

Les Forges

**La référence du level designer.
Et du développeur de jeux.**

<http://www.lesforges.org/>

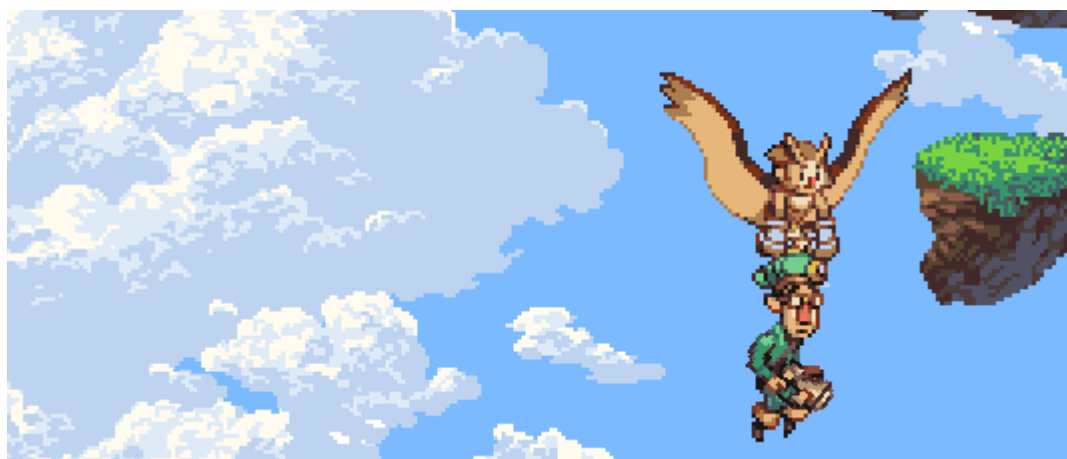
Créé librement pour la communauté (Yarflam)

Cours de pixel art 1 : préface

Par [Antoine Gersant](#) | le 11 juillet 2008 21:35:30 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Un peu de sérieux avant que je me mette à écrire mes bêtises

Le pixel art, même s'il a aujourd'hui perdu beaucoup de son importance à l'échelle de l'industrie du jeu vidéo, demeure un médium très utile aux créateurs indépendants et aux petites productions. Sa patte immédiatement reconnaissable a le don d'émouvoir une large frange de joueurs, tout en offrant aux créateurs l'occasion d'utiliser des techniques uniques en leur genre.



C'est de ces techniques que nous allons parler aujourd'hui, dans cette édition revue et augmentée du cours de pixel art original des Forges.

Définition du pixel art

Il existe de nombreuses définitions du pixel art mais nous n'en retiendrons qu'une ici : une image est faite en pixel art si elle est entièrement créée manuellement et que l'auteur a eu le contrôle de la couleur et la position de chaque pixel qui la compose.

En conséquence, en pixel art, on n'utilise ni brosses, ni outils flous, ni dégradés automatiques, ni un certain nombre d'autres outils que proposent les logiciels « modernes » et qui ont le défaut de produire des couleurs parasites. On se limite à des outils « propres » comme le crayon et le pot de peinture. Toutefois, ce n'est pas parce qu'on ne parlera dans ce cours que de pixel art que les graphismes non pixel art sont moins bien ou moins beaux. Ils sont simplement différents et se prêtent moins aux jeux « à l'ancienne » (dans la veine des graphismes de jeux *Super Nintendo* ou *Game Boy Advance*). Rien ne vous empêche d'ailleurs de combiner les techniques enseignées ici avec des effets non pixel art pour créer un style hybride.

Contenu de ce cours

Ici, vous apprendrez donc la partie technique du pixel art. Malheureusement, en aucun cas ce cours ne fera de vous un artiste pour la simple et bonne raison que je n'en suis pas un. Je ne vous apprendrai rien sur l'anatomie humaine, rien sur la composition d'une œuvre d'art et assez peu de choses sur la perspective. Vous trouverez surtout dans ces lignes des informations sur des techniques propres au médium pixel art (mais pas uniquement, je ne suis pas non plus obtus).

Au final, vous devriez être capable avec de l'entraînement de produire des décors et des personnages tout à faire corrects pour vos jeux, pour peu que vous soyez attentifs et que vous vous entraîniez régulièrement à appliquer les conseils donnés.

Par ailleurs, je tiens à vous signaler que seule une partie des images utilisée dans cette série d'articles vous est livrée « zoomée ». Quand ce n'est pas le cas, ce serait une bonne chose que vous preniez le temps de copier/coller ou d'enregistrer ces images afin de pouvoir les étudier en détail. Le pixel art est une affaire de pixels, inutile d'espérer les étudier de loin.

Enfin, je me dois de remercier tous les artistes qui ont participé avec moi à la création de ce cours d'une manière ou d'une autre : Pierre Guyot pour ses basses besognes et ses linearts (le dragon que vous verrez bientôt sous toutes les coutures, c'est aussi le sien), Xenodrogen pour son génie de la couleur, Lünnn pour ses connaissances en perspective, et enfin Panda, Poupi, Ahruon, Dyo et Kryon pour leurs œuvres généreusement prêtées pour illustrer ces pages.

Voilà, vous êtes fins prêts à rentrer dans le vif du sujet !

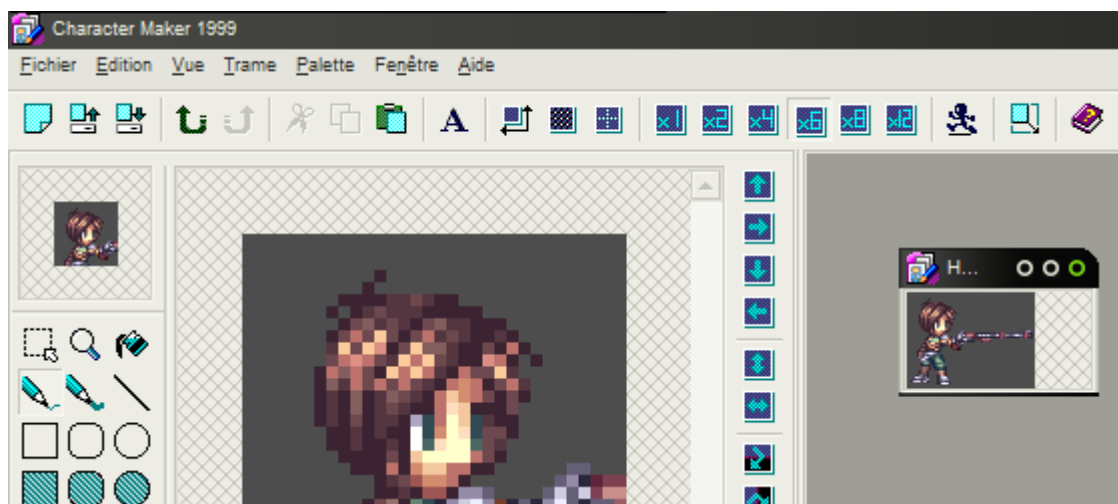
[Lire la suite →](#)

Cours de pixel art 2 : les bons outils

Par [Antoine Gersant](#) | le 11 juillet 2008 23:11:42 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

S'il y a bien un proverbe que je déteste c'est :

Il n'y a pas de mauvais outils, seulement de mauvais ouvriers.



Je crois même qu'il n'y a rien de plus faux (sauf peut-être que « tout ce qui ne tue pas rend plus fort » mais c'est pas le sujet) et le pixel art en est une très bonne preuve. Ce chapitre a pour but de vous présenter différents logiciels qui permettent de faire du pixel art et de vous aider à choisir celui qui vous conviendra le mieux.

Tout pourri.

Au moment de choisir son logiciel pour faire du pixel art, ce qu'on pense le plus souvent c'est :

Hein ? ben on s'en fout non ? Pour faire du pixel art il faut juste **Paint**.

Tragique erreur : je parlais de mauvais outils, en voilà un.

Paint a un avantage (et un seul) : vous l'avez déjà si vous êtes sous Windows. Par contre, il a des inconvénients à la pelle et en voici une liste (non exhaustive) :

- On ne peut ouvrir qu'un fichier à la fois.
- Pas de gestion de palette (remplacement de teintes, modification appliquées à plusieurs couleurs).
- Pas de calques.
- Pas de transparence.
- Pas de sélections autres que rectangulaire et elliptiques (ha ha).
- Peu de raccourcis claviers.
- Ergonomie catastrophique.

Bref, **Paint** vous pouvez oublier. Maintenant on attaque les vrais logiciels.

Éventuellement...

Ce qu'on pense ensuite souvent c'est :

Bon ok, **Paint** c'est bien trop limité pour moi, alors je vais utiliser mon ami **Photoshop** (ou **PaintShopPro** ou **The Gimp**, ça revient au même) qui a mille fois plus de possibilités.

Cela peut-être une bonne ou une mauvaise idée : si vous savez déjà bien maîtriser l'un de ces logiciels, vous pouvez faire du pixel art avec (en désactivant toutes les options d'anti-aliasing automatique et en vous privant d'une bonne partie des fonctions évidemment). Si en revanche vous ne savez vous servir d'aucun de ces logiciels, vous passeriez beaucoup de temps à apprendre comment ils fonctionnent alors que vous n'avez pas besoin de toutes les options qu'ils proposent et ce serait une énorme perte de temps.

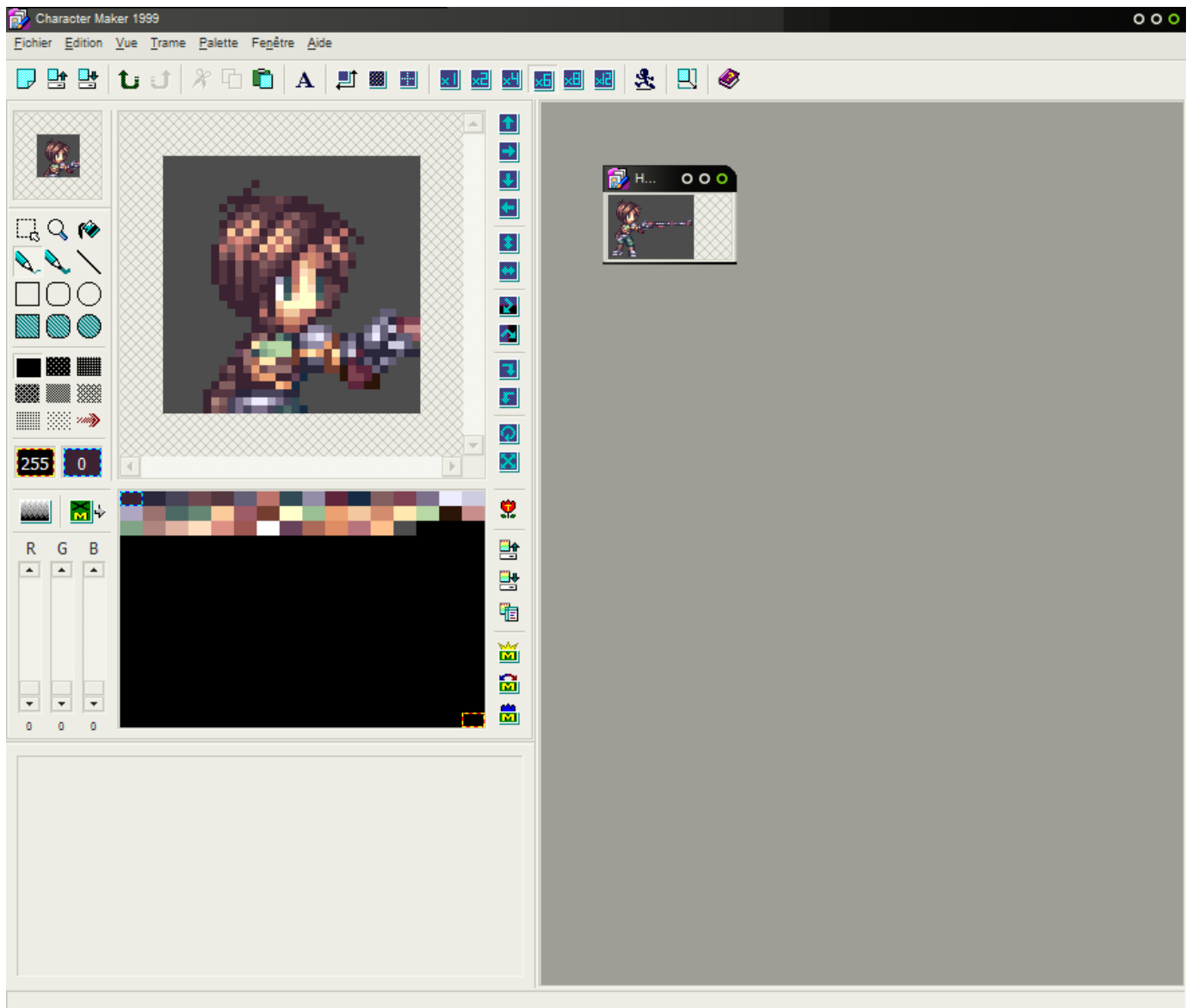
Pour faire court, si vous utilisez déjà **Photoshop/PSP** depuis longtemps, vous pouvez faire votre pixel art avec (j'utilise personnellement **Photoshop** par habitude), mais dans le cas contraire, mieux vaut utiliser un logiciel spécialisé dans le pixel art. Car oui, ça existe.

Le top du top.

Les logiciels dédiés au pixel art sont plus nombreux qu'on pourrait le penser mais nous ne retiendrons ici que ceux que nous jugeons les meilleurs. Ils ont tous des fonctionnalités très semblables (gestion efficace de palette, possibilité de prévisualiser ce que donnerait un *tile* mis bout à bout plusieurs fois, gestion de la transparence, gestion des calques, etc.). Toute la différence se situe dans l'ergonomie... et dans le prix.

Charamaker 1999

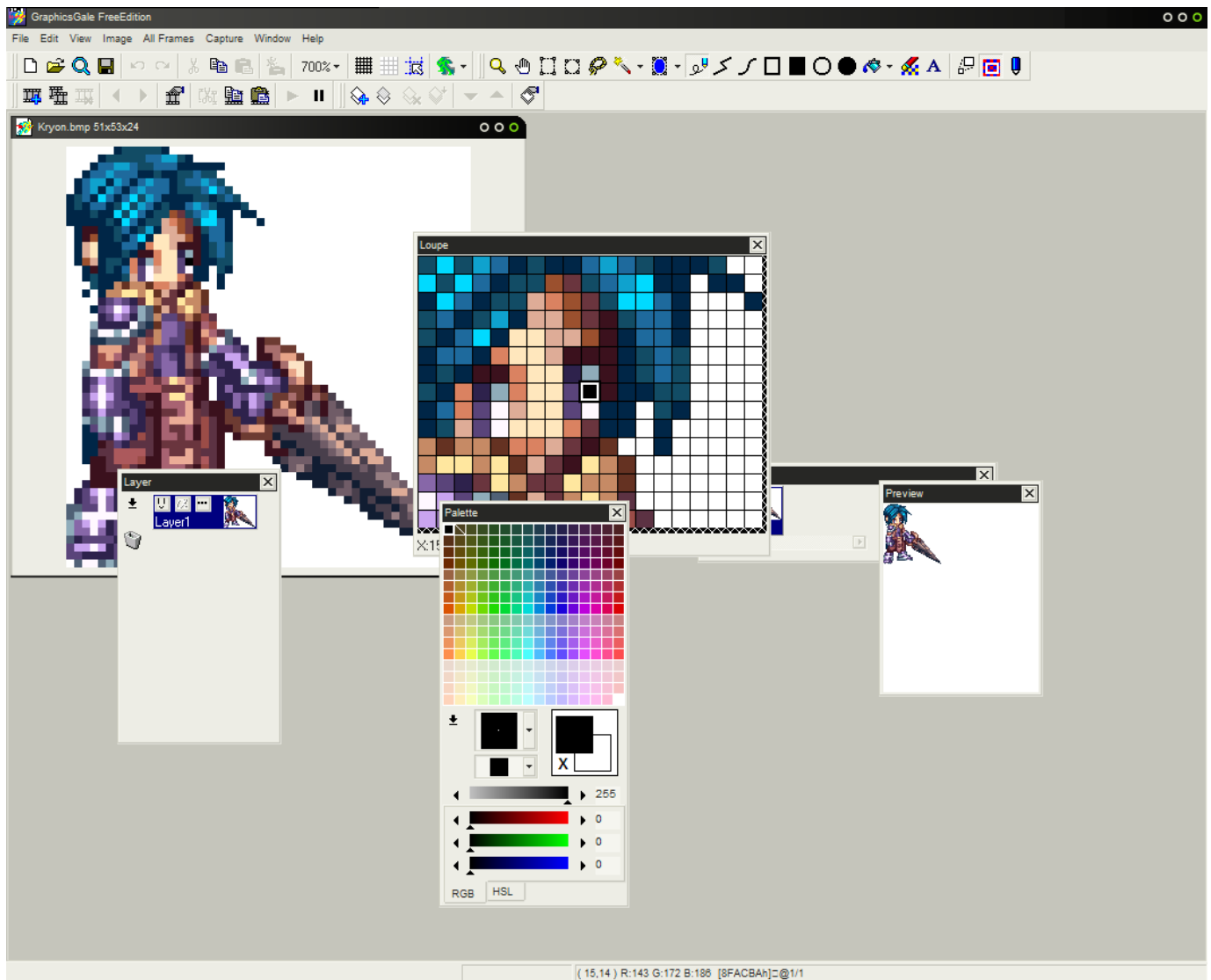
Charamaker 1999 a un gros avantage pour l'internaute pressé que vous êtes : la distribution officielle semble arrêtée depuis très longtemps et on peut le trouver en téléchargement direct aux quatre coins de la toile. La légalité du procédé n'étant pas très claire, nous vous laissons le soin de le trouver vous même (via [Google](#) par exemple).



Graphicsgale

De mon côté, je trouve l'interface de **Graphicsgale** beaucoup plus lisible et bien plus pratique. Néanmoins, ceci a un prix (12€).

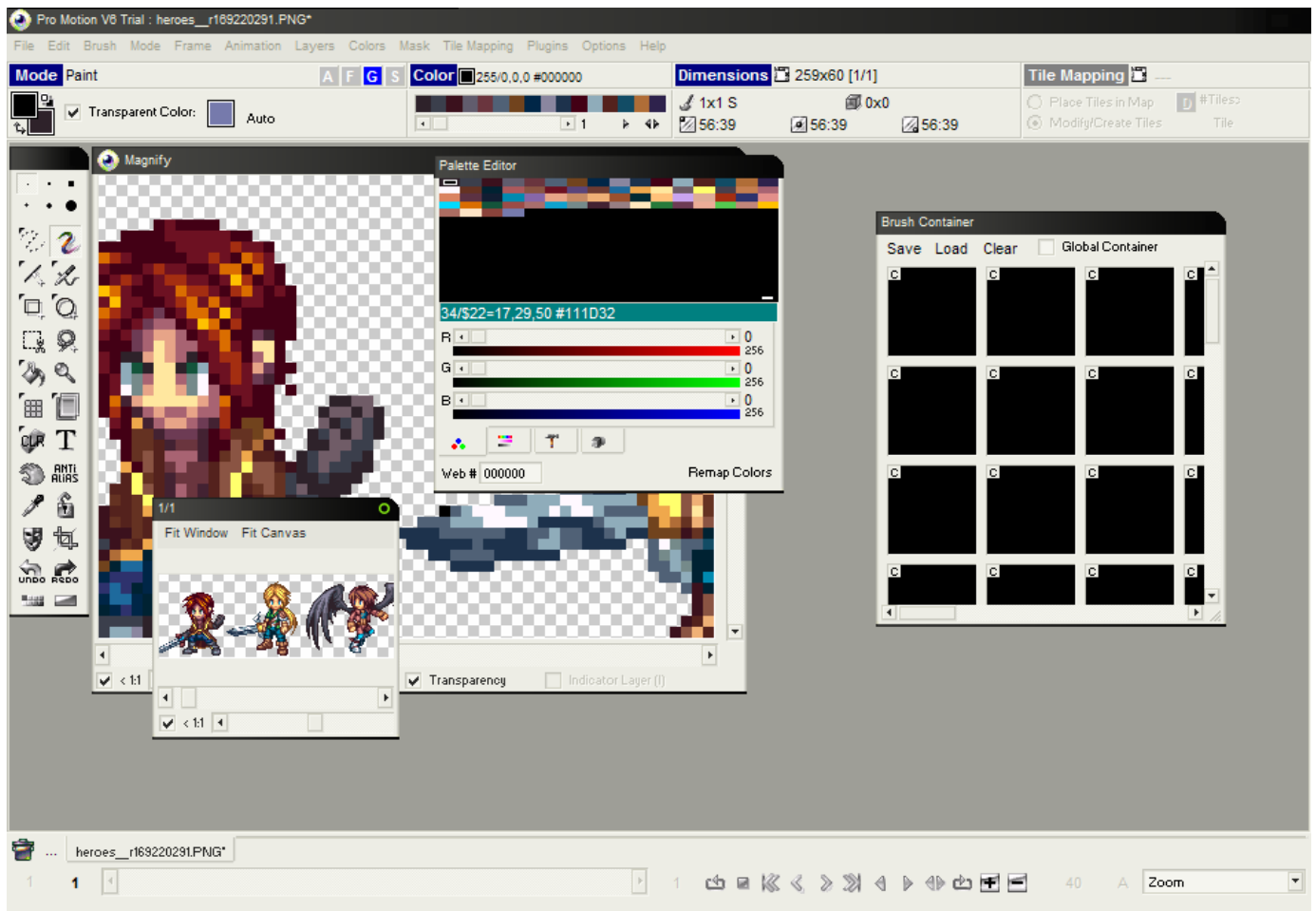
Cela dit, ça reste très abordable. De plus, la version d'essai comporte **absolument toutes** les options nécessaires pour faire des graphismes de jeux. La seule fonction absente de cette version est la gestion des images en *.gif* mais ce n'est pas du tout un problème puisque le format *.png* lui est préférable. Par ailleurs, cette version d'essai n'est pas limitée dans le temps et est donc tout à fait utilisable à long terme.



Vous pouvez télécharger la version d'essai depuis le [site officiel](#).

Pro-Motion

Le logiciel le plus souvent utilisé par les pixel artistes professionnels est **Pro-Motion** qui est (à ce qu'il paraît) plus ergonomique et plus rapide que **Graphicsgale**. Ah oui, il coûte cher ! Vous pourrez vous procurer la version complète pour la modique somme de ...50€ (78\$).



Vous pouvez télécharger la version d'essai depuis le [site officiel](#).

Pixen

N'oublions pas les malheureux utilisateurs de MacOS. **Pixen** est un logiciel open source de la même trempe que les précédents et il a le mérite d'être gratuit, même si son développement est un peu chaotique. Je ne peux malheureusement pas vous en dire plus puisque je n'ai pas de Mac.

Retrouvez-le sur [github](#)

Je vous recommande vivement de les essayer tous grâce aux versions de démonstration et de faire quelques essais dessus afin de voir lequel vous convient le mieux en termes d'ergonomie. Essentiellement, on peut dire qu'ils se valent et qu'il s'agit simplement d'une question de goûts. Sachez toutefois qu'une fois habitué à un logiciel, il vous sera très difficile d'en changer.

[Lire la suite →](#)

Cours de pixel art 3 : tracer des lignes

Par [Antoine Gersant](#) | le 13 juillet 2008 19:28:45 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Fin la parlote, on va enfin faire du pixel art.

Si vous n'êtes pas encore un artiste accompli (non vous n'en êtes pas un), la façon la plus simple d'aborder une illustration est de réaliser une première version au crayon, de l'encreur une fois que vous en êtes satisfait, puis de la mettre en couleur. Il en va de même pour le pixel art : la façon la plus simple d'aborder un pixel art est le tracé de ses contours, le **lineart**.



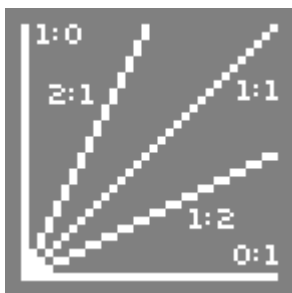
Le lineart est une étape extrêmement importante de la réalisation d'un pixel art. En effet, quelques pixels de votre image représentent un gros volume dans la réalité (contrairement à un dessin, où l'échelle permet une plus grande tolérance) si bien qu'une erreur d'un ou deux pixels peut donner à un personnage une malformation congénitale de 20 centimètres. La précision du lineart est **C-A-P-I-T-A-L-E** pour la réussite d'un pixel art (j'insiste).

Droites.

Un lineart est essentiellement composé de droites et de courbes et nous allons donc commencer par parler de droites.

Ceux d'entre vous qui ont écouté leurs cours de maths au moins jusqu'au collège savent qu'une droite est caractérisée par son coefficient directeur. Il s'agit du quotient des « variations en y » sur les « variations en x ». Pour ce cours, on représentera une droite par ce quotient, noté sous la forme $y : x$. L'idée intuitive est la suivante : quand je me déplace de x pixels vers la droite, je me déplace de y pixels vers le haut (ou le bas si y est négatif).

Ainsi, une diagonale parfaite de « en bas à gauche » vers « en haut à droite » est une droite $1 : 1$. Voici quelques lignes simples et leurs coefficients directeurs pour illustrer l'idée.



En pixel art, les droites qui utilisent ces coefficients directeurs simples (0 : 1, 1 : 2, 1 : 1, 2 : 1 et 1 : 0), sont appelées « lignes parfaites ». Elles sont composées de segments qui font tous la même longueur et elles contribuent à rendre votre image plus agréable car elle repose l'œil, qui les suit sans difficulté. On les utilise par exemple beaucoup lorsqu'on fait de la 2D isométrique.

Elles ne sont cependant pas les seules que vous pouvez tracer : vous pouvez par exemple faire des lignes « intermédiaires » en alternant des segments de longueur deux avec des segments de longueur un (pour simuler, de loin, des segments de longueur 1,5). Exemple en image.

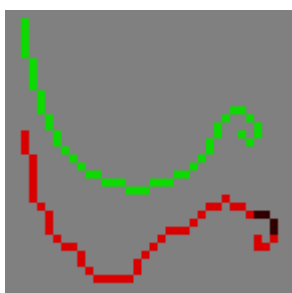


Le résultat est nettement moins esthétique (d'autant plus que l'image est ici agrandie 4 fois pour que vous puissiez voir ce qui se passe) et c'est pourquoi il est important de modérer l'usage des lignes comme celle du milieu. Rassurez vous : elles ont tout de même leur utilité, et vous apprendrez à les embellir dans un chapitre plus avancé de ce cours.

Courbes.

Bien, vous savez maintenant tracer des traits. Vous n'irez tout de fois pas bien loin avec uniquement des lignes droites. Passons donc aux courbes, qui sont un sujet autrement plus complexe. Contrairement à une droite, une courbe peut être réussie ou ratée.

Il n'y a qu'une seule règle à suivre pour réaliser une courbe bien lisse : la longueur des segments doit varier de manière progressive, et il faut éviter les angles droits injustifiés (non, sans blague ?). Bon, je vous l'accorde, ça fait deux règles, mais en fait la seconde est incluse dans la première. Des exemples seront cependant plus parlants.



- En vert, une jolie courbe qui suit cette règle à la perfection. De gauche à droite, les longueurs sont : **5 4 3 2 2 1 1 1 2 3 3 3 2 1 1 2 2 1 2 1 2 1 1**. La longueur des segments ne varie ici que de 1 en 1 (pas de **5 3** ou de **2 4** par exemple). Elle pourrait varier un peu plus mais jamais de façon brutale.
- En rouge, une deuxième courbe qui la bafoue sans vergogne (ne faites pas ça !).

- En rouge foncé, un segment qui bafoue plus précisément sa deuxième partie (« les angles droits »).

Au final, une bonne technique pour tracer une courbe consiste à dessiner un premier jet « à main levée » (on obtient quelque chose du genre de la courbe rouge) puis à repasser derrière et à la modifier localement jusqu'à ce qu'elle respecte « la règle ». Cela permet de s'assurer que la courbe aura l'allure souhaitée. Bon, les courbes, c'est fait.

En conclusion, je vous laisse avec de petits exemples d'applications de ce cours. Le premier est un dragon tout en rondeurs qui illustre les principes de tracé de courbes, le second est une épée qui montre que, parfois, des lignes plus anguleuses (regardez la pointe et le pommeau) peuvent aussi être utiles quand on n'a pas la place pour mieux.



Je vous recommande fortement de faire quelques linearts avant de poursuivre la lecture de ce cours : vous pourrez ensuite vous en servir pour mettre en application le reste du cours (de même que je me servirai du dragon ci-dessus). Si vous avez du mal, vous pouvez toujours partir d'un dessin scanné au préalable.

[Lire la suite →](#)

Lexique de Pixel art

Par [Antoine Gersant](#) | le 14 juillet 2008 22:07:20 | Catégories : [Graphisme](#), [Pixel art](#)

Le pixel art est un médium auquel sont associé de nombreux termes techniques, barbarismes et néologismes aux nuances parfois subtiles. Visite guidée.

Anti-Crenelage/AC (Anti-Aliasing/AA) : technique de Pixel art permettant d'adoucir la transition d'une couleur à une autre. Ce n'est ni un contour ni un dégradé.

Color ramp : traduction anglaise de rampe.

Dithering : technique de pixel art qui simule un dégradé en utilisant peu de couleurs différentes, à l'aide de motifs répétés qui entrelacent des couleurs. Le motif de dithering le plus simple est le damier 50/50.

Edit : action de modifier le Pixel art de quelqu'un d'autre pour suggérer des améliorations.

Glissé de teinte (Hue-shift) : il s'agit d'une technique de gestion de palette ; faire un glissé de teinte sur une rampe de couleur c'est ne pas donner la même teinte à toutes les couleurs de la rampe. Par exemple sur une rampe de marrons, on peut faire varier la teinte du orange (couleurs claires) au violet (couleurs foncées).

JPG ou JPEG : le JPEG c'est le maaaaaaaal ! C'est un format d'image destructeur (il compresse votre image pour en alléger le poids) adapté aux photographies et *surtout pas* au Pixel art.

Lineart : tracé des contours d'un Pixel art avant sa mise en couleurs.

Luminosité (Luminance) : l'une des trois composantes d'une couleur dans le système TSL (HSL). Elle désigne comme son nom l'indique la luminosité d'une couleur. Une luminosité de 0% correspond à du noir (quelles que soient la teinte et la saturation) tandis qu'une luminosité de 100% correspond à du blanc (quelles que soient la teinte et la saturation).

Non-Pixel art : graphismes qui ne sont pas du pixel art.

NPA : l'abréviation de non-Pixel art.

PA : l'abréviation de Pixel art.

Palette : l'ensemble des couleurs utilisées dans un Pixel art. En pixel art, on essaye d'économiser le nombre de couleurs utilisées pour avoir un rendu plus uniforme et pour faciliter la gestion de la palette.

Pixel : abréviation de "Picture Element". Une image bitmap est composée de pixels, les pixels sont les tout petits carrés qui composent l'image. Chaque pixel porte une couleur (et une seule).

Pixel art : image entièrement créée manuellement et où l'auteur a eu un contrôle sur la couleur et la position de chaque pixel qui compose l'image. En clair, en pixel art on n'utilise ni brosses, ni outils flous, ni dégradés automatiques et autres options que les logiciels "modernes" mettent à notre disposition. On se limite à des outils propres comme le "crayon" et le "pot de peinture".

PNG : c'est le meilleur format d'image pour faire du Pixel art : il est léger et conserve tous les détails de vos images.

Rampe (Color Ramp) : une palette est composée de plusieurs rampes. La rampe est un ensemble de plusieurs couleurs de teinte proche. Par exemple on peut faire une palette avec une rampe de bleus + une rampe de verts + une rampe de marrons.

RVB (RGB) : mode de sélection de couleur à partir de ses composantes Rouge, Vert et Bleu. Ce système est peu pratique pour le Pixel art dans lequel le système HSL est bien plus employé.

Saturation : l'une des trois composantes d'une couleur dans le système TSL (HSL). Elle désigne la vivacité de la couleur : une saturation de 0% correspond à un gris (quelle que soit la teinte) tandis qu'une saturation de 100% correspond à une couleur très (trop) vive.

Spritesheet : une spritesheet est une planche de personnages. Elle contient un ou plusieurs personnages et leurs animations.

Teinte (Hue) : l'une des trois composantes d'une couleur dans le système TSL (HSL). Elle désigne ce qu'on appelle au quotidien la couleur (jaune, rouge, bleu vert...). Les teintes vont de 0° à 360°.

Teinte (Tint) : une couleur dans une palette. Exemple : J'ai 5 teintes de bleu dans ma palette.

Texturage (Texturing) : technique de Pixel art qui permet de simuler différentes matières (roche, bois, herbe...).

Tile : un tile est un morceau de tileset et mesure 16*16 pixels. Il représente un élément de décor, par exemple un "tile d'herbe" ou un "tile de buisson".

Tileset : un tileset est une planche de décor qui contient côte à côte les différents tiles qui composent une map.

TSL (HSL) : mode de sélection de couleur à partir de sa Teinte, Saturation et Luminosité. Ce système est bien plus pratique que le système RVB(RGB). Consulter les définitions individuelles de Teinte, Saturation et Luminosité pour en savoir plus.

Eléments de Creature Design

Par [Pierre Guyot](#) | le 14 juillet 2008 22:50:28 | Catégories : [Creature Design](#), [Dessin](#), [Graphisme](#)

Dans un univers de jeu, plus particulièrement dans le RPG, le *Creature Design* est un élément important. Les monstres, et plus généralement les créatures, donnent corps à l'univers du jeu et se retrouvent partout : villages, forêts, grottes, immeubles, etc. Qui plus est, le joueur risque de s'en farcir un certain nombre tout au long de l'aventure, alors quitte à devoir vaincre quatre cents exemplaires du même gobelin (ou de sa version recolorée avec 20 niveaux en plus :3), autant que ce soit un gobelin superbe et original.



Ce cours, si on peut appeler ça ainsi, vise à expliquer la démarche de questionnement et de recherche nécessaire à la conception d'une créature, d'un monstre ou d'une bestiole (démarche qui peut s'appliquer à d'autres domaines, le *Character Design* entre autres). Je vais essayer de développer des notions comme les valeurs, les contrastes, etc. Je signale avant toute chose qu'il ne s'agit absolument pas de LA méthode à suivre, mais de la méthode que j'emploie personnellement : elle est parfaitement critiquable, il se peut tout à fait (et c'est quasiment sûr) qu'il y ait mieux et vous n'êtes pas obligés de la suivre. Si vous avez des suggestions à faire pour modifier ou améliorer cet article, je suis toute ouïe :)

Au fur et à mesure de l'article, je vais développer un exemple de *Creature Design* et essayer de montrer comment il s'est formé, histoire d'illustrer mon propos (ce sont les passages en italiques). J'ai essayé de forcer sur les étapes, histoire que ce soit bien compréhensible, même si je ne le fais pas toujours de façon détaillée. Néanmoins, même s'il n'est pas nécessaire d'avoir chaque étape de l'article en tête, ce qui serait passablement fastidieux, je pense que c'est une bonne chose que de garder cette démarche en tête, dans un coin.

Questionnement préliminaire

L'idéal pour commencer, c'est de savoir dans quelle direction vous voulez partir. Si vous n'êtes pas sûr de ce que vous voulez, essayez de vous poser des questions sur la créature que vous allez concevoir : dans quel environnement va-t-elle évoluer ? Quelles sont les caractéristiques que je veux voir en priorité chez elle ? Quelles dominantes de couleur, quelles textures, quels éléments ai-je envie d'utiliser ?

Ces questions sont de deux natures : d'une part, des questions de logique, qui relèvent de l'utilisation que vous voulez faire de la créature (si elle vit dans l'océan, il semblerait plus approprié qu'elle ait de quoi se mouvoir dans l'eau, comme des nageoires ou des pattes palmées, par exemple), d'autre part, des questions de goût (si vous avez envie qu'elle soit rouge, qu'elle ait une crête punk, etc.).

Si vous avez vraiment des difficultés pour vous lancer, une autre technique est celle de la « ligne folle » : vous gribouillez tout ce qui vous passe par la tête (ou plutôt par le poignet), jusqu'à voir surgir une forme (« Tiens, ce gribouillis informe me fait penser à un sublime dragon-méduse ! »). Une fois passé par cette étape de questionnement préliminaire et que vous tenez un peu mieux votre sujet, vous pouvez vous lancer dans le dessin à proprement parler.

Pour l'exemple, j'ai d'abord cherché de vagues formes, en gribouillant du bout d'un crayon, jusqu'à trouver un truc qui ressemblait à quelque chose d'intéressant. J'ai ensuite essayé de faire le lien avec une créature

qui me serait utile pour les besoins de mon scénario : j'ai pu relier la bestiole en question avec un gardien de temple. Les pistes que j'ai gardées sont celles d'un style un peu asiatique, avec des masques blancs, très raffinés, en contraste avec un gros corps très bourrin et assez difforme.



Allure générale

A partir de ce premier croquis, il faut essayer de corriger ce qui ne vous satisfait pas. Bien que le premier jet apparaisse souvent comme le meilleur (car il est plus naturel), il comporte souvent des défauts qui n'apparaissent pas tout de suite à l'œil : dès que vous sentez que quelque chose ne vous convient pas tout à fait, même si vous vous dites que « ça pourrait passer », recommencez. En effet, cette sensation sera souvent suivie, dans les mois – voire semaines – qui suivront de la désagréable découverte de défauts dont on ne s'était pas rendu compte au premier regard.

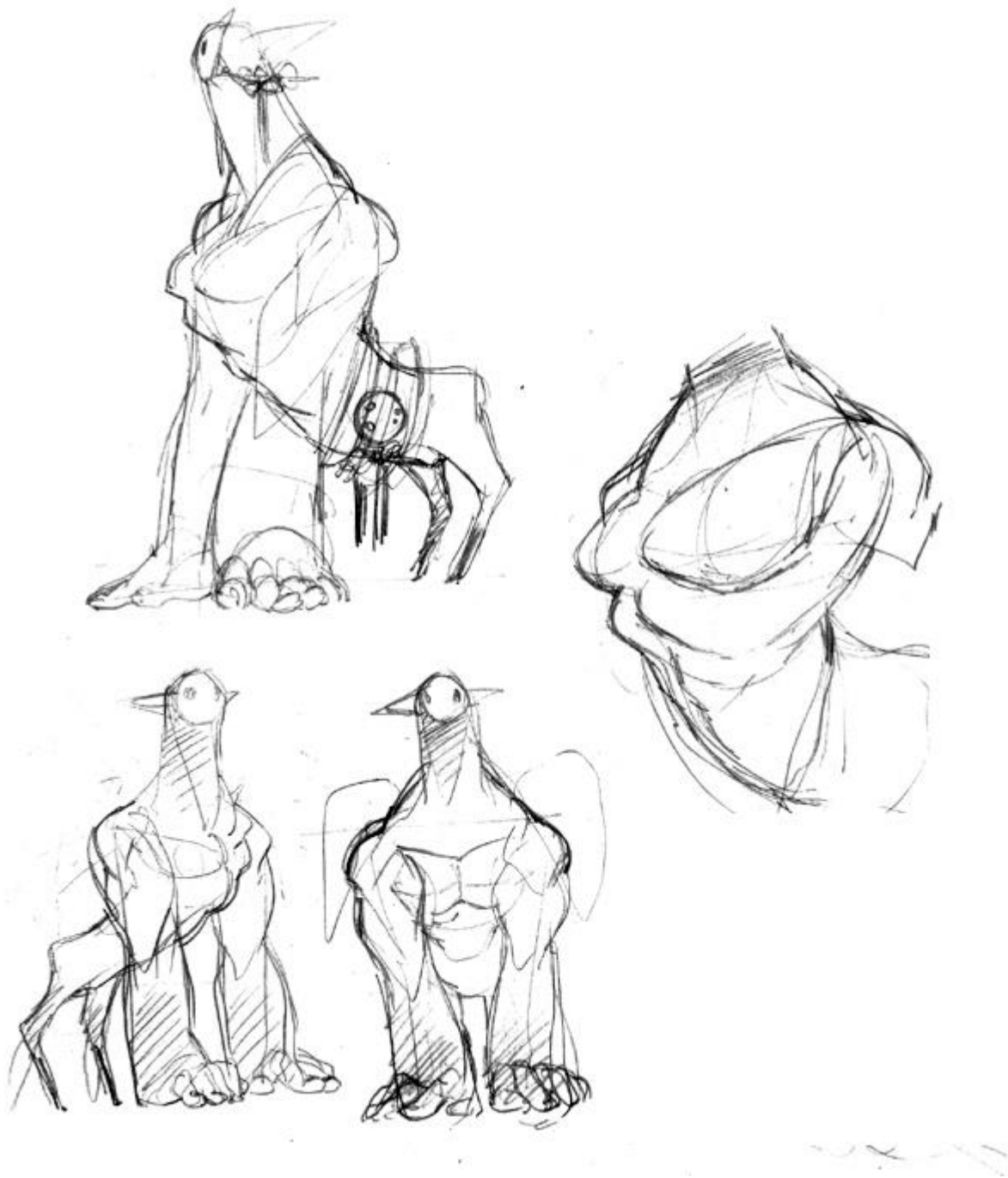
Il est important d'être satisfait de son travail à la fin de cette étape, dans la mesure où elle détermine la structure générale de votre créature, l'allure qu'elle aura au premier coup d'œil, sans faire attention aux détails. C'est à ce moment que des critiques extérieurs peuvent être utiles : même si cela peut sembler désagréable, les autres personnes ont par rapport à notre création un recul que nous n'avons pas, dans la mesure où nous avons, pour ainsi dire, « le nez dans le dessin ». C'est pourquoi il arrive qu'on se rende compte d'un défaut qu'on n'avait pas vu lors de la conception après un certain temps.

Cette première ébauche, qui donne l'allure du dessin, doit avoir deux qualités : d'une part, elle doit être claire, précise. Même si le trait n'est pas très soigné, on doit pouvoir lire sans ambiguïté la structure de la créature. Imaginez-vous que votre création doit pouvoir être modélisée en 3D et que donc tous les détails doivent être pensés (ce qui exclut toute tentative de masquer un détail qu'on n'arrive pas bien à décrire dans sa tête en le cachant :D). D'autre part, le design doit se suffire à lui-même, c'est-à-dire qu'il n'a pas besoin du dessin pour fonctionner. On peut tout à fait être un excellent designer et un dessinateur médiocre (l'inverse aussi d'ailleurs u__u'). C'est pourquoi il faut que, même si le crayonné est sale, le trait tremblant et la technique mauvaise, le design soit efficace.

Une fois que vous tenez votre ébauche, essayez de reproduire votre bestiole sous plusieurs angles, ou même plusieurs fois sous le même angle : il faut que la créature soit cohérente dans votre tête, que vous la pensiez

dans votre tête comme si elle était en 3D. Le fait de la dessiner plusieurs fois permet de la « fixer » dans votre tête.

J'ai ici modifié la silhouette de ma bestiole pour mieux coller à l'idée de gardien que je me faisais d'elle : j'ai ôté la queue qui donnait un côté trop animal et j'ai remplacé ses jambes par d'autres qui ressemblent un peu à celle d'un centaure, afin de donner un côté plus noble, plus approprié pour un gardien de temple. J'ai aussi balancé une partie de la masse musculaire sur les bras et réduit les jambes, pour renforcer le contraste entre l'aspect imposant du corps et le côté presque délirant du visage. J'ai fait plusieurs versions de façon à bien voir comment il rendait en 3D et j'ai zoomé sur le torse, histoire d'avoir une vision plus précise (attention, il y a des fautes d'anatomie sur ce dernier que n'excuse pas l'aspect cartoon U__u'). On peut remarquer que j'avais essayé des sortes d'ailes, mais j'y ai renoncé parce que ça gênait la répartition des masses et ça détournait l'attention (cf. suite :)).



Principe d'équilibre

Ce que je vais développer ici est un principe qui me semble extrêmement utile pour développer cette première version de la créature. J'appelle ça ici principe d'équilibre par commodité, mais ce nom n'est pas forcément le plus approprié et vous en avez peut-être déjà entendu parler sous une autre forme. Le but de ce principe, c'est de savoir « équilibrer » (ou savamment déséquilibrer ;)) les éléments de votre design afin de mettre en valeur ce qui compte dans le dessin. Ce principe d'équilibre peut prendre plusieurs formes :

Les valeurs

Les valeurs, ce sont les zones claires ou foncées du dessin (par exemple, les ombres), ou les niveaux de gris, si vous préférez. Ici, j'écarterai le cas des ombres justement, dans la mesure où elles font partie du dessin et non du design : les valeurs désignent ici les sombres et les clairs liés à la couleur même du monstre. L'idée est qu'il est important de savoir placer ses valeurs de façon à donner de la force au design. Un simple placement des zones suffit, pas la peine de s'attarder sur le travail de l'ombre et de la couleur : ce qui compte, c'est d'avoir des zones sombres et claires (pas des ombres, hein :o) réparties de façon à mettre en valeur l'essentiel du dessin. Il faut que vous ayez un bon contraste pour accrocher l'attention du spectateur : un dessin dont les niveaux de gris seraient uniforme n'a pas beaucoup d'intérêt. Une astuce pour voir si le contraste est bon est de passer l'image en niveaux de gris (Ctrl+Shift+U sous **Photoshop**) et de comparer avec les niveaux automatiques (Ctrl+Shift+L).

Ici, j'ai voulu mettre en valeur le contraste entre la lourdeur, la force, la brutalité du corps d'une part, et la finesse et la petite taille de la tête d'autre part. C'est pour quoi j'ai laissé le torse blanc au milieu de zones noires : il a l'air bien bombé, bien puissant, il ressort bien. En sombre, les mains ont l'air encore plus lourdes (ceci rajoute en plus l'idée qu'elles sont gonflées de sang, comme nos mains lorsqu'elles semblent lourdes :)). Enfin, le contraste du masque blanc sur le coup sombre fait ressortir de façon très nette la tête et la met en valeur, en contraste avec le corps.



Les détails

Il en va de même pour la répartition des détails, de leur finesse, des textures, des à-plats, etc. Si vous appliquez au dessin le même niveau de détail partout, des textures ultra-fines, l'œil du spectateur ne saura pas où aller, il n'y aura pas de zone qui attirera l'attention plus que d'autre : ceci reviendrait exactement au même que de tout laisser en à-plats, dans la mesure où aucun détail ne ressort plus que les autres au final. Il faut donc choisir de mettre en valeur certaines zones, quitte à en délaissé certaines (je sais, ça fait mal au cœur, surtout quand on a passé beaucoup de temps sur certains détails, mais en design, il faut toujours être prêt à tout virer pour recommencer :/).

Sur l'exemple, j'ai décidé d'accentuer encore plus le contraste masque/corps en mettant des détails au niveau de celui-ci (ainsi que sur la taille où je compte ajouter une ceinture), histoire que l'œil se focalise bien sur la tête (en rouge sur le dessin, mais vous vous en doutez ;)).



Les masses

Comme j'en avais parlé précédemment, la répartition des masses et des volumes est aussi importante : en choisissant de grossir ou de rétrécir telle ou telle partie de la créature, vous changez l'impression globale qu'elle donne. De plus, si c'est fait sans abus, accentuer les volumes (dans un style un peu cartoon par exemple) permet de donner une meilleure dynamique. Il faut aussi tenir compte des saillies : si une espèce de pic sort à un endroit d'une créature ronde, ce pic sera tout de suite très visible et au centre de l'image. Savoir jouer de la répartition des masses permet de donner une impression forte au spectateur. Attention tout de même, la déformation ne doit jamais se faire au détriment de l'anatomie : vous ne devez pas faire quelque chose de faux anatomiquement (n'importe qui en est capable), mais savoir faire quelque chose de juste anatomiquement et le déformer avec soin (et ça, peu y arrivent bien :/).

Ici, comme indiqué plus haut, j'ai réparti la plupart de la masse dans les bras, de façon à donner une certaine lourdeur à la créature. De plus, le coup en pointe, qui s'achève par la tête, permet d'accentuer la mise en valeur de celle-ci.

Les couleurs

Comme pour les valeurs, pas la peine ici de trop détailler, de simples à-plats sans nuance feront parfaitement l'affaire pour situer les couleurs. Une chose à retenir selon moi (mais sachez que je suis loin d'être un as de la couleur, faudra demander à Fil :D) : restreignez vous sur le nombre de couleurs (du moins au début). C'est vraiment délicat de gérer beaucoup de couleurs et on tombe dans le mauvais goût. Il vaut mieux se limiter à deux ou trois couleurs bien choisies. Si possible, choisissez une paire de couleurs complémentaires, une couleur et son complémentaire s'accordent toujours bien ensemble (à ce propos, n'oubliez pas la commande Ctrl+Alt+U sous **Photoshop**, ça peut être utile :)).

Ici, pas de couleurs, je vous renvoie vers Fil, qui vous sera beaucoup plus utile que moi sur ce sujet :).

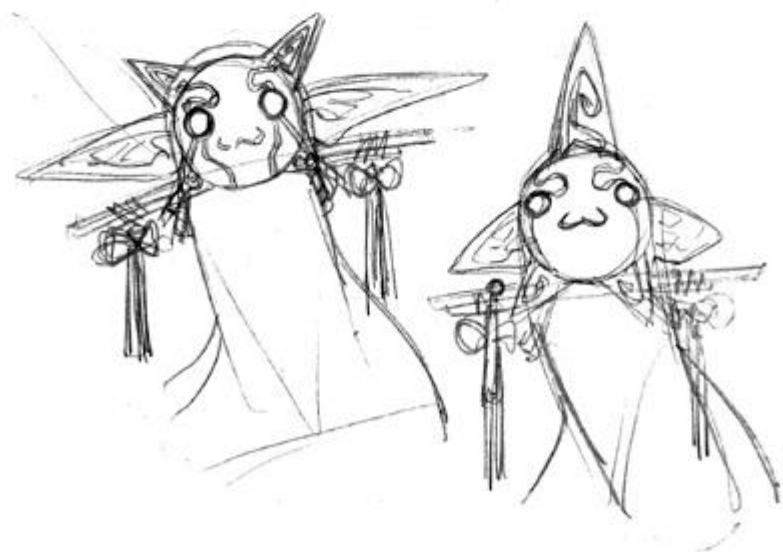
Détails

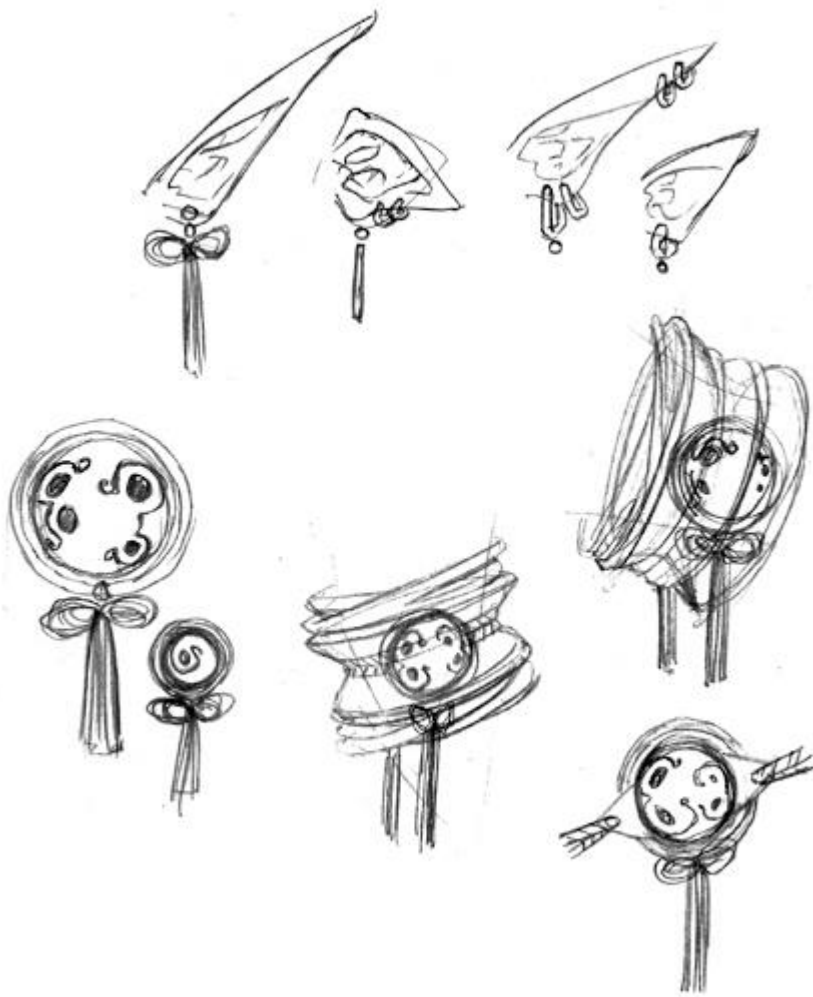
Une erreur qui revient souvent à ce niveau là de la conception est de se contenter de finir le croquis en rajoutant quelques petites choses et de lui donner son rendu final. Il ne faut pas s'arrêter, à part dans quelques cas rares où le résultat vous satisfait parfaitement du premier coup (et encore, on se rend

généralement compte peu après que ce n'était pas une très bonne idée). Au lieu de se contenter de la simple première version, essayons de pousser la démarche un peu plus loin : il faut essayer de fouiller plus, de voir comment on pourrait améliorer cette première version design.

Il est utile d'essayer de travailler sur les détails et de chercher des variantes (dans l'idéal, 5 versions pour chaque détail important : en réalité, ceci n'est pas faisable, mais il est bon d'avoir au moins deux idées, si possibles assez différentes pour les points importants de la créature). Essayer de ne pas vous cantonner à la première idée qui vous est venue et de ratisser le plus loin possible, même si ce n'est pas ce à quoi vous pensiez au départ. Pour le placement de ceux-ci, n'oubliez pas ce qui a été dit plus haut :D.

J'ai réalisé plusieurs versions des détails qui me semblaient important, à savoir la tête et le masque, les oreilles et les ornements de la ceinture.





Rendu final

C'est après toutes ces étapes de conception que vous pouvez enfin passer à un rendu finalisé. Je vous invite à vous rendre sur d'autres tutoriaux, plus portés sur le rendering que sur celui-ci. Néanmoins, même si le rendering n'est pas le sujet de ce tuto, je ne pouvais pas conclure sans poster une version finale :





Critiques

Un point important, et pas seulement en design – et ça, trop de gens l'oublient – c'est qu'il faut toujours être prêt à se remettre en question. Être ouvert à la critique, même si elle vous paraît injuste. Ce n'est pas parce

que vous avez passé 20 heures sur une œuvre que celle-ci est de qualité : vous devez toujours être prêt à tout effacer pour repartir de zéro et ce quelque soit le temps que vous avez déjà passé à travailler. Ceci peut sembler évident, néanmoins, c'est essentiel pour progresser et peu de gens l'appliquent vraiment. Je vous conseille vraiment d'y penser, et s'il n'y avait qu'une seule chose à retenir de ce tuto, c'est ça :).

Pour conclure, j'aimerais juste rappeler que ce tutorial n'est qu'une façon de procéder, la mienne, et pas LA méthode à suivre : il y a sans aucun doute plus efficace et plus intéressant. Si vous avez des suggestions à apporter sur ce tutorial, n'hésitez pas à m'en faire part. En espérant que ça vous sera utile ;)

Image d'ouverture par [njoo](#)

Méthodes et conseils pour bien travailler

Par [Dyo](#) | le 16 juillet 2008 21:14:08 | Catégories : [Graphisme](#), [Pixel art](#)

Dans cet article, je vais vous donner quelques règles importantes pour commencer (ou continuer) le « travail » sur de bonnes bases (difficile de parler de travail pour notre activité artistique favorite).

L'espace de travail

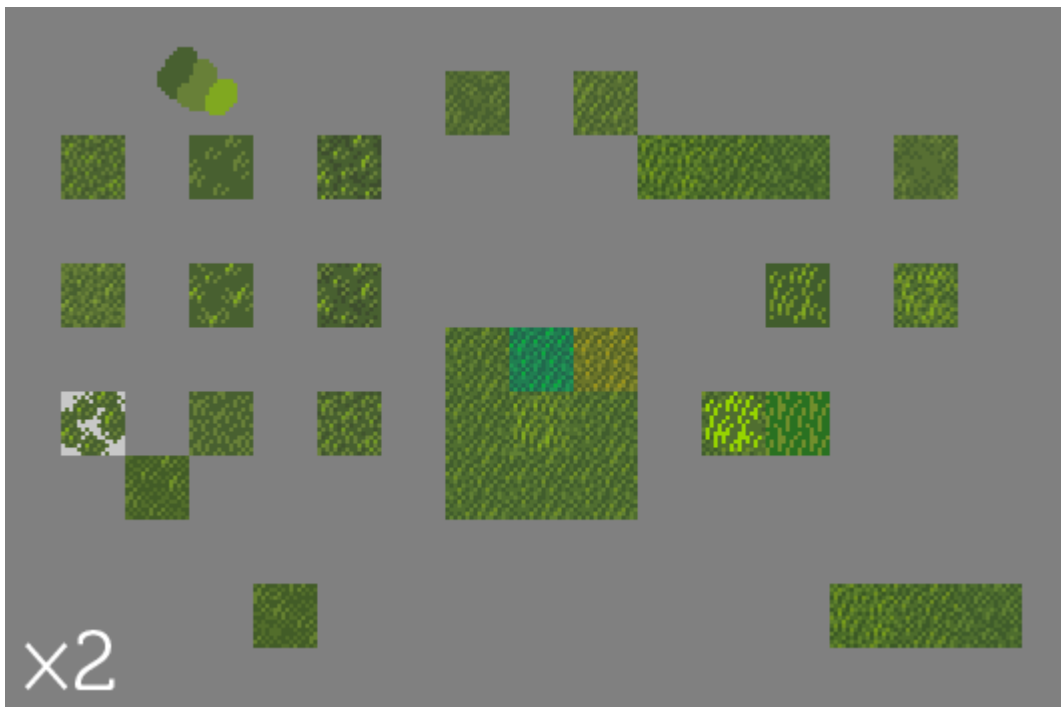
L'espace de travail « physique »

Sans aller jusqu'à vous dire comment vous placer sur votre chaise où à quelle distance de l'écran vous tenir, parlons quand même un peu de cet objet qui monopolise votre regard : le moniteur. Il faut faire attention au contre-jour (la lumière provenant d'une lumière ou d'une lampe qui se reflète directement sur l'écran). Même un petit contre-jour a vite fait de fausser l'appréciation des couleurs ou du contraste sans que l'on s'en aperçoive. Ensuite, pour le cas de mes collègues paléo-pixel artistes utilisant encore un écran CRT, attendez que l'écran chauffe avant de commencer à travailler (souvent la luminosité met un petit moment à atteindre sa vraie valeur).

Voilà pour ces quelques recommandations matérielles, passons aux recommandations virtuelles.

L'espace de travail « virtuel »

Quelque soit le logiciel utilisé, il est important de bien retenir cette règle : on ne travaille jamais sur le support définitif de l'image réalisée. Explicitons un peu les choses avec des exemples : vous souhaitez faire une GIF animée du sprite d'un personnage, ou bien réaliser plusieurs tiles devant par la suite être assemblés. Et bien, il est beaucoup plus utile de s'autoriser un vaste espace de travail, sur lequel on pourra rapidement comparer des variations de couleurs ou des essais de formes, déborder, superposer, etc.



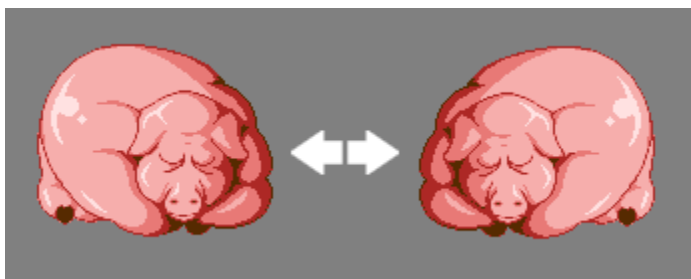
Notez que le fond n'est pas blanc, ni noir : il est « neutre ». Choisissez toujours un fond gris tel que celui ci (en valeur Rouge-Vert-Bleu : 128, 128, 128) pour ne pas être influencé et avoir une bonne appréciation du

contraste. Prenez aussi en compte le fait que même les barres d'outils et menus de votre logiciel, s'ils sont un peu clairs, peuvent fausser le contraste (le blanc à tendance à faire disparaître les nuances).

Essais et retouches

Quoique vous vouliez pixel arter, ne supprimez jamais vos brouillons, vos essais. Une forme qui ne s'adapte pas correctement (par exemple, l'accessoire d'un personnage pour un sprite, ou bien le motif décorant le mur d'une maison) pourra souvent être ré-adaptée ailleurs. Quelque fois aussi, on s'essaye à des choses trop compliquées mais le temps et le progrès aidant, on peut revenir sur un travail délaissé et le “remettre à neuf”. Lorsque vous travaillez sur une création, pensez aussi à ne pas vous acharner en retouches. Des fois il vaut mieux recommencer depuis le début (gardez votre essai dans un coin au cas où vous n'arriveriez pas à faire mieux).

Pour ce qui concerne les sprites complexes et/ou de grandes tailles, un moyen pratique pour repérer les défauts éventuels (et notamment, les défauts de proportion) consiste à retourner l'image horizontalement. En effet, l'œil s'habitue à un dessin ; les erreurs ne redeviennent visibles qu'en le retournant.



Par ailleurs, n'hésitez pas à créer en utilisant régulièrement cette technique pour votre image afin de ne pas laisser les erreurs s'insinuer dans la création (erreurs qui seront plus difficiles à rattraper une fois la création terminée).

[Lire la suite →](#)

Représentation et stylisation

Par [Dyo](#) | le 16 juillet 2008 22:13:02 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Cet article vise à vous donner quelques grandes lignes plutôt théoriques relatives à la création en pixel art.



Le dessin, nécessaire... ou pas ?

Commençons par le commencement : pour bien pixel art, il faut avant tout bien dessiner. Ne fuyez pas si vous êtes peu doué avec vos mains ! Car un avantage par rapport au dessin pur tient dans le fait qu'avec le pixel art, on travaille le plus souvent sur des surfaces réduites et les retouches sont aisées. Donc on peut largement compenser le manque de talent (travail serait un terme plus juste :p) avec un peu de patience et de motivation.

Mais reprenons. Vous vous trouvez devant votre “espace de travail” gris, prêt à commencer. Mais par où commencer ? Deux méthodes, chacune adaptée à la taille de la création visée.

Le dessin préparatoire

Pour un sprite de personnage, une illustration, un paysage, en somme quelque chose de relativement grand, le dessin sur papier ensuite scanné ou bien le dessin à la tablette s'avère quasiment obligatoire. Obligatoire pour structurer votre création, partir sur de bonnes bases. Il n'est pas nécessaire de faire un dessin préparatoire très détaillé ; un simple croquis, une esquisse de l'allure générale peut suffire. Les détails pourront être rajoutés après. En revanche, pas de secret ici pour avoir une illustration somptueuse, l'entraînement au dessin est inévitable.

[Exemple grand format](#)

Travailler directement

Pour le pixel art d'un élément de taille réduite, un tile ou bien une icône par exemple, le travail à l'ordinateur, directement à la souris, est possible. Le critère pour se passer d'un croquis, est de se demander : “est-ce que la structure de base peut être retouchée suffisamment rapidement ?”. Par exemple, une table devant s'inscrire dans un carré de 16x16 pixels : quelques pixels enlevés ou ajoutés, et l'aspect général est modifié.



En gros, vous devez être capable de définir la forme globale directement à la souris, en enlevant ici et là des pixels, mais sans avoir à « dessiner ». Rien n'empêche de dessiner à la souris la forme globale “à main levée” et de retoucher ensuite, mais vous perdrez beaucoup de temps.

Alternative...

Une méthode alternative possible consiste à se servir d'une photo, la réduire, et travailler par dessus. Mais je déconseille très fortement cette solution là, car la photo réduite donnera le plus souvent une « bouillie de pixels » inutilisable. Et bien sûr, travailler ainsi mobilise peu vos capacités créatrices, même si l'exercice peut être intéressant d'un point de vue strictement technique.

Reste une dernière méthode : se servir d'un dessin plutôt grand (et non d'une photo) et le réduire. En revanche il est impératif d'utiliser ici un dessin colorisé, sans line-art noir, pour la lisibilité.



Formes

Entrons dans le vif du sujet avec les formes. Tout comme dans le cas d'un dessin, deux voies s'offrent à vous : représenter de la manière la plus réaliste possible l'élément, ou bien le styliser, le déformer, lui donner une identité visuelle tout à fait particulière. A noter que même en essayant de représenter fidèlement un objet, cet objet portera toujours votre « coup de patte » personnel...

Je jure de dire la vérité... et uniquement la vérité

Personnellement, je conseille aux débutants de s'initier avec une représentation réaliste. Du moins, « la plus réaliste possible ». Déformer demande d'avoir l'œil pour ne pas tomber dans les erreurs, les excès, les irrégularités d'un élément à l'autre de la création. Tenter de représenter les choses telles qu'on les perçoit est donc une première approche assez indispensable.

Le jeu *Demon's Crest*, sur Super Nintendo. C'est un exemple de graphisme assez peu stylisé : les proportions du personnage sont cohérentes, les textures de sol et de mur se veulent réalistes. Cette absence de stylisation n'est pas un défaut et n'empêche pas le jeu d'être très réussi sur le plan esthétique.



Un concept fondamental, quand on essaye de faire réaliste, est de ne pas tricher. Il faut savoir renoncer à certains détails « qui tuent » pour rester cohérent ; ainsi on doit parfois résister à la tentation de mettre un simple pixel de détail, de reflet, pour ne pas tomber dans la stylisation (ce qui n'est pas un mal en soi mais dans ce cas là la stylisation doit être uniforme et... c'est là que les choses se compliquent).

Stylisation

La stylisation, si elle est maîtrisée, donne beaucoup de personnalité, on pourrait même dire qu'elle correspond à « l'âme » du pixel art. Par exemple, le style SD (Super Deformed) souvent adapté aux personnages des RPG japonais (la tête beaucoup plus grosse que le corps) leur confère un charme indéniable. Et au charme s'ajoute la lisibilité : déformer les parties essentielles (ici, le visage) permet de les mettre en valeur (on peut reconnaître les personnages autrement qu'avec la couleur de leurs vêtements). Mais il faut que tous les éléments du pixel art aient à peu près le même niveau de déformation, sinon le résultat sera hétérogène, déséquilibré.

La stylisation peut prendre plusieurs formes :

- Déformation de certaines parties des éléments
Ici, les têtes sont plus grosses que le reste du corps, les falaises apparaissent très rectangulaires.



- Éléments symbolisés
Dans Final Fantasy IV, les éléments sont peu diversifiés et symbolisent plutôt qu'ils ne représentent.



- Création épurée, avec très peu de détails
L'herbe ainsi que la falaise est réduite à un simple aplat de couleur.



Dans la plupart des cas de stylisation, un problème peut se poser quand on travaille sur un sprite extrêmement petit. En fait, les pixels peuvent se révéler trop gros. Dans ce cas, il faut savoir utiliser avec talent la palette pour que de loin, l'illusion fonctionne.

Prenons cet exemple :



Le but était de représenter en sprite le personnage de gauche, dont les yeux blancs sont entourés d'une ligne noire. Cette ligne étant très fine, si on utilise des pixels noir le personnage aura l'air de porter des lunettes (exemple A), un pixel est donc ici « trop gros ». C'est le recours à une échelle de gris qui va permettre de simuler des yeux blancs et leur contour (exemple B).

Vilain Game Designer ! Pas de biscuit pour toi !

Par [Antoine Gersant](#) | le 20 juillet 2008 22:54:38 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)
Cet article est une traduction de [Antoine Gersant](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mars 1998).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Je me suis remis récemment à jouer à de vieux jeux, et j'ai remarqué quelque chose d'intéressant par rapport aux jeux d'aujourd'hui : bien sûr, la technologie a énormément changé, mais certaines erreurs de conception de l'époque sont pourtant toujours présentes. Les énigmes mal conçues et les fausses-bonnes-idées irritantes appaues il y a de ça quinze ou vingt ans rôdent toujours.

Voici une liste de fausses-bonnes-idées que je ne supporte plus. C'est un point de vue hautement personnel et votre opinion peut être différente, mais pour moi, ces idées sont la marque d'un game design bâclé ou fait par un paresseux.

Les labyrinthes débiles et ennuyeux

Le premier jeu d'aventure textuel de l'histoire du jeu vidéo, *Colossal Cave*, comportait deux labyrinthes. Le premier était une succession de salles toutes décrites ainsi : « Vous êtes dans un labyrinthes de petits passages biscornus, tous semblables. » Le second était une succession de salles décrites par : « Vous êtes dans un petit labyrinthe biscornu de passages, tous différents » (ou par « Vous êtes dans un petit labyrinthe tordu de passages, tous différents », ou encore « Vous êtes dans un labyrinthes de petits passages biscornus, tous différents » etc etc...).

Ces deux labyrinthes étaient typiquement des labyrinthes débiles et ennuyeux, mais *Colossal Cave* étant le tout premier jeu d'aventure, on ne lui en veut pas trop pour ça. Par contre c'était il y a plus de vingt ans; il n'y a aujourd'hui plus aucune excuse pour faire ce genre de choses. Quelqu'un m'a donné un exemplaire de *The Legend of Kyrandia* il y a quelques années, et j'y ai joué avec plaisir...jusqu'à ce que j'arrive au labyrinthe.

Les labyrinthes ne sont pas nécessairement débiles et ennuyeux et il est possible de concevoir des labyrinthes divertissants en plaçant les salles selon un agencement que les joueurs peuvent comprendre. Un labyrinthe devrait être attirant, intelligent et surtout amusant à résoudre. Si un labyrinthe n'est pas intéressant ou si l'on apprécie pas d'y être, c'est une fausse-bonne-idée.

Les jeux sans carte

J'ai un sens de l'orientation notoirement déplorable en intérieur alors peut-être que c'est juste moi, mais dans le monde d'un jeu vidéo, où tous les murs et les sols utilisent les mêmes textures, je trouve que les endroits se ressemblent tous. Dans le monde réel, même les pires dédales de cloisons dans les immeubles de bureaux permettent de distinguer les pièces les unes des autres (une tache sur un tapis, une bande-dessinée affichée sur le cube de quelqu'un...). J'ai joué à *Doom* et je me suis bien amusé. Ensuite j'ai lancé une démo de *Quake* que j'ai abandonnée dès que j'ai vu qu'il n'y avait pas de carte. Je veux une carte ! Il n'y a pas de raison de m'empêcher d'en avoir une si ce n'est pour me ralentir, et c'est une arnaque pour rallonger le jeu sans offrir de gameplay en plus. Vilain Game Designer ! Pas de biscuit pour toi !

Les élément incongrus ou qui brisent la magie.

Parfois les jeux d'aventure vous mettent devant un puzzle ou un autre obstacle complètement déplacé par rapport à l'histoire fantastique que vous vivez. A mon avis, c'est le résultat d'un manque d'idées de la part des concepteurs et cela déçoit les joueurs. Si vous m'avez conduit dans un monde magique dans lequel j'incarne un chevalier héroïque qui mène une glorieuse quête pour sauver la princesse en détresse, je n'ai pas envie de m'asseoir pour jouer au *Mastermind* avec le dragon ! Si il faut absolument que je joue à quelque chose avec lui, débrouillez vous pour que ce soit le jeu du moulin, mais il serait franchement préférable de se débarrasser de lui à la main.

Cela nous mène tout naturellement au problème suivant...

Le Surréalisme bidon

Bon nombre de jeux sortis évitent les monde habituels de SF ou fantasy et préfèrent plonger le joueur dans le monde tordu et dérangeant du Blabla blabla. Je vais vous apprendre quelque chose sur le S majuscule de Surréalisme et le A majuscule de Art : ça ne sort pas de nul part. Le vrai Surréalisme cherche à choquer l'esprit pour le rendre conscient de la condition humaine / de la nature de Dieu / du véritable sens de la compassion / etc... en juxtaposant des objets et des idées qui n'ont semble-t-il rien à voir. La clé de voûte du genre réside dans « semble-t-il » : le vrai Surréalisme, même s'il semble bizarre et peut être même vide de sens au premier abord, véhicule sur un thème sous-jacent.

Je n'ai jamais vu dans un jeu vidéo de Surréalisme qui puisse prétendre à une telle noblesse d'intention. La plupart du temps on dirait plutôt que quelqu'un s'est dit « ...et en arrivant dans la salle de contrôle de l'apocalypse, il y aura un clown à l'intérieur ! Ouais! Ca va être cool ! ». Le Surréalisme est comme la prose en poésie : facile à faire mais extrêmement difficile à bien faire. « C'est du Surréalisme » n'est pas une excuse pour un concept mal ficelé à la base.

Nous arrivons donc en douceur aux...

Énigmes qui demandent des formes extrêmes de pensée latérale

Nous parlons ici des énigmes du genre « utiliser l'abat-jour avec le bulldozer ». Le concepteur pense peut être que c'est drôle ou sur-réel, mais ce n'est en fait rien de plus qu'un caprice fatigant. C'est de la paresse dans la conception de l'énigme que de la rendre difficile en lui attribuant une solution irrationnelle ou tirée par les cheveux. On peut comme ça rallonger la durée de vie en créant des obstacles ridicules, mais on ne fait pas plus plaisir au joueur avec ça alors que c'est censé être le but recherché.

Énigmes qui ne demandent aucune forme de pensée latérale

Vous arrivez devant une porte fermée. La solution évidente pour passer est de trouver la clé, mais c'est aussi la plus ennuyeuse alors peut-être le jeu propose-t-il une autre solution pour l'ouvrir. Cela étant rarement le cas, il n'y a qu'une solution, quelle qu'elle soit.

Dans les jeu d'aventure textuels, on rencontrait souvent le problème du « quel est le verbe à utiliser ici ? » et on était coincé jusqu'à ce que l'on trouve le verbe attendu par le jeu. Casser ? Frapper ? Endommager ? Démolir ? Défoncer ? Brûler ? De nombreux jeux d'aujourd'hui souffrent de ce problème : un obstacle que l'on ne peut passer que d'une seule et unique façon. Le jeu n'encourage pas le joueur à penser, il lui demande juste de deviner ce que pensait le concepteur.

Dans la réalité, les moyens d'ouvrir une porte verrouillée ne manquent pas :

- Trouver la clé
- Crocheter la serrure
- Forcer ou persuader la personne qui a la clé de l'ouvrir
- Duper quelqu'un de l'autre côté pour lui faire ouvrir (peut-être simplement en frappant à la porte !)
- Casser la porte, la brûler, la couper, la dissoudre avec de l'acide etc...
- La contourner en passant par une fenêtre ou un trou dans un mur

La seule limite à cette liste est votre imagination.

Ok, je sais que c'est plus facile à dire qu'à faire. En tant que développeur, il est coûteux et difficile pour vous de penser et d'implémenter toutes les façons de passer une porte. Tout de même, maintenant que nous avons la capacité de créer des « environnements déformables » (c'est à dire des explosions et des coups de feu qui agissent sur le décor en plus des ennemis), il est temps d'introduire un peu de diversité dans les univers, pour récompenser les joueurs qui font preuve d'imagination.

Énigmes qui demandent des connaissances pointues et sans rapport avec le jeu

Je dois ce paragraphe à mon ami génie et maître des énigmes [Scott Kim](#). Je n'y avais pas pensé jusqu'à ce qu'il lise le brouillon de cet article et me le fasse remarquer. C'est une astuce bidon, et encore plus frustrantes que les private-jokes. « Non, je ne connais pas le nom de la piste 3 sur Sgt. Pepper's Lonely Hearts Club Band (NdT : un album des Beatles) », et s'il est vital que je sache ça pour finir le jeu, alors le jeu est mal ficelé. (Les jeux de Quizz du style *You Don't Know Jack* sont évidemment une exception à cette règle, on sait à quoi s'attendre quand on démarre une partie).

Un interrupteur qui ouvre une porte à deux kilomètres d'ici

Et pas forcément une porte, je parle de n'importe quel obstacle sur le chemin du joueur. *Doom* a beaucoup utilisé cette arnaque, mais le pire exemple qui soit est *Le Guide du voyageur galactique*, un jeu d'aventure textuel créé par Infocom. Dans ce jeu, si vous ne ramassiez pas le « courrier de pub » au tout début du jeu, vous ne pouviez pas gagner tout à la fin. Cette fausse-bonne-idée est inutile et profondément frustrante. A part dans les raffineries et dans les centrales nucléaires, la poignée d'une porte se trouve le plus souvent — suspens suspens — sur la porte. C'est un autre exemple de paresse dans la conception d'énigmes en rendant le problème difficile non pas parce qu'il demande d'être intelligent mais parce qu'il est long à résoudre.

Un seule bonne combinaison sur les [Insérez ici un gros chiffre] possibles

Encore des énigmes faites par des paresseux. A la fin de *Infidel*, une autre aventure de chez Infocom, il fallait faire quatre action dans un certain ordre, ce qui nous fait 4! (factorielle 4, c'est à dire 24) combinaisons. Il n'y avait aucun indice pour vous aider à trouver la bonne séquence, il fallait toutes les essayer. Berk. Encore du temps perdu sans s'amuser.

(NdT : Le problème est apparu plus récemment dans *Resident Evil* sur PSX/Gamecube où aucun indice ne vous était donné pour trouver la bonne combinaison sur un digicode à 4 touches. Si vous cherchez encore, la bonne combinaison est 4231.)

Tuer un monstre / Ramasser l'épée / Vendre l'épée / Acheter une autre épée / Tuer un autre monstre

...ou le RPG dans toute son excellence. Vous avez peut-être entendu la blague de John F. Kennedy's qui dit que Washington D.C. est une ville qui a « l'efficacité du Sud et le charme du Nord », et bien à mon avis, la plupart des RPG combinent « L'adrénaline et l'excitation d'une simulation bancaire avec le challenge intellectuel d'un jeu de tir ». Je joue à des jeux d'aventure médiévaux pour être héroïque, tuer des princesses et sauver des dragons (NdT: chez moi c'est plutôt l'inverse) ; je n'y joue pas pour passer les deux tiers de mon temps à faire des affaires avec des marchands : je veux être un héros mais le jeu me force à être une sorte de marchand d'armes d'occasion itinérant. Ah et puis gagner de l'argent en dépouillant des cadavres ne me rend pas très noble non plus.

Vous avez 30 secondes pour résoudre ce problème ou vous êtes mort.

Vu le temps qu'un jeu met à recharger la partie, ce n'est pas très intelligent ni stimulant : c'est simplement frustrant. S'il faut trouver une astuce sans avoir eu d'indices, alors c'est encore une perte de temps sous la forme « essaye et crève ». S'il y a des indices, alors il faut donner un temps raisonnable au joueur pour y réfléchir. A titre de comparaison, les militaires ne chargent au hasard jamais sur un territoire qu'ils n'ont pas examiné au préalable (et si ils le font, ils le regrettent) : demander à votre joueur d'agir à l'aveugle n'est pas raisonnable. Si vous placez le joueur face à un danger imminent des l'instant où il voit l'écran, vous devez lui laisser au moins une chance sur trois de s'en sortir sain et sauf.

Les adversaires débiles

J'ai aussi marre des monstres qui avancent bêtement vers vous jusqu'à leur mort. C'est la méthode *Doom*, et celle d'un million d'autres jeux depuis l'aube des temps. Au lieu de vous offrir un défi intéressant, le jeu essaye de vous submerger par sa supériorité numérique. Argh. *Space Invaders* était peut être génial et addictif en son temps, mais il faudrait passer à autre chose.

Faisons preuve d'imagination ! Pourquoi ne pas mettre en place des monstres lâches qui vous donnent un coup facile avant de repartir se cacher pour combattre plus tard ? Ou un monstre qui peut alternativement s'exposer et se mettre à couvert ? Ou peut être un monstre qui s'échappe dès qu'il vous voit pour aller chercher ses copains (à moins que le joueur ne parvienne à l'abattre en chemin) . Pourquoi pas aussi un monstre qui essaye de s'infiltrer dans votre dos pour vous surprendre ? Ou un qui s'expose pour combattre mais part se cacher s'il est blessé plutôt que de lutter à mort bêtement ? Et peut-être pourrait-on faire des monstres dont le rôle serait de vous attirer hors de votre cachette pour que leurs alliés puissent vous tirer dessus. (C'est ce que faisait la soucoupe volante dans le jeu d'arcade *Battle Zone*.) Ou même — argh — des monstres assez intelligents pour faire tout ça à la fois, comme le font par exemple, mh... LES GENS ! Parbleu !

Aucune de ces idées n'est franchement neuve, mais on ne les voit pas souvent à l'œuvre. Pourquoi ? Encore de la paresse : les monstres idiots sont simples à programmer, les monstres intelligents ne le sont pas. Et puis il est facile de balancer un jeu dans lequel les adversaires sont débiles puisqu'il suffit de trouver le bon ratio entre monstres et trousse de soin. Pour rendre le jeu plus ou moins difficile, il suffit de changer le ratio. Mais c'est ennuyeux. Investissez un peu de matière grise en Monster Design, offrez un nouveau challenge aux clients.

Les deux dernières choses dont j'ai marre sont plutôt de l'ordre de l'esthétique que de la conception mais je les rajoute pour la forme.

Le mauvais jeu d'acteur

Le mauvais jeu d'acteur détourne le joueur de l'essentiel, et pas moins dans un jeu vidéo que dans un film. Cela brise sa suspension d'incrédulité, et c'est particulièrement visible quand un mauvais acteur est entouré de bons acteurs, si bien qu'on se retrouve à espérer la mort du personnage qui joue mal. Remarquez aussi que le jeu d'acteur est presque toujours mauvais dans les jeux.

Heureusement, c'est un problème qui se règlera tout seul au bout du compte. La compétition nous forcera à développer des compétences dans le domaine et si on atteint le niveau du 'téléfilm de la semaine', je m'estimerai heureux. Les talents John Gielgud et Katharine Hepburn (NdT: Un célèbre acteur de théâtre britannique et une actrice américaine de renom) seraient inutiles pour un jeu vidéo dont le but est de de toute façon d'être interactif. En général il vaut mieux pas de jeu d'acteur plutôt qu'un mauvais jeu d'acteur, c'est d'ailleurs plus simple et plus économique.

De belles explosions bien propres sur elles

Regardez de près une photo prise là où une bombe a explosé. C'est le bordel. Le gros bordel. Les choses sont brisées en morceaux de toute tailles, allant de l'objet presque entier à la poussière, en passant par divers fragments et copeaux. Elles sont aussi tordues et déchirées au point d'être à peine reconnaissables. Les objets qui subissent une explosion ne tombent pas gentiment en 4 ou 5 petits polygones : ils sont pulvérisés.

Je suppose que pour le bien de nos estomacs, il faut éviter de montrer à la télévision ou dans des films de fiction que les gens qui meurent de façon violente ne le font pas rapidement et sans bruit mais en hurlant et en étant éparpillé un peu partout aux environs, mais je ne vois pas de raison de prétendre que les explosions ne détruisent pas les choses de façon affreuse. Les bombes bousillent tout, les gens et les vies, elles laissent les bâtiments en piteux état et on a plus envie d'y entrer. Arrêtons de mentir sur les bombes.

Conclusion

Scott Kim me dit que je suis trop sévère en traitant certaines de ces fausses-bonnes-idées de paresse de conception. Il tient à faire remarquer que la conception d'énigmes est un domaine difficile et, à moins d'être familier des vieux jeux, il est facile de se planter et de faire les mêmes erreurs sans le savoir. En plus de ça, beaucoup de gens qui se retrouvent à concevoir des énigmes viennent d'autres domaines comme la programmation ou les graphismes, et n'ont pas d'expérience sur le sujet.

Je veux bien le croire. Mais maintenant que vous avez cette liste pratique vous ne risquez plus de commettre ces erreurs hein ?

[Lire la suite →](#)

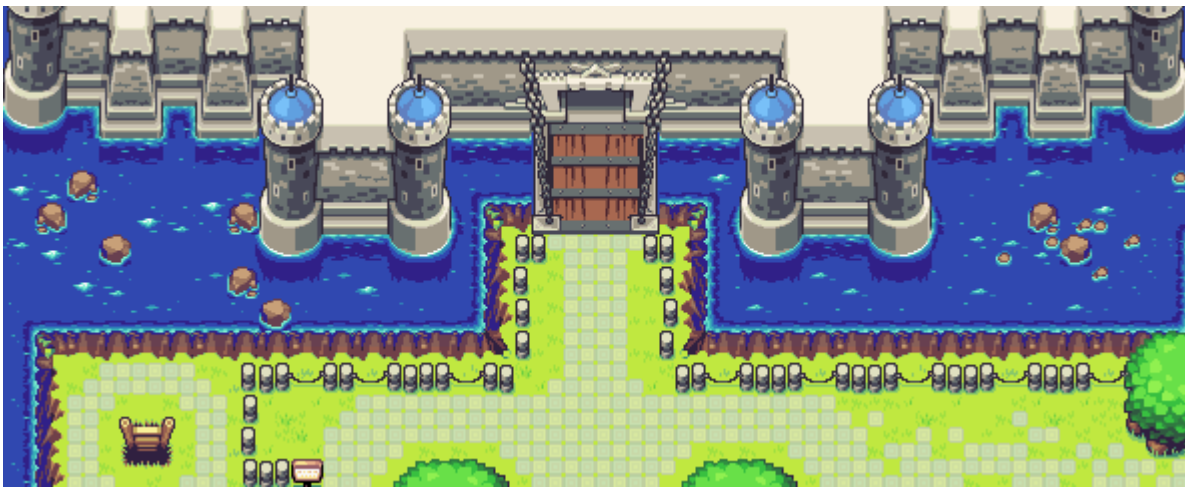
Cours de pixel art 4 : perspectives

Par [Antoine Gersant](#) | le 21 juillet 2008 23:50:13 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Maintenant on va apprendre à dessiner des décors de façon ramenarde. Si si.

La question de la perspective dans les jeux en 2D est une question que personne ne se pose, et c'est sans doute la raison pour laquelle beaucoup de jeux amateurs multiplient les incohérences dans ce domaine. Ce chapitre a pour but d'éclairer un peu votre lanterne sur le sujet, mais il ne traitera cependant que de perspectives RPG-esques. Pour tout ce qui est vue de côté ou vue à la première personne, il faudra aller chercher ailleurs.

Rappelons aux plus distraits d'entre vous que la perspective est « l'art de représenter les objets sur une surface plane (votre écran en l'occurrence) tels qu'ils apparaissent à l'œil de l'observateur. »

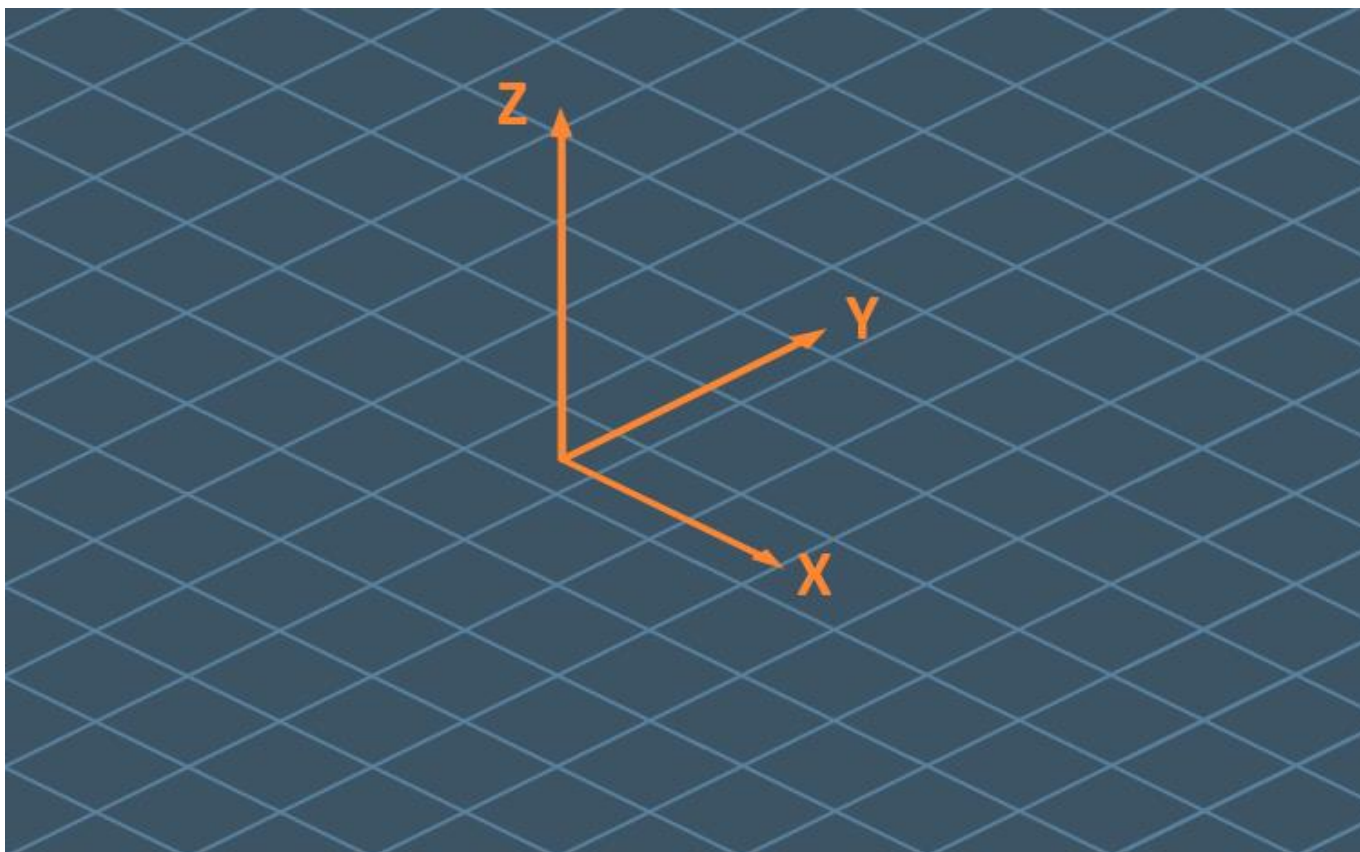


Avant de démarrer, je tiens à remercier Lünnsans qui ce chapitre n'aurait ressemblé à rien, puisqu'il s'agit en fait d'une version remise en page d'une conversation que j'ai eue avec lui sur le sujet.

Perspectives — et pourquoi au pluriel ?

Parce qu'il y en a plusieurs pardi ! Si vous avez déjà lu ou pris quelques cours de dessin, vous aurez probablement entendu parler de perspectives à 1, 2 ou 3 points. Ne vous inquiétez pas, car ce n'est pas de cela dont nous allons parler. Vous pouvez donc laisser tout cela de côté pour le moment =)

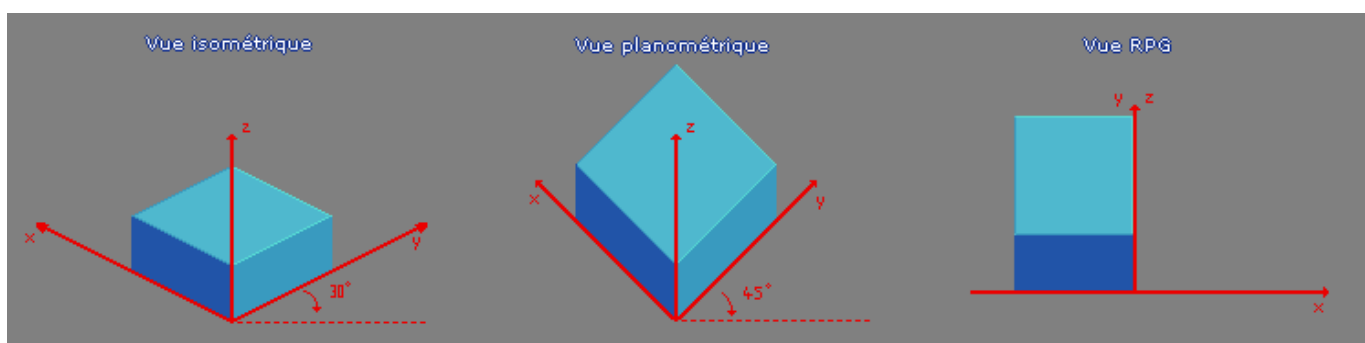
Dans les jeux vidéos en 2D, les perspectives les plus simples à utiliser sont les axonométries. Qu'est ce donc que cette bête là ? Une axonométrie est une représentation de l'espace dans laquelle un objet peut être repéré par 3 coordonnées (X, Y et Z) qui augmentent selon des **droites** et de manière **constante**. Mais une image sera sans doute plus parlante qu'un long discours :



Concrètement, cela signifie que votre perspective est définie par trois axes : Z sera la verticale, alors qu'X et Y seront deux axes qui permettront de « quadriller » le sol. Cela signifie aussi que deux droites parallèles dans la réalité sont représentées par deux droites parallèles en perspective et que la taille d'un objet ne dépend pas de la distance à laquelle il se trouve (les objets plus lointains ne sont pas plus petits, contrairement à une perspective avec un point de fuite par exemple).

La perspective cavalière, la 2D isométrique de jeux comme *Final Fantasy Tactics Advance* ou encore les vues en coupe utilisées pour les plans en sont des exemples.

Enfin, il existe une infinité de perspectives axonométriques, puisque l'artiste est complètement libre de choisir les trois axes comme il le souhaite. Nous allons maintenant nous intéresser aux trois axonométries les plus couramment utilisées. Hop, image.



Celle de gauche est appelée la perspective isométrique : l'axe Y devrait être incliné de 30° pour correspondre à la définition de la vue isométrique mais puisque ce n'est pas possible de manière simple en pixel art, il est représenté par une droite 1 : 2 (cf. [chapitre précédent](#)) et l'angle n'est que de $26,5651^\circ$ (je vous épargne le calcul qui mène à cette conclusion). Cette vue est particulièrement adaptée aux Tactical RPG, car elle permet de représenter de façon convaincante la profondeur et l'altitude. Elle n'est par contre pas adaptée à la plupart des éditeurs de niveaux classiques (genre RPG Maker) en raison de la forme des tiles.

Celle du centre est la perspective planométrique : l'axe Y est cette fois-ci incliné de 45° par rapport à l'horizontal et est une droite 1 : 1. Très peu de jeux utilisent cette perspective pourtant intéressante et riche en volumes, le plus connu d'entre eux étant *Boktai*. Cette perspective gagnerait à être répandue, alors il ne tient qu'à vous de vous en servir.

Enfin, la petite dernière (la moins jolie aussi) est la fameuse « vue de trois-quarts » : l'axe Y a encore tourné un peu vers la gauche jusqu'à être confondu avec l'axe Z. Il s'agit de la vue dans laquelle sont faits la plupart des RPG, et c'est d'elle que nous allons parler plus en détail.

Tout sur la vue de trois-quarts.

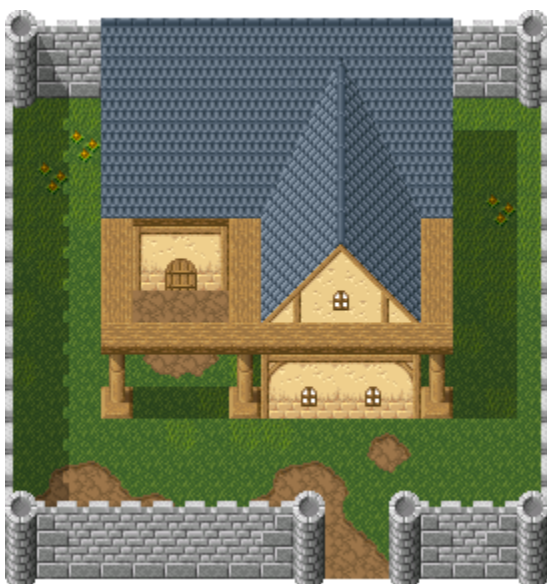
En théorie, je pourrais m'arrêter ici mais des exemples vous aideront à mieux saisir mon propos.



Commençons par un exemple simple : le baril (élément présent dans tout tileset de ville, village, mine abandonnée, maison, etc. qui se respecte). Que remarquer d'intéressant là-dessus ? LE COUVERCLE EST UN CERCLE. Vouï. Pas une ellipse comme on le voit souvent. Il est comme cela pour deux raisons (une seule ne suffirait pas) :

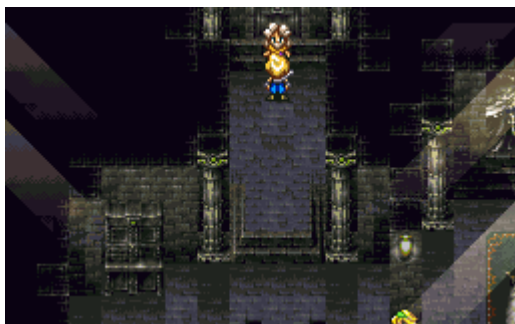
- C'est un cercle dans la réalité du jeu (le couvercle est rond quoi).
- Il est parallèle au sol et donc vu de dessus. De dessus, pas de biais.

Vous retiendrez donc avec application que les plans horizontaux sont représentés comme vus de dessus si l'on veut conserver une perspective de trois-quarts cohérente. Poursuivons avec un exemple un peu plus complet, généreusement fourni par Dyo.



Remarquez avant tout chose que le haut des tourelles est circulaire (comme quoi je ne raconte pas que des conneries). Cela fait, observez la porte, les murs et les fenêtres. Ils sont, ils sont, ils sont (non, ils ne sont pas moches) ? Je vous le donne en mille, ils sont... vus de face ! Vouï, purement de face. C'est encore une fois la magie de nos 3 axes qui est à l'œuvre : les plans verticaux sont représentés comme vus de face si l'on veut conserver une perspective de trois-quarts cohérente.

Mais alors, que se passe-t-il si on veut représenter des plans verticaux ET des plans horizontaux ? Et bien, rien de spécial : l'interaction des deux ne pose pas de problème particulier. Regardez les murets en pierre de Dyo, ou bien l'escalier (ou l'armoire, ou les murs) du superbe screenshot de *Bahamut Lagoon* ci-dessous pour vous en convaincre.



Pour faire court reprenez ça : les plans horizontaux sont vus de dessus, les plans verticaux sont vus de face. Voici donc la marche à suivre si vous souhaitez représenter une perspective axonométrique rigoureuse. Néanmoins, il est tout à fait possible de s'écarter un peu des règles, une fois qu'on les a bien assimilées.

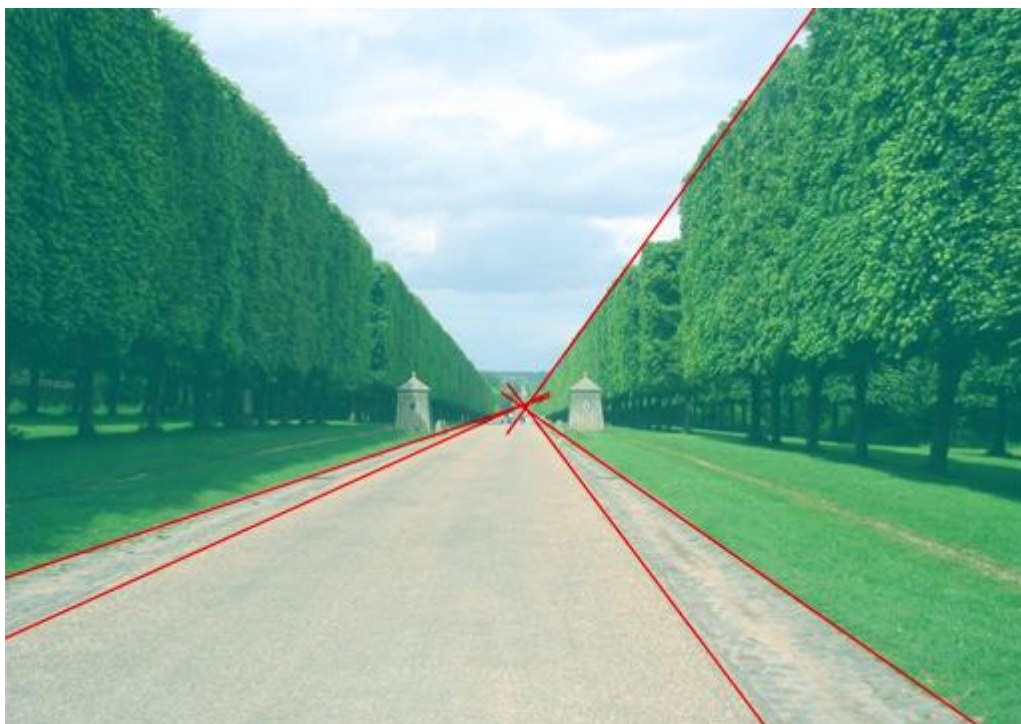
Pour les graphismes de base de RPG Maker et un certain nombre de jeux (y compris pro), les graphistes ont choisi de « tricher » et de représenter les cercles (par exemple le couvercle de notre baril ou l'ouverture d'un puits) par des ellipses. On peut trouver ça plus esthétique, mais il faut veiller ne pas en faire trop : les ellipses ne devraient pas être trop écrasées et elles devraient toutes être aplaties de la même façon (la plupart des jeux s'en tirent avec un rapport largeur sur hauteur d'environ un et demi).

Extension possible : des points de fuite souterrains.

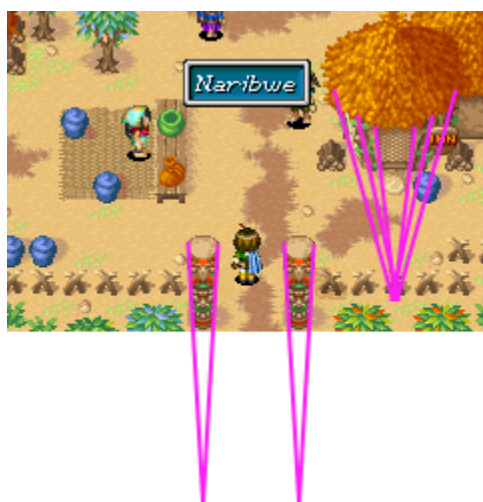
Bon, j'espère que vous avez suivi jusqu'ici parce que les choses vont se compliquer un peu. Je vous ai dit en ouverture de ce chapitre de ne pas penser aux histoires de perspectives à un point (ou deux, ou trois). Et bien nous allons maintenant nous en préoccuper un peu.

Un petit rappel visuel ci-dessous suffira sans doute à vous (re)mettre les idées en place pour ce qui est de la perspective à un point. Comme vous pouvez le voir, il existe un point quelque part (pas nécessairement dans l'image) vers lequel semblent converger toutes les lignes qui sont parallèles à la direction du regard du spectateur (ce seraient les lignes verticales sur une carte de la région, pour peu que la photo soit prise vers le Nord et qu'on tienne la carte à l'endroit) : c'est ce qu'on appelle un point de fuite.

Vous n'avez pas besoin d'en savoir plus pour la suite, mais si le sujet vous intéresse, je vous conseille de lire [cet excellent cours](#) sur la perspective en dessin — attention c'est en anglais et il faut vous inscrire sur le forum de *conceptart.org* pour le lire).



Mais revenons à nos moutons : à quoi vont nous servir ces points de fuite ? Et bien à améliorer notre perspective, pardi ! A la différence de l'image montrant une allée du château de Versailles (l'aviez vous reconnu ?), notre point de fuite va attirer les lignes verticales. Dans l'idéal, on aimerait pouvoir fixer un point de fuite quelque part et dessiner notre jeu par dessus, mais ce n'est pas franchement compatible avec les systèmes de tilesets utilisés par les RPG. La solution est d'attribuer un point de fuite personnel à chaque objet de notre tileset. C'est un peu moins élégant mais le résultat est là ! Voici deux screenshots de *Golden Sun* et *Golden Sun 2* qui sont deux des très rares jeux que je connaisse à utiliser cette technique. Pierre a gentiment mis en évidence quelques points de fuite de ces images.



Remarquez que les points de fuite sont tous situés sous les objets (pour qu'ils soient au dessus, il faudrait que la caméra soit sous le jeu ! un nouveau concept ? =D) et aussi tous à peu près à la même « profondeur » (question de cohérence). Une dernière petite remarque pour la route : plus vos points de fuite se situent en profondeur, plus vous vous rapprochez de la perspective trois-quarts standard (celle décrite au début de ce chapitre), si bien qu'en plaçant votre point de fuite « à l'infini », vous retombez complètement sur la perspective trois-quarts (c'est magique, hein ?).

Maintenant que vous en savez un peu plus sur la perspective, vous n'avez plus d'excuses pour ne pas vous lancer, d'autant plus que les erreurs de perspective ne sont pas trop difficiles à corriger et que vous pouvez tracer des lignes (et des points de fuite si vous tentez la méthode *Golden Sun*) pour vous aider.

[Lire la suite →](#)

Comment aller à l'essentiel et donner aux joueurs ce qu'ils veulent

Par [Antoine Gersant](#) | le 22 juillet 2008 19:14:33 | Catégories : [Game Design](#)

Cet article est une traduction de [Antoine Gersant](#).

L'[article original](#) est écrit par Joshua Dallman publié par Makeitbigingames (septembre 2007).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Il est important dans la conception d'un jeu de tout faire pour donner au joueur ce qu'il est venu chercher : de la satisfaction. Cet article décrit une technique qui vous aidera à centrer votre jeu autour de son concept et à donner aux joueurs ce dont ils ont envie (et aussi à éviter de leur donner ce qu'ils ne veulent pas). Il ne s'agit que d'une méthode, et pas de la vérité ultime sur le sujet, mais essayez la et voyez si elle vous convient.



Bien, vous avez une idée de ce que votre jeu va être, mais vous n'avez pas encore détaillé le concept à 100%. Vous avez parfois un peu l'impression d'être un musicien de jazz qui essaye différents riffs et qui choisit ce qui sonne le mieux. Au bout du compte, tant que que le jeu marche bien (c'est à dire qu'il est amusant), la façon dont vous êtes arrivé là importe peu. Ou peut être que si en fait : Dans les dernières phases de Game Design il faut être structuré pour pouvoir bien balancer le jeu et, contrairement à la peinture abstraite où tout se fait au pif du début à la fin, le Game Design est en bonne partie une affaire de méthodologie. La méthodologie poussée à l'extrême supprime la composante créative du Game Design, mais si l'on en use avec parcimonie, le jeu en bénéficie. Ce qui va suivre est une technique simple que j'emploie pour aller à l'essentiel qui a déjà aidé plusieurs développeurs en plus de moi. Il s'agit d'un nouvel outil pour votre arsenal de concepteur. Commençons sans plus attendre !

NdT : Cet article a été écourté. Referez-vous à l'original pour lire l'exemple final tronqué.

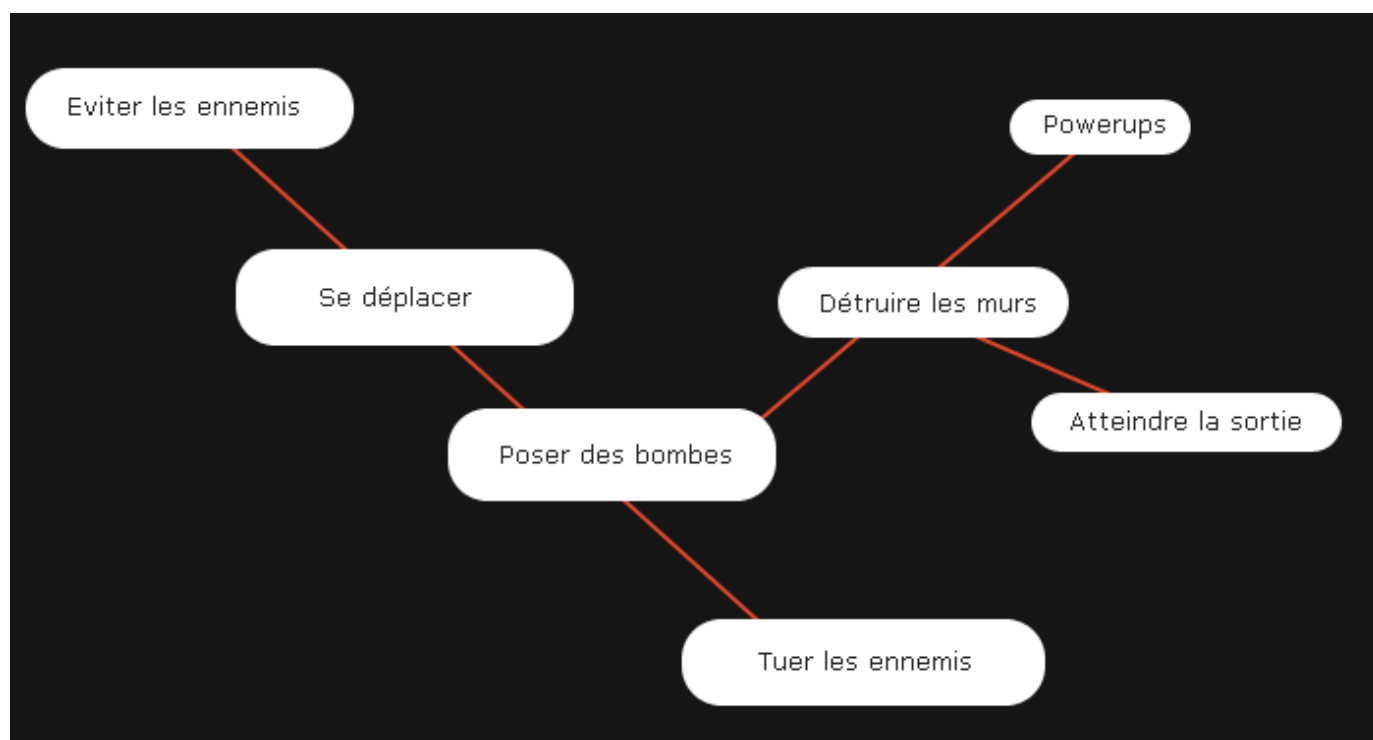
La manie de tout lister

Les Game Designers sont souvent aussi des programmeurs, chez qui les idées tendent à s'organiser de façon linéaire et textuelle. C'est une bonne façon de penser pour écrire du code, mais pas pour concevoir des jeux. La programmation vit dans un monde d'absolus, où x a une valeur précise et où y a une tâche précise. Le Game Design vit lui dans un monde relatif dans lequel les relations entre deux éléments sont aussi importantes que ces deux éléments en eux même.

Prenons par un exemple un jeu du style *Bomberman*. Pour un programmeur, le jeu se résume à une suite de fonctions qui sont des techniques à mettre en œuvre :

- Mouvement du joueur
- Capacité à poser des bombes
- Pouvoir faire sauter des murs
- Pouvoir détruire les ennemis
- Pouvoir éviter les ennemis
- Powerups
- Sortie de niveau

Cette façon de voir les choses est tout à fait pertinente pour programmer le jeu. Mais cette liste logique, linéaire et textuelle ne montre pas du tout les relations qu'entretiennent les différents éléments et leur importance relative. Il y a une meilleure méthode.



Ce que vous voyez ci-dessus est un diagramme visuel qui reprend les mêmes informations (mais ce n'est PAS un flowchart (organigramme de programmation)). Il s'agit d'une liste des fonctionnalités du jeu arrangées d'après leurs relations : ce n'est ni une représentation parfaite ni un diagramme technique, c'est simplement une approximation qui détermine les éléments importants (ou pas) du jeu et qui décrit leurs relations. Il y a moyen de faire un tel arrangement pour chaque jeu (y compris le vôtre) et vous ne pouvez pas vous tromper : on voit tout de suite quand c'est bon. Cela n'aura peut être pas un grand sens logique en terme de positionnement, mais il existe un arrangement qui vous semblera intuitivement être le bon.

Un seul jaune dans mon œuf

La première question à laquelle vous devez répondre est : « Quel est le cœur de votre jeu ? Quel est au fond le sujet de votre jeu ? Quelle est l'idée centrale ? Qu'est ce qu'on y fout ? » Dans le jeu style *Bomberman* ci-dessus, le cœur du jeu c'est de faire sauter des trucs ; vous savez...BOOM : des explosions. C'est ce qu'on fait le plus dans ce jeu, c'est ce qui est marrant dans ce jeu, et c'est ce autour de quoi tourne ce jeu. Dessinez le cœur au centre de votre diagramme et entourez le en gros. Tout le reste de votre concept devrait avoir rapport à ce cœur. Tout devrait y renvoyer et le soutenir.

Si votre cœur est dispersé entre plusieurs petits éléments, vous n'avez pas de cœur. De même, si votre cœur est réparti sur deux éléments parce que vous combinez avec maestria deux choses différentes, vous n'avez pas non plus de cœur. La terre a un noyau, une pomme a un trognon et mon œuf n'a qu'un jaune : un cœur est un cœur parce qu'il est unique. Le cœur en tant que sujet de votre jeu doit être unique, mais il peut par

contre être une idée abstraite comme « faire peur au joueur ». Ce qu'il faut c'est un cœur précisément défini et bien encadré. Vous l'aurez compris, le cœur de *Bomberman* ce sont les bombes.

Une fois votre cœur situé, arrangez les autres éléments du jeu pour que leurs rapports aient du sens. Il n'y a pas de méthode précise pour y arriver : essayez jusqu'à ce que cela vous semble bon.

Est-ce que c'est ÇA l'intérêt ?

Maintenant que vous avez votre diagramme sous les yeux et que vous savez autour de quoi votre concept est centré, posez vous cette question : « Est-ce que le cœur du jeu est aussi l'élément le plus intéressant du jeu ? Est-ce que l'on va faire dans votre jeu est ce qu'on a le plus envie d'y faire ? Y'a-t-il des éléments de divertissements qui émergent spontanément et qui sont plus amusants que ce autour de quoi le jeu est construit ? ».

Soyez honnête avec vous même et faites tester votre jeu, il se peut que vous ayez besoin de réaliser un prototype pour pouvoir répondre à ces questions. Le Game Design consiste aussi à expérimenter, et le divertissement n'aboutit pas toujours là où on a voulu le mettre. Utilisez comme cœur ce qui est divertissant plutôt que de forcer un autre élément à devenir divertissant, et votre jeu en sortira sûrement meilleur.

Soutenir le cœur

Maintenant vous avez un diagramme avec un cœur, et le cœur est aussi la partie la plus intéressante du jeu. La question suivante est : « Soutenez vous le cœur du projet ? » Tout d'abord, commencez par imaginer des moyens de soutenir le cœur. Remarquez que dans *Bomberman*, c'est l'action de placer des bombes qui a le plus de conséquences : elle pousse le joueur à se déplacer (pour éviter les explosions), elle peut briser des murs (ce qui conduit à des powerups ou à la sortie), elle peut éliminer les adversaires (la meilleure récompense possible). On aurait pu concevoir le jeu de façon à soutenir le déplacement du joueur en lui faisant briser les murs en courant à travers et en faisant mourir les ennemis de la même façon (moyennant un powerup qui permet de les vaincre, comme dans *Pac-Man*). Cependant, agir ainsi n'aurait pas soutenu le cœur du jeu qui réside dans les bombes.

Dans *Pac-Man*, ce qui est marrant c'est de manger les petits palets. Chomp chomp ! Du coup ils ont rajouté des palets qui donnent des capacités spéciales, et ensuite ils ont ajouté une capacité qui transforme les fantômes ennemis en palets ambulants. C'est un soutien très bien trouvé !

Si on pouvait avoir un deuxième bouton pour jouer à *Bomberman*, il faudrait permettre une autre façon d'utiliser les bombes. Il ne faudrait pas un bouton « courir » et encore moins un bouton pour utiliser quelque chose qui n'a rien à voir comme un pistolet. Un bouton courir soutiendrait le mouvement du joueur (ce qui n'est pas le cœur du jeu), tandis qu'un pistolet serait complètement isolé sur nos diagrammes et seulement en relation avec « éliminer les ennemis ». Un deuxième bouton pour utiliser des bombes pourrait permettre de les faire rouler jusqu'au mur le plus proche. Cela ajouterait un peu de profondeur au gameplay tout en soutenant le concept de bombe. D'ailleurs le titre du jeu est *Bomberman* et pas *CoursEtPoseDesBomberman*.

Pensez à d'autres façons de soutenir le cœur.

Pour *Bomberman*, cela pourrait signifier des bombes de différents types comme par exemple des bombes IEM qui paralyseraient les robots ennemis sans blesser le joueur, ou alors d'autres façons de lancer les bombes, comme par exemple un bouton pour les jeter par dessus les murs plutôt que de les poser devant soi. Du côté des graphismes, les explosions avaient intérêt à être réussies : un ou deux ennemis mal foutus auraient pu passer mais les bombes étant le cœur, c'est elles qu'il fallait soigner.

En bref pensez à différentes façons créatives de soutenir votre cœur, ensuite pensez à soutenir les plus intéressants de ces soutiens etc. ...c'est la partie amusante du Game Design : soyez créatifs et flexibles. A vous de voir si vous le faites détaillé ou seulement dans les grandes lignes, essayez un peu tout et amusez vous. =)

Dégraissage

Maintenant que vous avez planifié plein de fonctionnalités que vous pourriez inclure, il est temps de faire le ménage pour aboutir à un concept propre, élégant, centré et surtout fai-sa-ble. De toute façon un jeu avec trop fonctionnalités est comme un plat avec trop d'ingrédients. Il devient de plus en plus difficile de régler la balance à chaque nouvelle fonctionnalité et le résultat est souvent trop ambitieux, dilué ou bordélique. En plus de cela, le concept du brainstorming consistait à réunir plein d'idées pour ne garder ensuite que les meilleurs : le moment de faire le tri est arrivé.

Regardez votre diagramme et déterminez comment vos idées renvoient les unes aux autres. Idéalement, il faudrait que les éléments soient liés autant que possible, comme une toile d'araignée ou une tapisserie, plutôt que d'avoir quelques groupes isolés qui n'auraient plus de sens une fois reliés.

Il y a beaucoup de raisons qui peuvent pousser à supprimer une fonctionnalité et il se peut qu'il faille réaliser des prototypes pour déterminer le véritable intérêt d'une idée. Parfois les fonctionnalités à garder sont déterminées par le budget (un moteur physique par exemple coûte cher mais n'apporte pas toujours grand chose par rapport à de fausses relations physiques). Ne soyez pas trop attaché à vos fonctionnalités et montrez vous sans pitié avec elles : si une fonctionnalité n'en vaut pas vraiment la peine ou ne soutient pas assez le cœur (voire pire, détourne le cœur ou est orpheline sur le diagramme), envisagez de vous en débarrasser.

Aider et pénaliser le joueur

A mesure que vous modifiez le concept, en enlevant et en ajoutant des fonctionnalités, gardez à l'esprit quels éléments sont là pour aider et pénaliser le joueur, afin de maintenir un certain équilibre. Par exemple, si votre cœur consiste à poser des bombes, dans ce cas les aides doivent aider à placer des bombes : imaginez des tonneaux qui explosent aussi et provoquent des réactions en chaîne, des powerups qui augmentent la puissance de vos bombes, des murs qui s'écroulent tout seul à proximité de vos bombes pour révéler l'existence de powerups ou la sortie. Équilibrez les aides et les pénalités pour que le jeu ne soit ni trivial ni trop difficile ou sec. Si les joueurs ont du mal avec certains passages, essayez de les modifier avant de les supprimer : il leur manque peut-être juste un petit coup de pouce.

Utiliser des diagrammes pour centrer votre concept est une idée d'une simplicité désarmante mais qui peut mener à des résultats impressionnants. Au moins, cela vous force à réfléchir à votre concept et le voir d'une façon dont vous n'avez pas l'habitude. En vous interrogeant et en bricolant votre concept, vous ferez un meilleur jeu.

Cours de pixel art 5 : ombre et lumière

Par [Antoine Gersant](#) | le 23 juillet 2008 19:29:10 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Aujourd'hui on passe de la 2D à la 3D tout en restant en 2D. Si. Vous allez voir (ou pas).

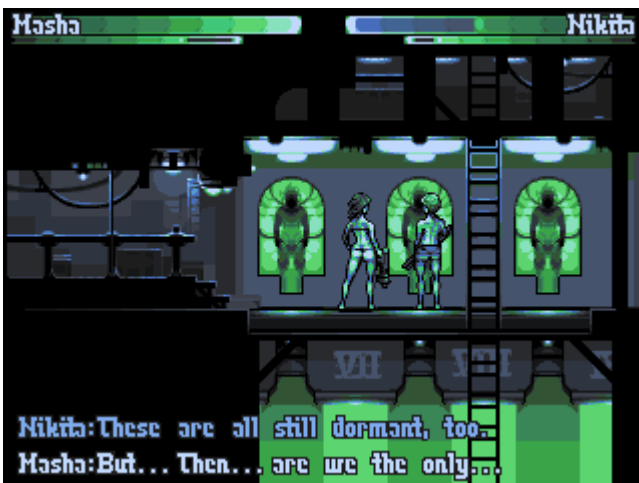
Avec ce dont nous avons parlé jusqu'ici, vous devriez être capable de faire de jolis linearts et de dessiner des décors avec une perspective correcte. C'est un bon départ mais vous vous doutez bien que ça ne suffit pas pour faire un pixel art.



Dans ce chapitre, nous allons nous occuper de shading. Il s'agit d'un ensemble de techniques qui vont nous permettre de sculpter nos décors et nos personnages pour représenter des reliefs ou des formes, bref des volumes (et je vous épargne mon traditionnel laïus sur l'importance capitale de ce chapitre, *c'est juste super important*).

Aujourd'hui on passe de la 2D à la 3D tout en restant en 2D. Si si. Vous allez voir (ou pas).

Avec ce dont nous avons parlé jusqu'ici, vous devriez être capable de faire de jolis linearts et de dessiner des décors avec une perspective correcte. C'est un bon départ mais vous vous doutez bien que ça ne suffit pas pour faire du pixel art.



Dans ce chapitre, nous allons nous occuper de *shading*. Il s'agit d'un ensemble de techniques qui vont nous permettre de sculpter nos décors et nos personnages pour représenter des reliefs ou des formes, bref des volumes (et je vous épargne mon traditionnel laïus sur l'importance capitale de ce chapitre, **c'est juste super important**).

Ça sort d'où ?

En fait, vous connaissez sans doute déjà la réponse à cette question. Ce qui fait que votre cerveau arrive à interpréter le volume des objets autour de vous, ce sont les variations de couleurs sur sa surface, qui résultent des différences d'éclairage entre les parties de votre objet. Pas besoin d'un projecteur halogène pour ça, le moindre petit reflet définit un volume : regardez autour de vous !

Ce croquis illustre le concept de manière simple : un objet (une sphère) est éclairé par une source de lumière (symbolisée par la flèche) et cela modifie ses couleurs. Les couleurs sont plus claires là où la sphère est directement exposée à la lumière et plus sombres dans les zones à l'ombre.

Remarquez que je ne parle ici que de couleurs claires et sombres, le prochain chapitre vous en dira plus sur la façon de les choisir précisément.



Ce qu'il est important de retenir c'est que j'ai choisi une source de lumière et que j'ai placé les zones d'ombre et de lumière (les highlights) en fonction de cette dernière. Si mon dessin avait été plus compliqué, j'aurais dû faire attention à conserver ma source de lumière sur l'ensemble pour que le tout soit cohérent. Bien entendu, vous n'êtes pas obligé de placer la source de lumière en haut à droite, vous pouvez la mettre n'importe où.

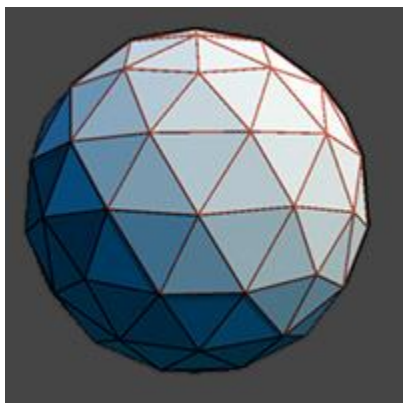
Malheureusement, les choses ne sont pas toujours aussi simples que pour cette sphère : d'abord parce que les objets peuvent être dans l'ombre les uns des autres, ensuite parce que plus vos objets ont des formes complexes, plus il est difficile de faire apparaître leurs volumes avec précision (surtout en pixel art) et enfin parce que la lumière a une fâcheuse tendance à rebondir (!) sur les objets, les murs et les sols. Au final, le bas de la sphère devrait un peu plus ressembler à ceci, parce que l'environnement a lui aussi tendance à renvoyer de la lumière sur la sphère :



Et moi je fais comment ?

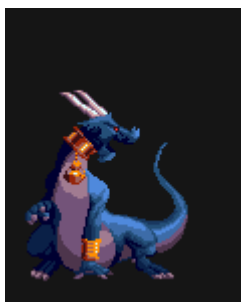
Vous ? Hah ben vous faites pareil mais en pixel art, quelle question ! Mais comme je suis sympa, je vais quand même vous aider un peu. La première chose à faire est de choisir la position de votre source de lumière (ce sera le plus souvent en haut à droite/en haut à gauche puisque votre source de lumière sera la plupart du temps le soleil).

A partir de là, il faut réussir à faire preuve d'un peu d'abstraction : vous devez penser votre objet en 3 dimensions (et pas simplement comme un tracé sur votre écran) pour réussir à déterminer les zones qui seront « frappées » par la lumière (un peu / beaucoup / passionnément / à la folie / pas du tout) et les mettre en couleur en tenant compte de cela. Pour simplifier un peu le problème vous pouvez penser comme une bonne vieille Playstation (merci à Poupi pour l'image de sphère en 3D à ci-dessous) et découper dans votre esprit l'objet en différents polygones et réfléchir à l'éclairage de chacun d'eux. Comme ceci :



D'une manière générale, il faut commencer par établir les zones qui seront « globalement dans l'ombre » et celles qui seront « globalement éclairées » plutôt que de s'attaquer directement aux détails (mauvaise idée !). Il serait bon qu'au long du processus les contours noirs de votre lineart soient peu à peu remplacés par des couleurs utiles, ça vous fera de la place en plus pour ajouter des détails (un pixel de plus, ça n'a pas de prix !) et ça vous évite de les enlever à la fin sans trop savoir quoi en faire.

Pour l'exemple, revoilà le dragon du chapitre 1 (et vous allez encore le revoir plus tard). Je lui ai appliqué les techniques de shading décrites dans ce chapitre pour lui donner du volume (attention, il n'est pas fini pour autant !). Ma source de lumière étant située à droite et un peu en hauteur, toute la partie à gauche du dragon est dans l'ombre, sauf sa main qui est plus proche de vous et qui s'extrait de la zone d'ombre créée par le corps. Pas grand chose d'autre à signaler : j'ai un peu amélioré le lineart à l'occasion mais rien de très important.

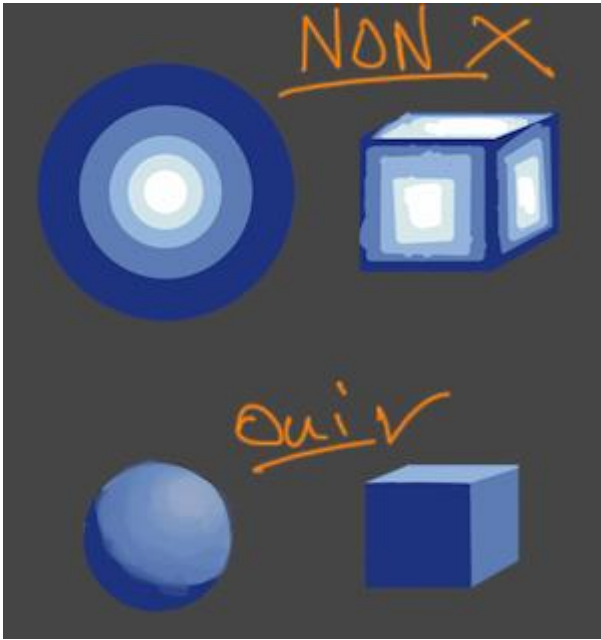


Deux erreurs à éviter

Le pillow-shading

Il n'y a qu'une seule excuse valable pour faire du pillow-shading : n'avoir jamais lu toutes ces théories sur les ombres et les lumières. C'est la démarche employée par les gens qui ont remarqué sur d'autres dessins qu'il faut utiliser des couleurs claires et des couleurs sombres, mais qui n'ont pas vraiment compris pourquoi ni

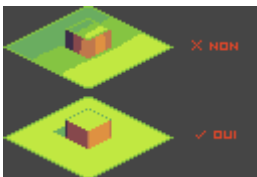
comment les placer. D'instinct, ces gens se mettent à placer les couleurs claires au milieu des surfaces et les couleurs sombres sur les bords. Le résultat est totalement incohérent et AFFREUX.



Le désastre saute aux yeux sur des formes simples comme une sphère ou un cube, mais faites bien attention à ce que vous faites sur vos pixel arts. Si vous n'êtes pas habitué à placer des ombres, vous risquez d'avoir une tendance naturelle à faire ainsi lorsque votre attention baisse.

Comprendre sans avoir compris.

La seconde erreur à éviter concerne les gens qui ont lu ce genre d'articles (donc vous par exemple). Le raisonnement qui conduit à commettre cette erreur est le suivant : « Bon très bien je place ma source de lumière : en bas à droite. Ça c'est fait. Maintenant la couleur, bon ben, plus clair en bas à droite et plus foncé en haut à gauche et tout ira pour le mieux ».



Et non, mauvaise pioche ! En faisant ça le résultat est désastreux et dénué de volume. Pourquoi ? Parce qu'en plaçant ainsi les couleurs, on pense en deux dimensions et on ne tient pas compte de l'éclairage réel des choses. Il ne faut pas vouloir rajouter à tout prix des ombres et des lumières partout et très souvent les surfaces planes sont éclairées de façon uniforme. Ne cherchez pas à tout prix à replacer votre gamme complète d'ombres et de lumières dans chaque zone de l'image.

Pour se la péter dans les dîners mondains : l'ambient lighting.

Terminons notre rapide tour d'horizon avec une technique « qui fait genre ».

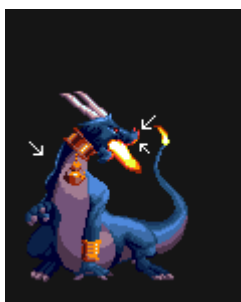


Dorénavant, vous êtes censé être capable de gérer une source de lumière. Bien. L'*ambient lighting* consiste à ajouter une seconde source de lumière le plus souvent colorée près de votre objet (ou de votre personnage). Il est préférable que cette seconde source de lumière ne soit pas placée dans la même direction que la première, pour deux raisons :

- Elle serait « noyée » dans la première source, et ne serait pas franchement visible
- Tout le charme de la seconde source de lumière vient du fait qu'elle éclaire les ombres et donne ainsi caractère extrêmement dramatique à la scène.

Attention toutefois, « éclairer les ombres » ne veut pas dire que les ombres sont plus claires (oui je sais, on aurait pu croire). Cela veut dire que l'on se retrouve avec des highlights « bien violents » au bord des zones les plus sombres et le cerveau humain appréciant les contrastes (si si), il a tendance à apprécier le résultat.

Voilà ce que ça donne sur la gueule de notre dragon, éclairée par les flammes (rawr rawr !), et sur son dos éclairé par une mystérieuse lueur bleue. Bien entendu, cette technique — comme toutes les techniques « qui font genre » est à utiliser avec parcimonie. Par ailleurs, évitez de vous jeter dessus dès vos premiers essais : commencez par maîtriser l'utilisation d'une seule source de lumière avant de compliquer les choses.



Enfin, un petit exemple de ce que la technique permet de réaliser à plus grande échelle : voici un screenshot de *Tales of Phantasia* sur Super Nintendo, dans lequel les murailles et les colonnes éclairées par les torches fournissent une belle démonstration d'*ambient lighting* (la source de lumière « principale » étant le ciel).



Et voilà, nous sommes déjà (enfin « déjà » c'est pour vous, moi ça fait des plombes que je suis dessus) au bout de ce chapitre. Comme pour les précédents, entraînez vous et relisez le si tout ne vous paraît pas clair. Je ne suis pas rentré dans le détail pour que ce tutorial reste centré sur le pixel art et pas sur l'art en général.

Si la lumière vous intéresse et si vous savez parler anglais (ou si ne regarder que les images ne vous dérange pas), vous pouvez [approfondir le sujet](#).

[Lire la suite →](#)

Cours de pixel art 6 : couleurs et palettes

Par [Antoine Gersant](#) | le 23 juillet 2008 22:49:35 | Catégories : [Couleur](#), [Cours](#), [Graphisme](#), [Pixel art](#)

Et non, on ne dessine plus rien ! Heh, revenez !



Pourquoi et quand choisir une palette ?

Bonne question, après tout pourquoi je vous bassine avec ça ? Tout simplement parce que la palette, c'est 50% de la qualité de votre travail (pourcentage réalisé avec pourcentage-maker). Au fil d'un jeu, il est important d'avoir des palettes cohérentes car elles contribuent grandement à installer une ambiance. En règle générale, ce qui fait la différence entre des graphismes beaux et des graphismes ratés, c'est en bonne partie la palette.

« Bon d'accord c'est important d'avoir une jolie palette, je ferai ça à la fin. »

Tututut, surtout pas ! La palette se travaille du début à la fin de la conception d'un pixel art. En ce qui me concerne, c'est la première et la dernière chose que je fais : je commence par déterminer une palette de départ (dans un coin de mon image) que j'arrange au fur et à mesure que j'avance, puis, une fois que tout est fini, je passe un bon moment à l'améliorer. Les améliorations étant essentiellement dues au « feeling », ce que vous allez apprendre ici peut être remis en cause : c'est juste un point de départ.

La magie du système TSL (HSL)

Présentation

En informatique, toutes les couleurs sont représentées par un code hexadécimal (c'est à dire qui utilise les caractères 0 1 2 3 4 5 6 7 8 9 A B C D E F) de 6 caractères. Par exemple, le code `#000000` correspond au noir, le `#FFFFFF` au blanc et le `#FF0000` au rouge pur. Rien de très compliqué jusqu'ici.

Ce code peut être divisé en trois segments de deux chiffres (le # sert juste à décorer) : les deux premiers chiffres correspondent à la composante rouge de la couleur, les deux suivants au vert, et les trois derniers au bleu. En combinant les 3 valeurs, on obtient n'importe quelle couleur. Par exemple mon rouge pur de tout à l'heure est `#FF0000` car sa composante rouge (`FF`) est maximale et les deux autres nulles (`00`).

Les codes hexadécimaux peuvent être convertis en valeurs plus courantes pour faciliter la discussion (tout le monde ne lit pas l'hexadécimal comme vous et moi) : chaque doublet compris entre `00` et `FF` est en fait un nombre entre 0 et 255.

Imaginons maintenant que j'ai choisi une couleur que je vais utiliser dans mon pixel art, par exemple le violet `#6A146A` (Rouge 106 — Vert 20 — Bleu 106). Bon, cette couleur est pas mal mais elle ne me convient pas tout à fait...j'aimerais bien quelque chose de plus vif et plus intense. Je me dis donc que pour la rendre plus vive, j'ai juste à augmenter les 3 composantes...pfff tu parles, ça ne fait que la rendre plus blanche et pâle. Tant pis, je vais l'assombrir un peu alors, je vais diminuer chaque composante de

10...encore raté ! J'ai assombri ma couleur mais je la fais tendre vers le gris...aïe aïe. Que c'est compliqué de choisir des couleurs avec précision, laissons tomber les palettes. Et puis Zorro est arrivé. Zorro, c'est le système TSL (HSL).

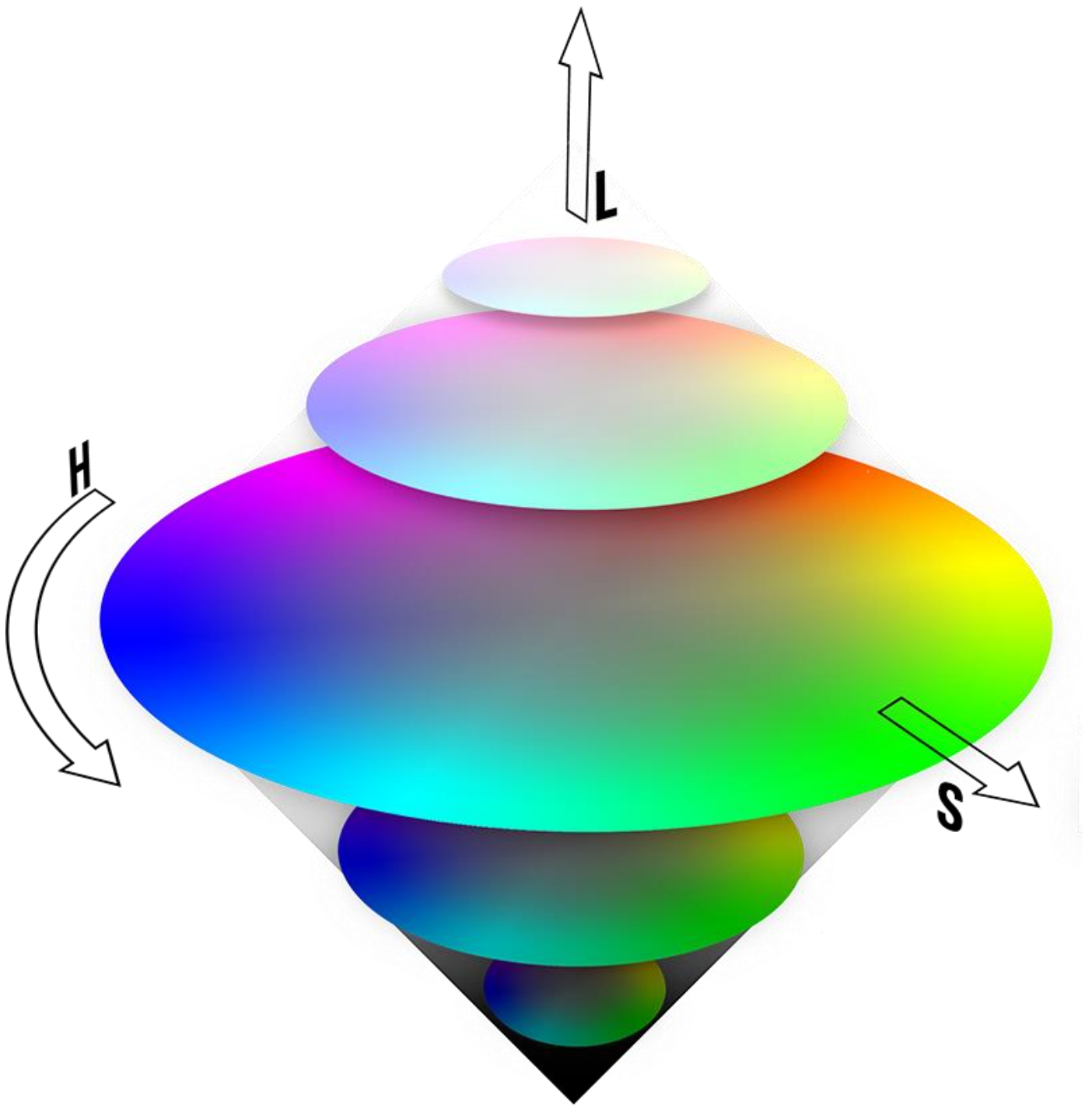
Vous l'avez surement compris, ce n'est pas naturel pour votre cerveau ni pour le mien de décomposer une couleur en trois composantes RVB (Rouge Vert Bleu). Le système TSL c'est tout simplement une autre façon d'identifier une couleur : on n'utilise plus les composantes RVB mais trois composantes visibles et reconnaissables à l'œil nu, à savoir la Teinte, la Saturation et la Luminosité.

La teinte, comprise entre 0 et 360, c'est tout simplement la couleur en elle même : rouge (0), jaune(60), vert(120), cyan(180), bleu(240), rose(300) et rouge (360). Si vous êtes observateur, vous aurez remarqué que les deux extrêmes se rejoignent (comme en politique). Bien évidemment, toutes les teintes intermédiaires, entre les couleurs que j'ai citées, existent.

La saturation, comprise entre 0 et 100, est la vivacité de la couleur. Plus une couleur est saturée, plus elle est intense. Une saturation de 0 correspond à un gris.

Enfin la luminosité, comprise entre 0 et 100, est la plus simple à comprendre : elle détermine si la couleur tend vers le blanc ou le noir.

Voici pour fixer vos idées une jolie représentation de l'espace colorimétrique HSL, généreusement fournie par [Alexandre Van de Sande](#) :

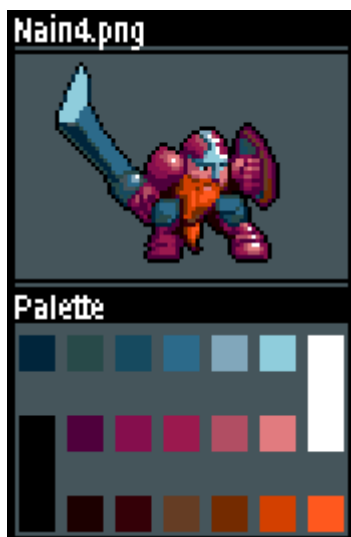


Tous les logiciels de pixel art présentés dans le chapitre 1 vous permettent de choisir vos couleurs avec le système TSL, donc vous n'avez aucune raison de ne pas l'utiliser. Avec un peu d'entraînement, vous pourrez très facilement obtenir une couleur à laquelle vous « pensez », tandis que cela aurait pris des heures en utilisant le système RVB.

Les rampes et les glissés de teinte

Maintenant que vous connaissez les outils à votre disposition pour choisir une couleur, entrons dans le vif du sujet : la constitution d'une palette. Une palette est constituée de plusieurs "rampes" : une rampe c'est un groupe de couleurs dont les teintes sont voisines.

Par exemple, une palette peut contenir une rampe de rouges, une rampe de verts et une rampe de marrons. J'ai déjà pas mal blablâté dans ce cours alors allons y pour un exemple.



Voici un pixel art de nain barbu (pas terrible mais ça fera l'affaire pour illustrer) et en dessous, la palette qui a servi à le réaliser. Cette palette contient trois rampes : une bleue, une rouge et une orange. Petite note en passant, il n'est pas indispensable de mettre en scène votre palette comme je l'ai fait. En général c'est plus bordélique que ça.

Sur chaque rampe, j'ai classé les couleurs de la plus foncée (à gauche) à la plus claire (à droite). Remarquez que le blanc pur et le noir pur (les grands rectangles) font partie de deux de mes rampes : étant donné que ces deux couleurs ont une saturation de 0 (le blanc et le noir sont des gris purs), elles peuvent appartenir à n'importe quelle rampe, et c'est le cas de tous les autres gris. Les plus vieux d'entre vous ont peut être déjà vu des images de jeux sur C64 : la palette de couleurs affichables sur cette machine contenait de nombreux gris pour permettre aux artistes de créer des rampes très diverses.

Passons à l'essentiel et intéressons nous au concept **fondamental** du glissé de teinte (hue-shift). On a souvent tendance à croire que dans une rampe, toutes les couleurs ont la même teinte et la même saturation et que seule la luminosité change (pour avoir des couleurs foncées et claires). Et bien en fait non !

Ce que je ne vous ai pas dit dans le chapitre précédent, c'est que toutes les lumières qui éclairent vos objets sont colorées, soit par le bleu du ciel, soit parce qu'il s'agit du jaune d'une ampoule soit par autre chose encore (l'article proposé en approfondissement du chapitre précédent parle bien de ce sujet). Ainsi, le long d'une rampe, la teinte et la saturation évoluent. Le plus souvent la réalité des choses est très (trop) complexe et demande des connaissances sur l'éclairage que je n'ai pas évoquées (et que je maîtrise assez mal), mais en pixel art on peut se contenter des règles suivantes :

- Plus on va vers le foncé, plus on augmente la saturation
- Plus on va vers le clair, plus on diminue la saturation
- Plus on est dans le foncé, plus la teinte va vers le bleu ou le violet
- Plus on est dans le clair, plus la teinte va vers le jaune

Si l'on regarde en détail la rampe rouge de la palette du nain, on retrouve ces différentes tendances :

Luminosité	0	31	52	61	69	88	100
Teinte	-	314	328	336	347	358	-
Saturation	0	100	89	84	56	45	0

Les chiffres parlent d'eux-mêmes, vous pouvez vérifier que tout concorde avec les principes énoncés ci-dessus. A noter, les règles concernant la saturation peuvent être inversées, on obtient alors un rendu plus doux.

Terminons avec un petit exemple de nos amis les professionnels : le très classique « arbre de *Seiken Densetsu 3* » dont la palette contient un énorme glissé de teintes du jaune au violet. L'image ci-dessous ne montre pas la palette entière, seulement la couleur la plus claire et les deux plus foncées.



Methode de Xenodrogen

La méthode de Xenodrogen, du nom de son inventeur, est une méthode permettant de choisir avec exactitude les couleurs de votre rampe et sans (trop de) risque vous tromper. Cette méthode n'est pas une règle absolue à respecter mais si vous avez des "gouts de toilettes" en matière de couleur, elle vous permettra de faire de jolies rampes.

Cette méthode repose sur une règle compliquée à formuler mais simple à respecter :

« Au sein d'une rampe, la teinte, la saturation et la luminosité varient en sens unique. Leurs variations (dérivée seconde, pour les matheux) varient elles aussi en sens unique. »

Nous avons déjà parlé de la première phrase, et nous avons même dit dans quel sens devaient varier teinte, luminosité et saturation les uns en fonction des autres.

La seconde phrase va nous permettre de quantifier ces variations : dans une rampe, chacune des composantes (teinte, saturation et luminosité) doit varier soit de plus en plus, soit de moins en moins d'un bout à l'autre de la rampe. Imaginons une rampe de 5 couleurs que l'on va appeler A,B,C,D et E. Je peux par exemple faire +1 en saturation entre A et B, puis +5 entre B et C, encore +5 entre C et D et enfin +12 entre D et E. Ce serait par contre incorrect de faire quelque chose comme +2 +3 +2 +1 +4. Pour résumer, on augmente ou diminue la composante de plus en plus ou de moins en moins mais sans faire de "yoyo".

Allons y pour un petit tableau d'exemple, avec une rampe de verts :

Teinte	-1	-4	-4	-8	
Saturation	+10	+10	+9	+5	
Luminosité	+6	+9	+9	+12	

Ici, la teinte diminue de plus en plus, la saturation augmente de moins en moins, et la luminosité augmente de plus en plus.

Remarquez en passant que comme annoncé dans le paragraphe précédent, on peut inverser la relation entre luminosité et saturation pour avoir un rendu doux : sur cette rampe, ce sont les couleurs les plus claires qui sont les plus saturées.

Un dernier détail sur cette méthode de Xenodrogen : plus vous avez de couleurs dans votre rampe, plus vous pouvez faire de petites variations. La rampe de vert du tableau a 5 couleurs donc on a fait de petites variations. Si on avait eu que trois couleurs, on aurait eu des variations de l'ordre de 15–20 unités.

Le bon goût

Noir et blanc

Le problème de l'utilisation du noir et du blanc mérite bien un paragraphe à lui tout seul. Généralement, l'usage de ces deux couleurs à l'état pur (`#000000` et `#FFFFFF`) est déconseillé. Cependant, il existe deux exceptions (une pour chaque couleur) et le nain présenté au début de ce cours est concerné par ces deux exceptions. On peut utiliser du blanc pur (`#FFFFFF`) sur des surfaces très lumineuses (effets de magies et sortilèges) ou très réfléchissantes (métal, pierres précieuses). C'est le cas du nain et de son armure en métal.

Le cas du noir pur (`#000000`) est plus subtil : il ne doit servir qu'à faire des contours de sprites et ce pour plusieurs raisons.

- Il ne peut pas servir à faire les ombres car dans la réalité rien n'est jamais parfaitement noir, il y a toujours une teinte très discrète (le plus souvent du violet).
- Il ne doit pas servir à faire des contours dans des décors. Un objet entouré d'un contour noir est particulièrement visible et si l'on veut faire un jeu lisible, ce sont les personnages qui doivent être mis en valeur et pas le décor.
- On peut donc se servir du noir pour faire des contours de sprites, et c'est le cas du nain. Cependant, il est fortement déconseillé de faire des concours noirs "intérieurs" : le noir sépare le sprite du décor mais s'en servir pour séparer les différentes parties du sprite entre elles serait une perte de place et rendrait le sprite brouillon.

Pièges à éviter

Les débutants ont deux tendances à écarter dès que possible, à savoir :

- Utiliser des couleurs trop saturées.
- Utiliser des couleurs trop typées. Rappelez vous que l'herbe n'est pas toujours verte, que l'eau et le ciel ne sont pas toujours bleus et qu'un personnage est plus couleur pêche que rose bonbon (par exemple). D'une manière générale, l'heure de la journée et la lumière ambiante modifient **considérablement** les palettes de couleurs, regardez autour de vous et sachez en tenir compte.

J'ai décidé de conclure ce cours comme je l'ai commencé, et de vous rappeler que les règles exposées ici ne sont en aucun cas des impératifs mais de simples conseils. Il existe de très nombreux moyens de faire de jolies couleurs : ce chapitre vous en a donné un mais le plus important reste... le bon gout et la pratique. Aussi, si vous pensez pouvoir améliorer votre palette en outrepassant ces règles, n'hésitez pas à le faire. Au fond c'est le résultat qui compte ;)

[Lire la suite →](#)

Rendons sa magie à la magie

Par [Antoine Gersant](#) | le 25 juillet 2008 23:39:24 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Antoine Gersant](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mars 1999).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



Si vous avez déjà joué à un vieux jeu sur ordinateur, un jeu vraiment vieux (sur un macro-ordinateur relié à un terminal d'impression), vous vous souvenez sûrement que ce qui en sortait ressemblait à ceci :

« Nous sommes en l'an 3 de votre règne. La population compte 1937 citoyens. Au cours de l'année écoulée, 24 citoyens sont morts et 92 citoyens sont nés. Vous avez 178 acres de terre cultivée et 241 sacs de grain dans les greniers.

Combien de sacs voulez vous semer cette année ? »

Ce qui saute aux yeux dans cet exemple c'est qu'il est plein de chiffres. Les ordinateurs ne manipulent que des chiffres, bien sûr, et c'est pour cela que tous les jeux vidéos sont nécessairement des modèles mathématiques. Dans les premiers jeux sur ordinateur, le modèle mathématique était proche de la surface : puisqu'il n'y avait pas de graphismes, les concepteurs ne pouvaient pas faire mieux que de vous renvoyer des nombres avec des petits bouts de texte au milieu pour les relier.

Ce n'est aujourd'hui plus le cas et au fil des années nous avons appris à cacher ou à déguiser les nombres, notamment en empruntant les techniques d'affichage de statistiques : nous utilisons des barres, des voyants colorés, des panneaux de contrôle, et tout un tas d'autres mécanismes pour présenter des données numériques sans afficher aucun nombre à l'écran. C'est parfois même fait de façon très intelligente, en utilisant un moyen d'afficher l'information direct et en rapport avec le jeu. Dans *Doom* par exemple, il y avait un portrait de votre personnage en bas de l'écran pour vous représenter. Quand vous vous faisiez toucher, le visage devenait de plus en plus ensanglanté, il piquait du nez et avait l'air de plus en plus épuisé. Quand vous vous soigniez, vous aviez l'air moins répugnant et plus alerte. Le portrait affichait aussi d'autres informations ; quand vous ramassiez des armes, il souriait comme un dément pendant une seconde ou deux. C'était un mécanisme intelligent et bien conçu pour afficher différentes variables sur un petit espace. En plus de ça, il y avait dans *Doom* une jauge de vie qui indiquait de manière chiffrée la vie qu'il vous restait. Étant donné qu'on pouvait se renseigner sur le portrait, le compteur était redondant et je pense que c'était une erreur.

A chaque fois que l'on affiche un nombre « brut » à l'écran, on se raccroche à notre héritage de terminal d'impression. Si votre monde de fantaisie fait vraiment intervenir des nombres, comme dans une simulation de business ou de conduite d'un véhicule militaire moderne, pas de problème. Mais si vous incarnez un space-marine ou un chevalier errant, vous ne devriez pas voir des nombres à tous les carrefours. Aucun chevalier n'a jamais regardé son armure pour conclure « Ouai, celle là peut encore recevoir exactement 37 coups ».

En plus de cela, il y a des variables de jeu dont la valeur devrait être imprécise. L'essence et les munitions peuvent et doivent avoir des valeurs précises, mais pas la santé et la résistance d'une armure. Je suis bien conscient que personne n'a envie de simuler la santé de façon complètement réaliste en s'occupant indépendamment de chaque membre ou organe et en simulant les effets des dégâts reçus de façons différentes selon les parties touchées, mais d'un autre côté, afficher une valeur numérique absolue nuit à la suspension volontaire d'incrédulité du joueur. Si vous tenez à afficher la santé du joueur dans une barre, vous pouvez appliquer du flou à l'extrémité de la barre. Ce n'est pas parce que la santé du joueur est stockée en dur par le jeu qu'il faut la donner telle quelle au joueur.

Tout cela est encore plus vrai en ce qui concerne la façon dont on gère la magie. Nous avons emprunté les mécanismes de Donjons et Dragons pour implémenter la magie, et cela nous a donné un modèle mathématique utile. Mais Donjons et Dragons est un jeu de rôle papier dans lequel les calculs doivent être effectués au vu et au su de tous pour que personne ne triche. C'est inutile dans un jeu -vidéo : les petits calculs et les valeurs devraient être enterrés assez profondément pour que le joueur n'en voit pas la trace.

La magie est une affaire de superstition et d'émotion, et la croyance en la magie émerge de l'intersection de deux besoins humains. Le premier est le besoin d'expliquer un univers hostile et incompréhensible. Nous ne sommes peut-être pas capables de contrôler la mort, les épidémies, la qualité des récoltes ou la météo, mais il est rassurant de penser qu'il existe des connaissances occultes qui donnent des réponses et des pouvoirs. L'autre besoin est le désir de croire qu'il existe une forme d'Être supérieur qui n'est pas sujets aux revers du destin et du hasard. L'idée d'un tel être nous donne l'espoir qu'Il vienne pour apaiser les souffrances et la détresse de l'humanité, ou qu'en L'imitant on puisse devenir comme Lui.

Pendant les siècles précédents, quand le paradigme dominant pour comprendre l'univers était surnaturel, l'Être en question était Dieu ou une panoplie de dieux et esprits. De nos jours, le besoin de croire en de telles choses est toujours présent, mais le paradigme a changé. Le paradigme dominant d'aujourd'hui c'est la science et la technologie, et Dieu a été remplacé pour beaucoup de gens par les « aliens », puissant et bienfaisant (du moins on l'espère). En fait, il existe un débat vieux de plusieurs siècles sur le sujet : quand Dieu reviendra, sera-t-Il un Dieu vengeur venu plonger l'humanité dans la damnation, ou nous apportera-t-Il la lumière, la paix et la félicité ? Aujourd'hui on se demande : quand les aliens viendront, seront-ils des êtres supérieurs et avides de conquêtes venus prendre possession de notre planète, ou seront-ils bienfaisants et venus nous offrir une technologie plus avancée pour améliorer nos vies ? C'est exactement le même débat ; seul le contexte a changé.

On remarque aussi le phénomène dans la terminologie utilisée par les charlatans pour vendre des soins aux gens les plus crédules. Pendant l'antiquité ils vendaient des amulettes et des charmes censés éloigner les mauvais esprits. Au moyen-âge, ils vendaient des reliques : des os des Saints et des fragments de la Vraie Croix qui étaient censés attirer la bienveillance de Dieu à l'égard de leurs possesseurs. Aujourd'hui ce sont les cristaux et les bracelets en cuivre, et le langage n'est plus celui de la spiritualité mais de la technologie. Ces objets « concentrent les champs biomagnétiques » ou « canalisent les flux d'énergie » (enfin il paraît), tout cela n'aurait rien voulu dire il y a mille ans. Ce sont toujours des charlataneries, mais des charlataneries de pseudoscience plutôt que de religion.

Ce que je veux dire par là c'est que rien de tout cela n'est rationnel. La magie, ancienne ou moderne, émerge d'une zone pre-rationnelle de l'esprit humain. Elle est imaginée depuis le système limbique du cerveau, à partir de la peur, des désirs, des attentes, des craintes, des interrogations et de l'horreur. En tant que développeurs de jeux, nous sommes mal placés pour comprendre cela. Nous sommes des gens qui vivent avec des ordinateurs et des ingénieurs. Nous faisons parti de ceux qui ont réussi leurs études scientifiques à l'école et à qui les mots « algèbre de Boole » ne font pas peur. Nous sommes les héritiers de cinq siècles de rationalisme... et cela se voit dans les jeux que nous faisons. Notre magie n'est pas magique, elle est mécanique.

Arthur C. Clarke a un jour écrit que n'importe quelle technologie suffisamment avancée est virtuellement indiscernable de la magie. Il ne parlait pas spécialement pour nous, utilisateurs des technologies modernes ; il voulait dire que la réponse culturelle à la technologie est relative. La distinction entre magie et technologie

repose sur l'attitude de l'observateur. Quand un paysan médiéval lambda était confronté à un phénomène mystérieux et inexplicable, il l'interprétait comme l'œuvre de Dieu, de sorcières ou de démons. Quand nous observons des phénomènes inexplicables, nous les mettons sur le compte de principes scientifiques pas-encore-compris.

Quand nous, développeurs de jeux, implémentons la magie dans les jeux, nous devons mettre notre rationalisme et notre point de vue technologique de côté un petit moment. Nous devons nous plonger dans ce marécage bouillonnant de superstitions et d'émotions que nous avons éliminé avec tant d'efficacité, mettez-y la main pour en ressortir une poignée de saletés et apprenez à travailler avec. Pensez à la magie comme y pensent les gens qui croient en la magie. Si vous incarnez un sorcier, vous n'êtes pas un genre de comptable médiéval, occupé à additionner ses points de mana sur un grand registre : vous êtes un être humain qui a été touché par la main du Tout-Puissant. Si vous incarnez une sorcière, vous ne passez pas votre temps à faire rentrer et sortir des sorts de votre cerveau comme si il s'agissait de balles dans un pistolet, en vous demandant lesquels vous devriez emmener avec vous : vous êtes une personne extraordinaire et terrifiante : les forces inconcevables de la nature tonnent à travers votre corps dans un torrent enragé à peine en votre contrôle. On ne devrait pas penser à vous comme « une personne pratique à avoir dans l'équipe » ; vous devriez être une personne à côté de qui est il est extrêmement dangereux et effrayant de se trouver.

Regardez la façon dont sont gérés les objets magiques dans la plupart des RPG fantastiques. Vous marchez dans la forêt et vous trouvez un bâton. une fois sur deux, vous savez immédiatement de quoi il s'agit, c'est tellement bien étiqueté que ça doit être gravé sur la tranche : « Bâton, magique, éclair+3. Propriété du gouvernement américain. Tout utilisateur non habilité sera poursuivi en justice. ». Ça ou alors il n'y a pas l'étiquette et vous devez l'apporter à une vieille chouette que connaissez qui a comme de par hasard la capacité d'identifier chaque objet magique jamais conçu.

« Salutations, bel aventurier ! Samantha peut-elle faire quelque chose pour vous aujourd'hui Messire ? Ah, oui. Messire possède un bâton, magique, éclair+3. Puis-je faire autre chose pour vous, Messire ? »

La réponse émotionnelle de nos personnages à ces objets est l'indifférence la plus glaciale. La plupart du temps, on ramasse ces trucs plein d'aplomb, en toute équanimité, et on se balade avec sans faire de commentaires. Si vous devez combattre, votre seule préoccupation est alors de planifier sur qui vous allez utiliser votre bâton. « Mh voyons voir, je vais te foudroyer toi, et toi aussi, et.... mh toi. »

Ce ne va pas. On parle d'un bâton magique de la foudre là, pas d'un moulin à poivre. La réponse émotionnelle appropriée à cette trouvaille n'est pas « Oh, super, ça peut servir ». Un bâton magique de la foudre est un objet dont le pouvoir est terrifiant, capable d'invoquer une des forces élémentaires de l'univers. Notre personnage convaincu de l'existence de la magie devrait répondre comme vous si on vous mettait dans les mains une bombe atomique armée :

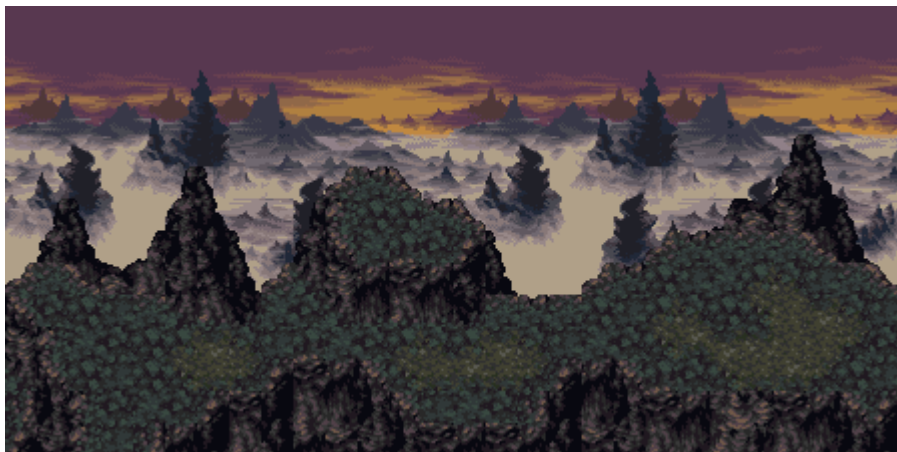
« Oh Pu#&%\$!! Je viens de ramasser un bâton rempli d'éclairs ! Merde ! J'espère qu'il ne va pas me faire exploser la tête. »

Nous devons changer la façon dont nous traitons la magie dans les jeux, et nous devons aussi changer, dans une certaine mesure, la façon dont nous la programmons. Pour l'instant, nous traitons les sorts et les objets magiques de la même manière que les autres objets et outils. Mais les objets magiques ne sont pas des armes. La magie est étrange et mystérieuse, elle est bizarre, dérangement et grotesque, elle est dangereuse, éblouissante et belle. La magie est surnaturelle, dans tout le sens extraordinaire du mot. Par dessus tout, la magie est imprévisible, la magie n'est pas fiable, la magie n'est pas mécanique. Nous devons rendre la magie magique à nouveau.

Cours de pixel art 7 : anti-aliasing

Par [Antoine Gersant](#) | le 26 juillet 2008 01:11:14 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Là où tous vos pixel arts deviennent beaux. Ou Pas. Ou si en fait, ça dépend à quel point ils étaient moches avant.



Si vous avez le courage de lire de nos jours la presse « spécialiste » du jeu vidéo, vous avez sans doute déjà vu des journalistes se plaindre de l'aliasing dont souffrent certains jeux. Il s'agit du phénomène qui survient quand un objet au premier plan se trouve devant un fond de couleur distincte : le contour de l'objet se détache du fond et « on voit les pixels » sous forme d'un affreux crénelage des bords.

Plan d'attaque

Les jeux en 3D sont les principales victimes du phénomène, étant donné que les algorithmes de lissage automatique sont coûteux en temps de calcul. Exemple ci-dessous avec *Final Fantasy III* version Nintendo DS. L'aliasing est particulièrement visible sur les bords des vêtements du personnage en noir.



Les jeux en 2D ne sont pas trop touchés car il existe depuis toujours une bonne vieille méthode de pixel art, l'anti-aliasing, pour lisser tous ces affreux contours et adoucir les transitions entre objets et arrière plan. En effet, ici pas besoin d'algorithme de la *muerte* : il suffit que les graphistes (ça c'est vous) y pensent à l'avance. Pensons donc un peu si vous le voulez bien.

On peut distinguer trois situations dans lesquels le phénomène d'aliasing peut survenir, et il va falloir agir en conséquence.

- J'ai deux couleurs différentes sur mon sprite et la transition entre les deux couleurs est aliasée (drame !).
Dans ce cas là, on va pouvoir appliquer les techniques d'anti-aliasing à loisir pour adoucir cette transition.
- Je veux adoucir la transition entre mon sprite et le décor et je connais à peu près la couleur du décor.
Dans ce cas de figure, vous pourrez anti-alier les bords de votre sprite pour que la transition avec le décor ne soit pas trop brutale.
- Je veux adoucir la transition de mon sprite avec les décors mais je ne sais pas à quoi ils ressemblent.
C'est par exemple le cas du héros de votre jeu, il va passer par toute une variété d'arrière-plans, ET BIEN NE FAITES RIEN. Voilà c'est dit. Il serait idiot de modifier les bords de votre sprite pour adoucir une transition puisque vous ne savez pas vers quelle couleur adoucir les contours.

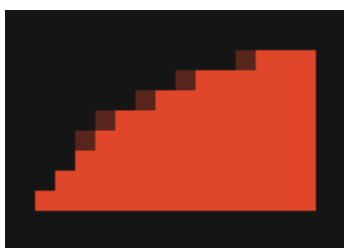
Voilà, maintenant vous savez pourquoi et quand utiliser l'Anti-Aliasing...reste à découvrir comment !

A l'assaut !

Bien. Comment adoucir ces fameuses transitions entre deux couleurs ? La réponse est plutôt évidente, il suffit d'utiliser une ou plusieurs couleurs intermédiaires (par exemple du gris pour aller entre du blanc et du noir), que l'on appelle teintes tampons (buffer shades). Toute la subtilité réside dans la façon de s'en servir. En effet, la brute sanguinaire et dénuée de tout sens artistique qui sommeille en vous a sûrement très envie de faire quelque chose comme ceci :



Argh ! C'est juste immonde. J'espère que vous vous en rendez compte aussi bien que moi, mais non seulement cette erm... « méthode » prend beaucoup de temps à mettre en œuvre (le pixel art est déjà assez fastidieux comme ça) mais surtout elle ne résout pas notre problème de transition. Bref, il ne suffit pas de faire N'IMPORTE QUOI avec les teintes tampons pour anti-aliaser. L'anti-aliasing est une technique DISCRÈTE, gardez ça en tête. Maintenant que vous savez ce qu'il ne faut surtout pas faire (d'infâmes contours), voyons ce qu'il faut effectivement faire. D'abord les images, ensuite les mots.



Occupons nous d'un petit arc de cercle orange qui tranche violemment avec son fond gris foncé. A gauche, la bête à l'état naturel. A droite, les étapes successives du processus d'anti-aliasing. Commençons par remarquer que le bord de l'arc de cercle est une suite de différents segments, ceux du bas sont de longueur 1, il y en a ensuite deux de longueur 2 et enfin deux de longueur 3 (cela devrait un peu vous rappeler le chapitre sur les courbes). Nous allons nous occuper séparément de chaque segment, en considérant les lignes

de l'image les unes après les autres. Dans le cas de segments verticaux (c'est à dire pas comme ici), on considèrerait les colonnes de l'image les unes après les autres.

Sur chaque ligne, on répartit de part et d'autre de la frontière des pixels des teintes tampons pour adoucir la transition (oui on fait un petit dégradé pour le coup), mais attention : si vous vous écartez trop de la « frontière » de vos deux teintes, vous allez détruire la forme originelle (et vous rapprocher de l'exemple de ce qu'il ne fallait pas faire). Vos teintes tampons ne doivent pas s'étaler plus loin que les extrémités des segments des lignes au dessus et au dessous, cela semble très compliqué mais en fait pas du tout, regardez les dessins : à la dernière étape, les segments de 1 pixels ne bénéficient que de deux pixels d'AA car sinon ils déborderaient sur les segments voisins et créeraient un effet appelé « banding » très déplaisant (et assez proche de l'exemple « à ne pas suivre » ci-dessus). Il ne faut surtout pas réaliser un « contour » autour de votre forme.

Au besoin, on peut se permettre quelques ajustements si malgré toutes les précautions prises l'AA dénature un peu la forme de l'objet mais cela n'a pas été le cas ici. Aussi, sachez qu'il n'est pas indispensable d'anti-aliaser les segments de longueur 1 (les lignes à 45°) qui s'en sortent généralement assez bien au naturel.

La victoire est totale chef !

Les résultats sont immédiatement visibles (ça ou alors vous avez immédiatement besoin d'une paire de lunettes). Voilà ce que cela donne sur notre cercle orange (en plus vous l'avez genre image de pub avant/après) et sur notre bon vieux dragon qui pour le même prix s'est vu élargir sa palette.



Remarquez également que j'ai un peu calmé l'ambient lighting sur le dragon. Comme annoncé précédemment, c'est une technique à utiliser avec parcimonie. Terminons par une petite note technique pour les plus débrouillards d'entre vous et que je ne savais pas trop où caser : vos teintes tampons ne doivent pas nécessairement être des couleurs dont la teinte, la saturation et la luminosité sont comprises entre celles des couleurs à lisser. Si vous voulez optimiser un peu vos palettes, vous pouvez tenter d'exploiter différents gris car ils ont le don de pouvoir servir dans des rampes de différents couleurs et font des merveilles pour l'anti-aliasing.

Concluons tout ce bazar avec un superbe pixel art généreusement fourni par Panda, qui maîtrise l'anti-aliasing bien mieux que moi (et que vous aussi). Cliquez sur l'image pour la voir en grand et n'oubliez pas d'essayer le filet de bave de votre clavier après avoir observé le détail.



[Lire la suite →](#)

La création de noms : tout un art

Par [Dyo](#) | le 5 août 2008 00:11:41 | Catégories : [Cours](#), [Univers](#)

Créer quelques noms propres c'est relativement facile ; on a tous de temps en temps des petites illuminations sur une syllabe qui déchire ou bien un nom commun détourné en nom propre. Là où ça devient compliqué c'est quand, pour un jeu où chaque PNJ porte un nom, il faut des tonnes de noms, et que ces noms doivent avoir des consonances communes selon la race/région... Sans oublier les noms de lieux. La création de noms est donc en fait une tâche complexe et longue (rassurez-vous, rien d'infaisable :p).

Attention : ce cours, qui permet de produire « en masse » des noms intéressants, est long. Si vous avez comme objectif de créer un jeu de type *Final Fantasy* où seuls les héros sont nommés (et où les PNJ s'appellent « Villageois 1 » par exemple :D), vous pouvez aller directement à la dernière partie. Elle explique les fondements d'un nom de qualité (ce qu'il faut faire — et ne pas faire) en vous épargnant le travail rendant crédible et homogène une longue série de noms (nécessaire si votre projet est plus vaste).

Attention (bis) : fuyez les générateurs tout fait (du genre ceux proposant noms elfiques, nanesques et consorts, à part pour un fan game du Seigneur des Anneaux, c'est pas bon) ou complètement aléatoires (prenant des lettres au hasard).

Ce cours est divisé en 4 parties :

- Dans la première, on va voir comment faire un brassage de sonorités pour en tirer une liste de syllabes.
- Dans la deuxième, on va produire à partir de ces syllabes une série de noms « bruts ».
- Dans la troisième, on classera par groupes des préfixes et des suffixes destinés à créer des prénoms, des noms de famille et des noms de lieux.
- Dans la quatrième et dernière partie, on sélectionnera et on retouchera les noms bruts pour en tirer les noms finalisés.

Pour ce qui est des outils nécessaires, le bloc-notes intégré à Windows est indispensable (et suffisant comme logiciel de traitement de texte). Quelques programmes supplémentaires (des générateurs) vous seront proposés dans la première et deuxième partie.

1. Sonorités

a. Par où commencer ?

Il existe plusieurs méthodes pour démarrer. Je vous en propose une qui marche particulièrement bien (et à vrai dire, je n'en ai pas trouvé de meilleure). En fait, il s'agit de débarrasser l'alphabet que nous utilisons de plusieurs lettres, afin de créer des sonorités particulières. Je vais prendre un exemple tiré d'un de mes propres projets, avec le peuple Cilde (exemple qui me servira plus loin dans ce cours). On va sélectionner un nombre restreint de lettres, le « brasser » pour en tirer des syllabes variées et à partir de là, essayer de construire de « vrais » noms.

On part avec : aeiouy — bcd fghjklmnpqrstvwxyz (il y a bien les 26, vous pouvez compter :D).

En gros je diviserai ces lettres en 3 groupes : celles qui vont amener des *sonorités dures* (*k, r, t*, etc.), celles qui vont amener des *sonorités douces* (*d, n, l*, etc.) et celles qui vont amener des *sonorités « étranges »* (*y, w, x*, etc.). Hé oui, mon cours est extrêmement franco-centré. Pour un Chinois, l'emploi de *z* et de *x* au milieu des noms est beaucoup plus habituel que pour un Français... Quoiqu'il en soit, il est plutôt conseillé de garder toutes les voyelles (qui « lient » les consonnes, bien évidemment. Ok, certaines langues d'Europe de l'Est me feront mentir). A la limite, vous pouvez en éliminer une voire deux.

Pour mon exemple (le peuple cilde), je veux des sonorités qui dégoulinent pas mal, avec juste quelques consonnes un peu dures (sinon ça ressemblera à de l'elfique :p) mais rien d'étrange. Je vais aussi m'amuser à supprimer le *u* et le *y* pour colorer encore davantage ces noms. En taillant dans le lard, j'obtiens : aeio — cdlmnrstv.

Sacrée réduction non ? Je suis passé de 26 à 13 (j'y suis allé volontairement un peu fort, car il est important que l'on reconnaisse bien les noms associés à ce peuple ; moins il y a de lettres au début, plus les noms se ressembleront à la fin — mais bien sûr vous pouvez choisir de ne supprimer que quelques lettres, voire aucune !). Si vous voulez, vous pouvez utiliser une méthode alternative, mais qui demande déjà de faire un premier pas imaginaire : dresser une liste de noms, en partant de rien, et relever dedans les lettres communes. Mais bon, l'intérêt de ce cours est d'échapper à cette contrainte de « partir de rien » (la feuille blanche, c'est jamais bien engageant). Une chose importante : conservez-bien, dans cette sélection, les lettres déjà présentes dans le nom même du peuple (par exemple, pour « cilde », je garde le *c*, le *i*, etc.). Et cela pour la cohérence.

b. Premières constructions de syllabes

Pour cette deuxième sous-partie, on va avoir besoin de l'aide d'un générateur. Le travail pourrait être fait de tête, mais le générateur nous facilite la tâche. Personnellement, j'utilise le petit programme *NewName*, qui fait parfaitement l'affaire.

[NewName \(595 ko\)](#)

Rentrez dans les champs appropriés votre sélection de voyelles et de consonnes. Décochez noms composés, numéros, majuscules. Laissez coché syllabes, de 1 à 3, et générez !

Le logiciel va vous donner divers noms (ainsi que quelques syllabes isolées) et dans le tas plusieurs qui pourraient être pris tels quels. Relevez tous les noms intéressants mais aussi toutes les syllabes et associations de syllabes qui vous plaisent, qui « sonnent » bien. Attention, c'est cette partie qui détermine la « couleur » finale des noms du peuple, donc il faut vraiment s'appliquer. Mais n'hésitez pas à générer vite et beaucoup ; laissez parler votre instinct, générez, générez. Et cueillez au vol tout ce qui sonne bien !

Avec ma sélection de lettres de départ, j'obtiens : Val — Memnel — Aramid — Nirteven — Tel — Torel — Ren — etc.

C'est le moment de regrouper toutes vos syllabes sélectionnées en une unique liste. C'est aussi l'heure du second tri, « à tête reposée ». Éliminez toutes les syllabes qui vous paraissent trop hétérogènes au milieu du reste. Gardez uniquement ce qui vous plaît vraiment : l'idéal est d'en obtenir une grosse quarantaine (une trentaine semble être un minimum, mais bien sûr tout dépend du nombre de noms que vous voulez créer). N'oubliez pas aussi de décomposer tous les noms relevés.

Par exemple, Nirteven me donne -nir, -te, -ven. Mais j'aurai tout aussi bien pu décomposer en -nirt -ev -en.

Pour ce qui concerne le nombre de lettres par syllabe (2, 3 lettres voire 4), tout dépend de ce que vous voulez obtenir à la fin. Ne garder dans votre liste que des syllabes à 2 lettres, par exemple, vous donnera des noms plus spéciaux, originaux, que si vous les prenez toutes sans distinction. N'hésitez pas à intégrer dans vos syllabes choisies des consonnes muettes, ou bien des successions de consonnes (*ett*, *ahj*, *lpo* peuvent être retenues si elles correspondent à l'aspect des noms que vous désirez).

2. Production

C'est la partie la plus facile (car uniquement technique) : il s'agit maintenant de configurer un générateur qui permettra de créer rapidement des tonnes et des tonnes de noms, qui auront donc tous des consonances communes. Fil m'a retrouvé une de ses créations, un générateur personnalisable (*Note de Fil : excusez la laideur, le petit format et le Flash Player prérequis. Cette chose doit bien dater de 2006.*).

[Générateur offert par le site](#)

Pour ce qui est de son utilisation : ouvrez tout d'abord le fichier .xml (avec le bloc-notes), et à l'intérieur des guillemets, inscrivez toutes les syllabes obtenues précédemment, sans oublier aussi les voyelles.

Pour mon exemple, ça donne :

```
<syllabe txt="a"/>
<syllabe txt="e"/>
<syllabe txt="i"/>
<syllabe txt="o"/>
<syllabe txt="val"/>
<syllabe txt="mem"/>
<syllabe txt="nel"/>
[...]
```

Pas de difficulté, il suffit de rentrer les syllabes telles quelles (par ordre alphabétique si ça vous amuse, mais de toute façon le générateur choisira aléatoirement :p). Et voilà, il n'y a plus qu'à lancer le fichier .html et à générer. On obtient rapidement et facilement une grande série de noms se ressemblant. Mais bien entendu, tous ne sont pas « exploitables », tous ne sonnent pas bien... Ce serait trop facile ! Il nous reste encore plusieurs étapes : un gros travail de catégorisation et de retouche.

3. Catégorisation des préfixes et suffixes

a. Quelques mises au point

Nous allons maintenant inventer et classer des préfixes et des suffixes qui nous serviront à transformer les noms bruts en prénoms, noms de familles ou noms de lieux. Pour bien faire, il faudrait d'abord inventer une langue, de laquelle ils découleraient tous. Cohérence et réalisme garantis, mais à part Tolkien je vois peu de gens capables de cette prouesse.

Analysons rapidement un exemple typiquement français : *Marie Dujardin*, habitant la commune de Bellerive.

- Le prénom Marie correspond à la francisation du prénom hébreux Myriam. Comme la plupart des prénoms donnés en France, celui-ci provient d'un saint (une sainte en l'occurrence). On voit ici toute la complexité de l'apparition d'un prénom dans une langue, car cette apparition peut avoir des origines très diverses : adaptation d'un nom commun, d'un nom étranger, d'un personnage saint...
- Le nom *Dujardin* est assez explicite : le patronyme renvoie à un lieu (donc à un nom commun). On peut d'ailleurs souvent rattacher un nom à une caractéristique (lieu d'habitation, particularité physique, habitude comportementale/morale...).
- Enfin le nom du lieu, Bellerive, est du même type : sans doute ce village est-il situé à proximité d'un cours d'eau.

Cette analyse rapide et primaire, très éloignée d'un véritable travail étymologique, n'a qu'un seul but : vous montrer toute la difficulté que l'on rencontre quand on veut donner du sens et une origine aux noms que l'on crée. L'invention préalable d'une langue est nécessaire à l'élaboration d'un système de noms poussé.

Mais une méthode légère et alternative est possible. On va en fait se contenter de reprendre des suffixes et des préfixes communs, pour une meilleure homogénéité au sein de chaque catégorie de noms (prénoms,

lieux, noms patronymiques) et pour plus de réalisme. Vous avez certainement déjà senti que la sélection de lettres au début de ce cours déterminait pour beaucoup l'aspect final des noms. Donc ce que l'on fait tout de suite, c'est juste une meilleure mise en forme. Et selon vos attentes, vos besoins, et un perfectionnisme plus ou moins aigu, cette mise en forme sera plus ou moins longue.

b. On catégorise !

Reprenons notre liste de syllabes créée dans la partie 1. Nous allons en prélever certaines et les répartir en plusieurs groupes. Généralement, les groupes seront :

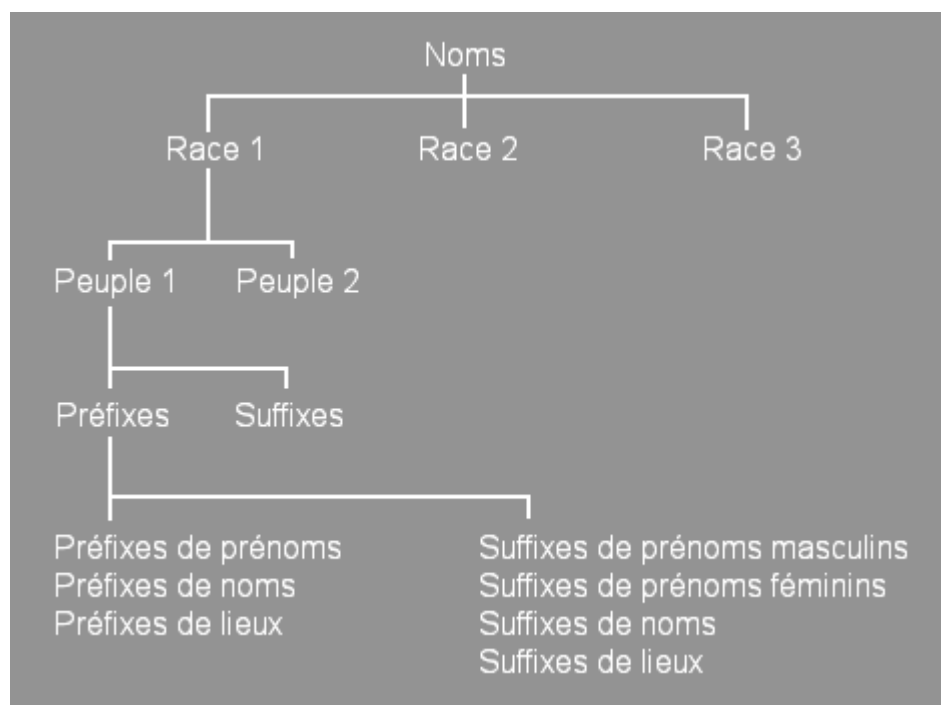
- Les préfixes de prénoms
- Les préfixes de noms de famille
- Les préfixes de noms de lieux

Et

- Les suffixes de prénoms masculins
- Les suffixes de prénoms féminins
- Les suffixes de noms de famille
- Les suffixes de noms de lieux

Attention ! A ce stade, tout dépend de vous : ce sont vos choix qui vont déterminer l'ampleur de la tâche ! En effet, vous n'êtes pas obligés de créer autant de groupes (en supprimant les préfixes de noms de lieux, par exemple), mais vous pouvez aussi en créer d'autres (en créant des préfixes masculins et féminins, ou en multipliant l'ensemble des groupes par le nombre de races/peuples/régions souhaité...).

Un très bon exemple est celui du Seigneur des Anneaux. Tolkien a nommé un de ses peuples avec des prénoms + noms patronymiques (ex : *Bilbo Baggins*), mais d'autres peuples ne possèdent qu'un seul nom « générique » (ex : *Aragorn*, fils d'*Arathorn*). De plus il a bien évidemment séparé les préfixes/suffixes humains des préfixes/suffixes elfiques et nanesques, mais à l'intérieur du groupe de noms « humains » il a encore séparé les hommes du Rohan de ceux du Gondor... C'est un travail titanesque mais qui apporte une profondeur et un réalisme incroyable à son univers.



Ce schéma vous montre comment s'articule l'arborescence des groupes nécessaire à une catégorisation étendue. Tout ceci nécessite beaucoup d'organisation ! Mais bien sûr, cette catégorisation peut être restreinte.

Pour une catégorisation superficielle, contentez-vous d'attribuer 3 ou 4 syllabes (tirées de votre liste) à chaque groupe, mais rien ne vous empêche d'en attribuer 20 (bien sûr, cela nécessite de se munir auparavant d'une longue liste de syllabes « exploitables » — n'hésitez pas à en générer davantage !). Réserver ainsi quelques syllabes pour les différents types de noms permet de donner des points de repère au futur lecteur/joueur. Il se sentira rassuré de reconnaître dans plusieurs prénoms ou dans les noms de lieux des préfixes/suffixes identiques, et vous gagnerez en crédibilité.

c. Donner du sens

Nous arrivons au moment de déterminer si vous voulez donner un véritable sens à tous vos noms (ou seulement quelques-uns). Vous pouvez en effet inventer une traduction pour chaque syllabe catégorisée.

Exemple : si je sors de ma liste de départ les syllabes *nir*, et *an*. - *nir* me paraît intéressante en position de préfixe. Cette syllabe signifiera « la terre ». Ainsi le prénom *Nirien* pourrait-il être traduit grosso modo par « le terrien ». Et *Nirteven*, en tant que nom de lieu, par « la haute terre » (notez que j'utilise le même préfixe pour deux types de noms — je réunis ainsi deux groupes en un seul). En procédant comme ceci, on introduit des éléments intéressants dans son univers sans avoir à élaborer un véritable système linguistique. - *an* semble mieux convenir comme suffixe. Je vais placer cette syllabe dans le groupe des suffixes de prénoms masculins, mais je ne vais pas inventer de traduction. Je réalise donc un travail partiel, mais toujours cohérent.

Attention, il ne faut pas trop abuser de cette méthode. Vos noms risquent de paraître redondants si vous ré-utilisez trop souvent les mêmes préfixes et suffixes. En fait, comme vous pourrez le constater si vous avez à générer énormément de noms, la plupart d'entre eux pourront se passer de ces ajouts. Il vous suffira simplement de les sélectionner avec soin.

Il serait logique à ce stade de se demander : quelles syllabes attribuer à quels groupes de préfixes/suffixes ? Et donc, autrement formulé : à quoi doit ressembler tel type de nom ? C'est l'objet de la dernière partie.

4. Finalisation

a. Types de noms

En fait, il n'existe aucune règle formelle. N'importe quel nom pourrait, en théorie, être attaché à une personne ou à un lieu. Mais en réalité, notre langue maternelle (et notre connaissance de plusieurs langues étrangères) nous conduit à privilégier certaines sonorités, certaines syllabes, selon le type de nom. Par exemple, la syllabe *-el*, utilisée en tant que suffixe, rappelle les prénoms *Daniel*, *Raphaël*, *Marcel*... De la même façon, la syllabe *-ster* en fin de nom de ville rappellera Gloucester ou Manchester (villes anglaises).

Ces syllabes familières ne sont pas un mal. Ce sont elles qui donneront de la crédibilité à vos noms. En effet, il est difficile de rendre accrocheur un nom dépourvu de toute ressemblance avec une langue existante. Par exemple, si je ne savais pas à quoi ressemblait le polonais et que je lisais « Krzysztof Zbikowski », je trouverais ça d'une rudesse incroyable (un assemblage maladroit incluant trop de consonnes). Or, *Krzysztof* m'évoque inmanquablement *Christophe*, et la terminaison en *-ski* est très répandue dans les noms polonais.

Le joueur a donc besoin de « s'y retrouver » . Dans ce but, on peut utiliser une méthode : calquer approximativement les types de noms sur une langue existante. Si vous avez choisi le français, par exemple :

- Le nom de famille sera plus long que le prénom (ou de taille égale) mais rarement plus court.
- Les prénoms et les noms de familles ne comporteront pas de successions trop longues de consonnes.

- Vous utiliserez des syllabes familières « françaises » pour distinguer les noms patronymiques des prénoms et des noms de lieux.

Exemple : pour l'un de mes peuples, les Lugdènes, je voulais des noms à consonances très françaises. Grâce à l'ajout de syllabes dites « familières » dans les groupes de suffixes, j'ai obtenu comme noms de famille : *Planguène, Lacière, Gulbelet, Lebord, Remillet...* Ces noms sont crédibles car ils respectent la « charte » des noms français.

Mais vous auriez pu choisir le chinois :

- Les prénoms et les noms de familles sont courts.
- La formulation en trois parties (ajout du nom maternel) est fréquente.
- Les lettres *z, w, ou x* sont plus fréquemment utilisées.
- Vous utiliserez des syllabes familières « chinoises ».

Exemple : Si vous avez déjà joué à un jeu de la série *The Elder Scrolls*, vous connaissez sans doute la race des Impériaux. Ces derniers ont leurs noms calqués sur les prénoms et patronymes romains (*Caius Cosades, Crassius, Darius...*). Le jeu (qui se veut réaliste) en tire un grand intérêt, et la légion impériale se présentant sous beaucoup d'aspects comme la légion romaine, ça ne choque pas. Toutefois, il faut bien remarquer que les Impériaux correspondent au seul peuple calquant d'aussi près une véritable langue (sinon cet univers fantastique ne le serait plus beaucoup).

Cette technique doit donc être appliquée avec retenue. En fait, il est préférable d'essayer de croiser plusieurs langues pour obtenir des mélanges intéressants. Le but est d'éviter l'emploi de syllabes familières au « mauvais endroit » (le joueur doit sentir que tel nom désigne une personne ou un nom de lieu). Mais il n'y a pas de formule magique : c'est à vous de jouer. Selon le nombre de syllabes catégorisées, vous pourrez rendre vos noms très exotiques (consonances inconnues du joueur) ou bien familiers (syllabes « familières » de la ou les langues choisies).

b. Des possibilités supplémentaires

Récapitulons : une fois le travail de catégorisation des préfixes et des suffixes achevé, il ne vous reste plus qu'à vous servir du générateur configuré dans la partie 2 pour générer rapidement des listes de noms (j'insiste sur le « rapidement » : le premier coup d'œil est le meilleur, vous ne devez pas avoir à vous « convaincre » vous-même que tel ou tel nom est acceptable — il doit naturellement vous paraître réaliste et plaisant à l'oreille).

Piochez ensuite dans vos groupes de préfixes : vous pouvez les rajouter en début de nom ou remplacer directement la première syllabe par le préfixe choisi. Il en va de même pour les suffixes. N'oubliez pas que ces préfixes et suffixes sont « accessoires », vous pouvez vous en passer. Comme vu juste au dessus, ces syllabes, plus ou moins familières, vont aider à créer les « types » de noms (et à leur donner crédibilité et sens).

Il nous reste encore quelques techniques de création importantes à aborder.

L'emploi direct de noms communs

Une technique très intéressante consiste à se servir de noms communs (français, obligatoirement — mais ils pourront être traduits) pour créer directement des noms propres (principalement des noms de lieux, mais aussi des noms patronymiques).

- Soit tels quels : « Les Hautes Plaines », « Les Marais des Morts »
- Soit par associations : « Rochefer », « Sombrefeuille »...

En revanche il est délicat de mélanger ces noms avec des noms créés de toute pièce. Utilisez-les pour des régions particulières si vous voulez être certains d'éviter l'hétérogénéité. Pour ce qui est du sens, il est bien sûr explicite mais vous pouvez jouer avec pour obtenir des effets divers ; par exemple, appeler une terre dévastée « Blanchelande ».

L'emploi de surnoms

Comme dit précédemment, un nom propre est souvent issu d'une caractéristique ; on peut donc attacher des surnoms (basés sur le même principe) assez facilement. Surnoms relatifs à une caractéristique physique (« le boiteux », « le borgne ») mais pas seulement ; ils peuvent concerner un comportement (« le brave », « l'indécis »), un talent ou un métier (« l'artisan », « le virtuose ») ou même une habitude vestimentaire (« le rouge », « le noir »).

Il faut noter qu'un surnom interpelle plus volontiers le lecteur/joueur. Attention donc à ne pas trop les utiliser et à les réserver à des personnages importants, ou tout du moins à des personnages sur lesquels vous voulez attirer l'attention.

c. Le bon, le beau et le mauvais nom

Il me reste enfin quelques recommandations à vous soumettre (rassurez-vous, c'est bientôt fini).

Un nom sera toujours « bon » si :

- Il a été travaillé (il ne doit pas avoir été inventé dans l'urgence pour pallier un manque d'inspiration).
- Il sonne bien à la première lecture (ou plutôt, "écoute"). A force de lecture, n'importe quel nom peut finir par devenir acceptable, pour autant s'il était mauvais au début il le restera.
- Il s'intègre bien dans son "groupe" (race, région, peuple). L'homogénéité est l'indispensable but à atteindre.
- Il se retient facilement : un nom trop compliqué, difficile à lire, ne restera pas longtemps dans la tête du joueur. L'utilisation de nombreuses consonnes contribue à alourdir la lecture.

Maintenant, voyons comment rendre nos noms excellents, en :

- Insérant des allitérations : répéter dans le prénom et le nom des mêmes syllabes (ou consonnes) donne un effet d'harmonie tout à fait plaisant. Par exemple : dans *Caïus Cosades*, on remarque une répétition du *c* en ouverture de nom et un *s* en fermeture. Encore une fois, cette méthode est à utiliser avec parcimonie.
- Équilibrant les longueurs : si le nom de famille est long, le prénom doit être court. Si le prénom est long, le nom de famille doit être court. Le but est que l'ensemble ne soit pas trop lourd à lire (surtout s'il est répété souvent). Un effet intéressant s'obtient en mettant côte à côte un nom très court (une syllabe) et un nom très long (trois ou quatre syllabes).
- Se servant du nom pour conférer une identité à un lieu/personnage : le nom, plus ou moins doux ou dur à l'oreille, plus ou moins porteur de sens, va aider à construire une identité pour celui (ou ce qui) le porte. La technique la plus simple est l'emploi d'un surnom ; mais plus encore, certaines consonances peuvent former des images dans la tête du joueur.

Par exemple, *Krarock* est un nom très dur (des *k* et des *r* qui râpent le palais). De plus, *kra* (« crasseux, crade ») évoque la saleté et *rock* (simplement *roc*), quelque chose de solide ou d'imperturbable. Et hop, voilà que l'image d'un guerrier robuste et sale se forme dans ma tête (et certainement pas l'image d'une fillette en robe rose). Il est impossible, bien sûr, de maîtriser complètement la connotation d'un nom et l'effet qu'il produira sur le joueur, mais on peut quand même prendre garde à la dureté plus ou moins grande d'un nom et aux homonymies.

Et au contraire, voici les écueils à éviter :

- Utiliser à outrance les lettres muettes et les apostrophes : le grand danger de cette technique sur-employée dans les univers fantastiques est de faire perdre toute saveur à vos noms. Il est facile en effet d'insérer un *h*

muet en plein cœur d'un nom ou bien de couper celui ci en deux avec une apostrophe. Mais c'est une méthode artificielle qui va avoir tendance à effacer le travail sur les syllabes au profit de la forme, ainsi vos noms seront redondants et banalisés (à la limite vous pouvez réserver son emploi pour un peuple ou une région, et en usant de la technique avec retenue).

- Utiliser des noms aux consonances trop proches pour les personnages principaux d'un jeu : plus les noms sont éloignés en terme de sonorités, plus le joueur pourra s'en rappeler facilement. Une exception concerne les personnages d'une même famille, qui peuvent avoir des noms ressemblants (ce sera un aide-mémoire pour le joueur, à moins que le lien familial doive rester secret). Pour un jeu où seuls les héros sont nommés, n'hésitez pas à vous libérer complètement des contraintes de réalisme ; vous pouvez toujours vous servir du générateur de syllabe (CF Partie 1 et 2) pour attraper au vol une association de syllabes intéressante, mais votre jugement personnel compte par dessus tout.
- Insérer des noms existants au milieu de noms inventés : il n'y a rien de pire que jouer à un RPG médiéval-fantastique et se rendre compte que le nom du héros est *Anthony*. Il est tentant, quand on crée un univers, quand on nomme à tour de bras des PNJ et des héros, d'inclure ici ou là le nom d'un ami par exemple, ou bien un prénom qui nous a toujours plu. Mais bien évidemment, cela déconstruit totalement la crédibilité et l'homogénéité du jeu. Attention donc à refréner parfois ses envies pour un résultat cohérent !
- Recourir systématiquement à la terminaison en *-a* pour nommer vos lieux et vos personnages féminins : cette technique, là aussi, est gratuite et sur-utilisée. Il ne s'agit pas de bannir l'emploi du *-a* mais juste de le remettre à sa place (il doit découler logiquement de votre travail de construction des noms et ne pas arriver comme par magie à la fin).

En conclusion...

Cet article est fini, j'espère qu'il vous aura aidé. Il me reste un dernier conseil à vous donner : la création de nom est un art, et on ne peut pas l'enfermer dans quelques lignes de texte. Les techniques proposées, les choses à faire et à ne pas faire seront des aides de chantier mais jamais elles ne se suffiront à elles-mêmes. Les noms que vous inventerez ne seront jamais perçus de la même manière par deux de vos lecteurs, il vous faudra avoir tout ceci à l'esprit et ne pas attendre d'office des retours dithyrambiques. J'ai toutefois la prétention de penser qu'en suivant de près ou de loin cet article, vous obtiendrez des retours positifs... ou pas.

Cours de pixel art 8 : textures et dithering

Par [Antoine Gersant](#) | le 5 août 2008 01:33:40 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Sauf qu'en fait on va parler de dithering avant de parler de textures. Mais ça sonnait mieux dans l'autre sens.



Après avoir fait une étape au pays de la douceur et de la volupté grâce à l'anti-aliasing, nous allons aujourd'hui nous attaquer à des techniques bien crades, sauvages et pleines de pixels (pour changer) — à savoir l'imitation de textures et le dithering. Commençons par une petite mise au point linguistique : le mot français pour parler de dithering est « tramage » mais les habitudes étant ce qu'elles sont (ça et mes préférences personnelles), vous n'entendrez pas le mot « tramage » dans ce chapitre. A vous de voir celui que vous souhaitez utiliser. Vous ne recevrez pas de coups de fouet si votre choix se porte sur « tramage » (juste un mépris sans fin de ma part).

Ditherer n'est pas jouer

Le dithering est une bonne vieille technique inventée du temps de Matusalem, à l'époque où les machines étaient encore trop péraves pour afficher plus de quelques couleurs à la fois et où les pixel artistes avaient du temps à perdre. Il s'agit d'une technique qui a pour but la réalisation d'un dégradé en utilisant un minimum de couleurs intermédiaires (oui c'est paradoxal, non ce n'est pas absurde).

Concrètement comment ça marche ? L'œil humain est en fait tout pourri et distingue les pixels de votre écran avec une précision douteuse, sur laquelle repose bon nombre d'illusions d'optiques, si bien qu'il a tendance à mélanger la teinte d'un pixel avec celles de ses voisins. Le carré vert en dessous de ce paragraphe est uniquement constitué de pixels jaunes et bleus entrelacés, et votre œil effectue le mélange pour lui donner sa couleur verte. Merveilleux n'est-ce pas ?

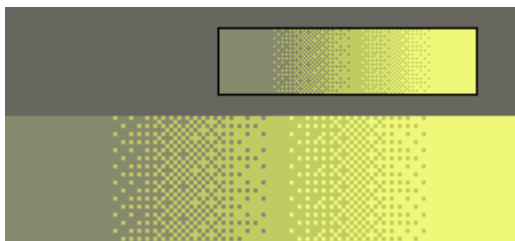


Nous allons donc utiliser cette propriété pour créer des dégradés de couleurs (par exemple du jaune au bleu, en utilisant l'entrelacs ci dessus pour créer du vert entre les deux). Il y a 3 bonnes raisons de procéder ainsi plutôt que d'utiliser un dégradé automatique comme en proposent les logiciels du type Photoshop : la première, c'est qu'en utilisant peu de couleurs, vous maintenez un style "pixel art" qui correspond au reste de votre jeu (pour peu que vous le fassiez en pixel art), la deuxième c'est que cela permet une gestion plus intelligente de votre palette : vous pouvez réutiliser certaines couleurs de votre palette dans le dégradé pour rendre l'image plus cohérente. Enfin la 3ème et certainement la meilleure, c'est que contrairement à un dégradé automatique qui réalise une transition parfaite mais monotone entre 2 couleurs, vous pouvez ici choisir des couleurs intermédiaires originales (hop, un petit coup de saturation en plus au milieu par exemple) pour avoir des dégradés intéressants pour l'œil.

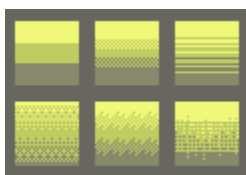
Sachez tout de même que l'heure de gloire de cette technique est finie depuis belle lurette et que mal employée, elle a plus de chances de ruiner vos chefs d'œuvres que de les embellir. Par ailleurs, c'est une

technique qui nécessite beaucoup d'espace pour être appliquée et elle est complètement inutile dans la création d'un sprite. Elle a beaucoup servi à réaliser les images de la demoscene ou les arrières plans de ciel dans certains jeux.

Voyez ci-dessous à quoi ressemble un dégradé fait avec 3 couleurs en utilisant du dithering. Je pense qu'il est plus simple pour vous de regarder le motif que pour moi de le décrire. La tendance générale consiste à utiliser de plus en plus une couleur à mesure qu'on se rapproche de "son côté". Il ne faut pas trop coller des motifs distincts les trop près les uns des autres mais les laisser s'étaler sur plusieurs lignes, sinon une fois dezoomé on a pas l'impression que ces motifs obéissent à une quelconque logique.



Bien entendu, il existe watmille motifs différents et tous plus compliqués les uns que les autres, histoire de faire passer votre petite sœur et son point de croix pour une débutante. En voici quelques uns parmi les plus simples, rien ne vous empêche de créer les vôtres. Vous n'êtes d'ailleurs même pas obligés de faire appel à des motifs ordonnés : un dithering chaotique fonctionne aussi et crée une certaine texture (sale et rocailleuse) sur une surface.



Ah tiens ça tombe bien, on allait justement parler de textures. Quel hasard.

Lumières et matériaux

Savoir réaliser des textures en pixel art, c'est savoir représenter différents matériaux indépendamment de leur forme. C'est ce qui permet à votre joueur de penser "je passe sur le petit pont en bois" plutôt que "je passe sur le petit pont en plastique". En effet, tous les matériaux ont une texture qui leur est propre et qui permet de les distinguer. Deux critères sont déterminants dans la réalisation d'une texture : le premier c'est le motif qui recouvre votre matériau. Il s'agit le plus souvent d'aspérités à sa surface (nœuds sur le bois, petites éraflures sur la pierre) qu'il convient de représenter en réalisant du shading à très petite échelle. Le second critère est la façon dont le matériau réagit à la lumière et la réfléchit, c'est ce qui fait que certaines matières sont plus brillantes que d'autres. Passons sans plus attendre à la pratique en parcourant ensemble une petite sélection de matériaux. Les bestiaux en image :



- **Le bois** : Pour le bois, tout repose sur le motif à la surface, qui est un réseau de lignes qui se coupent parfois pour former des nœuds. Il est peu sensible à l'éclairage, sauf si il a été vernis (ce n'est pas le cas ici).
- **Le granit** : Le granit a une surface bien plus marquée et irrégulière que le bois et une boule de granit ne s'ombre pas "simplement comme une sphère". En déformant les ombres, on déforme la surface de la pierre pour dessiner les aspérités. Les petites touches de couleurs isolées servent à représenter les petits trous/bosses sur la pierre : il s'agit en fait de shading à très petite échelle. Sa réaction à la lumière est un peu moins timide que celle du bois, mais toujours très matte.
- **Le marbre** : La technique pour le marbre est pratiquement la même que pour le granit. La principale différence est que le marbre est généralement plus lisse (on utilise donc plus de couleurs et plus d'antialiasing). Par ailleurs, il a tendance à recevoir des highlights bien plus clairs que ceux du granit.
- **L'eau (ou le verre)** : Ces deux matières sont semi-transparentes et ont donc tendance à réfléchir les lumières à l'intérieur d'eux même, d'où le second highlight en bas à gauche. En plus de cela, une grande partie de la lumière reçue est renvoyée et c'est pour cela que le highlight principal (en haut à gauche) est aussi gros et clair. Enfin, ils s'agit de matières très lisses qui demandent beaucoup d'anti-aliasing.
- **L'argile** : L'argile est une des matières les plus simples à représenter. Elle demande peu de couleurs car elle réagit peu à la lumière et son aspect grossier permet de ne pas trop s'embêter avec des détails. La boule a été entièrement faite avec un curseur de taille 2pixels * 2pixels (c'est dire si les détails ont peu d'importance).
- **Le cuivre** : Le cuivre, comme tous les métaux, réfléchit violemment la lumière dans ses highlights et devient très rapidement très sombre dans les zones à l'ombre. Il ne faut pas avoir peur d'utiliser une palette très étendue, non seulement pour rendre compte du contraste monstrueux, mais aussi pour donner un aspect lisse à la surface (antialiasing à la clé). A noter, les ombres sur le cuivre ont tendance à partir dans le VERT (oui, le vert. Ne me demandez pas pourquoi, regardez plutôt des photos pour vous en convaincre).
- **Le métal (acier)** : Comme annoncé juste au dessus, les métaux explorent toute la gamme de valeurs, du noir au blanc...mais sans passer par le gris. Il est courant de voir des débutants représenter les métaux par du gris, mais c'est une erreur. Les métaux sont toujours teintés, soit intrinsèquement soit par l'éclairage ambiant, mais il ne sont jamais simplement gris.
- **Pastèque** : En fait, la pastèque réagit presque comme le plastique mais je trouvais plus drôle de faire une pastèque qu'un bout de plastique. Les highlights du plastique sont très désaturés et partent vite vers le blanc, tandis que les ombres sont peu prononcées. Bien évidemment, il n'y a ni motif ni aspérités sur le plastique. Par contre pour les pastèques...

A POIL !

Les poils, cheveux, fourrures et autres viriles pilosités sont un cas à part et particulièrement difficile à traiter (les artistes 3D ont d'ailleurs autant de mal que nous avec, si ça peut vous rassurer). Tout le problème vient du fait qu'un poil est plus fin qu'un pixel et que par conséquent, on ne peut pas représenter de poils. C'est triste hein. Tant pis on laisse tomber alors. Ou pas.

La solution consiste à ne pas représenter des poils ou des cheveux, mais des mèches (et en plus y'a sacrément moins de mèches que de cheveux, alors c'est tout bénéf). Chaque mèche peut être représentée par une ligne ou un triangle dont l'épaisseur dépend la longueur des poils (plus les poils sont longs, plus les mèches sont fines) et dont l'orientation sert à définir le volume sous-jacent : les poils sont dans le sens de la surface qu'ils recouvrent ou perpendiculaire à cette direction (selon ce qui est le plus proche de la verticale / ce qui vous semble le plus logique). Par exemple, les poils sur le dos d'un animal sont horizontaux, tandis qu'ils tombent plus à la verticale (mais pas entièrement) le long de ses flancs.

Enfin, les poils sont des textures d'une grande douceur (pensez douillet :3) et cela doit se ressentir dans vos choix de couleur pour les dessiner. Les transitions entre teintes doivent être masquées autant que possibles, et à cet effet, il est intéressant d'adopter une disposition en mille-feuille. Il s'agit de considérer les poils comme une couche de manteaux que porte votre créature, certains (les plus foncés) "tombant" plus bas que

d'autres dans les zones d'ombres. Cela vous permettra d'entrelacer les différentes teintes tout en conservant la texture du pelage. C'est précisément ce que j'ai essayé de faire dans l'exemple du lion ci-dessous.



Enfin, je vous laisse cette fois encore avec un pixel art réussi à admirer. Remercions Ahruon qui nous fournit généreusement sa Moonlight Parade pour illustrer cette partie. je vous conseille d'y jeter un coup d'œil attentif (en zoomant très fort de préférence).



Pour conclure, revenons sur la retenue dont il faut faire preuve avec le dithering et les textures. Il est souvent préférable de maintenir certaines zones ou objets dans l'ombre ou la lumière suffisamment fort pour que leur texture soit indiscernable. Cela permet de renforcer la profondeur de votre image, cela vous fait moins de travail, et cela sert à définir les zones sur lesquels l'accent est mis (les plus détaillées donc). Remarquez que la patte arrière gauche du lion est pratiquement un aplat de couleur. Petite note pessimiste pour finir : si vous avez l'intention d'animer un truc pareil, vous allez en baver alors réfléchissez-y à deux fois avant de trop détailler.

[Lire la suite →](#)

Le commerce dans les RPG

Par [Pierre Guyot](#) | le 16 août 2008 02:15:14 | Catégories : [Game Design](#), [Univers](#)

L'un des éléments les plus importants du RPG, bien qu'on n'y pense pas tout de suite, c'est le commerce. En effet, qui dit jeu de rôle dit achat et vente d'équipement et de matériel divers (épées, potions, amulettes, etc.) et donc, commerce. L'article qui va suivre vise à développer une réflexion sur quelques aspects du commerce dans les RPG, son influence et la façon de l'implémenter, de façon à immerger le joueur dans l'univers sans que cela devienne un fardeau pour lui.



I. Les différents aspects du commerce

1. Aspect culturel

Le commerce intervient de différentes façons dans l'univers d'un RPG. En premier lieu, c'est un élément de background. Le commerce est un indicateur de développement économique, qui permet de distinguer un petit village dans une forêt coupée du reste du monde et une gigantesque ville portuaire au carrefour des civilisations : il est tout naturel de trouver des magasins plus nombreux et mieux achalandés dans les grandes villes que dans les petites.

Les objets vendus sont aussi des éléments de background. En jouant sur les objets produits uniquement dans certaines régions, on peut modifier l'ambiance d'une ville : par exemple, telle ville sera la seule à proposer la viande d'un animal qui n'est élevé que dans cette région. Les produits locaux donnent un certain exotisme à la ville visitée, ils lui donnent un côté « typique ». A l'inverse, une ville qui importe des produits dans le monde sera sans doute un carrefour commercial, un port par exemple. Ceci va de pair avec la façon dont sont répartis les savoir-faire : telle arme a été forgée à l'aide d'une technologie maîtrisée uniquement par les forgerons d'une certaine région.

Les marchands, enfin, sont aussi des éléments de background. Leur design doit coller avec leurs origines et si possible avec les produits qu'ils vendent. Ils peuvent donner un caractère rassurant ou inquiétant à un lieu selon leur caractère et ils peuvent constituer des personnages à part entière, comiques très souvent.

2. La monnaie

Les échanges commerciaux dans les RPG se font le plus souvent par le biais d'une monnaie. La plupart des jeux se contente d'un seul type de devises — souvent des pièces d'or — universel et la plupart du temps sans intérêt.

Les développeurs de *Chrono Trigger* ont proposé une solution intéressante à ce problème : à la préhistoire, la monnaie utilisée dans les autres époques — les pièces d'or — est remplacée par des objets apparemment

sans valeur — plumes, crocs, os, etc. — que le joueur peut échanger contre d'autres objets. Ceci permet d'éviter que le joueur ne « sorte » du jeu : imaginez-vous en train de payer des hommes préhistoriques avec des pièces d'or frappées à l'effigie d'un monarque ou d'un monument. Vous vous diriez immédiatement que les développeurs n'y ont pas pensé, que ce n'est pas crédible.

On peut aussi citer la série des *Zelda*, qui utilise des cristaux de différentes couleurs appelés Rubis. D'une part, ceci permet de particulariser l'univers et lui donner une touche qui le rend un peu plus unique. D'autre part, ce système de couleurs permet d'éviter d'utiliser des chiffres et permet donc une immersion plus efficace.

Intéressons maintenant au cas d'*Odin Sphere*. Le joueur doit jongler entre pas moins de cinq monnaies différentes. Celles-ci ont des designs, des histoires et des valeurs différentes et elles sont même intégrées au scénario. Elles constituent donc une forte plus-value pour l'univers. Mais elles ont aussi une importance véritable dans le gameplay. Les marchands ne feront pas l'effort de vous rendre la monnaie lors de vos achats, c'est donc à vous de faire l'appoint, d'où l'intérêt d'avoir la bonne pièce sur soi. De plus, l'acquisition d'expérience se fait par la cuisine dans *Odin Sphere* et chaque recette proposée possède son addition propre : si vous voulez préparer un plat, il vous faudra un certain type de monnaie, même si vous avez la somme équivalente avec d'autres pièces. Le joueur doit réfléchir un peu plus qu'à l'accoutumée quand il s'agit de ses achats : puis-je me permettre de m'acheter cet objet avec mes pièces ou ferais-je mieux de les économiser en vue de réaliser une recette très particulière ?

Une autre possibilité pour donner un rôle à la monnaie est la mise en place de taux de changes. *Secret of Evermore* proposait cette possibilité en permettant au joueur de changer sa monnaie des époques précédentes contre de la monnaie nouvelle, malheureusement à des taux de change lamentables. On aurait pu imaginer, pour pallier ce défaut, l'existence de PNJ bien cachés rachetant au prix fort votre ancienne monnaie ou un moyen de la transformer pour augmenter sa valeur, ce qui aurait pu donner lieu à des quêtes annexes.

Une dernière possibilité est de limiter la quantité de monnaie qu'il est possible de transporter, comme avec le système de bourses utilisé dans les *Zelda*. Notez que ceci est plus facile à mettre en œuvre dans un jeu où les quantités d'argent mises en jeu sont relativement petites, typiquement dans les A-RPG.

3. Les commerces spéciaux

Il existe aussi d'autres types de commerces qui ne font pas intervenir la monnaie. L'exemple le plus simple est celui du troc, qui permet une approche plus intéressante que l'achat simple : pour obtenir un objet convoité, le joueur devra aller chercher une monnaie d'échange très précise, ce qui peut donner lieu à de nombreuses quêtes — citons ici le marché de *Secret of Evermore*. Néanmoins, la généralisation du troc à tout un jeu peut poser problème si jamais le joueur doit gérer de gros stocks d'objets. Ceci fonctionne généralement mieux dans des jeux où l'inventaire est très réduit, la plupart du temps dans les A-RPG. Le marché permet par ailleurs au joueur de faire jouer la concurrence et de récompenser ses efforts, s'il a pris le temps de tout bien regarder et de comparer les prix intelligemment.

Une salle aux enchères constitue aussi une forme particulière de commerce. Elle propose des objets plus intéressants et en nombre plus limité. Mais surtout, elle fait intervenir le temps. Celui-ci permet de faire monter le stress et met en valeur le caractère stratégique de l'achat : le joueur doit savoir doser la mise et choisir le bon moment pour la placer, afin d'être sûr d'obtenir l'objet qu'il veut. C'est un moyen de donner plus d'intérêt à l'achat, en le présentant comme la récompense d'un petit challenge.

II. L'implémentation du commerce

1. Différents types d'objets

Le premier problème qui se pose à nous dans l'implémentation du commerce est le suivant : que peut-on acheter ? Essayons de distinguer les différents types d'objets que le joueur pourra acheter.

Les « consommables » (en anglais, consumables) : ce sont les objets qui peuvent être utilisés une seule fois — typiquement les objets de soin. On peut varier en proposant des objets qui peuvent être consommés en plusieurs fois — une potion bu en plusieurs gorgées par exemple.

Les upgrades immédiats : ce sont les équipements que le joueur peut équiper immédiatement après les avoir achetés et qui lui permettent de modifier ses caractéristiques — la plupart du temps, ce sont les épées, les armures, etc.

Les upgrades futurs : ce sont des objets qui ne peuvent pas être équipés mais qui peuvent devenir des objets d'équipement. Par exemple, une épée cassée ou un minerai, qui une fois transformés, deviendront des équipements de grande valeur pour le joueur. Ce sont des objets très importants si vous décidez d'implémenter un système de forge ou de création d'items.

Les objets qui permettent de gagner de l'argent: il s'agit la plupart d'objets rares, uniquement décoratifs, qui n'ont d'autre utilité pour le joueur que d'être vendus. Un exemple particulier : la bouteille de vin dans *Star Ocean III*, dont le prix ne cesse d'augmenter tout au long du jeu. Vaut-il mieux la vendre maintenant qu'elle est bien chère ou attendre encore qu'elle le soit beaucoup plus ?

Les objets funs et/ou inutiles : la plupart du temps, ce sont des bonus (par exemple des artworks ou des musiques du jeu), qui peuvent être achetés avec une monnaie spéciale (des points ou la monnaie d'un casino par exemple). On peut aussi penser aux costumes secondaires — souvent délirants — qui permettent de pousser l'aspect « customisation » plus loin.

Les magies : plus rarement, les magies peuvent prendre la forme d'objets, consommables ou pas. Par exemple, les pièces magiques de *Seiken Densetsu III* sont un exemple de magie consommable, alors que les magies vendues dans les échoppes de *Final Fantasy V* sont plutôt du type équipement, c'est-à-dire non consommable.

Les objets nécessaires au gameplay : un peu différents des objets nécessaires au scénario, ce sont les objets utiles à l'avancée du joueur, comme les bombes dans les *Zelda*.

Notez que les objets nécessaires au scénario ne peuvent classiquement pas être achetés (mais c'est peut-être une piste à explorer pour innover, qui sait ?).

2. Fonctionnalités

Deuxième problème : quelles sont les options proposées au joueur lorsqu'il parle avec un marchand et comment les rendre attrayantes ?

Un élément essentiel pour répondre à cette question est une bonne ergonomie. On doit toujours garder à l'esprit qu'un joueur ne vient pas passer quinze heures à marchander et à revendre la moindre queue de lézard, mais pour trancher du dragon avec une épée magique (ou quelque chose d'approchant). Par conséquent, les menus d'achat doivent être pratiques et simples d'utilisation. Si le joueur doit passer par une tripotée de menus chaque fois qu'il parle à un marchand, le moindre achat de potion risque de devenir un véritable calvaire pour lui. Je vous renvoie à un futur article de Fil pour l'optimisation de vos menus.

Je me permets tout de même de suggérer quelques petites choses, bien que ce ne soit pas mon domaine de prédilection :

- Minimiser les temps d'affichage des fenêtres.
- Alléger les fenêtres de façon à n'avoir que les informations nécessaires à l'écran
- Montrer les améliorations apportées par les équipements directement dans le menu d'achat — ceci évite au joueur d'insupportables allers-retours entre le menu d'achat et le menu d'équipement.
- Permettre de sélectionner rapidement un grand nombre d'objets, ce qui évite d'avoir à patienter au joueur s'il veut acheter une cinquantaine de potions d'un coup.
- Proposer une option semblable à l'option Troquer de *Grandia*, qui permettait d'échanger l'armure ou l'arme équipée contre un autre équipement du même type et de l'équiper directement en payant simplement la différence.
- Utiliser des bruitages agréables qui marquent l'achat ou la vente — *Final Fantasy XII* est un exemple du genre.

On pourra citer l'exemple de la série de *Tales Of*, dont les menus sont un modèle de clarté et d'ergonomie.

Ensuite, quelles sont les possibilités offertes au joueur ?

Dans la quasi-totalité des RPG, il peut acheter et vendre des objets. Bien que cela reste basique, on peut déjà travailler sur ces options, par exemple en jouant sur les prix d'achat et de vente au cours du temps ou selon le lieu où se trouve le joueur.

Il s'agit malheureusement souvent des seules options proposées, ce qui rend le commerce dans les jeux peu attrayant. Intéressons nous à la façon dont les actions du joueur peuvent rendre les choses plus intéressantes.

Une première façon de procéder consiste à donner un rôle à la réputation ou au charisme du personnage. Un personnage reconnu comme un bon samaritain aura droit à des réductions dans les magasins les plus classiques, mais aura droit à des prix prohibitifs s'il veut se fournir dans les bas-fonds. On peut aussi permettre au joueur de discuter lui-même avec le marchand pour essayer de l'amadouer, au risque de s'attirer ses foudres et donc une majoration des prix — horreur ! *Starfox Adventures* implémentait par exemple un système de négociation des prix qui donnaient lieu à de petites scènes amusantes.

Shadow Hearts II avait implémenté de façon intéressante le concept de ristourne. Le joueur peut obtenir de meilleurs prix d'achat ou de vente s'il réussit un mini-jeu d'adresse faisant intervenir une roulette. Ceci avait le mérite de rajouter un peu de piment aux achats et de conserver une cohérence avec le reste de l'univers — la roulette en question servant aussi dans les combats.

Pour intéresser le joueur, plutôt que de proposer une baisse de prix, on peut aussi lui proposer un objet plus précieux. C'est le concept à la base des « Affaires ». Imaginez vous, au détour d'une conversation, convaincre un marchand un peu méfiant de vous laisser voir dans l'arrière-boutique des artefacts moins licites mais beaucoup plus alléchants : c'est quand même bien mieux qu'un simple marchand de base, bonhomme et bien prévisible qui ne vous propose que des potions de soin.

Valkyrie Profile II propose un système de ce genre : de nouveaux objets, plus précieux, apparaissent au fur et à mesure que vous vendez les bons objets aux marchands. Ce système a le mérite d'inciter le joueur à faire du commerce et rend les achats plus intéressants : les objets à débloquent sont plus précieux et la collecte des objets à vendre s'apparente presque à une mini-quête en soi.

Final Fantasy XII va plus loin encore en proposant une option baptisée « Affaires » chez les marchands. Au bout d'une certaine quantité d'achats et de ventes, les marchands proposent au joueur des lots d'objets dont ils ne connaissent pas le contenu — celui est vaguement évoqué par le nom du lot — mais qui sont généralement très intéressants. Le joueur est alors récompensé de son investissement dans le commerce par des objets précieux et les transactions gagnent en intérêt grâce à une nouvelle facette du gameplay : le pari. Une récompense est d'autant plus précieuse qu'on a pris des risques pour l'obtenir.

Après l'achat, la vente, la négociation et les affaires, on peut enfin s'intéresser à la malhonnêteté. En effet, qui n'a pas rêvé dans un jeu, face à un marchand à l'air mauvais et aux tarifs repoussants, de pouvoir

s'emparer des multiples artefacts contenus dans ses coffres sans avoir à verser la moindre pièce d'or ? La malhonnêteté est particulièrement intéressante dans des jeux où le charisme ou la réputation ont des rôles importants à jouer — je pense plus particulièrement aux mécaniques de jeu des RPG PC.

Il faut cependant faire attention à bien doser : si voler un marchand vous rapporte l'intégralité de son stock, il vaut mieux avoir prévu que celui-ci ait des stocks finis, contrairement à un *Final Fantasy* par exemple. De plus, il peut être très intéressant du point de vue de la cohérence de l'univers d'intégrer le vol au reste du jeu, par exemple en faisant en sorte qu'un commerçant cambriolé ou trop souvent arnaqué refuse de vous vendre quoique ce soit, voire appelle à lui la garde de la ville. A vous aussi de voir comment présenter le vol : cambriolage discret ou braquage à main armée ? Rappelons pour mémoire la façon dont, dans *The Legend of Zelda : Link's Awakening*, le héros se retrouvait renommé en Voyou si le joueur volait dans la boutique du village Cocorico.

3. L'acquisition de l'argent

Troisième problème : comment récupérer-t-on de l'argent ?

Classiquement, c'est soit en effectuant des quêtes, soit en le récoltant sur le corps des ennemis tués que le joueur peut récupérer de l'argent. Néanmoins, on se retrouve vite très limité par cette vision des choses. Mais comment innover sur ce point ?

Je citerai deux exemples : *Final Fantasy XII* permet au joueur de gagner à la fin des combats, non pas de l'argent, mais des objets destinés à la revente — divers butins comme des cornes, des écailles, des griffes, etc. Ceci permet d'assurer un système équivalent au gain d'argent classique et en même temps, de résoudre le problème de cohérence suivant : comment se fait-il le moindre slime traîne sur lui une bourse dont la valeur est strictement définie ? De plus, la multitude d'objets à vendre et les descriptions qui vont avec ajoutent à la profondeur de l'univers du jeu. Il aurait été intéressant de pousser l'immersion plus loin en assurant que les ennemis humanoïde comme les soldats impériaux laissent quand même de l'argent, ce qui serait logique.

Final Fantasy VIII de son côté — et oui, cette série est beaucoup plus riche d'innovations qu'on pourrait le penser au premier abord — proposait tout simplement de supprimer l'argent gagné à la fin des combats et de le remplacer par un salaire reçu à intervalles réguliers, dont la valeur dépendait du rang du personnage — rang qui pouvait être augmenté à travers des examens de connaissance. Bien qu'il soit dommage que la façon de faire varier le rang soit si peu variée, on retiendra un élément très original : le jeu regorgeait d'idées très nouvelles pour un genre aussi codifié que le RPG.

III. Problèmes de vraisemblance

Maintenant que nous avons examiné différents aspects du commerce dans les RPG et la façon dont on peut les implémenter, intéressons nous aux différents écueils que l'on doit éviter. Pour la plupart, il s'agit de « clichés » ou plutôt de mécanismes tellement anciens qu'ils passent complètement inaperçus, alors qu'ils sont absurdes quand y réfléchit bien.

1. Le marchand trou noir

Commençons par un cas classique pour les RPG console — et peut-être moins pour les RPG PC : le marchand trou noir. Vous avez sans doute remarqué que dans de nombreux jeux, les marchandises vendues disparaissent immédiatement, alors que le marchand n'a pas eu le temps de revendre l'objet : c'est assez peu crédible. Une solution souvent consiste à permettre aux marchands de tout simplement garder les marchandises que le joueur leur a vendues, de façon à pouvoir les racheter en cas de pépin. On peut imaginer affiner cette solution en faisant disparaître les objets vendus au marchand à plus ou moins long

terme de façon à faire circuler les stocks, quitte à retrouver ces objets à d'autres endroits — par exemple un marché noir.

2. Inflation et progression

Un autre écueil caractéristique des jeux de rôle (et qui s'applique d'ailleurs à d'autres domaines comme la montée en puissance des monstres) est la façon dont l'évolution des marchandises et des prix est adaptée au joueur. Les armes sont de plus en plus puissantes à chaque nouvelle ville et offrent des capacités parfaitement ajustées au donjon du coin. Ceci est certes pratique, mais rend le commerce « automatique » et constitue une atteinte à la crédibilité de l'univers.

Une solution consiste à varier les prix en se rappelant que le joueur ne pourra jamais tout acheter : vous pouvez lui proposer dès le départ des armes extrêmement puissantes mais très chères — si par exemple la ville de départ est célèbre pour ses armes — de façon à ce qu'il ne puisse pas les acheter. Une autre façon de procéder consiste à jouer sur des armes dont les capacités ne peuvent être utilisées que si le personnage est suffisamment puissant : telle armure ou telle épée n'aura pas d'intérêt pour le joueur pour un niveau ou des caractéristiques trop faibles pour exploiter leurs véritables capacités.

Ce problème de la progression des prix est souvent lié à un autre écueil, celui du joueur trop riche. Souvent, à la fin du jeu, le joueur dispose d'une quantité d'argent tellement impressionnante qu'il pourrait sans doute acheter une ville (ayons une pensée pour la série des *Star Ocean*) ! Il faut veiller à ne pas faire pleuvoir l'argent, même en fin de jeu et surtout pas en tuant des monstres : plus fort ne veut pas dire plus riche. De plus, rappelez vous que l'achat d'une arme qui demande un effort financier — par exemple de faire des quêtes pour se procurer plus d'argent — est au final beaucoup plus valorisant pour le joueur qu'un achat automatique : rien de pire que les prix des équipements dosés pour que le joueur puisse les acheter de justesse. Si le joueur a sué pour avoir ses armes, il sera d'autant plus content quand il les obtiendra : on comparera à cet effet le plaisir d'obtenir une nouvelle arme dans un *Final Fantasy* et un *Dragon Quest* par exemple.

3. Gestion des stocks

Le summum de la crédibilité du point de vue de commerce, bien qu'il s'agisse de quelque chose de très difficile à réaliser, consisterait à programmer une gestion réaliste des stocks, c'est-à-dire de faire varier le type, le nombre et la valeur des produits disponibles en fonction des arrivages, voire des événements du scénario. Il semble logique qu'une guerre provoquera une pénurie de certains produits ou que la destruction d'une région amènera à une diminution de la quantité des produits locaux disponibles.

Il est aussi important de considérer la répartition des objets entre les différents types de commerce : dois-je faire un magasin unique ou une multitude de petites échoppes ? Séparer les armes et les armures ou les réunir ? A vous de voir, de façon à rester cohérent et à varier si possible. Il peut être intéressant de réserver certains types d'objets plus rares ou plus précieux à des commerces spéciaux, principalement les marchés et les enchères (licites ou non).

Au final, qu'est-il important de retenir de tout cela et que faut-il pour bien implémenter le commerce dans un RPG ?

Gardez en mémoire que vous devez parvenir à un équilibre entre l'aspect immersif du commerce (qui permet d'approfondir le background du jeu) et son aspect pratique (il faut que cela reste simple pour le joueur) : le joueur n'a pas acheté ou téléchargé le jeu pour passer son temps à compter des haricots. Le commerce doit être pratique (menus ergonomiques, mise en avant des aspects interactifs comme la négociation ou la spéculation, mais sans JAMAIS les imposer), il doit récompenser le joueur (un bon achat doit permettre de voir une modification significative des performances hors des menus et donner un sentiment de réussite au joueur) et il doit être dynamique (marchands sédentaires ou ambulants, arrivage de nouveaux stocks, influence de l'histoire sur les prix et les disponibilités).

Il est d'autant plus important de penser à cet aspect du RPG qu'il apparaît de plus en plus dans d'autres genres, à travers les upgrades d'un Shoot Them Up ou un système d'équipement dans un FPS : il serait dommage de passer à côté d'un moyen intéressant d'enrichir le gameplay du jeu.

Une leçon de symétrie

Par [Pierre Guyot](#) | le 17 août 2008 02:58:43 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (octobre 1998).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Les plus observateurs d'entre vous auront pu remarquer que je n'ai pas publié d'article le mois dernier. J'étais en vacances à Oxford, en Angleterre, où j'étudiais l'art, l'architecture, l'histoire et la sociologie de la maison anglaise. Vous vous demandez sans doute pourquoi un développeur de jeux PC voudrait être capable d'expliquer la différence entre décorations baroque et rococo ou pourquoi un libéral voudrait étudier les extravagantes propriétés d'un groupe de gens qui ont vécu des vies de dépenses ostentatoires tandis que les classes ouvrières opprimées trimaient au loin dans la chaleur... Où en étais-je ? Bien, de toute façon, pour faire court, la réponse est que si les développeurs de jeux n'apprennent pas quelque chose de nouveau de temps en temps, la totalité de l'industrie du jeu finira dans le pétrin, voilà pourquoi.



L'une des choses que la Renaissance a apportées à l'architecture anglaise est l'idée classique de symétrie, selon laquelle les bâtiments étaient plus beaux quand le côté gauche était l'image réfléchie du côté droit. Ceci ne constituait pas vraiment une idée nouvelle, mais jusqu'à la Renaissance, les gens n'avaient pas eu trop d'effort à fournir pour l'appliquer à leurs maisons. La symétrie s'applique à toutes sortes de choses en dehors de l'architecture : elle s'applique à la conception artistique, littéraire et même musicale, et elle s'applique de même au jeu vidéo. Ceci est un article sur le rôle de la symétrie dans le game design.

J'ai déjà écrit auparavant sur la façon dont l'équilibre constitue l'essence même du game design. Dans le cas d'un jeu multijoueurs, l'équilibre réside dans la condition fondamentale d'équité, la nécessité que tous les joueurs aient les mêmes chances de gagner au début du jeu. Dans le cas des jeux solos (comme la plupart des jeux vidéo), l'équilibre réside dans le fait que le jeu ne doit être ni trop simple, ni trop difficile.

Le problème est moins celui de l'équité que de celui de fournir un défi et une possibilité de gagner raisonnables. L'un des meilleurs moyens de garantir le bon équilibre d'un jeu multijoueurs au début est de le rendre symétrique, c'est-à-dire, de faire en sorte que tous les joueurs jouent selon les mêmes règles et commencent avec les mêmes ressources. Si vous essayez d'équilibrer les deux côtés d'une balance à plateaux, la façon la plus simple de procéder est de placer les mêmes objets de chaque côté, id est de les placer symétriquement.

Les échecs, les dames, le Monopoly et beaucoup d'autres jeux simples sont symétriques : chaque parti débute avec les mêmes ressources que les autres. Même dans un jeu parfaitement symétrique comme les échecs, il reste un élément d'asymétrie qu'il n'est pas possible d'éviter : quelqu'un doit commencer la partie. Dans beaucoup de jeux, comme le morpion, commencer la partie procure un avantage. Il existe plusieurs façons de réduire l'effet du joueur qui commence la partie. L'une d'entre elles consiste à configurer le jeu de telle sorte que le mouvement initial ne fournisse qu'un très petit avantage stratégique.

Aux échecs, par exemple, les règles du jeu sont faites de telle sorte que vous puissiez uniquement déplacer un pion ou un cavalier au premier tour. Celles-ci sont les deux pièces les plus faibles du jeu, si l'on ne

compte pas le roi. Ainsi, l'avantage conféré n'est pas significatif. De plus, les pièces du jeu sont séparées par quatre rangées, si bien qu'aucune pièce seule ne peut prendre ou même véritablement menacer une pièce ennemie au premier tour de jeu. Une autre façon pour réduire l'effet du commencement de la partie est de rendre le jeu suffisamment long, de façon à ce que faire le premier mouvement ne constitue qu'un très petit avantage sur l'ensemble du jeu.

Le morpion est un jeu très court, aussi commencer la partie constitue un avantage extrêmement précieux et celui qui joue deuxième joue habituellement en défensive pour tout le reste du jeu. Pour un jeu plus long comme les dames ou les échecs, cela n'a pas tellement d'importance. Enfin, un jeu peut faire intervenir le hasard pour réduire cet effet du premier tour. Le Monopoly et le backgammon sont des jeux dans lesquels les joueurs lancent des dés pour se déplacer et puisque le premier joueur pourrait très bien faire un mauvais lancer et le deuxième joueur un bon lancer, les déplacements sont beaucoup plus affectés par la valeur du lancer que par autre chose.

Dans les jeux vidéo, le problème de qui commence la partie ne se pose habituellement pas, puisque le nombre des jeux en temps réel a très largement dépassé celui des jeux au tour par tour. Et les quelques jeux basés sur le tour par tour qui existent, comme *X-Com*, prennent habituellement beaucoup trop longtemps pour que cela fasse une quelconque différence au final. Néanmoins, j'inclus ceci ici car je pense qu'un game designer devrait aussi être capable de concevoir des jeux papier, et il s'agit d'une question importante pour les jeux papier.

Une variante simple de la symétrie que l'on peut trouver dans les échecs, les dames, Stratego ou les autres jeux du genre est la symétrie « de rotation » que l'on trouve dans Rochambeau (pierre-feuille-ciseaux). Dans Rochambeau, deux personnes choisissent au hasard l'une des trois figures suivantes : la pierre, la feuille ou les ciseaux. Le vainqueur est déterminé par la formule suivante : les ciseaux coupent le papier et le battent ; le papier enveloppe la pierre et la bat ; la pierre brise les ciseaux et les bat. Si les deux joueurs ont choisi la même figure, ils recommencent.

Ce mécanisme existait aussi dans un vieux jeu *Brøderbund, The Ancient Art of War*. Dans ce jeu, les chevaliers avaient l'avantage sur les barbares, les barbares avaient l'avantage sur les archers et les archers avaient l'avantage sur les chevaliers (*NdT : c'est aussi ce principe que l'on peut retrouver dans la série Fire Emblem ou dans les simili-échecs de Gemity dans Star Ocean III**). Un élément supplémentaire de ce type de jeu était l'information cachée : le fait que l'un des joueurs ne sache pas laquelle des trois options l'autre joueur va choisir. Ceci aboutissait à plus de psychologie dans un autre jeu où l'information est complète, comme les échecs ou les dames.

Avertissement sur l'asymétrie

Comme je l'ai dit, la symétrie est la façon la plus simple de rendre un jeu équitable, mais elle a tendance à souligner la nature artificielle de la compétition. Les jeux sont souvent plus intéressants et donnent une plus grande sensation de « réel » quand ils contiennent des éléments d'asymétrie. Fox and Geese (*NdT : Renard et Oies en français*) est un exemple de jeu de plateau très ancien qui contient de l'asymétrie. Dans Fox and Geese, l'un des joueurs contrôle une pièce (le renard) et l'autre en contrôle dix-sept (les oies). Le renard peut se déplacer dans toutes les directions et sauter par-dessus les oies comme aux dames, en les enlevant du plateau. Les oies peuvent uniquement se déplacer vers le renard et ne peuvent pas sauter par-dessus. Les oies gagnent si elles coincent le renard entre elles de façon à ce qu'il ne puisse pas bouger. Le renard gagne s'il saute par-dessus suffisamment d'oies et qu'il n'en reste pas assez pour l'encercler. Les jeux de guerre, qui se présentent souvent comme des simulations d'événements historiques, sont souvent asymétriques parce que les effectifs, l'équipement et les positions sur le terrain sont asymétriques dès le départ. En conséquence, équilibrer un jeu de guerre de façon à ce que chaque joueur ait les mêmes chances de gagner est un défi considérable. Souvent, ceci est atteint en donnant des conditions de victoire différentes à chaque parti (ce que nous avons aussi vu dans Fox and Geese). Dans le cas d'une énorme armée qui assiège une petite garnison, la condition de victoire pour la garnison n'est pas de vaincre l'énorme armée, ce qui est clairement

impossible, mais de tenir un certain laps de temps ou un certain nombre de tours. Si l'armée se rend maître de la garnison dans le temps imparti, elle gagne ; sinon, elle perd.

L'un des jeux vidéo asymétriques les plus populaires est *Starcraft*. *Starcraft* est un jeu qui nécessite de construire une série de bâtiments à mesure que vous construisez une armée ; ceci se retrouve dans *Command & Conquer* et *Dungeon Keeper*. Dans *Starcraft*, les armées appartiennent à l'une des trois races suivantes : les Terrans, les Protoss et les Zerg. Les fonctions des bâtiments sont relativement semblables d'une race à l'autre, mais l'armement des unités (en particulier celles de haut niveau), leurs coûts de production, leurs mécanismes de production et leur durée de vie sont toutes très différentes.

En général, les unités Protoss sont très fortes mais aussi très coûteuses, alors que les unités Zerg sont beaucoup moins chers et plus faibles, les unités Terran se situant à mi-chemin entre les deux. Les unités Zerg se régénèrent d'elles-mêmes avec le temps ; les unités Terran peuvent être réparées facilement, mais seulement par une unité spéciale de réparation ; les unités Protoss ne peuvent être soignées ou réparées, mais utilisent des boucliers rechargeables pour se défendre. Cette asymétrie a, bien entendu, donné naissance à moult débats sur qui est la meilleure des trois races. Si vous lisez les ajouts à la fin du manuel de jeu, il est clair qu'il y a eu des changements de dernière minute afin d'accentuer les caractéristiques de certaines des races, car quelques-unes des fonctionnalités auxquelles le manuel fait référence ne sont pas dans le jeu.

Dans les jeux vidéo, il est plus simple d'équilibrer les éléments asymétriques, car le roulement de dé se fait loin du regard et vous pouvez truquer les probabilités sans que le(s) joueur(s) le sache. Je ne sais pas si cela a été fait dans *Starcraft*, mais il semble que oui.

Il y a une autre forme de symétrie, ou du moins d'équilibre, à considérer, et c'est l'équilibre entre les différents types de tactique qui permettent à un joueur de gagner le jeu. Les jeux contiennent souvent des erreurs de design qui permettent au joueur d'exploiter une lacune dans les règles pour gagner le jeu en répétant une seule tactique.

Ceci est particulièrement vrai pour les simulations multijoueurs online où intervient un élément économique. Le designer veut et attend que le joueur utilise tout un panel de tactiques, mais en raison d'une erreur de design, une tactique fonctionne tellement mieux que les autres que les joueurs abandonnent toutes les tactiques sauf celle-ci, ce qui rend le jeu plutôt ennuyeux. En général, on veut forcer les joueurs à adopter un panel de tactiques variées pour rendre le jeu plus intéressant. Une simulation de football (NdT : il s'agit de ce qu'on appellerait du football américain, et non pas de « soccer ») professionnel sur laquelle je travaille reproduit naturellement les règles du vrai football. Le football était autrefois un jeu particulièrement symétrique, tout particulièrement quand les joueurs jouaient à la fois en attaque et en défense.

Avec l'avènement de tactiques spécialisées dans l'attaque seule ou dans la défense seule, le jeu est devenu asymétrique, au moins pendant _ — bien sûr, il ne s'agit pas d'asymétrie dans le sens où les deux équipes jouent des règles différentes ou ont des objectifs différents. En conséquence, les règles sont constamment modifiées pour essayer de garder un équilibre entre l'attaque et la défense. De façon similaire, il y a un équilibre entre courir avec la balle et la passer. Dans le football américain professionnel, faire des passes est à peine plus important que courir, mais les équipes doivent maîtriser les deux pour gagner. Les règles du football canadien visent un équilibre différent, qui donne une importance significativement plus grande au jeu de passes et à l'attaque d'une façon plus générale.

Revenons à la maison de campagne anglaise pour quelques minutes : un autre développement intéressant est l'invention du « landscape gardening » (NdT : qui peut se traduire par *paysagisme en français*, mais fait surtout référence aux jardins « à l'anglaise », par opposition aux jardins « à la française »). Au lieu de concevoir des jardins de fleurs « formels » (et souvent hautement symétriques), des jardiniers comme Lancelot « Capability » Brown ont conçu de véritables paysages de pelouses, de sculptures, de chutes d'eau, de taillis et de petits bâtiments – habituellement des imitations de lieux saints romans. L'un des principes de ses conceptions était la surprise. Suivre un chemin amène à une vue inattendue ou à une statue qui était cachée depuis la maison. Le jardin-paysage encourage – et récompense – le penchant du visiteur pour l'exploration.

C'est un autre principe intéressant à considérer dans le game design. Un parallèle évident est à faire avec les jeux d'aventure, où l'exploration est le cœur du jeu, mais ceci peut aussi bien s'appliquer à d'autres sortes de jeux. Un jeu n'a pas exactement besoin d'avoir la même forme du début à la fin. Une surprise inhabituelle ou un twist inattendu qui surgit à mi-chemin peut ravir, encourager et récompenser le joueur. D'une certaine façon, ceux-ci représentent une asymétrie dans le temps : le début du jeu et sa fin ne sont pas symétriques. Cependant, c'est une épée à double tranchant : il est bon de ne pas changer le style de jeu de façon trop dramatique.

Par exemple, *Heart of China* commençait comme un jeu d'aventure plus ou moins conventionnel, mais, à partir d'un certain point, se ramenait d'un coup à une simulation 3D de tank assez grossièrement implémentée. Si vous ne réussissiez pas cette épreuve, vous ne pouviez pas continuer l'aventure. C'était très frustrant : si j'avais voulu une simulation de tank, j'aurais acheté une simulation de tank. Par-dessus tout, la surprise, quelle qu'elle soit, devrait coller naturellement au jeu, pour donner l'impression qu'elle en fait partie. La symétrie dans le game design est simple, facile et intuitive, mais elle amène souvent une certaine artificialité, le sentiment « d'être dans un jeu » que nous essayons de nos jours d'éviter. L'asymétrie est un outil très puissant pour générer de l'intérêt et du réalisme, mais elle complique de façon substantielle le design et la conception d'un jeu. Utilisez-en avec soin.

Trois problèmes de la narration interactive

Par [Pierre Guyot](#) | le 18 août 2008 14:04:06 | Catégories : [Ernest Adams](#), [Game Design](#), [Scénario](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (décembre 1999).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

L'article du mois dernier sur les jeux d'aventure a déclenché une si forte réaction que j'ai pensé discuter d'un important problème qui y est lié, pendant que j'ai l'attention de tout le monde.

La narration interactive est un sujet de débat très important depuis que les premiers jeux vidéo ont été créés. Beaucoup des premiers développeurs de jeux étaient des programmeurs sans expérience dans le domaine de l'écriture d'une fiction, si bien qu'il y avait une véritable pénurie de talents pour créer des éléments comme les personnages et pour rythmer une intrigue. Depuis lors, les écrivains professionnels sont entrés dans l'industrie et la qualité de notre narration a quelque peu augmenté.



En dépit de ça néanmoins, il reste une question philosophique plus vaste qui se dessine au-dessus du sujet : « Qu'est-ce que cela signifie quand on dit qu'une histoire est interactive ? ». C'est une question qui reste aujourd'hui sans réponse. Vous pourriez avancer qu'aucune réponse n'est nécessaire – les jeux d'aventures racontent des histoires et ils sont interactifs, par conséquent ils constituent une narration interactive, et il n'est pas nécessaire d'en discuter plus avant. Le problème est que la plupart des jeux d'aventure racontent des histoires plutôt pauvres. Nous n'avons jamais eu jusqu'ici vu de jeux d'aventure qui était du niveau des travaux de Dickens ou de Maupassant.

Je crois que la narration interactive souffre de trois très gros problèmes et ceux-ci sont clairement visibles dans les jeux d'aventure aujourd'hui.

Le problème de l'amnésie

Il s'agit du plus simple et du plus évident de ces problèmes. Dans une histoire normale, non interactive, les personnages appartiennent à un monde dont ils font partie. Ils comprennent ce monde. Ils savent ce qu'il y a dans chacun des tiroirs de leur appartement et ce qu'il y a dans toutes les boutiques de leurs villes. Quand ils se lèvent le matin, ils ne commencent pas la journée en ouvrant le moindre compartiment pour voir ce qu'il y a dedans, ni en prenant tous les objets qu'ils voient et en les fourrant dans leurs poches, au cas où ça leur serait utile plus tard.

Mais ceci n'est pas vrai pour les jeux d'aventure, n'est-ce pas ? Quand vous jouez à un jeu d'aventure, vous n'avez aucune idée de ce qui est en train de se passer. Vous souffrez d'amnésie. Même si le jeu débute dans votre propre maison, vous ne savez ce qu'il va vous arriver, alors, pour votre propre sécurité, vous prenez tout ce que vous pouvez voir et vous finissez par transporter une collection d'objets qui vous font ressembler à une clocharde folle (prenons l'exemple du jeu *Adventure* : une lampe, une cage à oiseaux, un bâton en bois, une hache, des pièces d'or, une bouteille d'huile...)

Quelques jeux ont réellement été écrits afin d'incorporer ce problème à l'intrigue. Il y avait un jeu simplement intitulé *Amnesia*, publié par Electronic Arts ; et il y avait un jeu basé sur une série de romans de fantasy de Roger Zelazny, *The Chronicles of Amber*, qui commençait avec un personnage amnésique. Mais soyons réalistes, ceci ne constitue pas un genre majeur de la littérature. Il y a très peu de romans au sujet des amnésiques. Dans la plupart des histoires, les personnages foncent juste et ont leurs aventures, et c'est à l'auteur de s'arranger pour qu'ils transportent tout ce dont ils ont besoin pour y survivre (s'ils sont censés y survivre).

Il y a trois types d'histoires dans lesquelles les personnages commencent les mains vides et ignorants et doivent deviner les choses par eux-mêmes. Le premier type est de celui de l'histoire le-bleu-dans-une-nouvelle-situation — la nouvelle recrue qui vient juste de rejoindre son bateau dans la marine ou le bandit armé qui vient d'être nommé shérif dans une ville de l'ouest. Dans ce cas, le fait que le protagoniste doive faire beaucoup d'exploration avant de pouvoir faire quoique ce soit fait sens. Les deux autres types sont les mystères à résoudre et les quêtes héroïques — deux situations qui impliquent de beaucoup parler à des étrangers et d'examiner des objets qui ne sont pas familiers au joueur.

Cela donne alors tout son sens au fait que la plupart des jeux d'aventures sont en fait des enquêtes, des quêtes héroïques ou des scénarios du type un-nouveau-en-ville. Il n'y a rien de particulièrement gênant avec ça, mais cela signifie que le genre est limité par le problème de l'amnésie. Nous pouvons créer des histoires interactives, mais nous ne pouvons pas créer n'importe quel type d'histoire dont nous avons envie.

Le problème de la cohérence interne

Quand nous jugeons une œuvre de fiction, nous la jugeons sur un certain nombre de critères : les descriptions sont-elles claires ? Les dialogues sont-ils crédibles ? L'écriture est-elle fluide ? Et ainsi de suite. Mais nous prononçons aussi un jugement bien plus fondamental. Si vous sortez d'une séance de cinéma ou si vous reposez un livre que vous venez de finir, et que vous vous dites « Je ne pense pas que j'aurais fait ça » ou « Je ne pense pas qu'elle aurait réagi de cette façon dans une telle situation », alors l'histoire présente un défaut. Il y a quelque chose qui ne va, ça ne fait pas sens. Toute histoire doit être cohérente vis-à-vis de ses propres lois. Elle doit être cohérente. A tout moment de l'histoire, la situation à ce moment doit être cohérente avec tout ce qui a pu se passer auparavant.

Les enquêtes sont un exemple intéressant de ceci, parce que dans une enquête, il y a beaucoup d'explications possibles différentes pour le crime et jusqu'à ce que le détective rassemble tout le monde dans une pièce à la fin et révèle quelle est la vraie explication, chacune d'entre elle doit sembler plausible. Mais les règles du genre imposent que seul l'un d'entre elles fonctionne réellement : le reste doit être logiquement impossible et de surcroît, l'auteur doit avoir montré tous les indices au lecteur. C'est une tâche très difficile de créer

quatre ou cinq explications possibles et apparemment cohérentes et de les introduire d'une façon telle que tous les indices soient là, mais que le lecteur soit tout de même surpris d'apprendre quelle est la bonne explication.

Cette nécessité d'une cohérence interne n'est pas un problème de pure logique, bien sûr. Je ne veux pas suggérer que tout point de l'histoire devrait être mécaniquement déductible, comme une preuve mathématique, de ce qui l'a précédé. Mais si vous revenez sur l'histoire, elle devrait être cohérente. Les histoires ne devraient pas être prédictibles mais elles devraient faire sens d'une façon satisfaisante.

Alors, qu'est-ce que tout cela a à faire avec l'interactivité ? La réponse est : rien. L'interactivité concerne la liberté. L'interactivité concerne le fait de donner au joueur des choses à faire et de le laisser les faire lui-même. Le cœur du média interactif est de laisser le joueur faire quelque chose de lui-même. Le problème est que dans beaucoup de cas, votre joueur va sortir des rails et faire quelque chose de totalement déroutant, qui ne peut pas être anticipé. Et ça ne fonctionne pas très bien dans un scénario.

Considérons Superman. Superman est un personnage qui est congénitalement incapable d'ignorer un bébé qui pleure dans un bâtiment en flammes. Il ne dit jamais : « En fait, je vais laisser quelqu'un d'autre s'en charger à ma place pour cette fois ». Mais que se passe-t-il si notre joueur est Superman dans un jeu vidéo ? Voici le bâtiment en flammes. Est-ce qu'il fonce à l'intérieur et sauve le bébé ? Et bien, il doit le faire s'il est Superman et s'il ne le fait pas, alors il contredit la nature même de Superman. Il y a un conflit qui apparaît entre le désir du joueur de faire comme il l'entend et votre désir de lui imposer une intrigue et un personnage. Et c'est un problème très difficile. Comment être sûr que le joueur va faire quelque chose qui soit cohérent, qui aille bien avec votre histoire ?

Le problème du flot narratif

Comme nous l'avons tous étudié en cours d'anglais au collège (*NdT : on pourra remarquer la différence avec le schéma narratif tel qu'on l'étudie en France*), toute histoire possède supposément une introduction, un mouvement ascendant, un point culminant (climax), un mouvement descendant et une conclusion. C'est le travail de l'auteur de la structurer de telle sorte qu'elle amène un point culminant dans le drame – une action, une confrontation ou tout autre événement à même de résoudre la tension intérieure à l'histoire. L'un des problèmes auxquels l'auteur fait face est de faire en sorte que tous les personnages impliqués soient prêts – psychologiquement et physiquement – pour le climax puisse avoir lieu. Si ce n'est pas la cas, alors nous lisons l'histoire et disons quelque chose comme « Attendez une minute, d'où est-ce que vient ce couteau ? » ou « Comment est-ce qu'il savait que le méchant se cacherait dans le placard du hall ? ».

Dans une fiction ordinaire, c'est un défi, mais au moins, vous, en tant qu'auteur, vous occupez de tout. Les personnages doivent aller là où vous leur dites d'aller, savoir ce que vous voulez qu'ils sachent, parce qu'ils font partie de votre tableau. Vous placez les éléments, vous les échangez comme les pièces d'un casse-tête, et quand votre puzzle est complet, l'image se forme : le climax peut avoir lieu.

Vous ne pouvez pas faire la même chose avec une histoire interactive. Il y a un personnage qui se trouve hors de votre contrôle en tant qu'auteur, et c'est le joueur. Le joueur fait ce qu'il veut et il prend autant de temps qu'il veut pour le faire. Comment faire en sorte que lorsque le climax dramatique a lieu dans votre histoire interactive, votre joueur est là et prêt à y assister ? C'est le problème du flot narratif.

Il existe trois solutions traditionnelles à ce problème dans les jeux d'aventure. Une très simple consiste à limiter l'interactivité. Vous élaguez l'interactivité de façon à ce que le joueur ne puisse s'écarter de l'intrigue ou vous lui donnez beaucoup d'interactivité, mais vous la videz de son sens – l'interactivité n'affecte alors plus rien.

Je ne pense pas que ce soit une solution acceptable. Si on se ramène au cas extrême, le jeu tourne au « Appuyez sur Entrée pour voir l'écran suivant ». Limiter l'interactivité n'est pas ce que nous sommes

supposés faire ici. Quelques jeux ont effectivement fait ceci, mais ils sont universellement reconnus comme de mauvais jeux – certainement pas l'exemple idéal de narration interactive.

La deuxième solution traditionnelle est de dire « Dommage, si le joueur n'est pas prêt pour le climax, et bien tant pis. ». Dans ce cas, vous créez un monde qui est vivant, qui évolue autour du joueur, sans égard à ce qu'il fait. Ceci peut faire des jeux d'aventure réellement très intéressants : la nuit tombe, les gens sortent des boutiques et rentrent chez eux, les voleurs sortent et ainsi de suite. C'est intéressant de voir les choses prendre place autour de vous dans l'un de ces jeux. La difficulté avec ceux-ci est qu'on tend à perdre beaucoup. Vous finissez par devoir recommencer tout le temps, parce que vous n'étiez pas prêt pour le climax quand il a eu lieu. Mais ce n'est pas une façon de présenter un travail de fiction ! Personne ne lit un livre en lisant la page un, puis en recommençant et en lisant la page un et deux, puis en recommençant et en lisant la page un, deux et trois, et ainsi de suite. Ça vous rendrait rapidement fou.

Il y a bien sûr une façon de contourner le problème et cela s'appelle une sauvegarde. Mais sauvegarder le jeu à tout instant détruit la suspension d'incrédulité. Si je combats les arbres maléfiques dans la forêt enchantée avec mon épée magique, je n'ai pas envie de m'arrêter toutes les cinq minutes et d'avoir une interaction avec mon disque dur. Sauvegarder permet d'éviter de recommencer sans cesse, mais au prix de me faire sortir de l'univers auquel j'essaye d'appartenir. Je ne pense pas qu'il s'agisse d'une solution satisfaisante non plus.

La troisième solution traditionnelle à ce problème est la solution classique du jeu d'aventure de faire avancer l'intrigue au fur et à mesure que le joueur avance. Ceci garantit absolument que le joueur aura tout ce dont il aura besoin lorsqu'il parviendra au climax de l'histoire. S'il a besoin de l'épée magique, alors il aura l'épée magique, et s'il n'a pas l'épée magique, il n'y pas de moyen pour qu'il puisse accéder au climax : l'intrigue n'avance tout simplement pas. C'est simple. Vous établissez juste un lien entre les actions du joueur et l'avancement de l'intrigue.

La difficulté de cette solution réside dans le fait qu'elle est mécanique. Elle transforme le jeu en une série de situations devant être résolues et une fois que vous avez joué deux ou trois fois à ces jeux, vous pouvez vraiment le voir. Si rien ne semble se passer, avez sans doute oublié de faire quelque chose. Quand vous faites quelque chose bien, des choses intéressantes se produisent. Le rythme est saccadé, marche-arrêt. Vous aussi bien que le joueur pouvez faire comme vous le voulez, mais vous n'avez pas la sensation d'être porté par l'histoire : en fait, il est plutôt clair que vous n'êtes pas dans l'histoire, l'histoire est un objet mécanique externe qui progresse uniquement quand vous faites les bonnes actions. C'est un peu comme essayer d'utiliser un magnétoscope avec des boutons sans symboles.

Conclusion

Vous pourriez penser qu'à ce moment, je suis sur le point de vous donner quelques solutions à ces problèmes. Mais je n'ai pas de solution et je ne suis pas certain qu'il en existe. Je n'irai pas jusqu'à dire qu'interactivité et narration s'excluent mutuellement, mais je pense qu'elles existent dans une relation de vases communicants : plus vous avez de l'une, moins vous avez de l'autre.

Dans sa forme la plus riche, la narration implique que le lecteur se soumet à l'auteur. L'auteur prend le lecteur par la main et l'emmène dans le monde son imagination. Le lecteur a un rôle à jouer, mais c'est principalement un rôle passif : faire attention, comprendre, peut-être penser... mais pas agir. Une bonne histoire se tient de la même façon qu'un bon casse-tête se tient quand vous le prenez : toutes les pièces sont solidement attachées les unes aux autres. Mais ça ne supporte pas bien d'être tripoté. Ôtez quelques pièces et tout risque de tomber en morceaux.

L'interactivité n'est pas comme ça. L'interactivité concerne la liberté, le pouvoir, l'expression de soi. Il s'agit d'entrer dans un univers et de le changer par votre présence. Dans beaucoup de jeux, l'univers est statique et mort jusqu'à ce que le joueur arrive ; le joueur est la seule chose qui fait bouger les choses.

L'interactivité est presque à l'opposé de la narration. La narration s'écoule sous la direction de l'auteur, alors que l'interactivité dépend des motivations du joueur.

Ceci ne signifie pas que je retire mon appel à l'industrie du jeu à créer plus de jeux d'aventure – loin de là. Mais je reconnais que les jeux d'aventure, du moins pour l'instant, ne racontent qu'une forme limitée d'histoire : le mystère ou la quête. Nous ne pouvons pas encore faire un jeu d'aventure sur une famille en décomposition ou sur la lente descente d'un jeune homme dans la folie. Les jeux d'aventure doivent sacrifier certains de leurs meilleurs éléments narratifs pour le bien de l'interactivité.

Je pense que les jeux d'aventure devraient simplement être ça : des jeux sur des aventures. Ils devraient donner au joueur l'impression d'une réalisation et d'un accomplissement. Ils concernent ça : faire, créer une différence. Ceci ne signifie pas qu'ils doivent devenir des jeux de shoot ou de réflexes, simplement que le joueur et ses actions sont les choses les plus importantes dans le jeu. Dans un jeu, subordonner le joueur à l'intrigue se fait à vos risques et périls.

Notre métier ne consiste pas à *écrire* des histoires. Notre métier est de construire des univers dans lesquels les joueurs peuvent vivre une histoire de leur propre création.

NdT : cet article étant relativement ancien, il ne présente pas forcément certains jeux ayant apporté des éléments de réponse à ce problème des rapports entre l'interactivité et la narration. J'invite donc le lecteur à se pencher sur des jeux comme Metal Gear Solid, qui a beaucoup joué sur la frontière intra/extradiégétique, id est entre ce qui apparaît dans le jeu au personnage et ce qui apparaît dans le jeu au joueur (par exemple, la musique est souvent un élément extradiégétique : le joueur l'entend, elle participe à la scène, mais, sauf exception, le personnage dans le jeu n'est pas censé l'entendre), Baten Kaitos I et II qui ont réussi à transformer le problème de l'amnésie en un ressort de l'intrigue particulièrement efficace (je n'en dis pas plus à ce sujet, histoire de garder un peu de surprise) et enfin The Legend of Zelda : The Majora's Mask, qui d'une certaine façon, réinvente de façon positive ce que l'auteur appelle la deuxième réponse traditionnelle au problème du flot narratif, en proposant au joueur de revivre inlassablement les mêmes trois journées.

La mort (et Planescape : Torment)

Par [Pierre Guyot](#) | le 18 août 2008 14:23:19 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mai 2000).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

La culture gothique, à ce qu'il me semble, est un peu passée à côté de son sujet. Le fard à paupières et les romans d'Anne Rice, c'est très bien tout ça, mais si vous êtes vraiment intéressés par la mort, je vous suggère d'aller travailler à la morgue de la ville le samedi soir, quand les victimes de tirs de balles arrivent. Là vous pourrez expérimenter la sordide vérité dans de bonnes conditions d'éclairage : les corps sont souvent encore chauds et les familles en sont à cette première phase d'incrédulité choquée qui précède la longue tristesse à venir. La mort violente n'a rien à voir avec des vêtements noirs à la mode ou des bijoux hérissés de pointes : il s'agit principalement de colère et de misère sordide, de brutalité et de mauvais jugement. Néanmoins, malgré leur vision romancée, je pense que les Goths ne sont pour la plupart pas bien méchants. Un peu de jeu théâtral ; un peu d'autodramatisation ; ça énerve leurs parents (ce qui constitue sans doute une grande part de l'intérêt de la chose) mais ce n'est intrinsèquement pas plus sinistre que de s'habiller avec des costumes de Star Trek. Des cultes de la mort sont apparus de nombreuses fois dans l'histoire humaine, des momies de l'ancienne Egypte aux soldats d'argile enterrés en Chine. Porter du vernis à ongles violet est en comparaison une expression assez douce d'une très ancienne pulsion.



Il y a quelques années, j'ai suggéré lors d'un cours magistral à la Game Developer's Conference qu'il était temps pour les jeux d'explorer une partie plus vaste de l'expérience humaine, ceci incluant la tristesse et la mort. La mort a été le sujet d'un certain nombre de débats dans les cercles du game design, mais la plupart du temps, nous parlons de la mort dans un sens purement symbolique : « Vous avez trois chances » est une phrase qui a précédé chacun des jeux de foire au Moyen Âge et sans doute bien après, et quand vous incarnez un personnage dans un jeu vidéo, il semble assez naturel de voir ces trois chances comme des vies qu'on peut perdre – l'échec étant une « mort » métaphorique.

Mais mon court n'était pas au sujet de la mort au sens métaphorique du terme ; je pensais à la mort au sens littéral du terme et particulièrement à la façon dont elle nous affecte émotionnellement. Nous pensons principalement de la mort qu'elle apporte de la douleur, mais en fait, les émotions qui l'entourent sont très complexes. Dans les familles malheureuses, il y a souvent de la colère, de la culpabilité et du ressentiment ; et dans celles qui sont heureuses, ces sentiments ne sont pas toujours un regret sans mélange. Il n'y a pas si longtemps, j'ai assisté aux funérailles du père d'une proche. Il avait presque cent ans lorsqu'il est mort et son décès a été à la fois sans douleur et prévu, aussi bien par lui-même : il avait lui-même planifié les funérailles. Lors de la cérémonie, la femme s'est mise à pleurer. « Je ne suis pas triste », a-t-elle dit, et je l'ai crue. Elle ressentait quelque chose d'autres, ou plusieurs quelques choses – de l'amour, de la nostalgie, de la gratitude ? Je ne lui ai pas demandé. Il n'est pas évident de savoir comment l'on devrait inclure la mort, la mort réelle, dans un jeu vidéo. « Vous-avez-trois-chances » est si profondément caractérisé comme la « mort » que le premier obstacle qui s'oppose à vous est de faire clairement comprendre au joueur que ce n'est pas là ce que vous entendez. Probablement, la meilleure façon passe par la mort, non pas du personnage principal, du personnage du joueur, mais d'autres personnages qui apparaissent dans le jeu.

Il y a eu une grande avancée dans le jeu d'aventure lorsque les game designers ont arrêté de traiter le personnage principal comme un Monsieur Tout-le-monde (ce qu'ils ont fait initialement car ils savaient que le joueur pouvait en réalité être n'importe qui) et ont commencé à créer des personnages principaux qui avaient un sexe, une voix, une apparence, un background et surtout, une personnalité propre. La répugnance initiale à faire ceci était fondée sur le problème de savoir si les hommes accepteraient de jouer des rôles féminins et inversement. Cette question a été pleinement résolue par Lara Croft et ce n'est plus un problème à présent. C'est une bonne chose, car il est beaucoup plus simple de créer une intrigue à résoudre par un personnage si ce personnage est une personne avec une histoire qui lui est propre.

Néanmoins, en dépit du fait qu'on nous donne quelqu'un à qui on nous demande de nous identifier – qu'il s'agisse de Sonic le hérisson ou de Duke Nuke'Em – je pense que nous nous occupons de cet individu d'une façon fondamentalement différente de celle dont nous nous occupons des autres personnages dans le jeu. Le personnage principal constitue une extension de nous-mêmes, une sorte de prothèse capable d'atteindre le monde du jeu. S'il meurt avant la fin du jeu, c'est irritant, frustrant, mais nous savons en nous-mêmes que ce n'était pas la façon dont les choses étaient censées se dérouler, et nous redémarrons le jeu et ressuscitons le personnage sans aucune perte de sens réelle.

Quand un autre personnage meurt, néanmoins – un personnage non-joueur, pour utiliser la terminologie du jeu de rôle – nous ne pouvons pas être surs qu'il ne s'agissait pas de l'action d'une destinée cruelle ; que ce personnage était destiné à mourir quoique nous faisons. Ceci a en partie à voir avec le sens du contrôle. Dans la vie réelle, nous aimons les autres différemment de nous-mêmes, précisément parce qu'ils ne sont pas nous-mêmes. Dans les jeux, nous faisons le deuil des autres différemment de la façon dont nous faisons notre propre deuil, encore une fois parce qu'ils ne sont pas nous et que nous ne sommes pas les maîtres de leur destinée. Pour donner du sens à la mort dans un jeu vidéo, il faut que ce soit, non pas le joueur, mais ses amis qui meurent.

Planescape : Torment est un jeu principalement tourné vers la mort. Ce n'est pas mon problème de faire la review de jeux, et de toute façon il est un peu tard pour celui-ci, mais le jeu développe certains aspects si bien que je pense que cela vaut la peine d'y jeter un œil si vous ne l'avez pas déjà fait. Je ne l'ai pas découvert par moi-même : il m'a été spécialement recommandé par des lecteurs de certains de mes précédents articles, auxquels je suis profondément reconnaissant.

Pour ceux qui ne l'ont pas vu, *Planescape : Torment* est un RPG fantasy de Black Isle Studios, une division d'Interplay. Il utilise une version modifiée de l'Infinity Engine qui fait marcher *Baldur's Gate* et est basé sur l'univers *Planescape* de TSR. Malheureusement, cela signifie aussi qu'il est flanqué du système de jeu de rôles Advanced Dungeons & Dragons de TSR. AD&D a été conçu pour un jeu de rôle papier et bien qu'il soit tout à fait adéquat pour ça, il utilise une foultitude de nombres inutiles pour le joueur PC (cf. « Rendons sa magie à la magie »). Néanmoins, je trouve le système moins intrusif que dans *Baldur's Gate*, où j'étais constamment en train de vérifier qui avait quelle statistique et qui avait telle ou telle capacité ou sort enregistrés. Bien que ces mécaniques de jeu soient implémentées dans *Planescape : Torment*, la nature du jeu semble demander moins de cette assommante comptabilité de la part du joueur.

Mais ce qui est le plus intéressant au sujet de *Planescape : Torment* et ce qui mérite notre attention en tant que designers, ce sont sa composition, ses personnages et son intrigue. L'expression « RPG fantasy », bien évidemment, appelle immédiatement des images d'un groupe de personnages Tolkienesques marchant à travers la forêt à la recherche de dragons. *Planescape* est miraculeusement dépourvu de ces stéréotypes – j'y ai joué plusieurs heures maintenant et il n'y a pas un elfe ou un nain en vue, ni, car cela est important, de forêt. Les designers de l'univers *Planescape* ont depuis longtemps abandonné la mythologie de l'Europe du Nord et imaginé quelque chose de plus riche, de plus sombre et tout à la fois plus frais. Si *Baldur's Gate* est une bière blonde, *Planescape* est une bière brune maison.

L'histoire tourne autour d'un personnage immortel et sans nom à la recherche de son passé oublié. Le jeu utilise le cliché de l'amnésie pour expliquer pourquoi il ne semble rien connaître au sujet du monde où il vit, mais je dois dire que c'est manié au moins aussi bien dans *Planescape : Torment* que dans n'importe quel jeu ou livre que j'ai jamais vu. Notre héros est à la recherche d'informations qui puissent expliquer et

finalement annuler son immortalité, pour l'autoriser enfin à mourir définitivement. Au moins, c'est ce que je recherchais : les motivations et la morale sont tout sauf clairs.

Ce n'est pas seulement le personnage principal qui est concerné par la mort. Le jeu débute dans une morgue, avec ses entrepreneurs des pompes funèbres et son fluide d'embaumement. A partir là, on passe à une cité grotesque remplie de zombies, de goules, de squelettes et d'autres, moins familiers, « travailleurs de la mort » : les collecteurs et les hommes-poussières pour n'en nommer que deux types. Mais il ne s'agit pas d'une horreur de pacotille ; en fait, il s'agit rarement d'horreur réelle, car le jeu n'emploie aucune tactique de choc. En dépit des nombreux cadavres mutilés et pourrissant qui apparaissent très tôt dans le jeu, les morts sont souvent dépeints avec sympathie, comme des victimes dignes de pitié, possédant une certaine dignité.

Une autre raison pour laquelle j'apprécie le jeu est qu'il n'emploie pas un vocabulaire simili-médiéval. Au lieu de cela, il tire son langage d'une source différente : l'argot des classes ouvrières du dix-neuvième siècle. Un grand nombre de mots employés dans le jeu sont encore d'un usage courant en anglais britannique (par exemple « berk » [fou] et « barmy » [timbré, maboul]), mais cela peut-être assez difficile pour les Américains de suivre. Néanmoins, il existe plusieurs glossaires sur le Net et au moins, c'est différent, intéressant et cela crée la sensation bien particulière d'être au cœur d'une culture étrangère.

L'univers *Planescape* est loin d'être nouveau – selon un site de fans que j'ai trouvé, TSR l'a développé pour la première fois en 1994, donc l'univers n'aura besoin d'aucune introduction aux fans de jeux de rôle. Pour ce que j'en sais, *Planescape : Torment* a été la première tentative de l'adapter en jeu vidéo. C'est un monde immensément riche, créé précisément pour des adultes et rempli de dilemmes philosophiques. Il y a un grand travail d'écriture derrière le jeu et une partie en est excellente. Cela ne signifie en aucune façon que le jeu est ennuyeux, et il peut être joué à la façon d'un hack-and-slash bête et méchant si vous en avez envie, mais il vous donnera matière à réfléchir, si réfléchir est quelque chose que vous appréciez.

Une remarque au sujet du travail de conception graphique : je n'ai pas assez de superlatifs pour le décrire. J'ai été agréablement surpris par la beauté des forêts et des canyons de *Baldur's Gate*, mais je suis complètement bouleversé par l'imagination dont il est fait preuve dans la Cité des Portes de *Planescape*. C'est si incroyablement varié que c'en est littéralement indescriptible – bien que les conduites et les tuyaux semblent être un thème récurrent, ce que l'on peut normalement attendre d'une cité qui connecte n'importe quel lieu de l'univers à n'importe quel autre lieu. Il faut que vous le voyiez par vous-mêmes.

Si j'avais une remarque à faire sur le travail de conception graphique, ce serait que les femmes sont plutôt sous-habillées et elles semblent apparaître sous des déclinaisons moins diverses que les hommes. Je suppose que ceci peut être attribué aux habituels problèmes hormonaux de l'équipe d'animation. Néanmoins, le background spectaculaire rattrape largement le coup à mon avis. Dégotez le jeu et jouez-y. Si vous n'êtes pas portés sur les RPG, trouvez une solution sur le net afin de pas avoir à vous embêter avec la résolution d'énigmes, lisez juste le texte et regardez les images.

Si vous voulez voir un game design bien fait, *Planescape : Torment* est un jeu qui peut vous enseigner pas mal de choses. Comme il utilise le modèle AD&D, il y a assez peu de choses nouvelles au niveau des mécaniques sous-jacentes, mais il y a surtout là un monde à explorer, dont je pense qu'il contient la plus intense concentration de créativité que j'ai jamais vu dans un jeu, passé ou présent.

Cours de pixel art 9 : un monde de tiles

Par [Antoine Gersant](#) | le 18 août 2008 14:43:31 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#), [Tiles](#)

Tout ça pour ça...



Si vous m'avez lu jusqu'ici, vos connaissances théoriques en Pixel art sont désormais à peu près correctes. Cependant, vous êtes sans doute déçu que je n'ai pas répondu à vos préoccupations les plus pressantes, du genre « Comment je fais un tile d'herbe ? », « Comment je fais la tête de mon héros ? », « Comment faire un arbre pas trop moche ? », « Où trouver une nouvelle copine ? ». En fait, vous en savez déjà assez pour faire tout ça (sauf pour la copine, et là vous vous débrouillez mieux sans moi je pense)... mais ne rêvez pas trop ! Sans pratique, on n'arrive à rien, et si vous vous posez encore ce genre de question, vous avez besoin de pratique (et de critiques). Histoire de remédier un peu à cela, nous allons passer en revue les éléments de base qui constituent un tileset de RPG et qui posent beaucoup de problèmes en pixel art. Ou pas.

La hantise du Pixel artiste : le TILE D'HERBE (*cris et pleurs*)

Comme les pros !

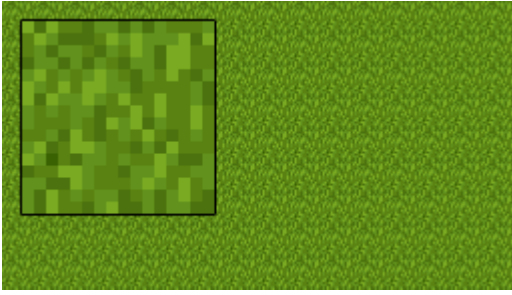
Le problème qui se pose pour représenter de l'herbe est le même que celui pour la fourrure au chapitre précédent : les brins d'herbe sont trop fins pour être représentés individuellement en Pixel art. Il va donc falloir recourir à une stratégie semblable à celle des touffes de poils... j'ai nommé la stratégie des touffes d'herbe. Si.

Étant donné que des photos d'herbe ne vous seraient d'aucun secours dans votre tâche, il va falloir s'inspirer d'autre chose... à savoir des tiles d'herbe réussis (ou pas) de jeux professionnels. Ce qui va suivre n'est en aucun cas une incitation à la copie, il s'agit d'une collection de screenshots contenant des tiles d'herbes que je vous suggère d'observer de très près (enregistrez les et zoomez très fort) en vous posant des questions techniques : combien de couleurs sont utilisées ? quels motifs dessinent les couleurs ? dans quels sens sont orientées les « touffes » ? le résultat est-il convaincant ? Voilà sans plus attendre la collection en question. Enjouaillez.



Cliquez sur les images si vous voulez voir les noms des jeux.

Comme je suis sympa, je vais vous mâcher un peu le travail et disséquer en votre compagnie le tile d'herbe de *Rudora No Hihou*. Voilà le tile d'herbe en question.

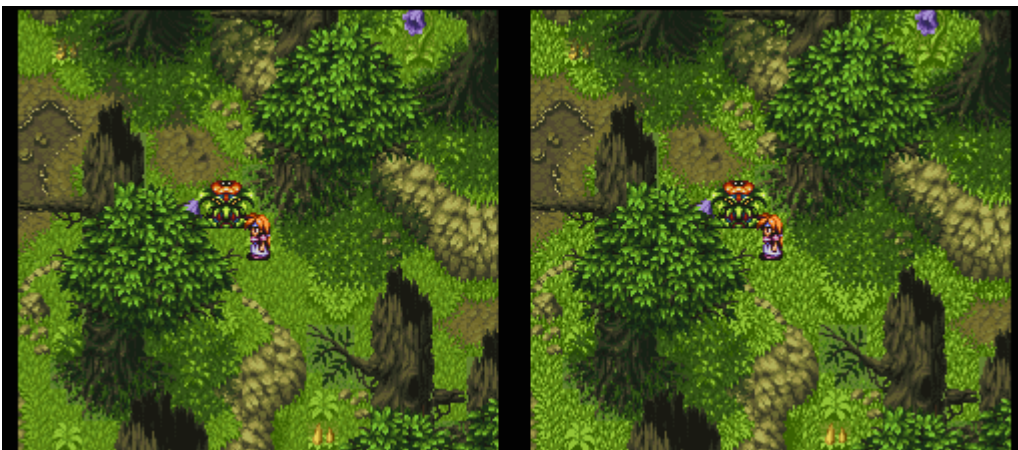


Il est impératif de remarquer que le tile n'est pas fait « n'importe comment » en plaçant au hasard des pixel des 4 couleurs. Souvenez vous du cours sur la texture. Les blocs d'une même couleur sont pour la plupart verticaux ou un peu en biais et cela représente grosso-modo la direction des touffes d'herbe. De plus, il n'y a pratiquement pas d'anti-aliasing (seulement un peu sous les highlights) car il n'y a pas franchement la place ou le besoin d'en faire (les teintes se ressemblent assez pour ne pas s'entrechoquer violemment).

Regardons justement un peu la palette histoire de savoir de quoi on parle : l'artiste a utilisé 4 teintes de vert pour créer ce tile d'herbe, à savoir une pour les highlights, une pour les ombres, et deux teintes médianes assez proches l'une de l'autre comme base. Les highlights servent à faire « le haut des touffes d'herbe » tandis que les ombres représentent les zones les plus enfoncées et près du sol.

La gestion du contraste est primordiale pour faire un tile d'herbe : moins le tile est contrastée, plus le tile est plat ; plus le tile est contrasté, plus le tile exhibe son volume et la richesse de la flore locale (couche épaisse d'herbe). Il faut réaliser un compromis entre un sol « plat » et sol « en volume » car ce dernier complique la lisibilité de l'écran et attire l'attention. Il ne faut pas oublier que le tile d'herbe ne doit pas noyer les personnages qui sont dessus ou paraître plus éclairé et volumineux que les arbres et les maisons.

Pour l'exemple, comparez le screenshot original de *Rudora No Hihou* (à gauche) avec une version modifiée (teinte sombre de l'herbe plus sombre et teinte claire plus claire). A droite, il y a plus de volume mais moins de lisibilité.



Trucs et astuces

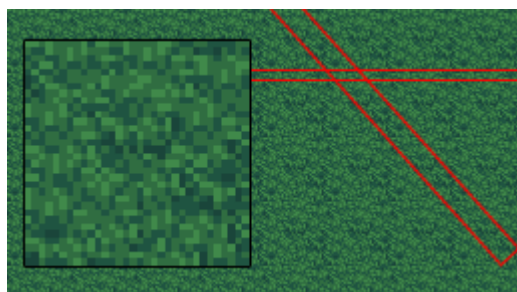
Voici deux astuces qui devraient vous aider à mieux réussir votre tile d'herbe (la seconde ne s'applique d'ailleurs pas qu'aux tiles d'herbe).

Premièrement, il est très avantageux de ne pas utiliser un tile d'herbe... mais plusieurs tiles d'herbe ! Si vous avez observé attentivement les screenshots proposés ci dessus vous avez sans doute remarqué que chaque jeu utilise plusieurs tiles différents afin d'avoir des variations dans la texture du sol. Cela permet de limiter l'effet de quadrillage dû à la répétition d'un même motif sur une large surface.

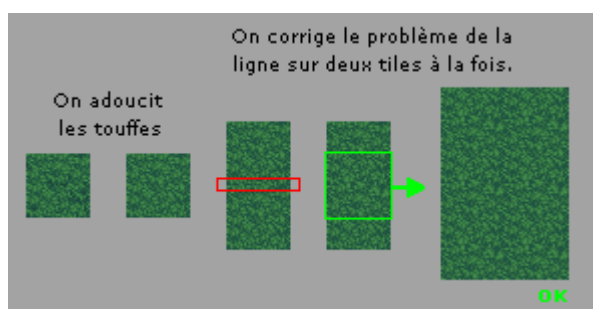
La plupart des jeux utilisent 4 tiles d'herbe (un carré 32*32), une variété d'herbes hautes et différents tiles de fleurs. De plus, les développeurs essayent en général de ne pas avoir de grandes étendues d'herbes (même

pleines de variation) et remplissent leurs cartes de falaises et d'arbres pour se débarrasser une bonne fois pour toute du problème.

Deuxièmement, il est possible d'éliminer la grille de façon efficace en corrigeant à la main les petits détails de votre tiles qui rendent visible la répétition. Considérons si vous le voulez bien le tile (32* 32 pixels, 4 tiles donc)à votre gauche (réalisé exprès pour ce cours). Il n'est pas trop mal, mais j'ai encadré en rouge une ligne horizontale qui se répète de façon clairement visible tous les 32 pixels, et une ligne de touffes en diagonale qui souffre du même problème. En modifiant cela, je peux rendre mon tile plus uniforme et plus à même de boucler.

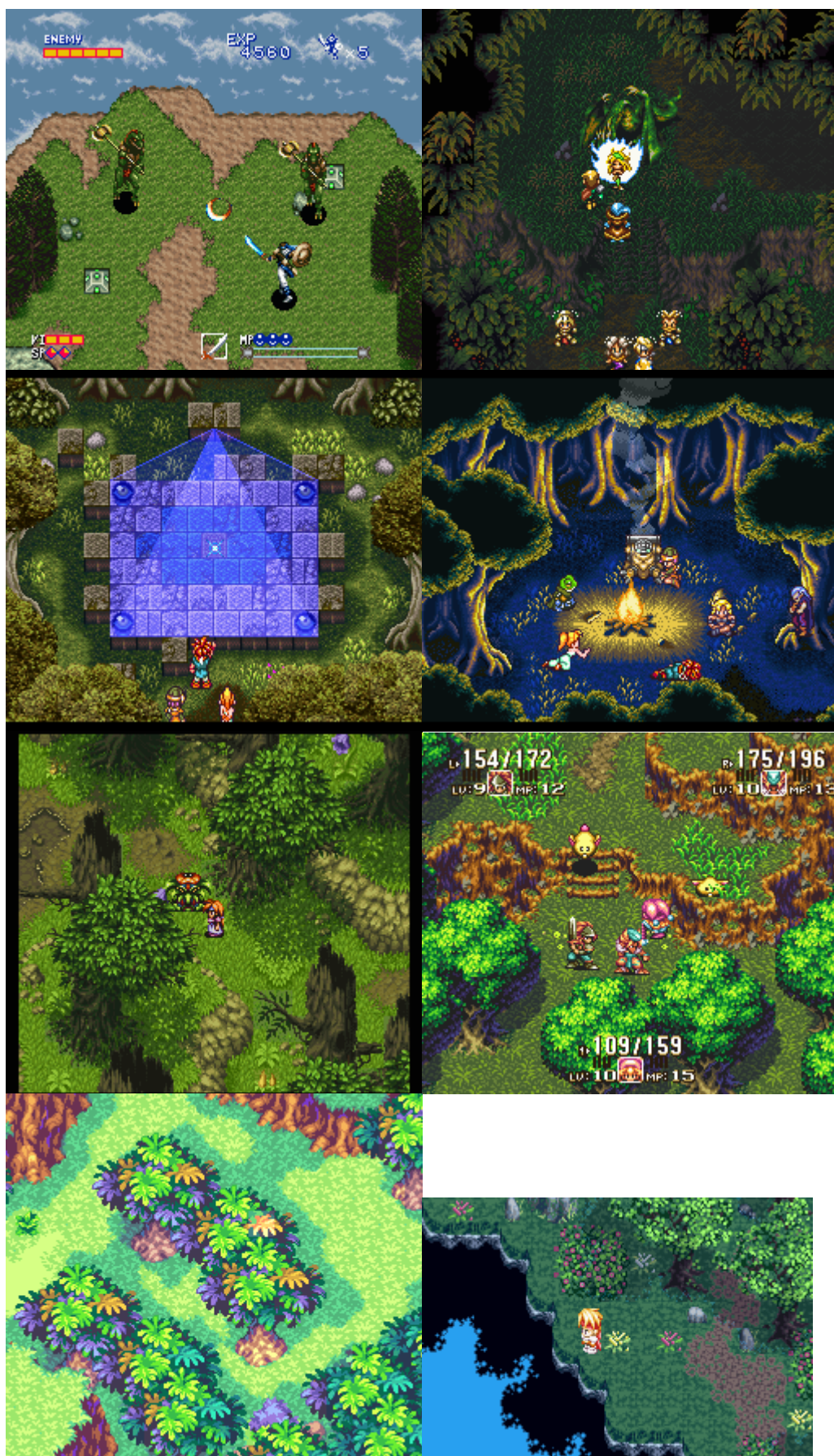


Le problème des touffes en diagonale se trouve dans la partie en haut à droite du tile donc il me suffit d'aplanir un peu les touffes pour corriger le problème. Par contre, les lignes horizontales sont à cheval sur le haut et le bas de mon tile (qui boucle mal donc) et il faut recourir à une petite astuce pour y remédier. L'astuce consiste à placer une copie du tile sous lui, histoire de bien voir la transition, et de faire disparaître la ligne en travaillant à la fois sur le haut du tile du bas et le bas du tile du haut. Une fois la ligne supprimée, on colle ensemble le bas du tile du haut avec le haut du tile du bas...et on a un tile qui boucle correctement. Le procédé est illustré ci-dessous, Le résultat n'est pas encore parfait mais déjà bien plus convaincant.



Auprès de mon arbreuuuh

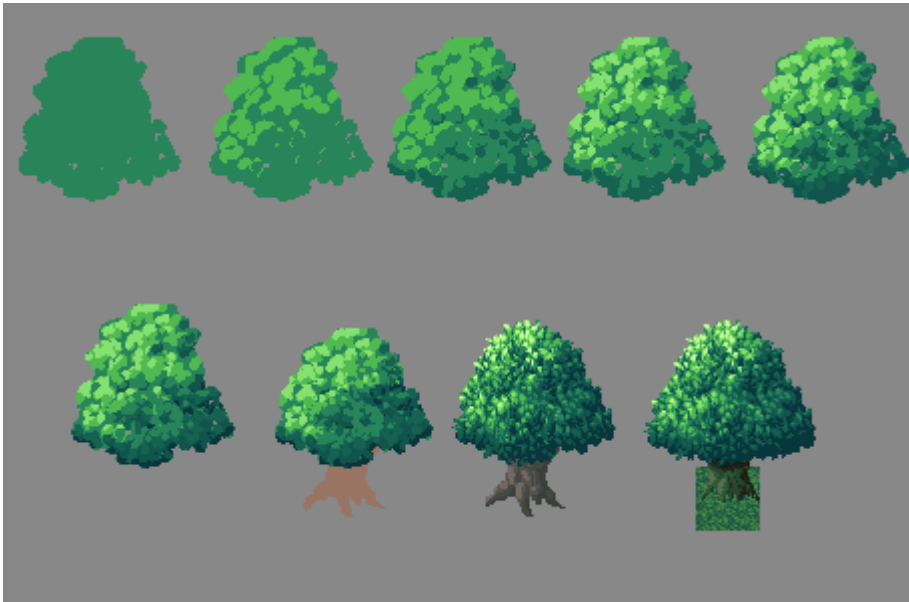
Bien, maintenant que vous avez de la verdure, il est temps de passer au gros œuvre : un arbreuh (ou un arbre, pour les non initiés). Je ne vais pas vous faire l'affront de vous expliquer qu'un arbre est constitué de branches et de feuilles, et que ces dernières sont généralement vertes. Je commence par vous laisser regarder une pelletée de screenshots avec des arbres réussis (ou pas ! *Chrono Trigger* n'est pas un exemple à suivre dans le domaine) à observer, les explications sont en dessous. Comme précédemment, il serait très bon pour vous de passer du temps à analyser à la loupe quelques uns de ces screenshots et de procéder à des « dissections ».



Comme vous pouvez le voir, il existe de nombreuses façon de représenter des arbres, du très réaliste *Tales of Phantasia* au plus exotique et stylisé *Sword of Mana*. En fait, je n'ai en tout et pour tout que quelques importants conseils à vous donner pour réussir un arbre : La premier c'est que les racines d'un arbre sont enfouies dans le sol et qu'on a souvent tendance à l'oublier et à en faire d'interminables filaments qui courent entre les brindilles. Les arbres sont PLANTES dans le sol, une petite ballade en plein air sera l'occasion de le vérifier. Tant que vous êtes dehors, profitez-en pour jeter un œil aux feuillages autour de

vous, c'est ce qu'il y a de plus important sur un arbre. Il y a deux erreurs à éviter sur les feuillages : la première c'est de faire un feuillage « plat », comme une tranche de carton. Il faut garder à l'esprit que le feuillage est un objet en 3 dimensions, qui est soumis aux règles sur l'ombre et la lumière que l'on a évoquées précédemment. Le second piège est de trop penser à cela et de réduire son arbre à une sphère éclairée. Il ne faut pas non plus oublier que le feuillage est constitué de branches et de feuilles, qui ne forment pas une sphère parfaite, mais le plus souvent un empilement de sphères incomplètes ou tordues sur lesquelles les feuilles bataillent pour recevoir de la lumière en se passant les unes devant les autres. Regardez l'arbre de *Seiken Densetsu 3* : il est constitué de 4 sphères sur lesquelles sont dessinées les feuilles.

Allons-y pour un petit mode d'emploi :



Je commence par dessiner grossièrement la forme du feuillage (étape 1), avant de consacrer 5 étapes à la création des volumes du feuillage. Ma source de lumière est à gauche et je place couleur après couleur des ombres et des lumières. L'étape 3 est celle où le procédé apparaît le plus clairement. Cela fait, je dessine une esquisse de tronc et trouve que mon arbre a trop des airs de champignon géant. Je décide de descendre le feuillage un bon coup (et oui, un peu de travail effectué pour rien). Je détaille un peu le tronc et beaucoup le feuillage (c'est TRÈS TRÈS long et TRÈS TRÈS ennuyeux. Soyez patients et appliqués) pour arriver à l'avant dernière étape.

Enfin, je coupe franchement le tronc pour éviter le problème des « racines tentacules » et bien planter mon arbre dans le sol, j'ajuste la forme du feuillage pour que l'arbre soit moins difforme (mais sans être parfaitement symétrique pour autant), je change la palette du tronc en quelque chose de plus sombre et plus vert (pour mieux s'accorder au feuillage), et je n'oublie pas que le feuillage fait de l'ombre au tronc. J'ai collé un de mes tiles d'herbes en dessous pour tester l'homogénéité des couleurs, mais j'ai eu la flemme de faire l'ombre portée du feuillage sur l'herbe : regardez le screenshot de *Rudora no Hihou*, ils font ça bien mieux que moi de toute façon.

Une dernière remarque en passant, je me suis ici permis de m'occuper séparément du feuillage et du tronc car j'ai fait un feuillage très dense qui masque complètement les branches. La logique voudrait que l'on dessine d'abord toutes les branches d'un arbre (son « squelette 7) avant d'y ajouter les feuilles, et c'est ce qu'il faut faire si vos arbres sont plus dégarnis.

Vamos a la playa !

Passons sans plus attendre à des tiles plus ensoleillées, à commencer par un tile de sable. Conservons la même démarche que précédemment et observez de près les images ci-dessous :



Si votre observation a été fructueuse, vous aurez remarqué que le tile de sable est bien plus simple à réaliser que les précédents, et aussi que la plupart des conseils données pour le tile d'herbe sont également valables ici (utiliser plusieurs tiles, faire attention à briser la grille, trouver un compromis entre volume et lisibilité). La plupart des tiles (les plus réussis surtout, notamment ceux de *Seiken Densetsu 3* et de *Sword of Mana*) sont en majeure partie constitués d'un aplat d'une seule couleur (vacances pour le pixel artiste) accompagnée de petites dunes ondulées. Remarquez également que de petits motifs de dithering sont de bon ton pour les tiles de sable, afin de donner une texture granuleuse.

Question palette, il faut faire attention à éviter le jaune fluo crève-rétine, et ne pas hésiter à tirer vers des tons crèmes (*Seiken Densetsu 3*) ou marrons (*Rudora No Hihou*). Si vous faites tendre un peu la palette vers le rouge, vous obtiendrez un joli sable éclairé par le soleil couchant.

Étant donné que je n'ai cette fois-ci aucune technique particulière à illustrer et que tout ce que j'avais à signaler se voit dans les screenshots (et dans mon explication) ci-dessus, je passe mon tour et ne fais pas de tile de sable.

Sea, Sex And Sun !

Terminons notre petite liste par les tiles d'eau. Il existe plein de façons de traiter leur cas (bien plus que pour l'herbe) donc vous allez avoir du boulot si vous voulez tout analyser. En ce qui me concerne, je vais tenter de mettre en valeur les grandes lignes. Voilà pour les images :





Je vous ai mis la dose mais il faut bien ça pour illustrer la diversité des méthodes utilisables. Procédons à une classification.

Les tiles de *Sword of Mana* et *Tales of Phantasia* sont les plus minimalistes du lot. Ils utilisent peu de couleurs par tiles et illustrent à merveille le premier point que je veux vous faire remarquer : l'eau étant une matière à peu près transparente, on la repère généralement plus à son entourage qu'à sa propre présence. Je m'explique : regardez cette photo du lac de Fabrège (dans les Pyrénées). La surface du lac est avant tout la surface où se reflète la montagne. Vous voyez une montagne à l'envers sur le sol, donc vous en concluez qu'il s'agit d'une étendue d'eau où la montagne se reflète, alors que rien ne vous l'indique vraiment.



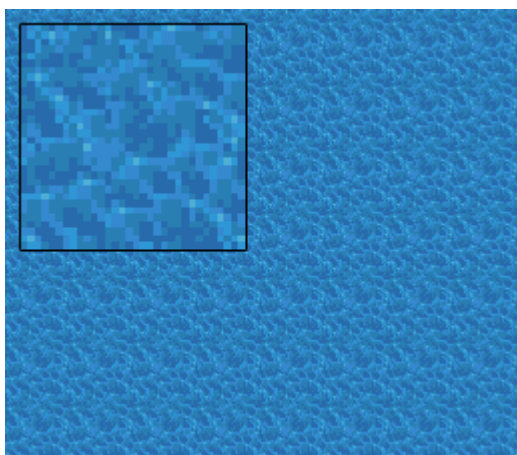
Bien qu'il soit difficile d'atteindre de tels extrêmes dans la réalisation d'un tile (encore que, ça mériterait d'être essayé), je pense que vous comprendrez l'idée en voyant que l'eau de *Tales of Phantasia* est d'un bleu uni sur lequel s'étalent les ombres des branchages qui la surplombent. De même, *Sword of Mana* dispose de toute une palette de profondeurs différentes, en jouant sur la palette utilisée. Plus l'eau est sombre et opaque, plus la distance qui sépare le fond de la surface est grande. Enfin, en regardant un peu les autres screenshots, vous remarquerez qu'il n'est pas rare de voir des falaises s'y refléter (ou visibles en transparence à travers l'eau), notamment dans *Rudora No Hihou*.

Les tiles de *Chrono Trigger* et *Bahamut Lagoon* sont les moins réussis du lot, et ne constituent pas des exemples à suivre (sauf pour ce qui est des cascades). Leurs textures sont avares en couleurs et se limitent à des barbouillages informes de points, à tel point que si l'eau de *Bahamut Lagoon* n'était pas bleue, il serait très difficile de deviner qu'il s'agit d'eau (une texture réussie est reconnaissable en noir et blanc).

A l'inverse, *Alcahest*, *Terranigma* et *Rudora No Hihou* (celui dans la caverne) ont des textures bien plus évocatrices, qui dessinent des motifs souples et arrondis (et bien anti-aliasés aussi). Regardez par exemple sur ce détail du screenshot de *Rudora No Hihou* comment la texture trace des courbes qui ondulent sur la surface de l'eau.



J'ai gardé le meilleur, et le plus difficile, pour la fin : l'eau de mer et les vagues. Il est courant de représenter les tiles de mer en utilisant une ou deux couleurs très claires pour tracer les crêtes des vagues, formant ainsi une sorte de trame ondulée et très irrégulière à la surface de l'eau. Il faut ensuite remplir cette trame en utilisant les couleurs plus sombres et sans oublier de choisir une source de lumière (les vagues sont tout sauf plates !). Ci-dessous, ma tentative personnelle après une séance d'observation.

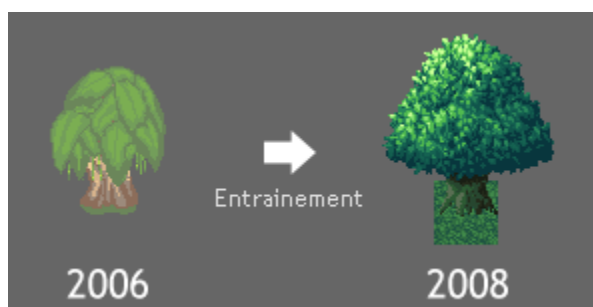


J'ai tenté de garder discret le système de crêtes mais je pense que le résultat final manque un peu de personnalité. L'excès inverse se trouve dans *The Legend of Zelda : The Wind Waker* dont la mer se limite en tout et pour tout à ce système de quadrillage (mais le résultat est pourtant excellent !).



Le grand secret de la vie et de la mort

Nous en avons maintenant fini avec les tiles, et ce cours touche également à sa fin. J'ose espérer qu'il vous a été utile et que vos jeux seront désormais un peu plus beaux. Je tiens à conclure en vous donnant un dernier conseil, et c'est sans aucun doute le plus important que vous trouverez dans ces 8 chapitres : c'est en vous entraînant que vous progresserez. Le pixel art est un domaine dans lequel vous pouvez devenir très bon sans avoir aucune forme de talent et sans être un artiste très doué avec les médias plus traditionnels (dessin, peinture...). La persévérance couplée au travail acharné est un duo gagnant à tous les coups, pour peu que vous réfléchissiez un peu à ce que vous faites et que vous soyez appliqués. La paresse et l'attentisme sont eux un duo perdant à tous les coups. C'est seulement en enchaînant tileset sur tileset et personnage sur personnage que vous finirez par exceller en la matière. N'attendez pas de miracle du jour au lendemain, il s'agit d'une progression lente et organique que vous ne remarquerez qu'en comparant vos travaux récents à ceux réalisés plusieurs semaines ou plusieurs mois auparavant. Pour preuve, je vous laisse en compagnie d'une image qui illustre ma propre progression entre 2006 et 2008 et qui vous en donne la clé.



[Lire la suite →](#)

Equilibrer les jeux grâce à la rétroaction positive

Par [Pierre Guyot](#) | le 12 octobre 2008 15:56:53 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (janvier 2002).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

La plupart du temps, cette chronique est remplie de notes dogmatiques sur la créativité, la narration ou les effets sociologiques du divertissement interactif – en d'autres termes, la pluie et le beau temps. De temps à autres néanmoins, je me sens obligé d'écrire quelque chose d'abstrus et de technique sur le game design, quelque chose qui relève plus du « comment » plutôt que du « pourquoi » ou du « pourquoi-ne-pas ». Là il s'agit de l'un de ces moments. Ce mois-ci, je vais vous parler des effets que la rétroaction positive a sur l'équilibre d'un jeu.

Qu'est-ce que la rétroaction positive ?

Quand nous parlons de feedback (*NdT : c'est ce mot que nous traduisons dans le reste de l'article par rétroaction*) dans la vie de tous les jours, nous faisons habituellement référence à cet horrible grésillement qui se produit chaque fois qu'un micro dans un système de sonorisation se trouve trop près de l'un des haut-parleurs (*NdT : c'est ce qu'on appelle l'effet Larsen*). Le micro récupère tout ce qui sort des haut-parleurs et essaye de l'amplifier à nouveau. De façon plus générale, une rétroaction intervient chaque fois que la sortie d'un système est aussi connectée au système sous la forme d'une entrée. Ce qui intervient avec le micro et le haut-parleur est un exemple de réaction positive – une situation qui tend à amplifier ce qui sort du système

Bien qu'on n'entende pas beaucoup de concepteurs en parler, la rétroaction positive joue un rôle important dans le game design. Elle peut gravement nuire à un jeu si c'est mal implémenté, mais elle a aussi des avantages significatifs. C'est un élément de game design que tout concepteur doit comprendre et apprendre à utiliser.

Avant que je ne m'occupe de son fonctionnement, regardons une minute la façon dont on perd ou on gagne dans les jeux. Quand vous arrivez devant deux de vos amis en train de jouer à un jeu, quelle est la première chose que vous dites ? « Qui est-ce qui gagne ? », bien évidemment. Il ne s'agit pas toujours d'une question simple. Certains jeux ont une métrique qui détermine qui est en tête à n'importe quel moment ; d'autres non. Au ping-pong par exemple, c'est évident : celui qui a le plus de points est en train de gagner. Aux échecs, c'est beaucoup moins clair, parce que la condition de victoire – coïncider le roi de toutes parts – n'est pas définie en termes de points accumulés. Vous pouvez dire de façon très approximative que c'est celui qui a pris le plus de pièces qui est en train de gagner, mais il est parfaitement possible de gagner aux échecs avec moins de pièces que l'adversaire n'en a.

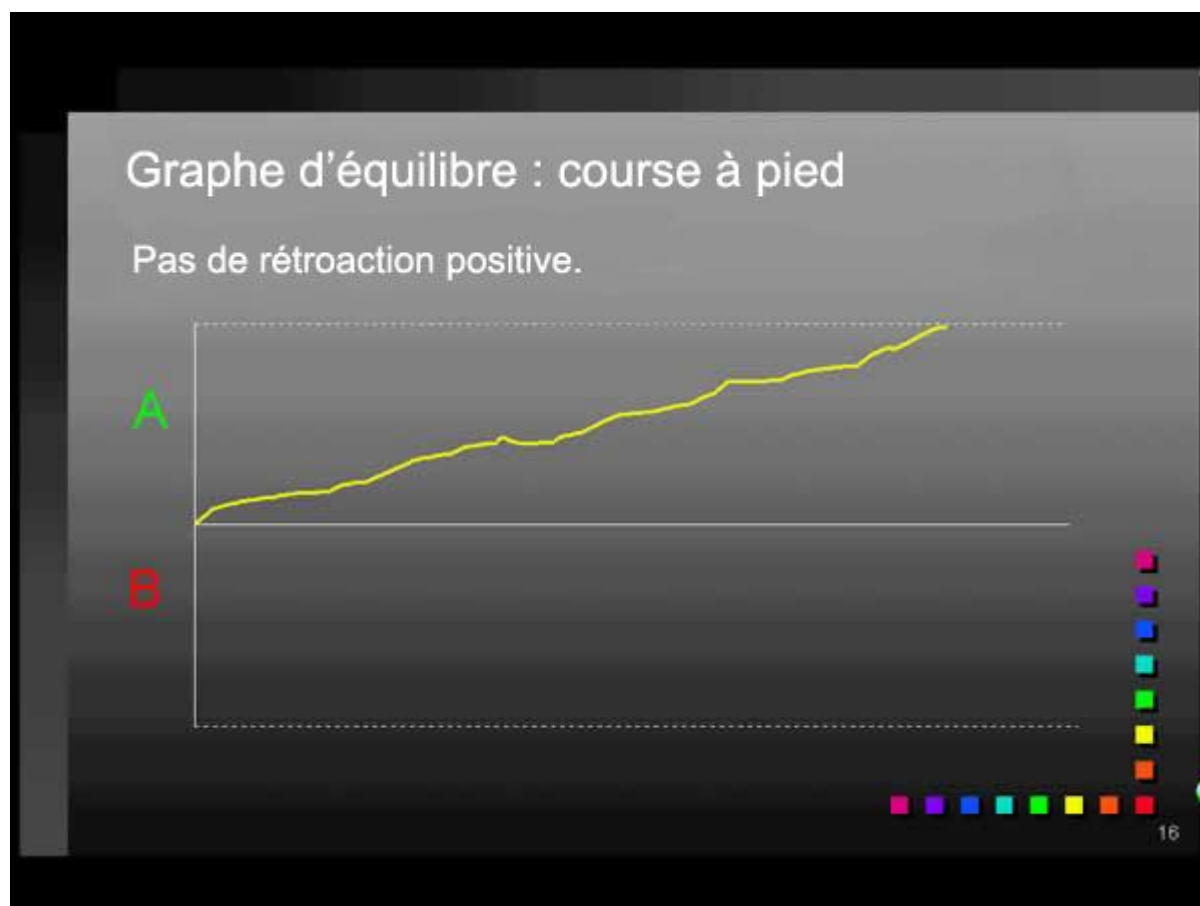
Pour ce qui est game design, on peut parler de rétroaction positive chaque fois que la réalisation d'un objectif rend les objectifs suivants plus faciles à réaliser. En d'autres termes, chaque fois que quelqu'un gagne quelque chose dans un jeu, cela devient plus facile de gagner. Si le rôle de la rétroaction positive est trop important, alors celui qui obtient le premier un avantage, aussi petit qu'il soit, dans le jeu a la garantie de gagner, parce que celui ne va pas cesser de s'amplifier. Ceci donne l'impression que la rétroaction positive est à éviter, mais ce n'est pas le cas ; la question est juste de l'employer correctement.

La rétroaction positive apparaît principalement dans les jeux où la condition de victoire est définie en termes numériques et dans lesquels vous travaillez tout au long du jeu à remplir cette condition de victoire en accumulant quelque chose. Dans le Monopoly, par exemple, c'est l'argent. Obtenir de l'argent vous permet d'acheter et d'améliorer des propriétés, ce qui permet de gagner plus d'argent. La rétroaction positive apparaît aussi dans des jeux où une certaine sorte d'avantage numérique permet de remplir une condition de

victoire non numérique. Bien que la condition de victoire aux échecs soit non-numérique, cela aide généralement d'avoir plus de pièces que votre adversaire.

Graphes d'équilibre

Alors, quels sont les effets d'une rétroaction positive sur les jeux ? Je pense qu'il est utile de jeter un œil à ce que j'appelle un graphe d'équilibre, bien qu'il existe probablement un nom différent pour cela dans la théorie formelle des jeux. J'en ai inclus plusieurs ci-dessous. Un graphe d'équilibre représente l'évolution d'un jeu à deux joueurs de somme nulle. Le temps est représenté par l'axe horizontal. L'axe vertical indique qui est en tête selon une certaine métrique numérique, habituellement la différence de points entre les deux joueurs. Si le joueur A est en tête, le nombre est positif ; si le joueur B est en tête, il est négatif. Tout à gauche du graphe se trouve le début du jeu, les deux joueurs sont ex-æquo. Les lignes en pointillés en haut et en bas du graphe indiquent les conditions de victoire pour les joueurs A et B.



Mon premier graphe d'équilibre représente une simple course de sprint, dans laquelle le joueur A est un meilleur coureur que le joueur B. A part immédiatement en tête et y reste pour toute la durée de la course. Les courses en ligne droite n'ont pas de rétroaction positive. Être en tête ne rend pas plus facile le fait de conserver ou d'augmenter son avance (en fait, certains faits psychologiques suggèrent que c'est plutôt le contraire qui est vrai : les coureurs essaient plus fort s'il y a quelqu'un juste devant eux. Quand Roger Bannister s'entraînait pour battre le record d'un mile en quatre minutes, il s'entraînait avec des meneurs qui couraient juste devant lui. Ce phénomène a aussi été observé dans des courses de chevaux et de chien de traîneaux. Néanmoins, ceci ne fait pas partie des règles du jeu, c'est-à-dire de ce qui nous intéresse ici.)

Jeu inéquitable (c'est-à-dire mal équilibré)

En supposant que A et B ont les mêmes capacités.

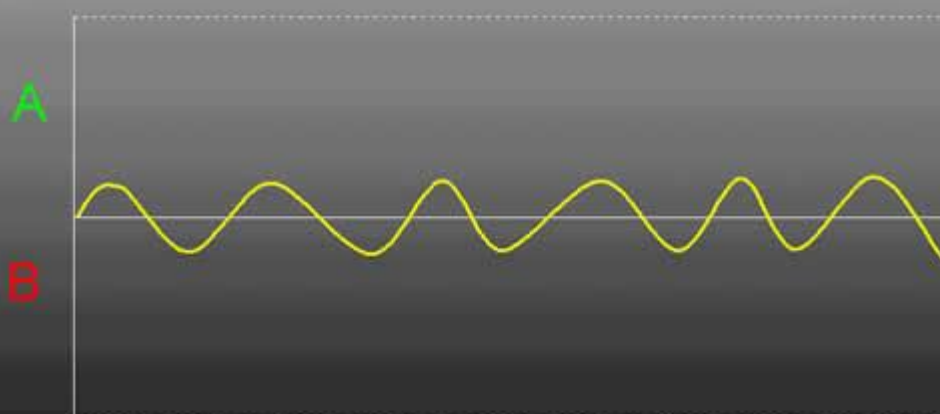


18

Ce graphe est un exemple de jeu non équilibré, *id est* inéquitable – qu'il fasse appelle à une rétroaction positive ou pas. Si on suppose que A et B ont les mêmes capacités et qu'il n'y a pas d'élément lié à la chance qui intervienne, quelque chose dans les règles donne à A un avantage, de telle sorte qu'il prend la tête et maintient son avance tout au long d'un jeu très court.

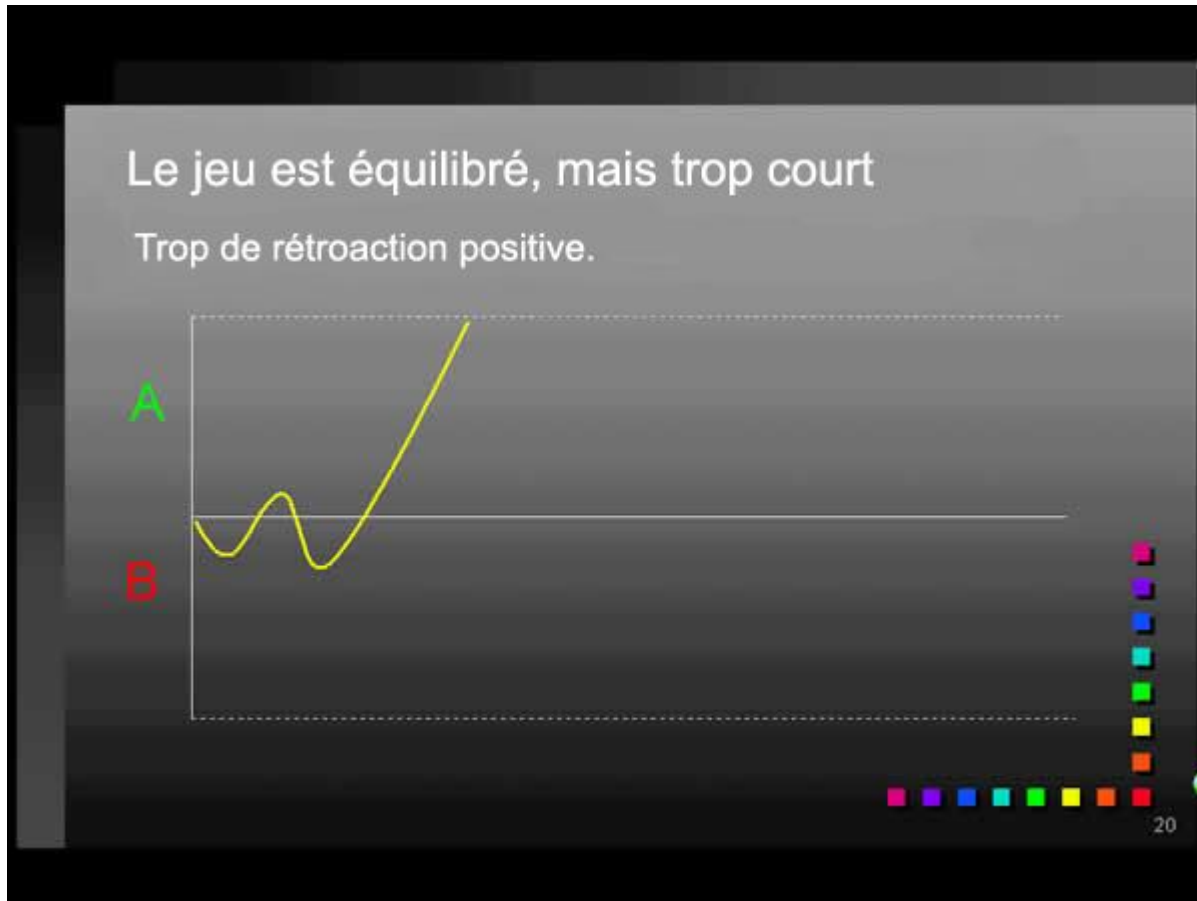
Impasse

Trop peu de rétroaction positive.



19

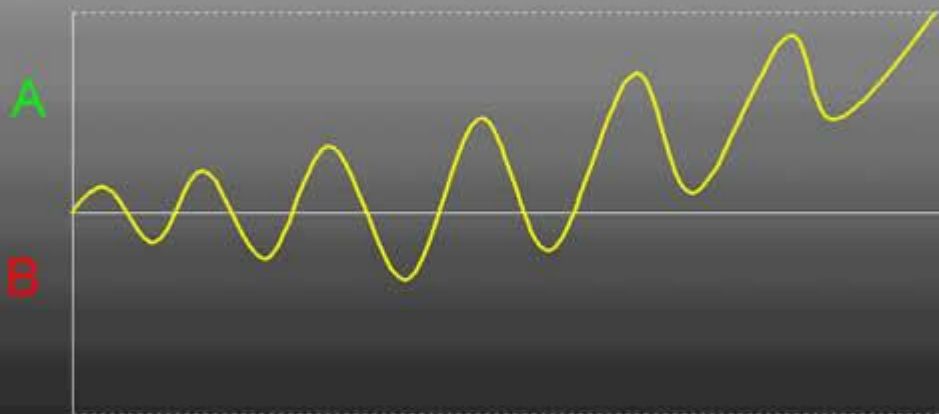
Ici, nous avons une impasse, un jeu qui continue éternellement sans que ni l'un ni l'autre des joueurs n'arrive à garder la position dominante. C'est un jeu trop équilibré : aucun des deux joueurs n'est capable d'atteindre la victoire. Le jeu de cartes pour enfants La Bataille est un bon exemple de cette sorte de jeu : tout repose sur la chance, rien sur les compétences et il n'y a pas de rétroaction positive. Ceci montre pourquoi la rétroaction positive est une chose utile : elle permet d'empêcher ce genre d'impasses. Une fois qu'un joueur a suffisamment gardé la tête, l'avantage que lui confère la rétroaction positive garantit qu'il va gagner.



La figure 4 représente un jeu qui est équilibré, mais où la rétroaction positive intervient trop tôt, ce qui aboutit à un jeu équitable mais très court. B obtient un léger avantage, puis A reprend une courte avance, puis B à nouveau, puis A garde l'avance plus longtemps et gagne rapidement le jeu.

Progression idéale du jeu

La tête change de joueur mais le meilleur gagne à la fin.



22

Le jeu idéal, selon moi, commence de façon équilibrée, mais devient lentement déséquilibré avec le temps, jusqu'à ce que l'un des joueurs gagne inévitablement – si possible le meilleur joueur ! La figure 5 en montre un exemple, bien que dans ce cas, le joueur B combatte vaillamment pour un assez long moment.

Une fois que l'avance d'un des joueurs devient suffisante, le jeu ne devrait pas mettre trop de temps à se terminer. C'est l'un des (très rares) problèmes du Monopoly. Entre le moment où il est clair que l'un des joueurs va gagner et celui où il a effectivement conduit les autres joueurs à la banqueroute, il s'écoule une demi-heure ou plus. Les autres joueurs n'ont plus qu'à rester assis et attendre tout du long de leur lente descente vers la faillite.

La rétroaction positive dans les jeux

Maintenant, regardons quelques exemples de jeux avec et sans rétroaction positive. Comme je l'ai mentionné précédemment, les courses et la plupart des autres compétitions sportives ne font pas intervenir de rétroaction positive – du moins, il n'y en a pas qui soit voulue par les règles. De même, la plupart des jeux de cartes non plus : le cribbage (*NdT : jeu de cartes sans équivalent en français*) et le rami, par exemple. D'autre part, les jeux de guerre font très nettement intervenir la rétroaction positive, particulièrement si la condition de victoire est simplement d'éradiquer toutes les unités des autres joueurs. Détruire une unité ennemie confère un avantage au joueur qui le fait. L'unité n'est plus là pour contre-attaquer, il est donc plus simple de détruire l'unité suivante et ainsi de suite.

Un autre genre de jeux vidéo qui fait intervenir la rétroaction positive est le jeu de rôle à un seul joueur. Vous commencez avec de mauvaises armes, vous tuez des monstres, vous récupérez de l'or et vous l'utilisez pour acheter de meilleures armes. Ces nouvelles armes vous permettent de tuer plus de monstres, de récupérer plus d'or, d'acheter des armes encore meilleures et ainsi de suite.

La rétroaction positive n'a pas besoin d'avoir une influence directe sur la façon de gagner ; cela peut tout aussi bien apparaître dans d'autres domaines du jeu. Lorsque j'étais chez Bullfrog Productions, j'étais lead designer sur un nouveau jeu (jamais sorti, hélas) appelé *Genesis : The Hand of God*. *Genesis* était un god

game dans l'esprit du *Populous* original de Bullfrog dans lequel vous pouviez affecter le ciel d'une région avec un pouvoir divin, en utilisant la mana généré par vos fidèles. L'une de nos innovations était qu'il y avait différents types de mana, qui dépendaient du terrain sur lequel vivaient vos fidèles. Par exemple, s'ils vivaient sur un terrain humide vous aviez beaucoup de mana d'eau, que vous pouviez utiliser pour faire pleuvoir. Bien sûr, nous nous sommes rendus compte immédiatement qu'il y avait là une boucle de rétroaction positive : plus vous aviez d'eau, plus vous pouviez obtenir d'eau. D'autre part, si votre peuple vivait dans le désert, il était très difficile de faire pleuvoir parce que vous produisiez très peu de mana d'eau.

Finalement, nous avons conclu que ce n'était pas un problème pour le jeu. Faire pleuvoir dans une zone humide la rend plus humide, et alors ? Si vous faisiez pleuvoir tout le temps, vous noyiez votre propre peuple et il n'y avait pas d'avantage à faire cela. Nous avons aussi pensé que faire fleurir le désert ne serait pas facile au départ, mais qu'une fois commencé, cela deviendrait plus facile. Bien qu'il s'agisse de rétroaction positive, cela ne mettait pas l'équilibre du jeu en danger, parce que cela ne vous conférait pas un avantage direct sur vos opposants.

Contrôler la rétroaction positive

Jusqu'ici, j'ai regardé à la fois les avantages et les dangers de la rétroaction positive. L'avantage est d'empêcher les impasses et de permettre au jeu d'arriver à une conclusion. Le danger est de déséquilibrer le jeu trop rapidement et de l'amener à se finir trop vite. Alors comment pouvons nous limiter la rétroaction positive ? Il y a plusieurs moyens de le faire.

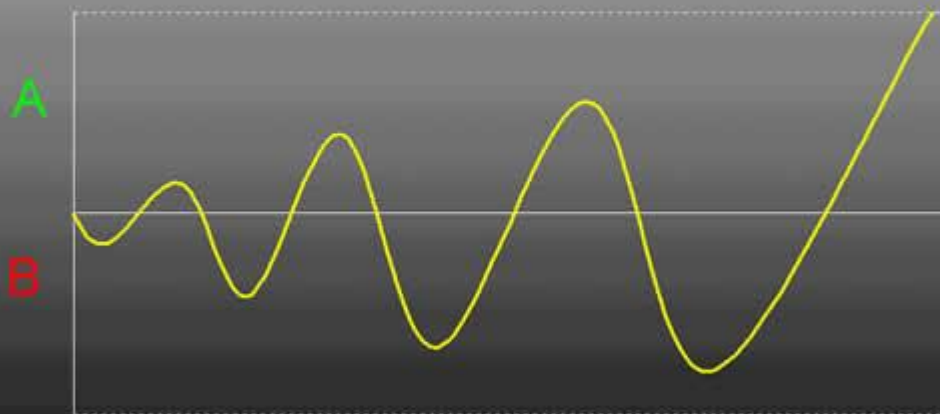
Utiliser la rétroaction négative

La rétroaction négative est l'opposé de la rétroaction positive ; c'est un effet qui tend à diminuer et non à amplifier la sortie d'un système. Un bon exemple de l'utilisation de la rétroaction négative est la façon dont la sélection des joueurs fonctionne dans le football américain professionnel. En général, une équipe qui gagne beaucoup aura beaucoup plus d'argent qu'une équipe qui perd beaucoup. Une équipe gagnante pourrait utiliser cet argent pour surenchérir sur les autres équipes et engager les meilleurs nouveaux joueurs qui arrivent du lycée. Les meilleures équipes gagnent plus d'argent, ce qui leur permet d'engager les meilleurs joueurs, et ainsi elles continuent à gagner. Les équipes pauvres ne peuvent s'offrir que de mauvais joueurs, et continuent donc à perdre – un exemple clair de rétroaction positive.

De façon à empêcher ceci et à essayer d'équilibrer la force de toutes les équipes, la National Football League a introduit un système de recrutement. Les nouveaux joueurs ne peuvent pas juste se proposer aux enchères et être achetés par les plus offrants ; au lieu de cela, les équipes prennent des tours pour choisir des joueurs parmi ceux disponibles, et, ce qui est plus important, les pires équipes choisissent en premier. Ceci signifie que les pires équipes ont la priorité pour le choix des meilleurs joueurs et la qualité des équipes est ainsi nivelée d'une certaine façon. Bien sûr, ce n'est pas aussi simple en pratique ; les équipes sont autorisées à échanger leurs positions dans l'ordre de sélection et la qualité de leur jeu dépend beaucoup de la qualité de leur entraînement, sans parler des autres joueurs qui sont déjà dans l'équipe. Mais le principe est bon. Le système de recrutement aide à empêcher qu'une équipe se maintienne dans une position d'invincibilité grâce à la rétroaction positive.

De violentes variations dans l'avance

Les effets d'une forte rétroaction négative.



21

Soyez néanmoins prudent avec la rétroaction négative – si elle est trop forte, elle peut aboutir à des impasses ou même des oscillations brusques dans la domination du jeu, comme l'illustre cette figure. Dans cet exemple, prendre la tête confère un désavantage fort ce qui permet à l'autre joueur de prendre la tête et ainsi de suite.

On voit souvent cela dans les jeux multijoueurs au tour par tour pour adultes, où le but n'est pas vraiment de récompenser la maîtrise du jeu, mais de passer un bon moment sans trop s'occuper de qui est sur le point de gagner. Tout le monde parvient à être en tête à un moment ou à un autre et qui est le vainqueur est principalement une question de chance.

Limiter le bénéfice apporté par la rétroaction positive

Aux échecs, cela aide évidemment beaucoup d'enlever l'une des pièces de votre adversaire du plateau. Mais imaginez comment seraient les échecs si pouviez, non pas simplement ôter la pièce, mais en faire l'une des vôtres. Prendre une pièce adverse vous conférerait alors un bien plus grand avantage, en d'autres termes, une plus forte rétroaction positive. Les parties d'échecs seraient plus courtes. Ceci se produit effectivement dans les échecs japonais, qui sont appelés shogi. Un joueur qui a ôté l'une des pièces adverses du plateau peut la réintroduire plus tard comme l'une des siennes (avec certaines restrictions).

C'est pourquoi la plupart des jeux de stratégie en temps réel ne vous autorisent pas à prendre les unités ennemies et à les utiliser comme les vôtres, ni ne vous autorise à vous emparer des dispositifs de production adverses. Si vous pouviez mettre la main sur les usines de votre adversaire et commencer à y fabriquer des unités pour votre compte, le jeu ne durerait pas très longtemps. Il s'agit d'une limitation de l'avantage apporté par la rétroaction positive.

En plus du processus de recrutement, le football américain professionnel dispose d'autres règles qui aident à empêcher les équipes les plus riches de dominer toutes les autres année après année. L'une d'entre elle est la restriction des salaires, qui limite la somme totale que peut dépenser chaque équipe pour payer ses joueurs. Peu importe combien d'argent l'équipe a, elle ne peut pas en dépenser plus pour les salaires que ce que fixe

cette règle. Une autre règle fixe la taille des équipes. Aucune équipe ne peut avoir plus de cinquante-trois joueurs durant la saison de jeu régulière, même si elle peut se le permettre financièrement. Ces restrictions artificielles servent à contenir les effets de la rétroaction positive. Dans les faits, elles permettent de s'assurer que « l'argent ne fait pas tout ».

Définir la victoire en d'autres termes

Comme je l'ai mentionné précédemment, la condition de victoire n'a pas besoin d'être définie en termes directement influencés par la rétroaction positive. Aux échecs, la victoire est définie en termes stratégiques plutôt que numériques. Dans les jeux de stratégies en temps réels, vous pouvez créer des missions dans lesquelles la victoire doit être atteinte furtivement, ou bien en détectant une certaine faiblesse dans les défenses ennemies, ou bien en survivant un certain laps de temps, ou bien de nombreux autres scénarios. L'une des faiblesses des RTS en ce moment est que trop d'entre eux reposent sur le principe de submerger l'ennemi avec d'énormes effectifs – dans les faits, l'efficacité de la production – plutôt que sur celui de récompenser les compétences stratégiques. Ceci conduit à une mentalité « chair à canon » chez les joueurs qui rappelle désagréablement Field Marshal Haig dans la Somme (NdT : *un commandant anglais surnommé le « Boucher de la Somme » pour le nombre de morts dans ses rangs lors de la Première Guerre Mondiale*). Espérons qu'ils ne s'entraînent pas avec ce genre de jeux à West Points (NdT : *l'une des deux plus prestigieuses écoles militaires des États-Unis*).

Augmenter la difficulté pour compenser

C'est exactement ce que font les jeux de rôle. Comme je l'ai décrit plus haut, il y a clairement une rétroaction positive dans l'évolution du personnage : gagner des batailles vous permet d'acheter de meilleures armes qui vous permettent de gagner encore plus de batailles. Si vous faisiez toujours face à la même sorte d'ennemis, vous deviendriez rapidement invincible, et le jeu ne serait pas vraiment très difficile. En conséquence, à mesure que la force et les compétences de votre personnage se développent, le jeu augmente aussi la résistance de vos ennemis. La difficulté à gagner chaque combat reste à peu près constante (avec des variations locales) tout au long du jeu.

Augmenter l'influence de la chance

Ce n'est pas la meilleure façon de réduire les effets de la rétroaction positive, mais ça marche. Le Monopoly le fait d'une certaine façon. Un mauvais lancer de dés peut faire régresser le joueur en tête de façon significative. Bien sûr, ceci peut faire autant de mal que de bien, à moins que vous ne pipiez les dés contre le joueur en tête – et si vous le faites, vous feriez mieux de ne pas leur en parler ! Les jeux pour enfants ont tendance à utiliser la chance plus que les jeux pour adultes. Ceci permet d'équilibrer les différences de maîtrise et permet au perdant de blâmer la malchance plutôt que le lui-même (je ne cautionne pas nécessairement ceci, je ne fais que le remarquer).

Conclusion

Équilibrer un jeu vidéo à un joueur est un peu différent d'en équilibrer un à plusieurs joueurs. Dans un jeu à un seul joueur, il n'est pas nécessaire d'être « équitable » de la même façon que dans un jeu à plusieurs joueurs. Le défi dans les jeux de stratégie temps réel à un seul joueur et les jeux de rôle repose plus sur l'ignorance du joueur sur ce qui s'oppose à lui que sur les compétences de stratégie et de gestion de l'ordinateur – souvent parce que l'ordinateur n'en a pas tellement. Mais la plupart des RTS sont conçus avec des modes multijoueurs de nos jours et dans ces cas, il est nécessaire de les équilibrer proprement et de faire en sorte que vous êtes juste avec chaque joueur, tout particulièrement s'ils disposent de forces asymétriques. En mode multijoueur, la rétroaction positive a un rôle important à jouer. Utilisez la avec sagesse !

La leçon de dessin de Arne

Par [Antoine Gersant](#) | le 27 octobre 2008 16:33:00 | Catégories : [Cours](#), [Dessin](#)

Cet article est une traduction de [Antoine Gersant](#).

L'[article original](#) est écrit par Arne publié par itch_studios (janvier 2005).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Je crois qu'il existe une logique qui dicte si une image est réussie ou pas. Je crois aussi que les humains sont des machines faites de viande, et qu'un jour les ordinateurs seront capables de faire ce que nous faisons. Puisque la logique peut être mise sous forme de règles, de lignes de conduites et de théories, on peut la partager. C'est ce que je vais essayer de faire ici.

Gardez à l'esprit que j'ai déduit empiriquement les théories que je présente ici, et que je suis une machine de viande qui a de grandes chances de se tromper. A un certain niveau, une peinture est assez réussie pour que nos préférences personnelles et nos capacités intellectuelles et cognitives ne remarquent plus ou ne s'attardent plus sur les imperfections. Plus vous en saurez sur l'art et plus vous remarquerez les petits défauts (et plus vous passerez pour un vieux grincheux aux yeux des autres).



De nombreuses règles vont à l'encontre les unes des autres, s'annulent mutuellement, ou sont à oublier pour certaines approches stylisées. Ce dont je vais parler ici n'est que le large spectre des règles de base sur lesquelles nous sommes généralement d'accord. Les extra-terrestres pourraient trouver étrange la partie sur les ombres froides et la lumière chaude...sans parler de la partie où j'évoque les visages humains.

Le conseil de loin le plus utile que je puisse donner aux artistes en cours d'apprentissage est : Entraînez-vous.

Étudiez tout et sur tous les supports. Oui, cela veut aussi dire dessiner ce que vous détestez. Alors à quoi bon ce cours si tout ce qui compte c'est l'entraînement ? Et bien je pense qu'il y a quelques choses qu'il est bon de savoir dès le début, mais vous ne remarquerez aucune amélioration simplement après avoir lu ce cours. Il ne fait aucun doute qu'il vous faudra du temps pour digérer ce que j'ai écrit ici, alors vous pourriez avoir besoin de jeter de nouveau un œil à ce texte dans quelques mois.

1. Conditions d'utilisations

Vous êtes libre de traduire et héberger ce cours tel qu'il apparait pour l'instant. Vous êtes également libre de le distribuer à des fins éducatives (pour le distribuer à des étudiants par exemple). Il doit cependant demeurer gratuit.

Étant donné que je risque de le mettre à jour et que les nouvelles versions sont en général mieux que les précédentes, je préférerais éviter la multiplication des miroirs de cette page. Je ne peux pas mettre à jour les miroirs que je ne contrôle pas. De plus, exposer ce cours entouré de publicités ou pour gagner du trafic sur un site est une idée de débile. Vous êtes sur internet et un lien vers cette page devrait suffire si vous voulez attirer l'attention de quelqu'un sur ce cours.

Je crois que ces conditions d'utilisation disent à peu près la même chose :



Le contenu de cette page est mis à disposition selon les termes de la [Licence Creative Commons Paternité — Pas d'Utilisation Commerciale — Partage à l'Identique 3.0 non transcrit](#).

Si vous avez traduit cette page, ou si vous repérez une traduction, merci de me le faire savoir pour que je puisse mettre un lien vers cette traduction ici.

Par Arne Niklas Jansson — 2005 & 2007

Contact : [Rébus e-mail](#) (indice : la réponse est diglett (*NdT : Taupiqueur*) et est hébergée chez le célèbre client de messagerie de Google)

Traduction : Antoine Gersant (texte) et Shin (images)

2. Terminologie

Je vais essayer de me tenir éloigner du vocabulaire compliqué. Voici quelques mots de base.

HSL — Trois dimensions de la lumière : Teinte, Saturation et Valeur. On peut les représenter sur un cylindre en tant que l'angle, le rayon et la hauteur.

RVB — Trois autres dimensions de la lumière : Rouge, Vert et Bleu.

Exposition — Tout comme l'œil, un appareil photo est un dispositif qui attrape les photons. Quand il enregistre une image, il compte les photons « par pixel ». Quand une image est transformée en une image plate, son support n'est pas assez lumineux pour briller comme le soleil. Les valeurs de lumière doivent être tronquées dans les extrêmes les plus clairs et les plus sombres. Les appareils photos peuvent enregistrer les images dans un format appelé RAW qui conserve toutes les valeurs. Ces images peuvent être observées sur des « moniteurs à grande gamme dynamique » qui sont capables d'émettre des valeurs plus claires que les moniteurs usuels. Quand on peint sur un support normal, l'impossibilité de peindre des choses brillantes nous oblige à raccourcir à différentes astuces qui simulent la luminosité. Le peintre Anders Zorn faisait ça très bien.

- **Teinte** — Les variations de couleur. Par exemple, la peau a des variations de teinte comme les joues qui sont plus roses que le reste.
- **Saturation** — La saturation décrit la quantité de couleur présente. Les couleurs « néons » sont très saturées. Les couleurs claires et sombres peuvent aussi être saturées, mais dans l'espace de couleurs RVB, la saturation est plus visible pour les valeurs moyennes. Cela est dû au fait que les couleurs paraissent plus saturées quand les trois composantes RVB ont des valeurs très différentes, ce qui nous place souvent dans des valeurs moyennes.
- **Valeur, Luminosité** — Il y a plusieurs façons de calculer la valeur. Dans le système RVB, le vert est en général le plus clair, suivi du rouge puis du bleu. Dans un programme que j'ai un jour codé, j'ai utilisé la formule

suivante pour calculer la valeur d'un pixel depuis ses trois composantes RVB : $Valeur = 0,299 * Rouge + 0,587 * Vert + 0,114 * Bleu$.

- **Lisible** — Se dit quand on peut comprendre ce qu'une image est censée représenter.
- **Radiosité, ambiance et lumière réfléchie** — J'emploie ces termes pour parler du fait que la lumière qui atteint une surface est réfléchie pour aller atteindre ensuite une autre surface. Cela arrive en permanence. Dans une salle verte, tous les objets prendront une teinte verte par exemple. La radiosité est un terme pour les rendus 3D, j'utilise « lumière réfléchie » pour les reflets dont je connais la source et « Ambiance » pour ceux qui caractérisent une couleur un peu partout dans les environs.
- **Speculars (sans équivalent français)** — Les points blancs de lumière qui apparaissent sur les surfaces réfléchissantes.
- **Aplanir** — Enlever les textures et changements de valeurs inutiles
- **Texture, bruit** — Poussière et autres petits détails de l'image qui ne décrivent rien en particulier.
- **Forme, modelage** — La forme en 3 dimensions de quelque chose
- **Plan** — Je ne sais pas exactement, je m'en sers pour dire « premier plan », « second plan » et « arrière plan ». Parfois je m'en sers pour parler d'une surface plane.
- **Perspective atmosphérique** — L'air n'est pas parfaitement transparent. Comme vous l'avez peut être remarqué, les choses ont tendance à s'effacer avec la distance et les couleur dérive parfois vers le bleu du ciel. La perspective atmosphérique peut servir pour séparer visuellement les différentes parties de votre image à des distances différentes. On peut également l'exagérer. Certains artistes, dont moi, utilisent la perspective atmosphérique pour séparer les différents membres d'un corps qui se superposent ou pour assurer la transition avec l'arrière plan.

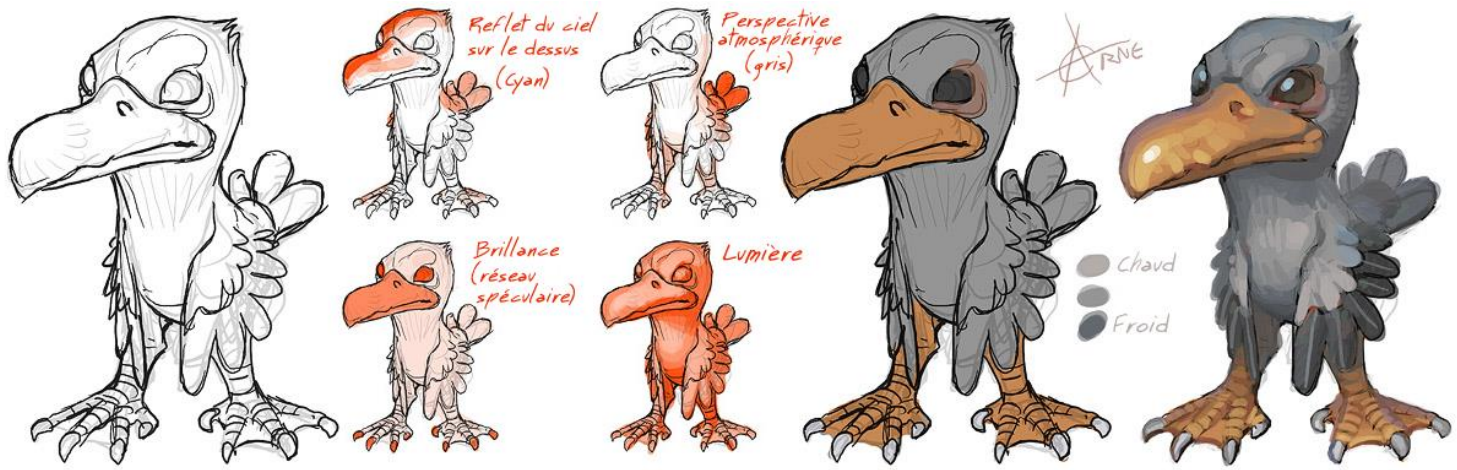
3. Apprendre à voir

La partie consciente du cerveau humain ne peut pas gérer beaucoup d'informations, il est plus efficace de développer des réflexes pour la plupart des tâches courantes et répétitives. Une partie de notre cerveau traite les images venant des yeux. Une fois traitées, très peu d'informations sont en fait envoyées à notre conscience (certains autistes ont d'ailleurs un trouble de ce processus, qui les surcharge d'information). Malheureusement pour nous autres artistes, notre cerveau modifie ainsi notre perception de l'image et nous ne voyons plus vraiment ce que nous voyons. Dans une certaine mesure, nous devons apprendre à réaliser le traitement inverse à celui de la perception.

- **Déplier** — Notre œil parvient à voir toutes les surfaces comme de face ou de plusieurs vues à la fois. Par exemple, le côté d'une boîte vue de travers ne nous apparaît pas déformé.
- **Exposition** — Nous parvenons à restituer la couleur d'un objet qu'il soit dans l'ombre ou à la lumière.
- **Balance des blancs** — Nous voyons toujours les feuilles de papiers comme du blanc, que l'on soit à l'intérieur ou à l'extérieur. Les appareils photos eux font mal la balance des blancs. Si un photographe ne fait pas attention, ses photos d'intérieurs finissent souvent orangées (à cause de la lumière dans ampoules) et ses extérieurs trop froids (à cause du bleu du ciel).
- **Redimensionnement avec la distance** — Une personne au loin nous semble être de la taille d'un humain. Toutefois, j'ai entendu l'histoire d'un homme qui aurait grandi dans une jungle épaisse et qui n'aurait donc jamais rien observé au loin. Le jour où un scientifique l'a emmené dans la savane où un rhinocéros se tenait à l'horizon, l'homme le prit pour un insecte. Lorsqu'ils s'approchèrent, l'homme crut que l'insecte « grandissait ».

Quand nous dessinons depuis une image plane comme une photographie, il nous est plus facile de voir les proportions, les alignements et les distorsions de perspective. D'une certaine façon, une partie du travail est déjà faite (et sans doute assez mal). Il est bien plus difficile d'étudier depuis la réalité, mais dans l'état actuel des choses les photographies ne nous donnent pas « toute l'image ». Les appareils photos tronquent des valeurs et modifient les teintes subtiles, sans parler du fait que les photographies ne sont pas en stéréo.

4. Pelures d'oignon (penser en calques)



Avant de donner un coup de pinceau, il faut penser à un bon paquet de choses. En fait, vous ne devriez même pas y penser et en faire des automatismes.

- Sentir le volume et l'angle de la forme.
- Déterminer d'où arrive la lumière sur cette forme.
- Essayer de déterminer si certaines ombres atterrissent sur cette surface
- Existe-t-il une lumière réfléchie (radiosité) qui arrive sur cette surface ?
- Quelle est la couleur d'ambiance de la scène (lumière globale réfléchie) ?
- A-t-on besoin d'ajouter des speculars ? La surface est elle mouillée ou réfléchissante et orientée de sorte à réfléchir une source de lumière comme le ciel ?
- Quel est le niveau d'exposition ? Suis-je en train de peindre un objet si éclairé qu'il est plus que blanc ? L'objet est-il tellement dans l'ombre que même son point le plus clair est plongé dans l'obscurité ?
- Du brouillard intervient-il ?
- Quelle est la texture de la surface ?

Notez bien que tout cela concerne surtout les styles réalistes. Chaque coup de pinceau devrait être efficace et cohérent avec le reste de la peinture et votre choix de couleur. Vous pourriez aussi bien avoir un style ou une idée qui vous interdit certaines couleurs ou textures et qui donne la priorité à d'autres éléments. Toutefois, même les illustrations des Super Nana sont des simplifications de représentations réalistes. N'espérez pas vous cacher derrière un « Ça ne fait pas partie de mon style donc je n'apprend pas à le faire ».

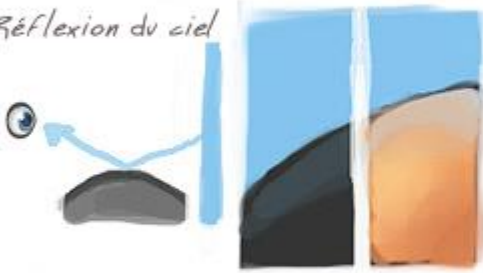
5. Lumière

Il n'existe dans la réalité qu'un seul type de lumière, et elle rebondit. La part de lumière (les photons en fait) que vous percevez est celle qui après avoir rebondi finit par rentrer dans vos yeux. La lumière fait deux choses quand elle frappe une surface : tout d'abord une partie de la lumière est absorbée, et c'est ce qui donne aux choses leurs couleurs. Une pomme réfléchit surtout les longueurs d'onde correspondant au rouge et le reste est absorbé et transformé en chaleur ou autre chose. C'est pour ça que les objets noirs chauffent au soleil d'ailleurs. Bref, la lumière rebondit différemment selon la nature de la surface qu'elle frappe. Si la surface est irrégulière, la lumière va rebondir de façon un peu aléatoire, comme quand une balle de tennis tombe sur du gravier. Si au contraire la surface est lisse, la lumière rebondira selon une trajectoire prévisible. Un miroir est une surface très lisse, si bien que la lumière repart non déformée, et on peut y voir son reflet. Remarquez que toutes les surfaces ont des speculars, puisque ces derniers ne sont que de la lumière réfléchie. C'est juste qu'ils sont très très dispersés et dilués sur les surfaces mats.

Réflexion spéculaire



Réflexion du ciel



Types de surfaces

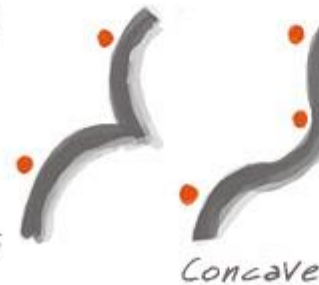
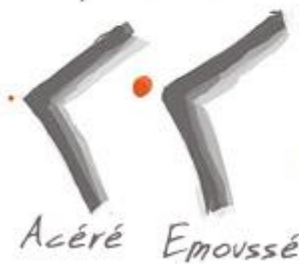


Brillant



Mat

Speculars



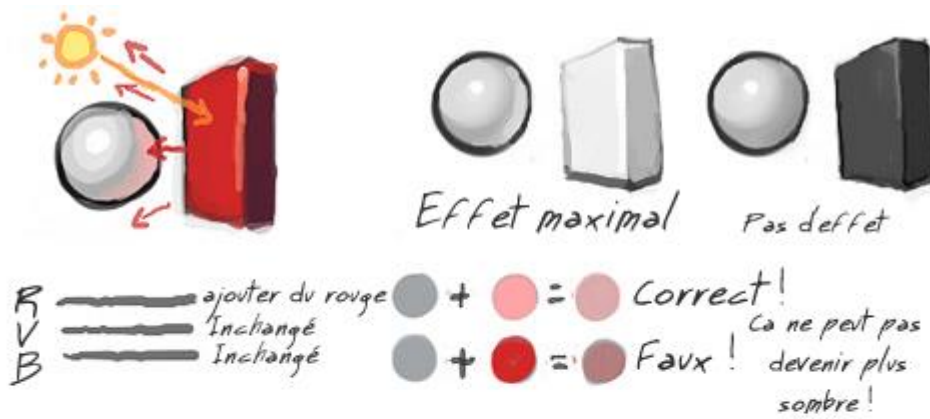
Selon la position de l'œil ou du tyranoeil, il verra la lumière différemment et les speculars a des endroits différents sur une surface courbe comme celle-ci. Une flaque d'eau n'est pas courbée (mise à part les bords, à cause des tensions de surface) donc vous ne verrez un reflet brillant que depuis un certain angle. Les speculars ne peuvent apparaître que dans un environnement contenant une source de lumière ponctuelle comme le soleil, une ampoule ou une petite fenêtre.



Il y a aussi des speculars sur les vêtements, plus dilués et plus subtils. J'ai tiré sur la manche de ma chemise avec deux doigts pour obtenir une surface plane entre les deux points indiqués (et j'ai ensuite bougé la caméra, pas la chemise).

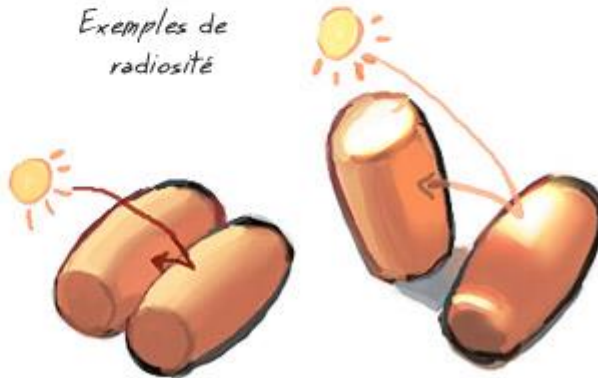
Sur terre il y a en général un environnement dense sur lequel la lumière peut rebondir, du coup les objets sont plus ou moins éclairés de tous les côtés. Par exemple nous avons le ciel qui est comme une source de lumière bleue en forme de dôme. Nous avons aussi le sol, les murs et d'autres surfaces. Dans l'espace il n'y a en gros qu'une seule source de lumière, le soleil. C'est pour ça que la lune a une face éclairée et une face dans l'ombre, d'où son air plat. Cependant, si vous regardez attentivement vous pouvez voir la lumière réfléchiée par la Terre sur la face ombragée de la lune. A cela s'ajoute la lueur des étoiles autres que le soleil, qui est je suppose encore plus faible.

La deuxième chose que fait la lumière quand elle frappe une surface et rebondit, c'est changer de couleur. Si elle frappe une nouvelle fois une surface de la même couleur, cette dernière gagnera en saturation.



La radiosité peut sur peau
augmenter la saturation

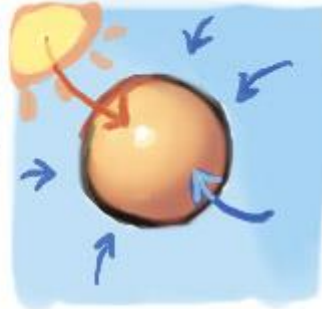
Exemples de
radiosité



Extérieur

Intérieur

Chaud/Froid



6. Exposition

La lumière du soleil est bien plus intense que celle du ciel, qui est elle-même plus forte que la lumière en intérieur. Nos yeux s'adaptent automatiquement après un certain délai, et nous pouvons également les adapter manuellement en plissant les yeux ou en fixant notre attention sur un objet. A cause de cela, il est difficile de comprendre que nos yeux sont en fait assez limités. Leurs limites deviennent évidentes grâce aux appareils photos. Si vous prenez une photo en intérieur, les fenêtres seront surexposées (blanches). Si vous essayez d'ajuster le niveau d'exposition en fonction de la lumière de la fenêtre, l'intérieur de la pièce sera alors sous-exposé.

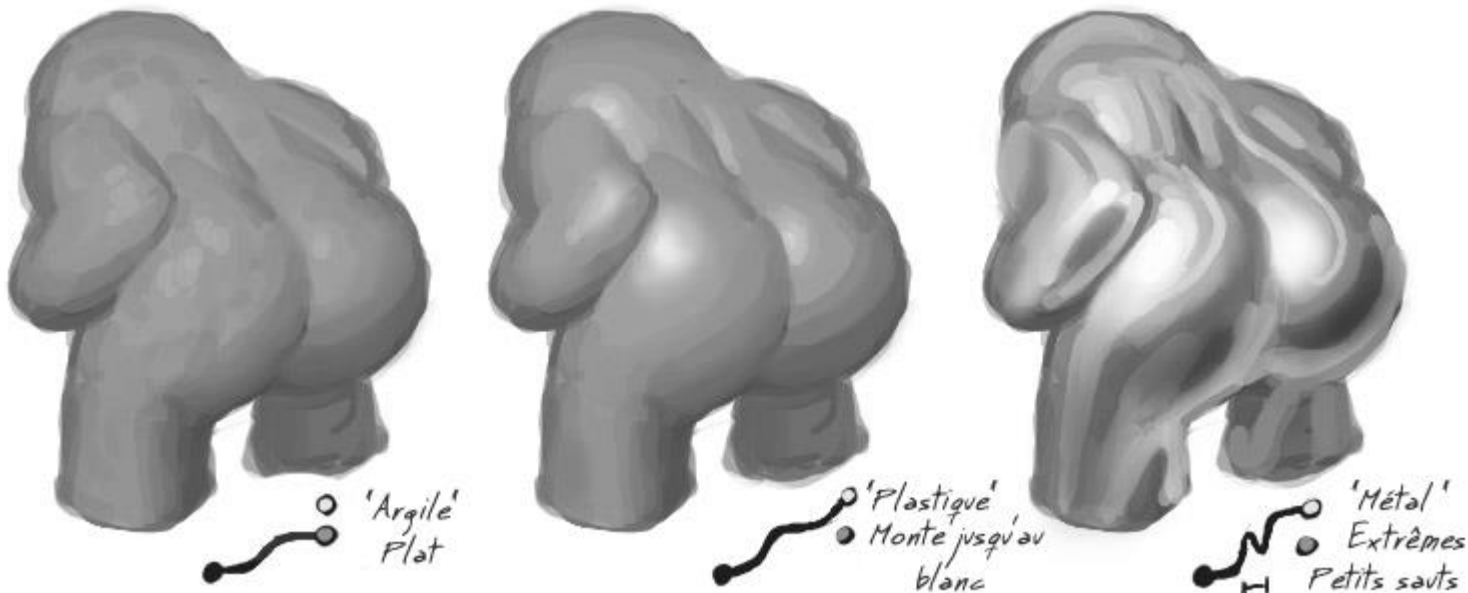
Ce phénomène peut être utilisé à votre avantage, par exemple en plaçant un personnage ou un objet au premier plan là où il sera sombre, de telle sorte que sa silhouette se lise bien avec une pièce très éclairée en fond.

Les variations d'exposition peuvent aussi faire rendre certaines parties d'un corps très claires ou très foncées, et les éloigner très loin des couleurs de peaux habituelles. Quand l'ombre est très sombre et que le côté éclairé est surexposé, la couleur ne peut se loger qu'à la frontière des deux.

7. Matières

Voici quelques exemples de matières et la façon dont je fais leurs rendus.

- **Vêtements** — Très peu de speculars, juste des zones d'ombre et de lumière. Parfois, une vive lumière peut traverser un vêtement fin et être diffusée sous la surface.
- **Cuir** — Il peut parfois être un peu brillant et du coup avoir quelques speculars. Attention à ne pas le faire trop saturé.
- **Arbres et bois** — Mat. Et peu saturé aussi (un peu gris brun / terre de sienne).
- **Pierre** — Un peu comme les vêtements. La surface est en général trop irrégulière (aussi bien à l'échelle microscopique que macroscopique) pour avoir des speculars visibles.
- **Plastique** — On dirait que les speculars et les reflets sont de la même couleur que le plastique. Les plastiques sont parfois aussi un peu transparents.
- **Or** — L'or n'est pas orange. J'utilise du noir, parfois du orange désaturé avec des pointes de vert, puis ensuite plus clair vers le jaune et le blanc.
- **Argent** — Plus ou moins comme un miroir.
- **Metal** — Pour les armures, boostez un peu les valeurs, et pas trop de teintes médianes.
- **Métal brossé** — Un intermédiaire entre une surface grise et une surface en argent.
- **Verre** — Souvent simplement transparent avec des distorsions. Les speculars apparaissent brutalement et sont souvent blancs. Dans le cas des vitres de voitures, vous aurez peut-être remarqué qu'il est plus simple de voir ce qui est derrière si il y a un ombre sur la fenêtre, la réflexion la plus claire masque ce qui est derrière.
- **Surfaces mouillées** — Plus de speculars ! Les vêtements peuvent devenir transparents (*NdT : pensez Miss T-Shirt mouillé*) et les pierres deviennent plus saturées (et leurs détails ressortent plus).



8. Ombres

Les ombres sont plutôt plates et en général moins saturées que le côté éclairé. La lumière d'ambiance se repère plus facilement dans l'ombre. Les ombres deviennent floues avec la distance (phénomène de diffraction).



9. Les tons de peau

Considérez l'environnement. la lumière est plus forte dehors, et la peau a tendance à y être moins saturée à cause de la lumière ambiante du ciel bleu et des speculars bleus. Parfois la couleur de la peau dérive vers le violet à cause du bleu du ciel qui s'en mêle. C'est particulièrement vrai si le sujet est à l'ombre.

En intérieur (sans fenêtres, et avec un éclairage à base d'ampoules), la lumière plus chaude augmente la saturation et amplifie les tons oranges et rouges. Les ombres sur la peau peuvent parfois s'éloigner vers le vert, particulièrement si la pièce comporte des objets verts (papier-peint, plantes, meubles).

Dans une pièce blanche ou une salle de bain, les teintes de peau sont plus pâles, proches des couleurs du lieu et moins contrastées en raison de la forte lumière d'ambiance.

Une pièce éclairée par une seule source de lumière intense donnera probablement des ombres presque noires.

...en bref l'environnement dans lequel se trouve la peau affecte beaucoup la façon dont vous devez en faire le rendu.

10. Les teintes de peau

Le corps humain a de nombreuses teintes différentes. Les parties qui sont couvertes par des vêtements sont par exemple moins bronzées : le pubis, les hanches et le torse sont plutôt pâles. Les épaules et les avant-bras (côté extérieur) sont un peu plus bronzés. La peau des genoux et des coudes est plus foncée que le reste. Le visage comporte de nombreuses teintes, comme les joues rosées, et les hommes peuvent avoir du gris voire presque du vert sur la mâchoire s'ils sont adeptes de la barbe de trois jours. La meilleure façon d'apprendre quelle sont les teintes du corps humain est évidemment de l'étudier. N'oublier pas que les animaux, les monstres et les objets ont aussi des teintes diverses. Si vous peignez tout avec la même teinte et la même saturation le résultat sera ennuyeux.

Certaines teintes sont dues aux lumières réfléchies et aux lumières d'ambiance. Les épaules et autres surfaces pointées vers le haut peuvent ainsi recevoir un peu du bleu du ciel.

Les dégradés saturés

Les dégradés entre ombre et lumière ne sont pas de simples fondus mélangeant les deux couleurs. De tels fondus ont l'air sans vie. Si vous regardez les images ci-dessous, vous verrez que les dégradés réussis sont ceux saturés au milieu.



La diffusion sous les surfaces



Les lumières intenses peuvent pénétrer les surfaces de certaines matières et rebondir à l'intérieur avant de sortir à nouveau. Cela augmente la saturation et la surface a l'air éclairée depuis l'intérieur. Dans le cas de la peau humaine, on le voit parfois sur les contours nets entre ombre et lumière.



Les feuilles sont brillantes sur le dessus, ce qui veut dire qu'il peut parfois y avoir un specular bleu dessus. La lumière brillant à travers la feuille augmente la saturation du dessous, c'est aussi vrai pour les oreilles et les doigts, qui peuvent prendre une couleur « super rouge » quand ils sont très retroéclairés.



Diffusion sous la surface du bout des doigts. La lumière sur le côté gauche du pouce est sans doute de la lumière réfléchie depuis l'index.

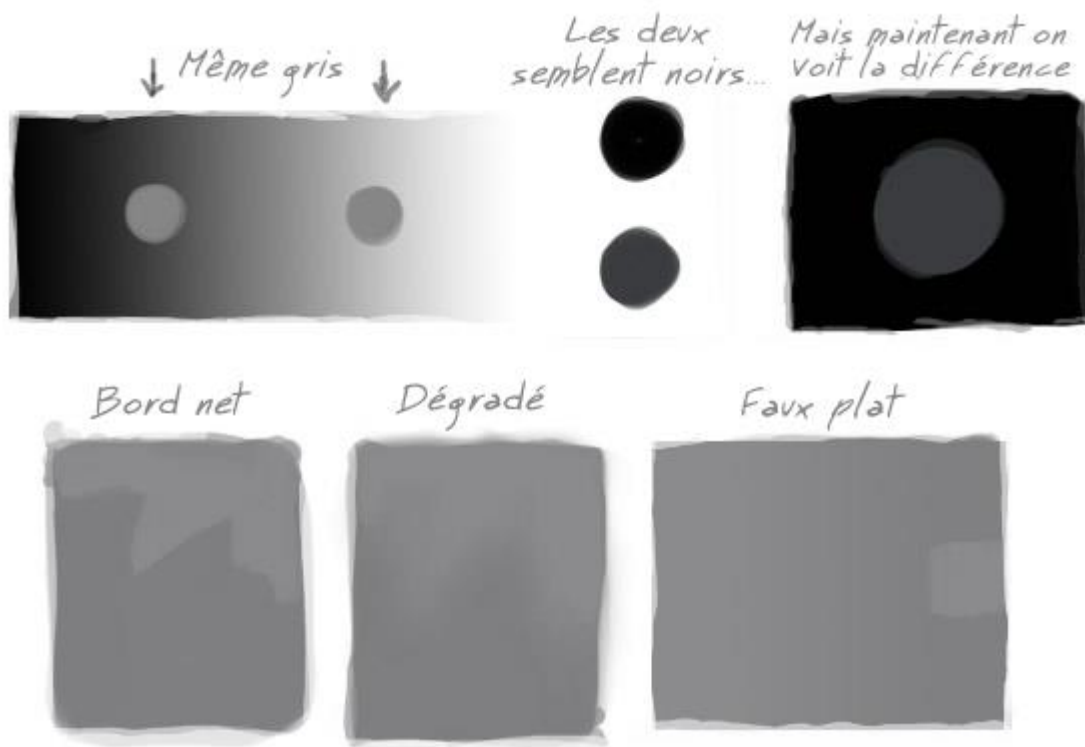


Remarquez que le contour n'apparaît que si l'objet est surexposé. Ce n'est pas aussi prononcé sur le pouce.

11. Relativité des couleurs

Les couleurs et les valeurs sont relatives. En utilisant différentes astuces, il est possible de faire croire à l'observateur qu'une couleur en est une autre, ou une valeur différente. Malheureusement, les artistes se font aussi souvent avoir et utilisent plus de valeurs ou de couleurs que nécessaire.

Un contour net se remarque plus qu'une transition douce. Vous devrez apprendre quand utiliser l'un plutôt que l'autre. Parfois vos choix de valeurs seront très limités, par exemple quand vous travaillerez sur des zones d'ombres. En utilisant des contours durs, vous pourrez alors décrire plus de détails en utilisant moins de valeurs différentes. Toutefois, les dégradés sont bien pratiques pour changer de valeur sans que l'observateur ne s'en aperçoive. Les « faux-plats » ont bel et bien l'air plats alors que ce sont des dégradés. Sur l'illustration du faux-plat, le carré à droite est de la couleur la plus à gauche du rectangle.



Les couleurs de même valeur sont perçues de façon relative à leur teinte. Une erreur courante est de dessiner un détail trop saturé, ce qui rend grisâtre quelque chose d'autre au voisinage de ce détail on augmente alors la saturation ici aussi pour composer et au final tout est trop saturé.

La patate a l'air ennuyeux



la patate ressort



Le gris apparaît froid

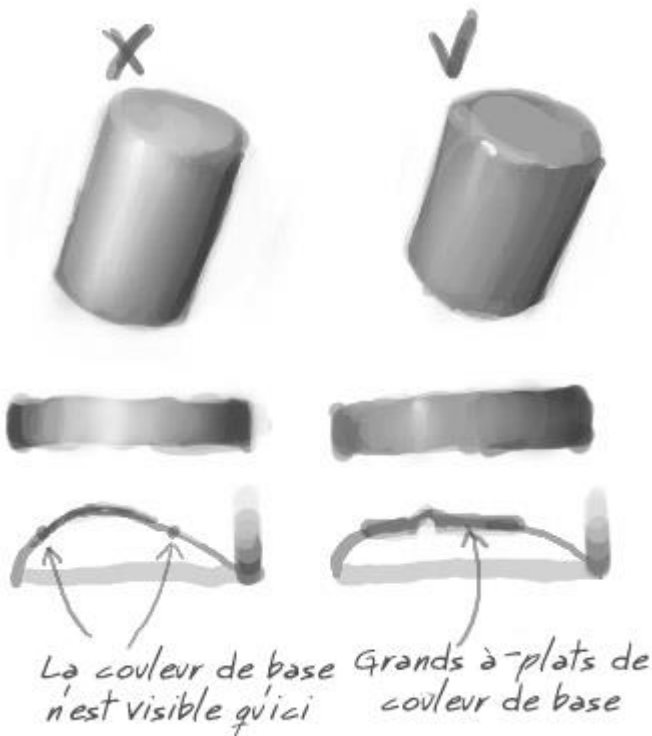


Le gris apparaît chaud



12. Identité des couleurs

On se laisse souvent aller à mettre trop de highlights. Cela masque les couleurs réelles des objets, et vous feriez mieux d'utiliser les ombres pour dépeindre les volumes du sujet.



13. Aplanir et simplifier

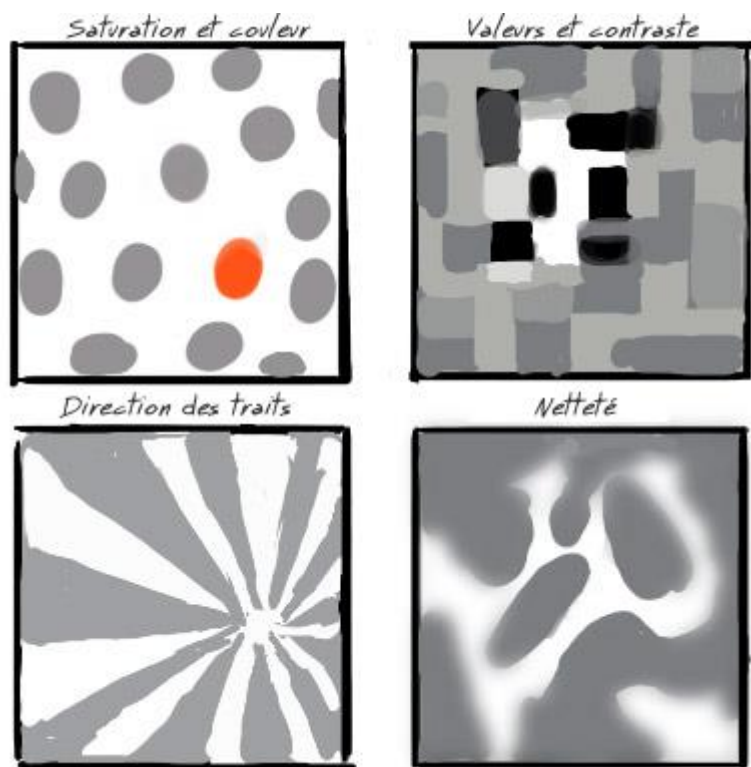
Travaillez avec des brosses grosses et supprimez les coups de pinceau inutiles. Regardez ci-dessous l'exemple de ce qu'il faut faire (ou pas). Je n'ai pas fait grand chose sur le deuxième, il est en fait simplifié. Il est surprenant de voir comment un peu d'aplanissement ici et là peut tout changer. J'ai aussi passé un peu plus de temps sur le visage car un visage raté peut tout foutre en l'air. L'image est basée sur une photo.



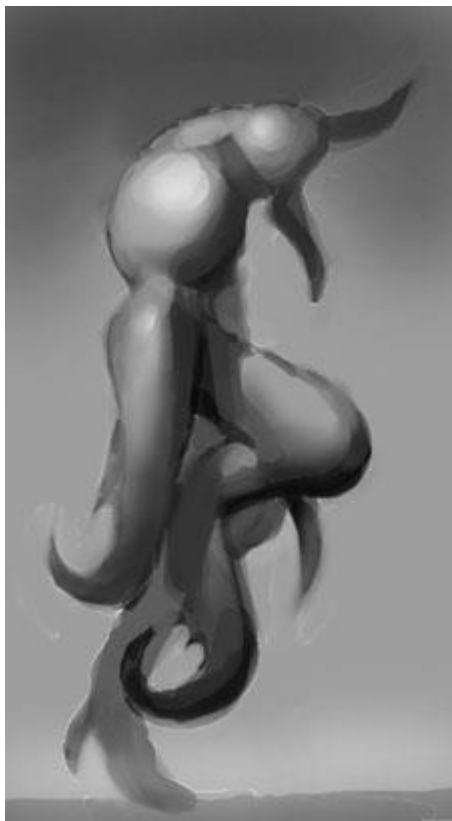
14. Point focaux

Une peinture est une hiérarchie de détails d'importance variable. Si vous êtes en train de peindre une pin-up, alors le personnage et sa silhouette sont ce qu'il y a de plus importants. Dans les bandes dessinées, un contour épais entoure souvent les silhouettes tandis que les détails sont dessinés avec des lignes plus fines par ordre d'importance. Quand vous peignez, ce doit être la même chose mais à coups de pinceau ! Vous devez utiliser des teintes, des saturations, des valeurs, des nettetés, des compositions et des niveaux différents pour guider l'œil de l'observateur vers le point focal de la peinture.

Si votre rendu est le même partout, le résultat sera plat. Vous pouvez guider l'œil vers les points importants, mais une fois l'œil en place il faut quelque chose pour l'y maintenir, comme par exemple des détails bien placés. La quantité de détails sur un endroit doit être proportionnelle au temps que l'œil y passera.



Voici un exemple que j'ai réalisé : (A) Formes importantes | (B) Texture | (C) Les deux à la fois



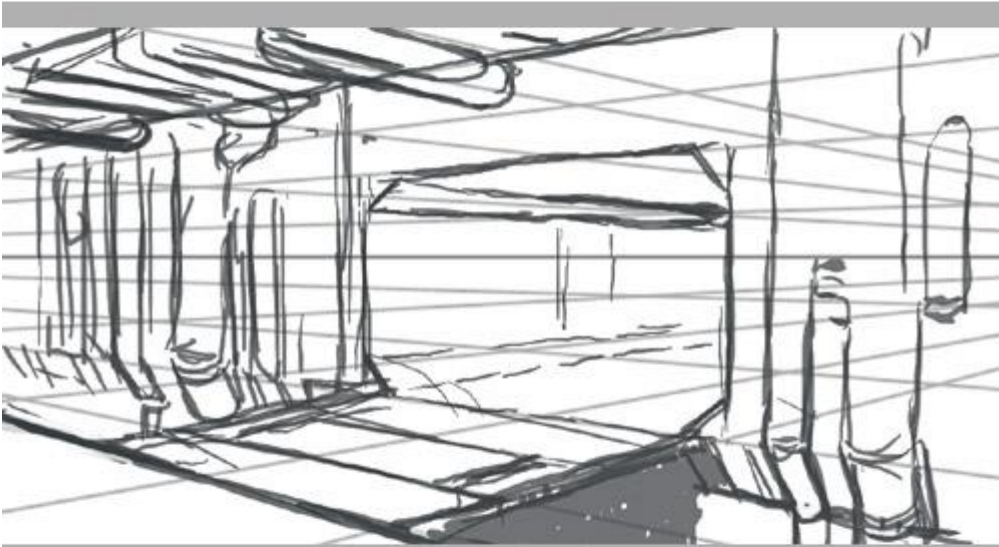
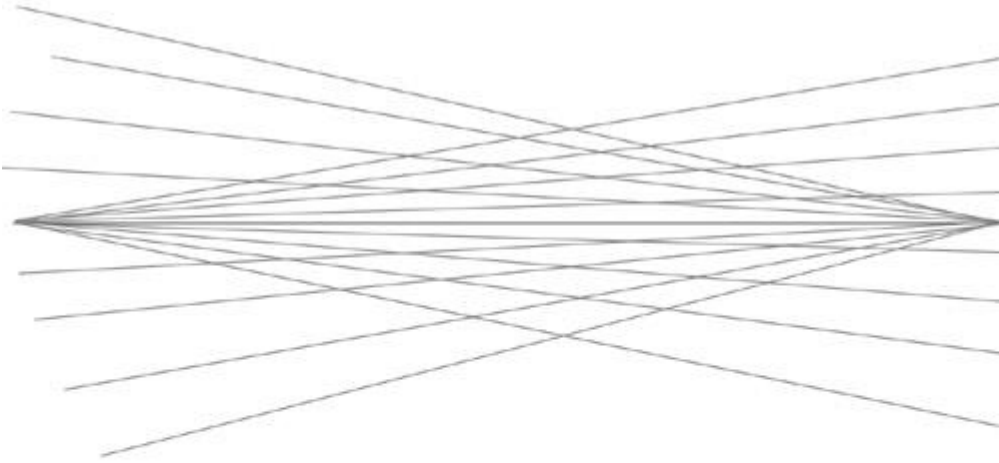


Il faut faire attention à ne pas se laisser emporter par le rendu des détails, surtout au début d'une peinture. Vous ne pouvez pas détailler de la même façon dans l'ombre et dans la lumière. Sur la seconde image (B), j'ai fait apparaître tous les détails pour montrer ce qui arrive quand on barbouille tout son dessin de détails sans penser aux formes importantes (A). (C) n'est pas parfait non plus mais c'est plus un problème de construction qu'autre chose. On peut difficilement faire mieux avec une vue de côté, et l'anatomie bizarre complique la lecture de l'image.

A est la forme sans détails, B représente les détails et C la combinaison de A et B. Soyez attention à ne pas exagérer B, on doit toujours pouvoir lire la forme !

15. Perspective et construction

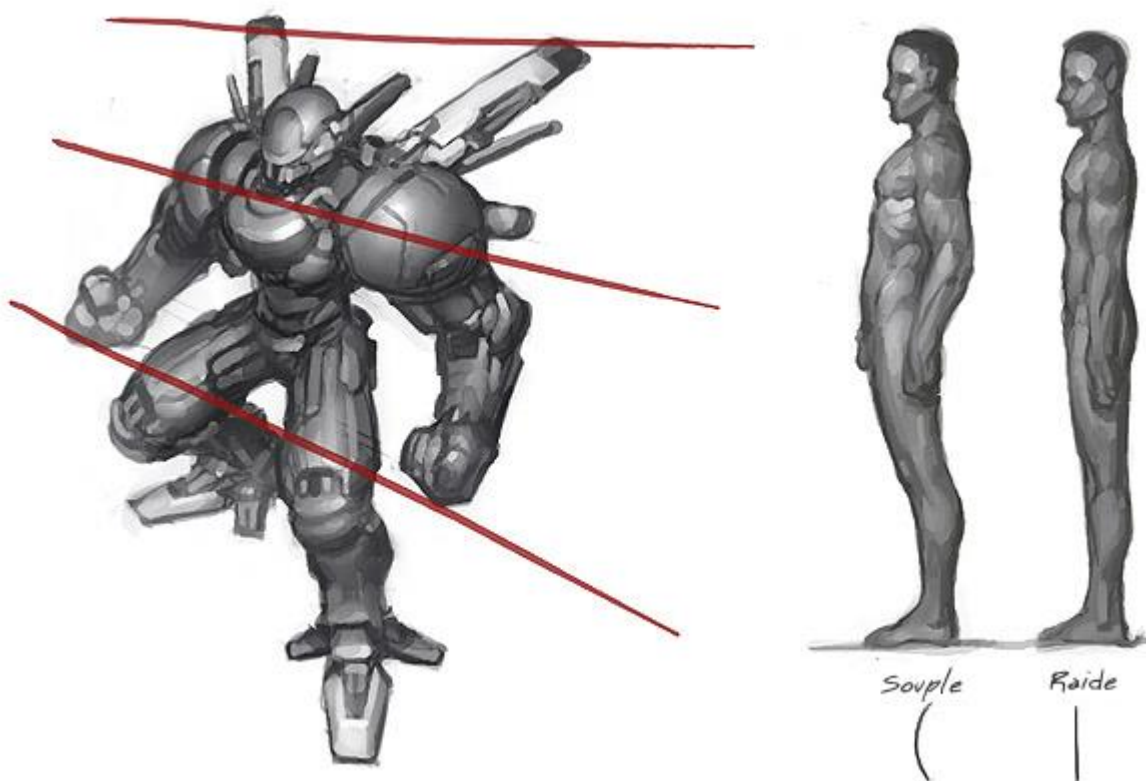
Je ne vais pas me lancer dans un long cours sur la perspective ici. Je vais me contenter de vous dire à quel point il est plus facile de dessiner après avoir dessiné quelques lignes de référence pour aligner vos personnages ou vos environnements.



Si vous renâclez à dessiner des environnements juste parce que la perspective est pénible à mettre en place, vous pouvez utiliser une simple perspective à deux points et deviner un peu les détails (*NdT : mais faudra pas pleurer si vos décors sont de travers*). Les choses n'ont pas besoin d'être parfaites pour avoir l'air parfaites. Vous aurez d'ailleurs des problèmes si vous n'êtes pas capable d'improviser et que vous avez

besoin de votre règle avant et après chaque coup de pinceau. Commencez par dessiner une ligne d'horizon puis faites partir de chaque extrémité de la ligne d'horizon des lignes dans des directions aléatoires, débarrassez vous ensuite des bords de l'image et vous êtes paré à dessiner.

Voici quelques lignes rapides et à mains levées qui m'aident à aligner les épaules et le reste. Quand on dessine des personnes, cela peut aider de les aligner selon le tracé de leur colonne vertébrale.



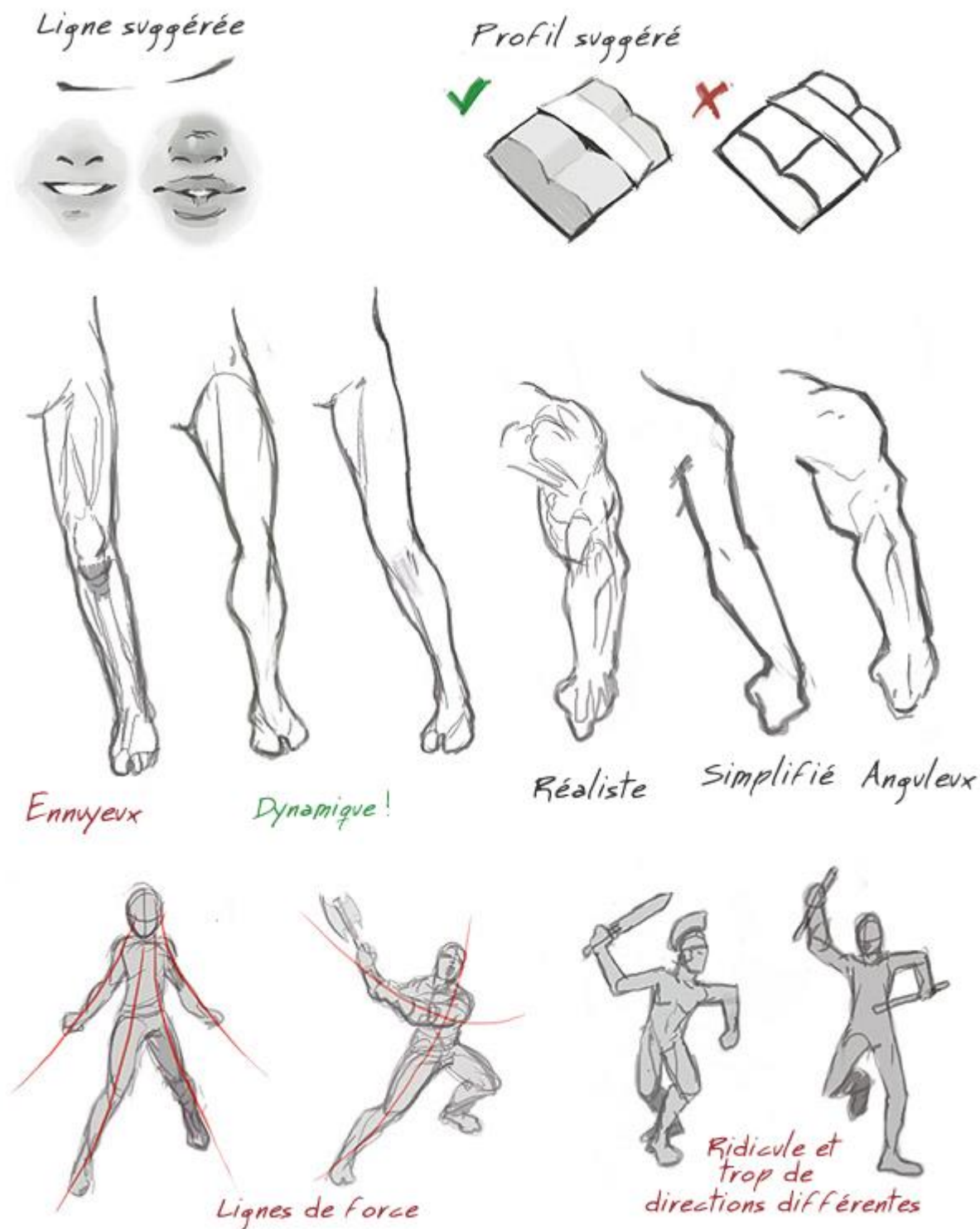
16. Tracés

- **Exagérer** — L'une des choses merveilleuses avec l'art c'est que l'on peut exagérer les choses, comme les hanches et les seins. Haha, non sérieusement. C'est une bonne chose que les courbes importantes soient plus prononcées.
- **Simplifier** — L'avantage du dessin sur la photographie est la simplification. Une photographie est remplie de détails qui détournent l'attention. En dessin, on peut se passer de tous les objets qui ne sont pas pertinents pour la scène représentée. Une erreur fréquente consiste à dessiner tous les muscles abdominaux à grand renfort de croisillons. Il vaut mieux conserver de simples lignes à cet endroit, particulièrement si vous allez colorier l'image puisque le contraste entre deux couleurs peut servir de ligne.
- **Harmoniser** — Un autre mot pour cela est « dynamique ». Contrairement au paragraphe ci-dessus, nous parlons cette fois de relations entre les détails et la façon dont les lignes se croisent et se dominent. Essayez d'utiliser quelques lignes dynamiques sur lesquelles vous pourrez aligner plusieurs éléments.
- **Styliser** — Quand on décide d'utiliser un style particulier, il est essentiel d'être cohérent. Vous pouvez par exemple transformer les courbes en traits droits ou vous pouvez décider de tout remplacer par des sinusoïdes ondulantes. Je préfère un mélange des deux en transformant les courbes en lignes droites à partir d'un certain seuil.
- **Épaisseur des lignes** — A quelques exceptions près, j'apprécie peu les dessins où l'épaisseur du trait est fixée. Je vais essayer d'introduire quelques conseils pour savoir quand et comment modifier l'épaisseur de son trait.
 - Les traits sont plus épais dans les ombres que dans la lumière.
 - Les traits sont plus épais près de l'observateur qu'à l'arrière plan.
 - Les contours de silhouettes sont plus épais que ceux des détails intérieurs.
 - Les objets plus clairs demandent des traits plus fins.
 - Les traits fins sont les amis des motifs détaillés.

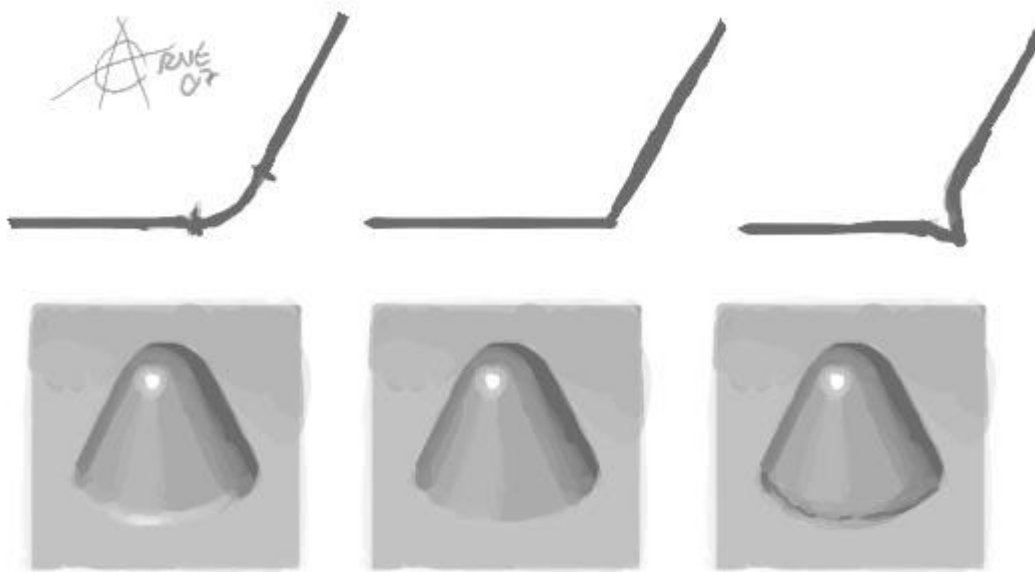
- Les traits épais sont les amis des formes simples.
- Les traits épais sont aussi les amis des aplats de couleur.
- Les traits fins aident à obtenir des volumes prononcés et des rendus réalistes.

Aussi, vous n'êtes pas toujours obligé de dessiner une ligne. Il suffit parfois d'en tracer l'extrémité pour que l'œil la complète naturellement. Les endroits où la peau s'agglutine comme la bouche, les fesses ou les seins (NdT : *malheur à vous si vous êtes arrivé sur cette page en recherchant un de ces mots clés*) sont de bons exemples de ce phénomène. Il est souvent de bon ton de rendre la ligne un peu plus épaisse là où se trouve un trou dans la ligne. Cela se voit bien là où les vêtements sont étirés, par exemple au dessus des espaces entre les muscles... ou sur les décolletés.

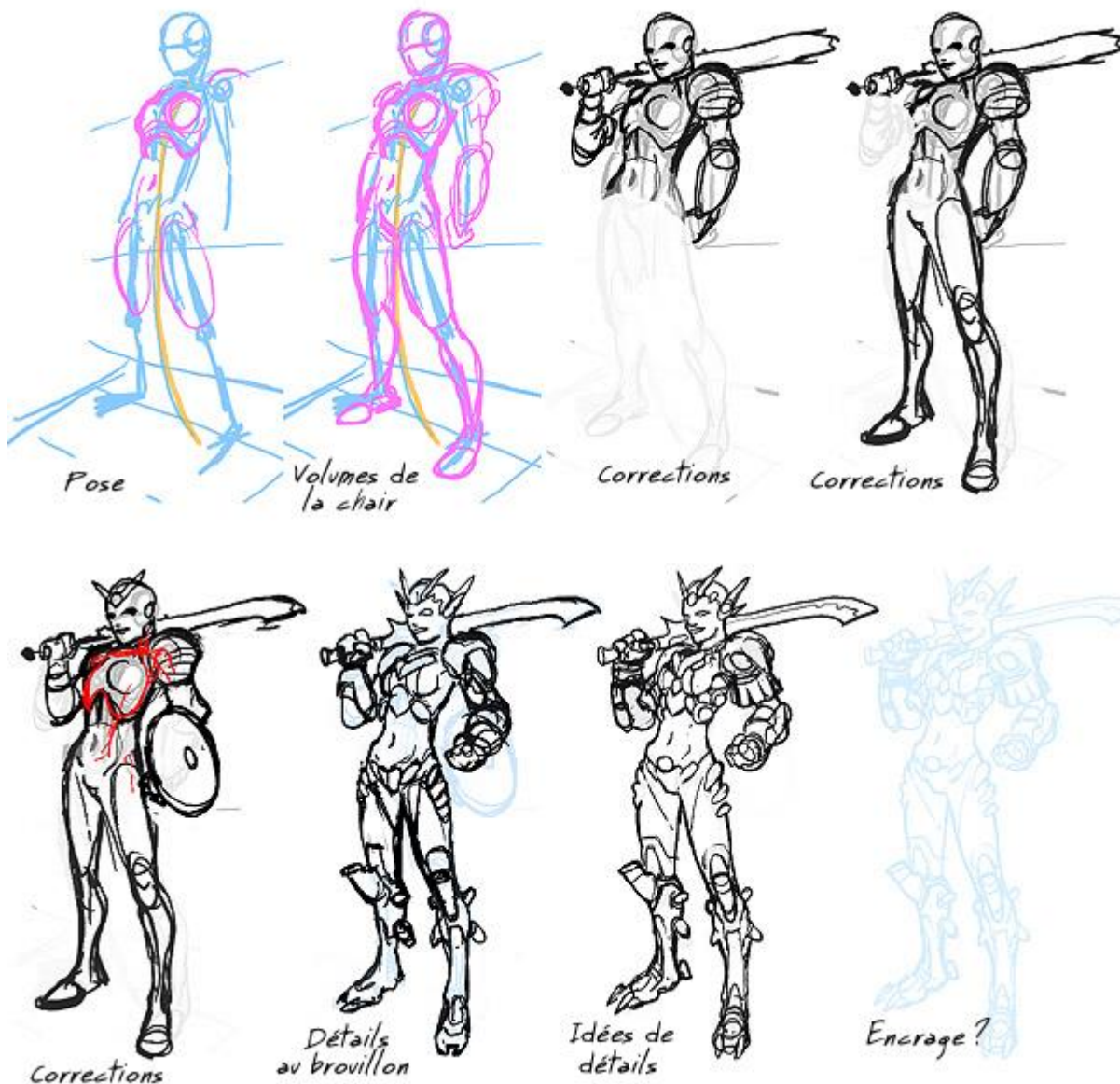
Pour finir, voici quelques illustrations. La première a été faite avec Photoshop (5.5) et une tablette Wacom, on ne peut pas appeler ça des traits de bonne qualité mais vous comprendrez sans doute l'idée. La deuxième illustration est un truc que j'ai encré il y a quelques années.



Quand vous peignez, les lignes sombres qui restent du tracé initial peuvent vous pénaliser. Vous pouvez séparer les formes autrement qu'avec des lignes noires. Vous pouvez par exemple utiliser des valeurs différents ou une lignes claire. Voici une illustration de quelque possibilités et des effets produits.



La vie d'un tracé comporte plusieurs niveaux de raffinements au fur et à mesure que les versions précédentes, plus brouillonnes, se font recouvrir par de nouveaux tracés. La version finale n'est pas représentée ici.



17. Etudes

J'ai commencé à faire des études il y a seulement quelques années et je le regrette. Vous ne tomberez jamais « sur la ligne juste » avec des devinettes et des barbouillages aléatoires. C'est avec des références que vous avancerez. Pour dessiner rapidement et naturellement vous devez parvenir à dessiner sans penser à la construction générale des choses (comme l'anatomie de base). Alors vous pourrez vous concentrer sur les concepts et les détails. Construisez vos dessins/peintures. N'avancez pas détail par détail avec votre crayon. Vérifiez systématiquement les proportions générales et ne vous laissez pas entrainer par les détails trop tôt. Placez des points de repères là où sont les caractéristiques importantes, elles vous serviront de balises sur lesquelles vous aligner.

Essayez de n'apprendre qu'une chose à la fois. Il est impossible de s'entraîner en même temps au jonglage et au 400 mètres haies. Si vous voulez apprendre à maîtriser un médium ou un outil, faites des études de choses simples comme des fruits. Si vous voulez apprendre à dessiner des visages, utilisez un médium que vous connaissez bien pour ne pas avoir à lutter avec lui en plus. Si plusieurs facteurs d'incompétence entrent en jeu, vous ne pourrez pas identifier ce qui ne va pas. La quantité est importante aussi, je ne vous recommande pas de faire des études d'anatomie avec des outils pour gravures sur bois.

Tracés

Les études au crayon n'ont pas besoin d'être autre chose que des petites esquisses sur du papier. J'y passe à chaque fois quelques heures et j'en mets 10 à 30 sur une feuille A4. Je ne fais que quelques études par mois, mais je remarque des progrès à chaque fois. Imaginez un peu quels seraient les résultats si vous le faisiez plusieurs fois par semaine pendant des années.

Quelques [vieilles planches](#) d'études faites d'après une bande dessinée (poses) et des photos (visages).

Peinture

Quand je fais des études digitales, j'utilise Photoshop et une image trouvée sur le net. Je duplique la fenêtre et j'en vide une, où je commence à placer les masses de couleur prédominantes. Après ça, j'augmente graduellement le niveau de détail, les valeurs et les couleurs avec le plus de précision possible. Quand de loin ou avec les yeux plissés on elle ressemble à l'original, c'est terminé. Je travaille toujours avec la plus grosse brosse possible permettant de rendre un détail donné. Je ne prélève pas les couleurs depuis l'original, mais je conserve les deux fenêtres de la même taille pour mieux voir si je rate le placement de quelque chose. Si vous voulez compliquer l'affaire, vous pouvez toujours essayer de dessiner dans une fenêtre de taille différente, avec une symétrie ou en peignant ce que vous observez autour de vous plutôt qu'une photo.

Études de statue : La [première](#) est faite sous OC (Open Canvas), et la [seconde](#) sous PS (Photoshop).

[Ici](#) j'ai essayé d'être le plus économique possible. La [deuxième](#) n'est pas une étude mais un exemple de nettoyage, ce qui n'est pas indispensable pour les études.

[Une de mes premières études](#)

18. Sujets d'études

Étudiez tout ! Vous devez construire dans votre tête une grande bibliothèque de formes pour pouvoir dessiner intuitivement. Cela prend une vie voire plus alors vous devriez commencer dès maintenant !

Anatomie humaine

L'une des choses les plus importantes à connaître. Même les monstres ont des traces d'anatomie humaine.

- Le corps entier. Utilisez des photos, des livres d'anatomie, des statues ou des gens.
- Le visage est ce que l'on regarde en premier. Si vous vous déplacez une ligne d'un poil, toute l'expression du visage peut changer. Étudiez des photos vos amis et vous même.
- Les mains aussi sont importantes (et difficiles) à maîtriser.
- Les pieds peuvent poser problème, non pas à cause de leur forme, mais parce qu'il faut enraciner le personnage sur le sol de telle sorte qu'il ne soit pas en train de tomber et que le sol n'ait pas l'air penché.
- Les vêtements de tous les jours. Il est important de savoir comment les vêtements se plient et à quoi ressemblent différents types de vêtements se comportent.

Gestes et styles

Vous devez avoir des activités variées et trouver des idées neuves. Apprendre des styles différents peut être une bonne idée.

- Faites du dessin d'observation en utilisant vos amis, les gens au café, dans le bus ou ailleurs. Quelle est la posture d'une personne qui ouvre une porte en cherchant ses clés tout en étant intimidé par l'artiste ?

- Marvel. Comment les artistes de Marvel représentent-ils l'anatomie humaine avec des lignes ? (*NdT : Ils la représentent mal ! Haha !*) Quels sont les détails importants et lesquels sont simplifiés ?
- Modesty Blaise, ou un autre style de bande dessinée réaliste. Faire des dégradés avec simplement du noir et du blanc n'est pas chose aisée.
- Le manga ou un style qui vous plaît. Encore une fois, comment l'artiste transforme-t-il le corps humain en une série de lignes et des tâches de couleurs ? Quelles parallèles pouvez vous établir entre les différents styles ?

Environnements

Placer un personnage dans un environnement le rend tout de suite plus vivant. C'est d'ailleurs quelque chose que je me dois d'apprendre.

- Les paysages avec des champs, des montagnes ou autre chose.
- Les forêts denses ou les jungles.
- Les paysages urbains ou industriels.
- Un environnement intérieur, comme une pièce avec des meubles. Oui c'est chiant. Pour vous dire la vérité je ne l'ai pas encore fait.

Allez chercher un livre sur les animaux

...et dessinez des animaux. Une bonne façon de concevoir un monstre est de fusionner plusieurs animaux en un.

- Tout ce qui vit. La nature perfectionne ses concepts depuis des millions d'années alors vous feriez bien de les étudier.
- Les chevaux, les chats, les chiens, les oiseaux. Ce sont les plus importants car les plus courants.

Machinerie

Vous devez aussi vous entraîner à dessiner des machineries. Cela peut servir pour dessiner des robots et droïdes de combats avides de vengeance et explodeurs de planètes.

- Différents modèles de voitures.
- Les machines permettant de creuser ou les machines d'assemblage.
- Les véhicules militaires.

Natures mortes classiques

C'est à dire n'importe quoi. Ce sont de bons sujets pour apprendre à peindre et dessiner, à cause de leurs formes simples. Vous n'aurez pas trop à lutter avec les formes et pourrez vous concentrer sur les textures.

- Fleurs, fruits, squelettes, sculptures, bouts de bois et de métaux rouillés.

19. Auto-critique

Analysez vos erreurs. Il est fréquent de s'aveugler en regardant trop longtemps la même image (ce que vous êtes bête obligé de faire pour travailler dessus). Essayez de renverser les images (symétrie horizontale), de les regarder à l'envers, à travers un miroir (j'utilise un CD), de vous reculer ou de zoomer/dézoomer. Ne restez pas assis et les yeux rivés sur le papier ou l'image zoomée. Sous Photoshop vous pouvez aussi utiliser « Créer une nouvelle fenêtre pour *Nom du document* » (ou « New View » en anglais).

Vous devez aussi accepter le fait que ce n'est pas parce que vous avez travaillé sur une image pendant longtemps qu'elle vaut quelque chose. Vous devez savoir sacrifier le temps passé sur quelque chose si le résultat est bancal. Même si vous êtes content des détails il se peut que vous deviez les retravailler. Parfois ce n'est pas le détail sur lequel vous vous concentrez qui cloche mais quelque chose qui y a trait, comme la valeur de son arrière plan ou la perspective d'un autre détail.

20. Ordre d'importance

Pour parler de façon très très générale, certains aspects d'une peinture sont plus importants que d'autres.

Construction

Qu'essayez vous de peindre ? Votre sujet et votre composition devraient être réussis au niveau fondamental, sinon même le meilleur rendu du monde ne sauvera pas votre image de la médiocrité. N'ayez aucune illusion du genre « ça va s'améliorer avec les étapes suivantes ». Si une pose est foirée, elle aura toujours l'air foirée une fois la peinture finie, même si Rembrandt fais le rendu à votre place.

Valeurs

Pour qu'une peinture soit réussie vous devez utiliser les valeurs pour sculpter les formes. Les valeurs peuvent très bien regrouper et dissocier les formes.

[Exemple](#) : la première version est évidemment ratée. Chaque forme a reçu toute la palette, de l'ombre à la lumière. La seconde est mieux mais chaque forme n'a qu'une valeur. La troisième a différentes valeurs sur différents formes. La construction n'est peut-être pas terrible par contre ; ce n'est pas une pin-up très intéressante.

[Exemple 2](#) : ici les valeurs et les couleurs sont utilisées pour séparer premier plan et arrière plan, mais je n'aime pas trop le résultat non plus sans doute à cause de sa construction (encore).

Couleur

On peut toujours s'en sortir avec des couleurs imparfaites. Si vous n'arrivez pas trouver les bonnes couleurs, ce sont en fait probablement les valeurs qui ont un problème. Petite note en passant, si vous avez réussi les étapes précédentes il est alors facile de faire des images qui auront l'air différentes à vos yeux simplement en réglant la balance des couleurs. Dans mon cas les couleurs de départ sont souvent les meilleures.

21. Critiques et erreurs fréquentes

Quand j'écris des critiques sur des forums, je me retrouve souvent à taper les mêmes remarques à chaque fois. Voici une liste des erreurs courantes que font les artistes débutants.

Problème : Utiliser ombre, teinte médiane et highlight sur chaque objet indépendamment de sa forme et de son angle.

Solution : Essayez de dézoomer, de retourner l'image ou votre tête. Ne faites pas le rendu de chaque détail un par un. Des petites formes éclairées équitablement ont tendance à aplatir l'image et à masquer les formes importantes.

Problème : Ajouter du noir pour les ombres et du blanc pour la lumière, puis flouter entre les couleurs pour la transition (par exemple avec l'outil « doigt » de Photoshop).

Solution : Cela rend les peintures grisâtres et ternes. J'ai croisé quelques personnes qui cherchaient des tutoriaux de « shading ». Il n'existe rien de tel. Il n'y a pas de méthode simple qui consiste à dire qu'il faut commencer avec une couleur foncée et aboutir à une couleur claire pour chaque détail. Vous devez apprendre comment vit la lumière et faire le rendu presque comme si vous étiez un logiciel de modélisation

3D. N'en déplaise à certains, c'est vrai au moins pour moi puisque j'obtiens souvent de meilleurs résultats en me débarrassant de mes habitudes et des mes préjugés pour ne suivre que les « règles de rendu ».

Problème : Faire le rendu et mettre des highlights sur des petits détails qui détournent l'attention. Utiliser l'outil Densité +/- (« Dodge ») de Photoshop.

Solution : Vous aller devoir sacrifier de nombreux détails qui auraient pu être magnifiques une fois rendus. Ce qui compte c'est la peinture dans son ensemble alors vous pouvez oublier les étincelles d'éclat lumineux sur la boucle de ceinture du petit type sans importance dans le coin de l'image. L'œil s'attarde sur les détails, les highlights, les grands contrastes et la saturation. Vous devez guider l'œil vers les parties importantes de votre peinture.

Problème : Poses plates et raides, personnages écartelés.

Solution : C'est un de mes propres problèmes. C'est si facile de dessiner les gens de face ou de profil avec les bras étendus... Ça aura toujours l'air ennuyeux et sans dynamique à moins que vous ne fassiez une icône ou que vous ayez une idée géniale de composition (*NDT : Pensez Jésus*). Il est souvent tentant de vouloir mettre en valeurs tous les détails d'un personnage dans un seul dessin mais c'est rarement compatible avec une pose dynamique. Cacher un bras derrière le corps est un sacrifice parfois nécessaire pour obtenir une bonne pose. De plus, en cachant un détail, vous laissez l'imagination de l'observateur décider à quoi il ressemble, et ça peut être une bonne chose ! Le résultat est parfois meilleur dans l'esprit de l'observateur qu'il ne l'aurait été sur papier. En bref, apprenez à gérer des poses dynamiques et vue de travers à partir de la réalité, de photographies ou de bandes dessinées. Vous n'irez pas bien loin avec des personnages écartelés.

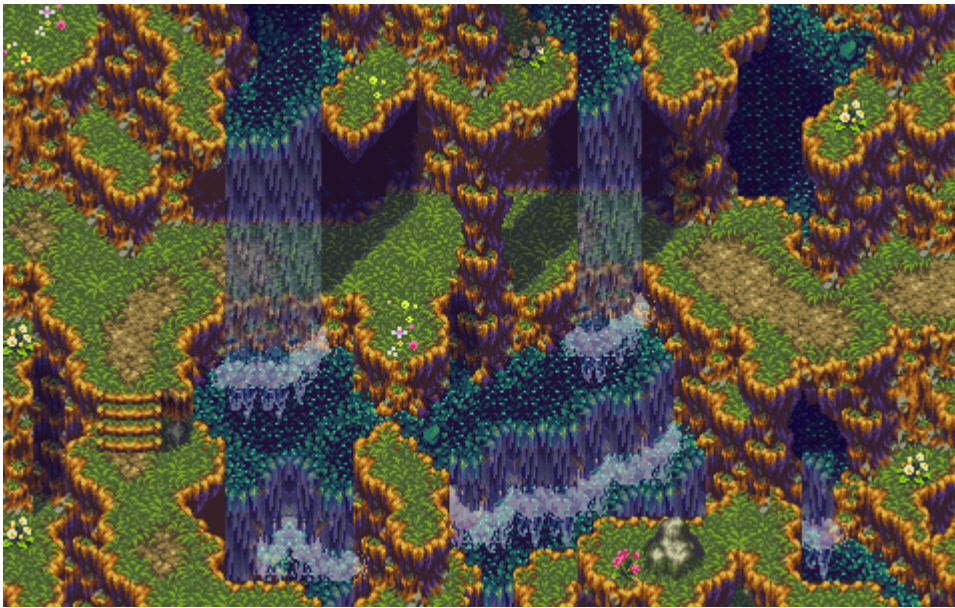
Problème : Coups de pinceaux inutiles ou tentatives de coups de pinceaux artistico-« Oh regarde comme c'est spontané ! »

Solution : Chaque coup de pinceau doit contribuer au résultat final. Certains espèrent s'en tirer avec des coups de pinceaux aléatoires et désorganisés. Les traits aléatoires créent des détails aléatoires, et les détails aléatoires détournent des vrais détails de l'image. Je sais bien que certains artistes donnent l'impression de peindre une scène parfaite avec quelques traits bien sentis et légers, mais gardez à l'esprit qu'il passent beaucoup de temps à styliser et à nettoyer pour atteindre ce résultat. Revisitez toute votre peinture pour supprimer les traits inutiles qui ne contribuent pas à l'ensemble. Soyez sûr que ce que vous représentez attire l'attention plus que des traits pseudo-artistiques. Les formes et les volumes sont toujours plus importants et beaucoup de jolis traits devront être sacrifiés pour que le résultat soit là.

Méthodes et conseils pour le mapping

Par [Dyo](#) | le 28 décembre 2008 18:44:22 | Catégories : [Cours](#), [Graphisme](#), [Level Design](#), [Pixel art](#), [Tiles](#)

Le mapping est une activité que je trouve extrêmement divertissante et ludique. Elle succède souvent à la conception des graphismes ; c'est alors l'occasion d'exploiter le fruit de longues heures de travail. Bien sûr, mapper correctement c'est, comme pour toutes les disciplines de la création d'un jeu vidéo, *un travail de patience* (rien ne s'obtient en claquant des doigts). Mapper à partir de graphismes somptueux ne vous garantit pas un succès assuré... Et un mauvais mapping peut lourdement pénaliser des graphismes de qualité. Mais, et j'y reviendrai plus loin, le bon graphiste aura vraisemblablement un maximum de chances d'être en même temps un bon mappeur. La raison ? Non pas un talent inné, mais un œil entraîné à repérer l'anomalie.



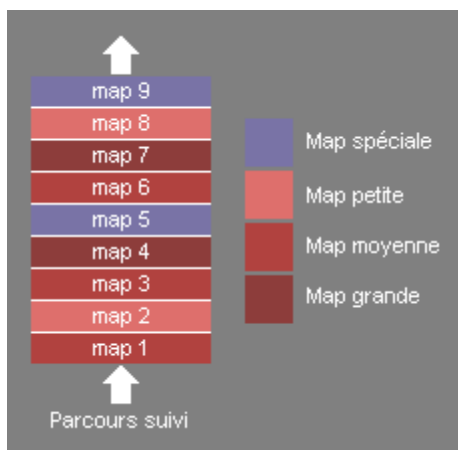
Je vous propose dans ce tutoriel des *principes assez généraux*, forcément imprécis car devant s'adapter à une multitude de cas. Son application directe concerne les graphismes 2D du type console 16 bit (ce sera le support des illustrations proposées). Toutefois la plupart des idées devraient aider quelque soit le support de création.

L'équilibre, rien que l'équilibre.

Le mapping est un domaine de création qui, comme tant d'autres, repose essentiellement sur l'équilibre (cela me paraît si important que j'ai décidé d'en faire ma première partie ; si vous ne deviez lire qu'une seule chose de toute ce tutoriel, c'est bien cette partie là). Cette notion d'équilibre peut être abordée à quatre échelles différentes, un peu comme si l'on zoomait successivement quatre fois sur une map :

- Au niveau le plus large, une vue d'ensemble de toutes les maps
- Au niveau intermédiaire, une map dans sa totalité
- Au niveau rapproché, un équilibre appliqué zone par zone
- Au niveau le plus fin, une zone dans la zone nécessitant un micro-mapping

Commençons par le niveau le plus large :



Il faut penser le parcours du joueur vis à vis de l'ensemble des maps, et donc faire alterner de manière équilibrée les maps grandes, moyennes et petites (représentées par les différentes teintes de rouge) mais aussi les maps « ordinaires » — en rouge (partie de forêt, pièce de donjon, village...) et les maps « particulières » — en bleu (lieu spécifique comme un gouffre, une immense cascade, une fontaine magique...). Bien sûr, mon schéma est très simplificateur et une succession de petites maps (exemple : pour un labyrinthe) n'est pas un mal en soi ! Mais il faut bien avoir en tête que c'est une alternance réfléchie de la taille et du type de maps qui va éviter au joueur de se lasser. Pour résumer, sa motivation sera d'aller de lieu particulier en lieu particulier (on fait tout le donjon pour atteindre la salle du boss contenant moult trésors...) : il faut donc doser la proportion de ces lieux pour ne pas trop en mettre (conserver leur statut attractif) mais en mettre suffisamment (si le joueur doit traverser une quinzaine de maps ordinaires avant de tomber sur un coin sympa, soyez assurés que sa motivation pour avancer dans le jeu va en prendre un coup).

A tout ceci se superpose la nécessité, bien sûr, de faire varier les lieux visités (ville, forêt, grotte...) mais le principe vu ci-dessus peut concerner un seul type de lieu. Par exemple, pour un village découpé en plusieurs maps : alternance de grands et petits quartiers, lieux particuliers disséminés avec soin (statue du roi sur la place centrale, manoir hanté plutôt isolé, etc.). Et un autre facteur est à prendre en compte : l'état d'avancement dans le jeu. Les grandes maps ne doivent pas arriver dès le début du jeu (cela va décourager le joueur). Il faut réussir à faire une gradation de la taille des maps en commençant par quelque chose de très petit.

Mais passons maintenant à l'analyse de l'équilibre pour une seule map :



Je me sers d'un tileset personnel (image de gauche) dont l'aspect très basique (éléments tous égaux en taille et peu nombreux) me permet d'illustrer de manière vraiment concrète les idées proposées dans ce tutoriel. Quelque soit la complexité des graphismes utilisés, les principes fondateurs sont toujours applicables (comme le montre la capture d'écran de *Secret of Mana* que j'ai mise à côté). Ici je pense que le principe d'équilibre est respecté. Pourquoi ? Parce tout est réparti. Les arbres, qui ceinturent le village, ne sont pas groupés en une seule masse qui dominerait le reste. Les maisons sont disposées de manière à couvrir l'ensemble de la map. A cet ordonnancement régulier des bâtiments vient se greffer un « élément perturbateur » : une colonne isolée dans une clairière. Mais surtout, l'effet d'équilibre provient de l'opposition entre une grande maison d'un côté, et deux maisons plus petites de l'autre. Et la clé d'un bon mapping, c'est la répartition des masses.

Schématisons l'exemple :

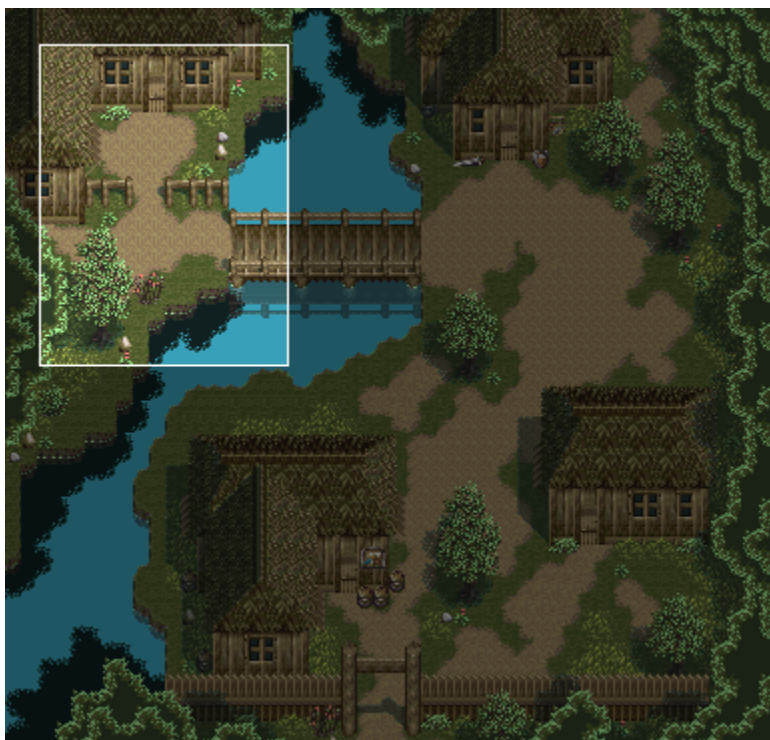


Pour équilibrer il faut répartir, mais pas répartir de manière régulière. Une régularité parfaite, une symétrie systématique peut rendre un jeu complètement fade et sans aucun intérêt pour l'exploration (voir plus loin pour la différence entre le mapping de la nature et le mapping de constructions humaines). Car le but est encore une fois le même : amener le joueur à explorer tout ce qu'on veut qu'il explore et éviter qu'il se lasse. Le secret consiste donc à mettre en opposition des masses. Si on me demandait de résumer en une seule image un « bon mapping », je ferais ce schéma :



Trois éléments disposés sur une surface de 3x3 positions d'éléments possibles, s'organisant selon le regroupement de deux d'entre eux d'un côté et un élément isolé de l'autre. Cette agencement donne une dynamique, un angle de lecture suggéré, et c'est bien là le souhait de tout mappeur : faire que le joueur suive l'un des parcours suggérés (j'y reviendrai plus loin). Histoire d'être bien clair, je ne vous conseille pas d'appliquer de manière primaire cette répartition là mais d'avoir en tête cette illustration : éléments regroupés et éléments isolés. Quand on doit mapper un village avec dix maisons, par exemple, recourir à ce genre de principe est indispensable. Une objection pourrait être opposée à tout ça : quand on explore, on ne voit quasiment jamais la map dans son ensemble. Le joueur a devant lui une vision cadrée, un fragment de la map. En fait c'est le déplacement qui va donner une vision globale ; si le déplacement est particulièrement lent, cette vision globale sera plus difficilement perceptible, mais dans tous les cas le joueur va ressentir l'équilibre général – ou le déséquilibre...

Mais puisque l'on parle de cadrage, passons au niveau rapproché de cette analyse de l'équilibre, avec une étude de zone (ce que le joueur aura toujours sous les yeux, puisqu'il visite la map « zone par zone »).



Voilà ce qui pourrait être qualifié de « zone ». Globalement, c'est un « lieu dans le lieu » (l'exemple parfait est l'intérieur d'une maison mais on peut penser à une clairière, un petit lac, un pont et ses environs...). Là encore, c'est le même discours que je répète et que vous devez mettre en application : penser « équilibre » avant tout. La répartition des arbres, de l'herbe haute ou courte, l'emplacement d'un tapis de fleurs, tout doit être pensé dans la modération et la nuance. Le joueur doit pouvoir faire une capture d'écran de n'importe quel endroit de la map et toujours se dire « c'est équilibré ».

Et ce n'est pas le sur-emploi d'un tile en particulier qui pose problème (l'exemple type : un champ de fleurs ou une forêt). Ce qui doit faire l'objet de toute votre attention ce sont les zones de cohabitation des tiles. Pour dire les choses autrement, et toujours du point de vue de l'équilibre, rien ne me dérange à la vue d'un lac faisant se répéter 40 fois un même tile d'eau ; en revanche la bande de sable parsemée de rochers qui fait jointure entre l'eau et la terre correspondra à une zone de « mapping mixte » qui requiert plus de travail et de soin.

Avant de passer à des choses moins théoriques, il nous reste à voir la question du micro-mapping.



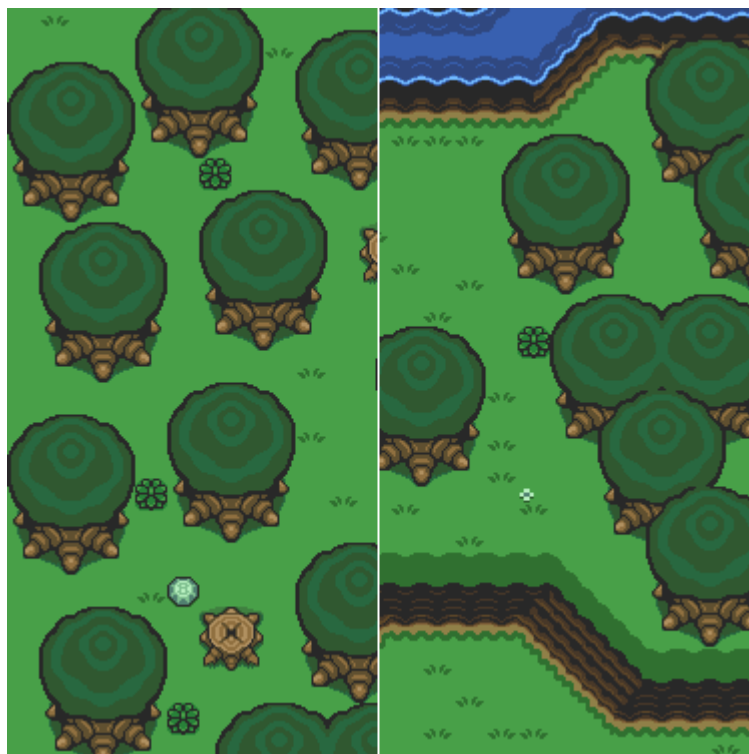
Le micro-mapping désigne les espaces extrêmement petits qui nécessitent de réfléchir à la position du moindre élément (si vous partagez ma vision d'un bon mappeur, vous me rétorquerez – à juste titre – que n'importe quel élément sur n'importe quelle map doit être placé après une réflexion, plus ou moins longue, ce à quoi j'adhère totalement ; toutefois dans le cas du micro-mapping il s'agit de mettre en évidence la difficulté à placer un lit dans une pièce qui fait cinq tiles de large par rapport au placement d'un tile de fleurs dans un vaste champ de fleurs). Encore une fois, les mêmes règles s'appliquent. On ne met pas trop d'un côté s'il n'y a pas quelque chose de l'autre côté pour faire contrepoids.

Finalement, vous aurez compris que l'objectif de cette partie (trop longue étant donnée la simplicité du message délivré – mais pour moi c'est le cœur de la discipline) était de vous faire voir que l'équilibre d'une map se pense sur une multitude d'échelles. Je vous recommande, bien sûr, de travailler du plus gros vers le plus petit. Pensez en premier lieu à l'organisation de toutes vos maps. Quand vous décidez de mapper l'une d'entre elles, brossez d'abord un rapide croquis de sa structure (pas besoin de papier et de crayon pour ça, servez-vous de tiles de couleur). Où seront les masses ? Où seront les éléments isolés ? Puis vous pouvez attaquer le mapping zone par zone, en veillant à porter un soin majeur aux espaces restreints.

Mapping de la nature, mapping de constructions humaines.

Mapping de la nature.

Attaquons quelque chose d'un peu plus concret, un peu plus pratique. Je disais plus haut qu'il faut fuir la symétrie ; en fait, tout dépend du type de lieu mappé. La nature déteste l'alignement et la régularité. Vous avez ici un lien direct avec la première partie : le regroupement de certains éléments et l'isolement de certains autres bénéficie non seulement à l'équilibre mais aussi à donner un aspect naturel, crédible, à la composition.

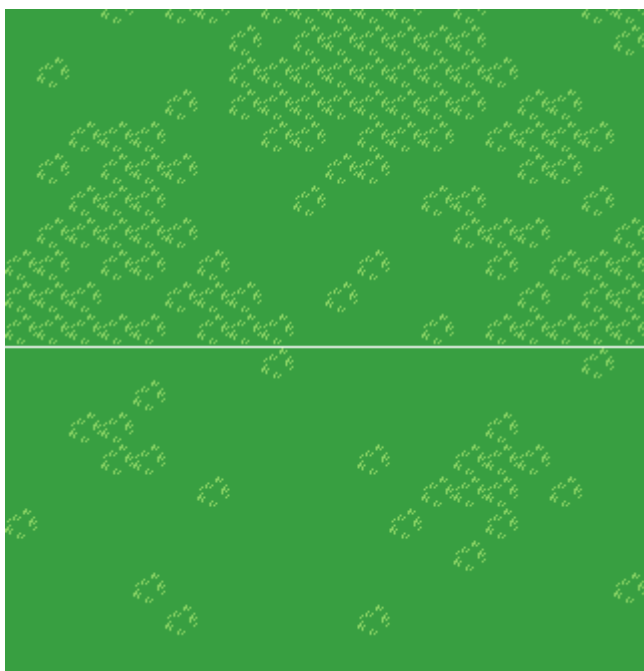


La partie de droite fait beaucoup plus naturelle que la partie de gauche. Bien sûr, les deux maps étant tirées du même jeu, on sait que la map de gauche correspond à un choix particulier des concepteurs, mais ça représente quand même ce que l'on trouve fréquemment dans des productions de type RPG amateurs pour lesquelles la disposition des éléments laisse trop d'équidistances. On retrouve ce problème spécialement pour les éléments considérés comme mineurs (cailloux, buissons, fleurs...), posés ici et là de manière complètement aléatoire, dans l'espoir inconscient que le résultat fera « naturel » grâce au « hasard » du placement. Or, et c'est là un joli paradoxe, pour faire du naturel il faut de la réflexion. Et le pseudo-hasard n'existe bien évidemment pas (on est enclin « par instinct » à mapper avec trop de régularité). Retenez donc bien ce conseil : l'équidistance, c'est le mal.

Un deuxième aspect concerne la variété des éléments utilisés. Je disais plus haut que la répétition d'un même tile sur de grandes surfaces n'était pas un problème par rapport à l'équilibre ; en revanche, du point de vue du naturel cela pose quelques soucis. Une forêt mérite au moins quelques types d'arbres différents ; un sol,

quelques textures ; une falaise, des creux et des saillies. Sans faire tomber ce tutoriel dans des questions relevant du graphisme, il faut quand même souligner l'intérêt d'un chipset à contenir des variations d'un même tile.

Prenons l'exemple du sol. Cet exemple est particulièrement pertinent ; en effet il est question ici de casser les régularités en introduisant de la variété, or le sol est, dans le mapping, créé en juxtaposant plusieurs tiles identiques sur de grandes surfaces. Même les jeux vidéos à la pointe de la technologie vont utiliser une même texture qui se répète, aucun cm² ne sera jamais unique. Et le public admet sans broncher cette « symbolisation ». Pour un RPG en 2D, l'herbe peut même être suggérée par un simple à-plat vert. Or, plus on abuse de cet symbolique admise, plus on s'éloigne du naturel. Il faut donc avoir au minimum deux tiles différents pour créer un « tramage ».



Cet exemple m'amène à parler d'un autre critère à prendre en compte quand on mappe du naturel : l'adoucissement, l'anti-aliasing. Oui, vous avez bien lu, ce maudit anti-aliasing (je sais que Fil vous a déjà pas mal suriné avec ça, mais j'en remet une couche, ça fait pas de mal). Et pourtant, jusqu'à présent on ne parlait pas de pixel art. Mais souvenez-vous : au début de ce tutoriel, j'avais évoqué les chances qu'un bon graphiste soit en même temps un bon mappeur. Pourquoi ? Parce que le même principe s'applique dans les deux domaines. L'idée, c'est d'adoucir la rupture trop brutale entre des surfaces trop différentes. Vous me direz : c'est encore un problème de graphisme... Mais regardez l'exemple de cette capture d'écran.



Les éléments correspondant à un « mapping d'anti-aliasing » sont les quelques arbres qui se détachent de la masse de la forêt, le petit bloc de falaise au sud, les petits cailloux éparpillés... Si ces éléments n'étaient pas là, la masse de la falaise serait en rupture avec la zone d'herbe en dessous, idem pour la forêt. Bien sûr, certains éléments doivent être mis en valeur et dans ce cas vous pouvez ne pas anti-aliaser, mais la plupart du temps vous devrez y recourir.

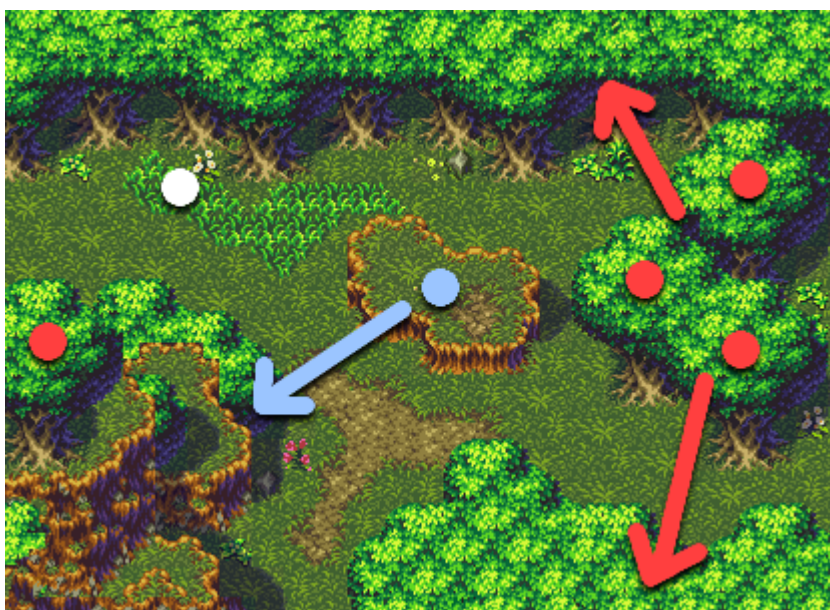
Sur cet exemple de sol :



Ce sont les tiles d'herbe situés entre les masses qui viennent adoucir la transition herbe dense — terre. Ces éléments sont vraiment indispensables, même si le chipset est parfait à la base. Le challenge consiste ici à faire cohabiter ce mapping d'anti-aliasing avec la variété des tiles utilisés (par exemple, l'arbre qui se détache de la forêt doit être associé habilement aux éléments périphériques — buissons, rochers, etc. — et ne pas surcharger la composition).

Dans le même esprit, une falaise ne sera jamais découpée au couteau, avec des angles droits, la bordure d'une forêt jamais parfaitement rectiligne. Donc quand on mappe « au naturel », il faut constamment avoir en tête ce principe d'adoucissement.

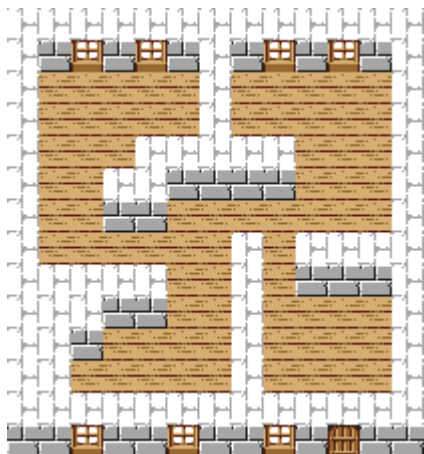
Un dernier exemple :



Les arbres isolés (ronds rouges) correspondent aux éléments d'adoucissement de la forêt. La petite élévation au centre (rond bleu) est extrêmement importante pour introduire la falaise plus élevée à gauche. On a donc ici une bonne map qui applique correctement le principe d'adoucissement, en revanche on peut contester l'aspect trop rectiligne de la forêt sur la partie supérieure. Une suggestion serait de placer un arbre isolé juste devant, sur le rond blanc.

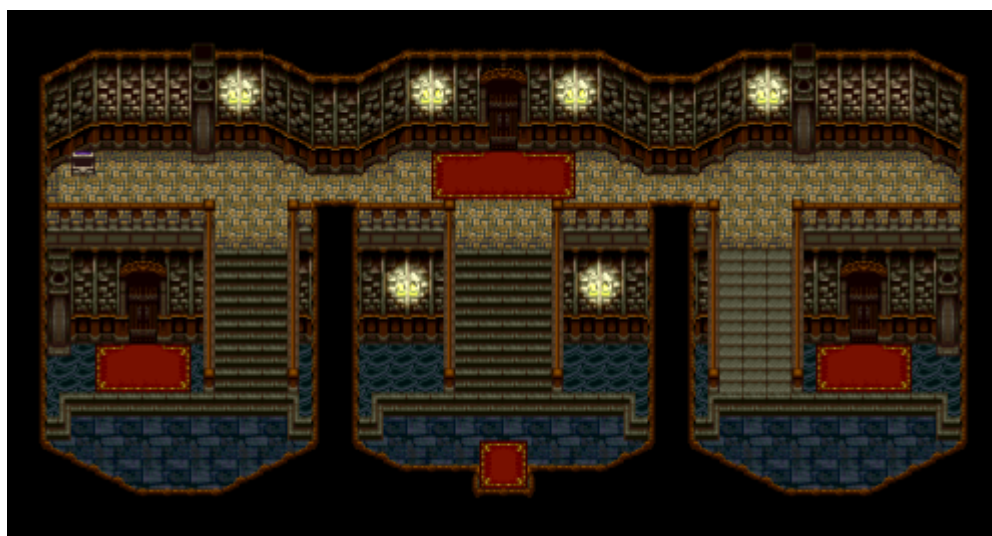
Mapping de constructions humaines.

L'avantage ici, c'est de pouvoir mapper « ce que l'on veut ». En revanche, les mots-clé qui s'imposent sont réalisme et logique. On a tendance à oublier que la loi de l'économie joue un rôle majeur. Dans une maison, on découpe les pièces de manière à ne pas perdre de place, on ne réserve pas une pièce immense juste pour un tapis et quelques caisses dans un coin. Il y bien quelques cas d'exception, tels que les palais, châteaux, sanctuaires, où les plans de bâtiments peuvent suivre d'autres logiques et gaspiller volontairement de l'espace (église en forme de croix, hall d'un palais dont la seule fonction est de montrer la grandeur, la richesse du lieu, etc.).



Cet exemple de map cumule les erreurs : perte de place en bas à gauche, murs non alignés, décrochement sans raison d'un mur commencé droit... Un vrai cauchemar d'architecte.

Toujours dans cette idée de réalisme architectural, il faut penser à la question de la portance. Sans aller trop loin dans le sujet, on peut quand même trouver bizarre de traverser une pièce gigantesque ne contenant aucun point d'appui tel qu'un prolongement de mur porteur ou quelques piliers... Alors bien sûr, beaucoup ne s'embarrassent pas de ce genre de questions, le plus souvent à tort (gagner en réalisme est toujours un plus). Par contre, aucun problème ici avec la symétrie. C'est même un double atout que de l'utiliser : d'un, pour son impact visuel, son esthétique ; de deux pour les points de repère que cela donne au joueur (s'il visite une pièce d'un côté, il sait qu'une pièce similaire est à visiter de l'autre).



La symétrie donne de la force à une composition. L'usage de la symétrie pour un mapping de nature, comme on a pu le voir dans de vieux jeux vidéos, peut apporter un côté irréel à un décor, lui conférer du charme... Mais toujours au détriment du réalisme.

Attention aussi aux principes de réalisme les plus fondamentaux (tellement fondamentaux que j'ai hésité à les rappeler) : pas de fenêtre sur un mur si derrière il y a une falaise, pas de foyer de cheminée si la maison ne possède pas de tuyau de cheminée sur le toit, etc. En bref, malgré l'aspect souvent simples des graphismes utilisés et la faible diversité des tiles utilisés, mettez-vous à la place de ceux qui auraient bâti les constructions que vous mappez.

Le mapping de constructions humaines pose généralement moins de problèmes que celui de la nature. Tout est justifiable, d'une manière ou d'une autre (en revanche, comment expliquer que cette falaise est naturellement carrée ou que ce lac a pris mystérieusement une forme parfaitement circulaire...).

Relief et volume.

Tout mapping nécessite de penser au relief (quelle évidence, mais une évidence bonne à rappeler !). Dans notre monde, n'importe quelle plaine, n'importe quel champ n'est jamais complètement plat (or cas exceptionnels comme une banquise). Mais il est souvent difficile de recréer un relief complexe lorsque les outils de création sont limités. Le danger est donc de tomber dans une alternance de maps totalement plates et de maps en relief.

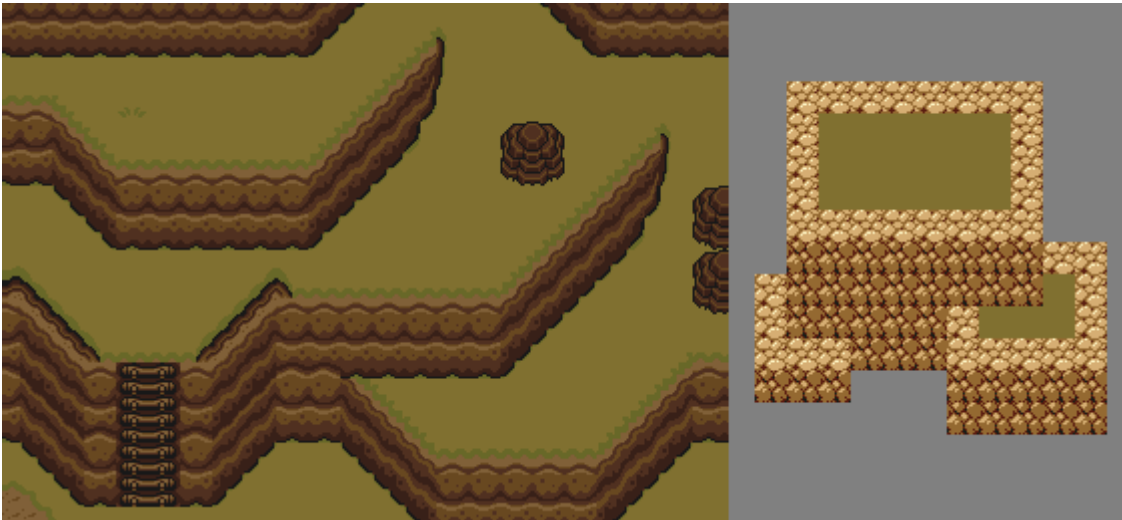
Comme d'habitude, il faut d'abord penser à l'ensemble des maps (alternance bien dosée de maps aux reliefs différents) puis s'attaquer à l'une des maps avec une vue d'abord très générale (encore une fois, c'est une question d'équilibre et je vous renvoie pour cela à la première partie). Pour le relief, la question du mapping par zone se pose moins (idem pour le micro-mapping).



Cette map est vraiment très bien réalisée. Elle est pensée à la fois en « pleins » et en « creux », et les escaliers donnent beaucoup de profondeur. Mais là, on reste encore dans le cas d'un relief assez marqué. La représentation d'une petite dénivellation est possible mais souvent trop « crue » pour suggérer une pente. On est donc obligé de passer par une construction en terrasses.

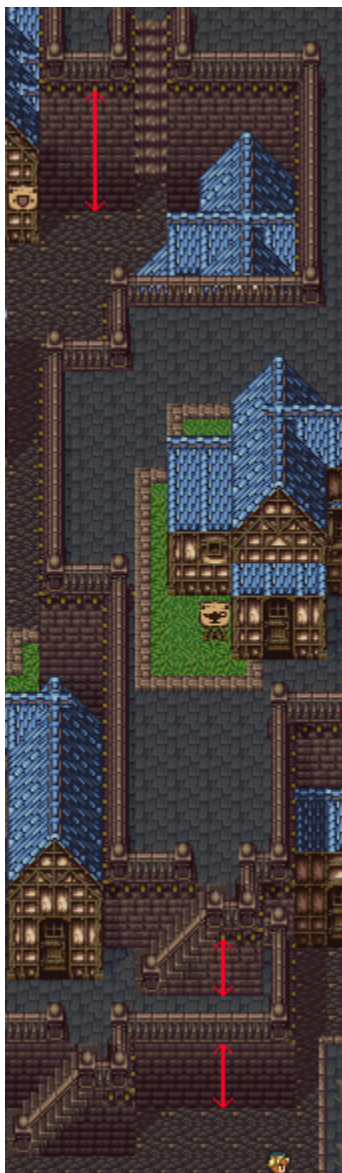


N'hésitez pas à recourir à cette technique pour échapper au « plat total » plutôt ennuyant lors de l'exploration. Une autre technique à mettre en œuvre consiste à associer des reliefs de hauteurs différentes pour éviter l'effet « mur » (pour les falaises le plus souvent).



Dans le même ordre d'idée, vous pouvez insérer de petits plateaux dans vos montagnes, plateaux que le joueur ne pourra pas forcément explorer mais qui contribueront à l'aspect esthétique et à l'équilibre général.

Et j'en viens à parler du volume (d'autres points importants touchant au relief seront abordés dans la partie suivante, car ils font intervenir la question du gameplay). Ce problème du volume ne se pose que pour les perspectives particulières ; l'essentiel est de ne jamais oublier de penser en 3D même pour des graphismes en 2D. Ainsi il faut faire attention aux décalages de hauteur qu'on observe parfois sur des reliefs complexes

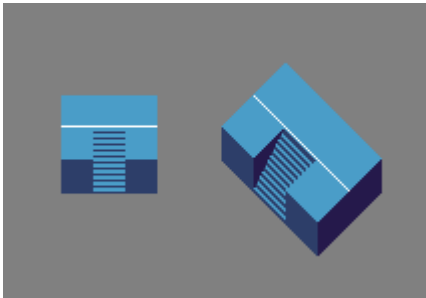


Sur cette capture d'écran de Final Fantasy VI, la hauteur du mur en haut doit correspondre à la somme des deux hauteurs de mur (deux escaliers) en bas (tout est correct, on a une hauteur de 4 tiles reportée partout). Ici le volume est relativement simple (avec un mur). Mais la difficulté va apparaître dès qu'on essaye de complexifier le volume en faisant s'empiler des niveaux de hauteurs différentes.



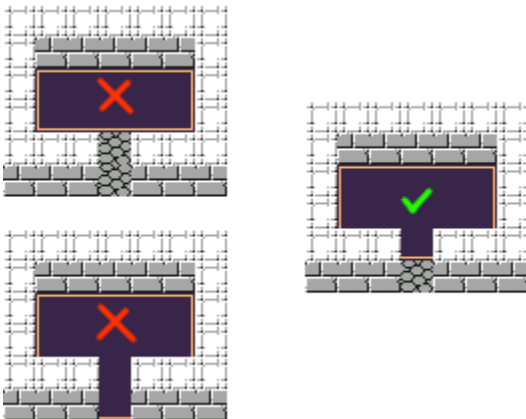
La complexité du relief sur cette map est assez hallucinante. Obtenir ce genre de résultat n'est pourtant pas très difficile du moment que l'on est méthodique, que l'on pense bien au volume, que l'on compte toutes les hauteurs soigneusement (et que l'on possède les graphismes adaptés, mais c'est un autre sujet).

Voyons maintenant quelques points souvent problématiques, toujours avec cette perspective en 2D, et tout d'abord l'éternel effet « les plans inclinés ont l'air de face ».



L'exemple de gauche montre un escalier créé en juxtaposant deux tiles l'un au dessus de l'autre. Or, si on se représente correctement le volume (en 3D, exemple de droite), on se rend compte que cet escalier est en fait extrêmement raide ! Il convient donc de rajouter un carreau d'escalier supplémentaire (au dessus ou au dessous, au choix).

Toujours dans cette idée de bien penser « en volume » :



Ce petit exemple du placement d'un tapis parle de lui même. Pour que le tapis occupe toute la surface du sol, il doit être positionné en tenant compte de la largeur des murs (ici, un tile). Bien sûr, l'exemple en haut à gauche n'est pas « incorrect » au sens strict, mais il apparaît illogique que le tapis soit placé comme ça dans une pièce et surtout la juxtaposition « bas du tapis / sommet du mur » est extrêmement déplaisante d'un point de vue esthétique et elle prête à confusion (le joueur se demandera si c'est un choix délibéré ou un mauvais mapping, or un bon mapping ne doit soulever aucune question de ce genre).

Enfin, pensez à ces petites choses quand vous mappez :



La partie gauche correspond aux choses à ne pas faire, la partie de droite montre les erreurs corrigées.

- Un élément de petite taille (tel qu'un caillou, une plante) a tendance à casser le volume et la perspective. Mieux vaut placer ces éléments périphériques plutôt devant les structures principales ; toutefois ce n'est qu'une recommandation générale et il n'y a rien de dramatique à ne pas la suivre constamment.
- Deux sols de hauteurs différentes ne devraient jamais être exactement similaires ; cela crée une confusion, une perte de repères chez le joueur. Au contraire, il vaut mieux employer deux types de sol différents pour marquer l'élévation (rien n'interdit ensuite de varier les textures au sein même de ces sols, ce qui manque d'ailleurs cruellement à l'exemple).
- Un arbre isolé placé derrière une falaise, entièrement visible, casse le volume. Je conseille personnellement de placer le plus possible de marques du volume (par exemple, un arbre dont la moitié inférieure est cachée comme sur l'exemple de droite, mais ça pourrait être un rocher, le mur d'une maison...).

Mapping et gameplay.

Nous abordons dans la dernière partie de ce tutoriel la question du gameplay. Comme on l'a vu, le mapping doit paraître réaliste, il doit être agréable à l'œil et varié pour que le joueur ne se lasse pas. Mais les contraintes du gameplay sont nombreuses et vont déterminer pour beaucoup la structure d'une map.

La notion fondamentale ici, c'est le parcours. Il faut suggérer un ou plusieurs parcours au joueur pour éviter qu'il passe dix fois par le même endroit, qu'il se perde, qu'il se lasse. Suggérer un parcours, c'est aussi amener le joueur à rencontrer des personnages, trouver des coffres, combattre des ennemis là où vous le voulez et dans l'ordre que vous voulez.

Comment suggérer des parcours ? D'abord avec des techniques de contrainte et d'incitation.

L'une de ces techniques majeures, c'est la mise à profit du relief. Vous pouvez créer des couloirs de circulation en bloquant le joueur à différents endroits avec le relief ou des éléments bloquants (arbres, pierres), ou tout simplement en plaçant des signes visuels (non bloquants, donc) comme un chemin. Le joueur a toujours tendance à emprunter les passages les plus dégarnis en éléments, les moins denses (la simplicité attire, la complexité — ou densité — rebute). Ainsi pour des intérieurs de maison, je conseille de placer la majorité des éléments contre les murs, pour dégager de grands espaces de circulation plus attractifs (et plus agréables à parcourir du point de vue de la jouabilité).

Mais les techniques de motivation sont beaucoup plus intéressantes (et difficiles) à mettre en œuvre. Il s'agit de faire voir quelque chose au joueur, tel qu'un coffre, un objet étrange, un personnage, mais de telle façon que cette chose paraisse totalement inaccessible. Le joueur va chercher à arriver jusqu'à cette chose et pour cela il sera forcé de suivre le chemin que vous aurez prévu pour lui. Ce cas de figure est extrêmement utilisé dans les jeux professionnels de toutes sortes. Dès le début de la map, on nous aguiche avec une porte grandiose nécessitant une clé, on nous fait miroiter la sortie de la map... Ce qui motive le joueur. Il a alors

une raison de faire l'exploration (autre que le plaisir des yeux à regarder le décor ou le combat contre des monstres), et va même y prendre plaisir.

Rien de pire que de ne suggérer aucun parcours !

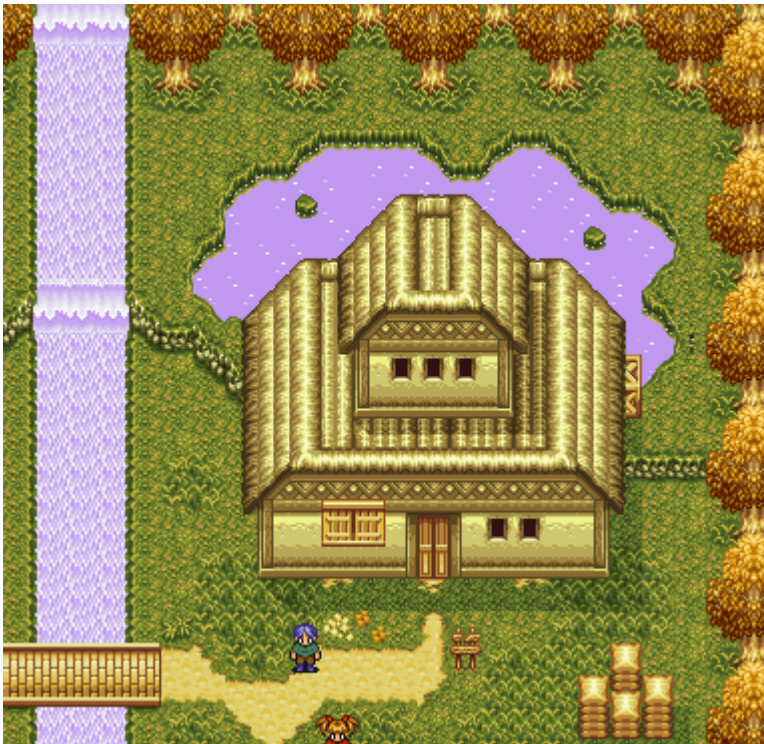
Le pire cas d'exemple possible est celui de la place de village bordée de maisons en rayons (place d'où pourrait démarrer le héros). Le joueur n'est donc pas vraiment perdu puisqu'il sait qu'il doit visiter une à une ces maisons ou seulement l'une d'entre elles pour progresser, mais sans aucune indication il va se sentir totalement découragé. Ce sentiment de devoir « tout visiter » alors que les chemins sont extrêmement nombreux constitue un danger majeur qu'il faut à tout prix éviter.



Une seule exception à cette règle : les lieux particuliers comme un bazar (un souk), un palais-labyrinthe... Où perdre le joueur en le saturant de parcours disponibles est alors un choix de game design. Mais même pour ces cas d'exceptions il faut prévoir un fil directeur, choisir un parcours principal à suivre et disséminer des éléments de repères (par où était l'entrée, par où sera la sortie) pour que le joueur puisse se raccrocher à quelque chose.

Nous allons maintenant voir quelques recommandations d'ordre pratique, toujours en relation avec le gameplay.

A commencer par les bords de maps (cette recommandation concerne plus spécifiquement le mapping de graphisme 2D mais s'applique aussi aux maps 3D). Le joueur ne doit jamais être bloqué par un « mur invisible » qui serait en fait le bord « virtuel » de la map. Une solution pour remédier à ce problème : ceinturer systématiquement les maps avec du relief et des éléments bloquants (une forêt, une rivière, etc.). Attention toutefois à ne pas tomber dans l'aspect « muraille ».



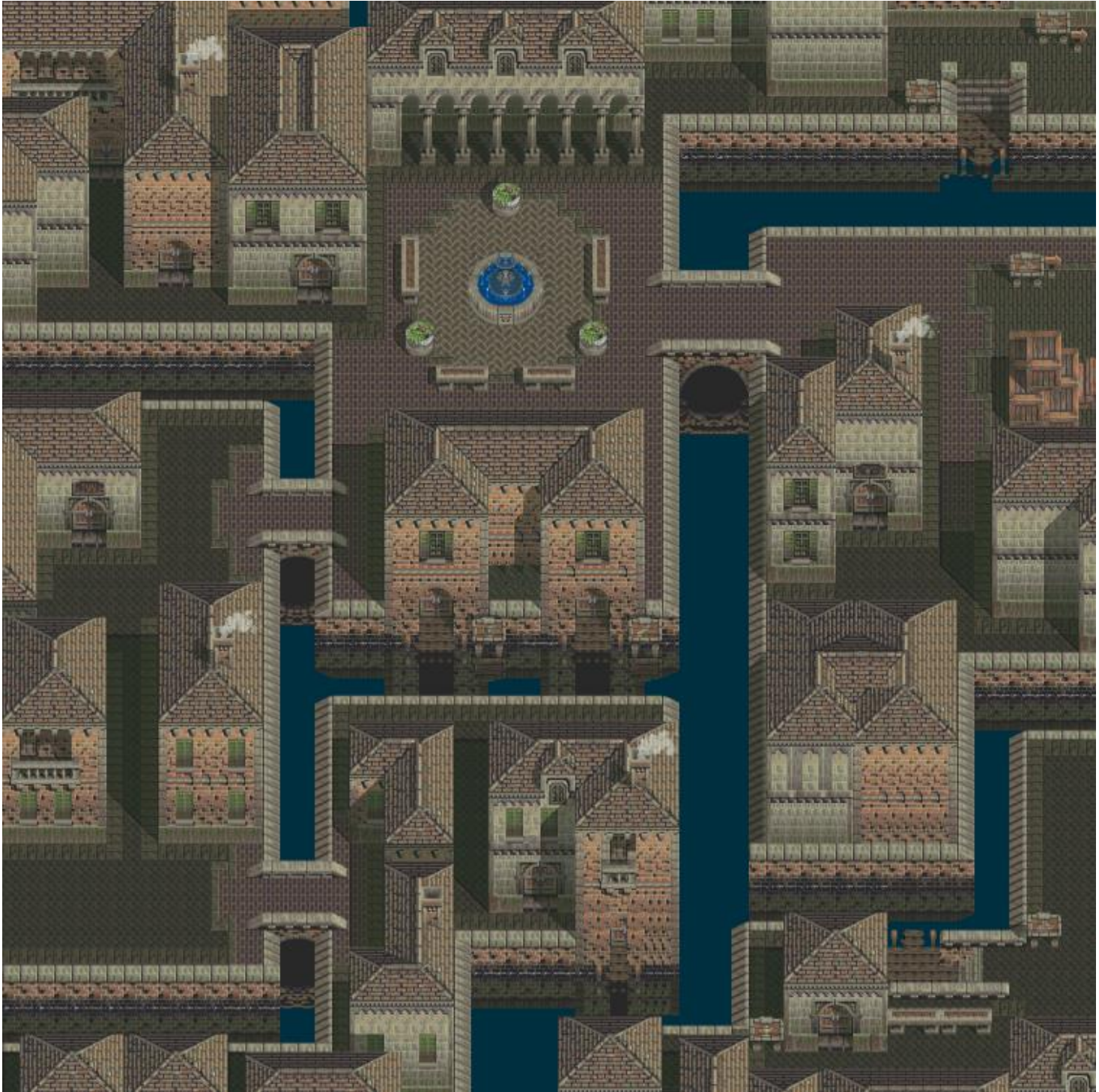
Cette map est correctement ceinturée avec des arbres, mais à cause de leur alignement parfait cette fonction devient trop évidente. Le joueur ne devrait jamais remarquer qu'on lui bloque le passage « consciemment ».

Si le joueur est bloqué par un mur invisible, non seulement ce sera pour lui une incohérence (visuellement, rien ne lui interdit l'accès) mais ce sera aussi le dévoilement de « l'envers du décor », du support logiciel du jeu. On voit parfois ce problème dans les jeux 3D : vous vous êtes sans doute déjà retrouvé dans cette situation où l'on fait brusquement face à un vide immense ou à un « mur » ; en cas d'erreur de programmation, on peut même « sortir de la map » et la contempler de l'extérieur. Il faut bien réaliser que dans ce cas là il y a rupture complète de l'illusion, l'immersion est anéantie.

De cette recommandation particulière je propose donc une recommandation plus générale : évitez à tout prix les erreurs de mapping qui vont faire voir au joueur l'envers du décor (autrement dit, le support logiciel du monde fictif que vous avez mis en place). Certaines erreurs, vous l'aurez compris, sont plus graves que d'autres... Un autre conseil que je vous donne : pensez à adapter la taille des maps en fonction de la vitesse de déplacement. Je crois qu'il n'y a pas besoin d'en dire beaucoup plus. Si vous voulez que le joueur passe du temps sur une map, faite la grande ou complexe, mais ne trichez pas en contrariant le déplacement du joueur.

Enfin, si vous avez à mapper des infrastructures très grandes (villes, bâtiments possédant de nombreuses pièces, etc.), ayez à l'esprit que l'exploration de chaque point d'entrée sera fastidieuse (intérieurs des maisons ou intérieurs des pièces). Vous devez recourir à des astuces pour limiter le nombre d'entrées à l'intérieur d'une map. J'en vois deux :

- Tout d'abord, interdire, tout simplement, l'accès à une entrée (barrière, porte fermée à clé, etc.) mais cela risque de créer un sentiment de frustration (et le joueur risque de ne pas comprendre ces interdictions, il va peut être penser qu'il y a quelque chose à faire même si ce n'est pas le cas).
- Une deuxième astuce, bien meilleure : cacher l'entrée, tout simplement. Dans le cas d'un mapping 2D, il suffit de ne pas montrer l'entrée (ne pas placer de porte sur la face visible d'une maison ; le joueur s'imaginera qu'elle est derrière). Les maps 3D posent plus de problème puisque le joueur peut étudier un bâtiment sous tous les angles. A vous d'improviser en plaçant des murs, en accolant les structures pour limiter le nombre d'entrées. La map ci-dessous utilise ces techniques extrêmement bien.



Conclusion

Voilà, ce tutoriel touche à sa fin. Je pense avoir abordé la plupart des notions clés ; il est possible que vous constatiez quelques oublis, si c'est le cas n'hésitez pas à m'en faire part. Certaines parties sont sans doute contestables, mais je vous demande de considérer tout ce que j'ai écrit comme une simple proposition de

méthodes et non pas l'affirmation de règles d'or fonctionnant quelque soit le contexte. Avant de clore ce tutoriel, j'ai une dernière recommandation à donner à l'apprenti mappeur : les jeux professionnels ne sont pas obligatoirement bien mappés (à titre d'exemple, je citerai le jeu *Terranigma* du développeur Quintet), et il ne faut pas avoir peur de sortir des figures classiques du mapping afin de proposer quelque chose d'innovant.

Vilain Game Designer ! Pas de biscuit pour toi ! II

Par [Pierre Guyot](#) | le 30 octobre 2009 23:15:53 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mars 2000).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Il y a à peine deux ans, j'ai écrit un article intitulé « Vilain Game Designer ! Pas de biscuit pour toi ! » qui traitait de ce que je considérais comme des fautes de conception – des choses qui m'irritaient dans certains jeux vidéo. Celui-ci a déclenché un important afflux d'e-mails, la plupart disaient « Oui ! Moi aussi ! », mais d'autres faisaient remarquer que j'avais été injuste par moments. Dans tous les cas, il s'agissait d'un thème populaire (et potentiellement sujet à débat) et j'ai donc commencé à établir une liste de choses qui m'irritaient dans les jeux. J'en ai cinq sous la main, et plus pour lesquelles je n'ai plus de place, c'est pourquoi le temps est venu pour un « Vilain Game Designer ! Pas de biscuit pour toi ! II ». Il s'agit pour certaines d'erreurs de programmation et non de design, mais vous avez saisi l'idée.



Les jeux qui tournent trop vite

Je sais que cela peut sembler bizarre – depuis quand le fait de tourner trop vite est un problème ? Mais j'aime jouer à de vieux jeux, et les vieux jeux, conçus pour de vieux supports, tournent souvent trop vite sur des machines plus récentes. Évidemment, ce n'est pas un inconvénient pour les jeux console, mais il s'agit d'un problème majeur pour les jeux PC.

D'un point de vue financier, bien sûr, il est peu probable que les éditeurs en aient quelque chose à faire. Pour eux, un jeu vieux de cinq ou dix ans, c'est de l'histoire ancienne. Ils en ont tiré tout le profit qu'ils pouvaient ; celui-ci ne tourne plus correctement ; les clients sont censés allonger la monnaie pour de nouveaux jeux, pas jouer à de vieux jeux. Mais du point de vue technique, il n'y a aucune excuse. C'est de la programmation bâclée. Si votre jeu tourne tellement vite que cela en devient injouable, cela signifie que vous ne gardez pas de trace d'à quelle vitesse le jeu devrait tourner et, pire, vous gaspillez l'une de vos plus précieuses ressources, le temps de calcul disponible. Il y a des tonnes de choses que vous pourriez faire avec du temps de calcul disponible supplémentaire, comme :

- Générer des étapes d'animation supplémentaire et fournir ainsi une animation plus fluide. Ceci est possible uniquement si votre système d'animation est capable de générer des intervalles supplémentaires, mais c'est de plus en plus souvent le cas aujourd'hui.
- Passer plus de temps sur l'intelligence artificielle. Aller plus loin dans les calculs de l'arborescence du jeu ; permettre à votre algorithme de pathfinding de regarder plus loin ; donner à vos monstres un plus gros cerveau. Une IA adaptative n'est pas triviale à mettre en place, mais si une partie quelconque de celle-ci, comme le pathfinding, implique une suite de calculs répétitifs, alors cette partie constitue un candidat idéal pour bien utiliser le temps de calcul disponible supplémentaire.
- Générer des polygones supplémentaires. Un bon nombre des jeux compensent la lenteur des machines sur lesquelles ils tournent en diminuant le nombre total de polygones qu'ils affichent, ce

qui a des résultats aussi laids que l'on peut l'imaginer. Pourquoi ne pas aller dans la direction opposée ? Concevoir un jeu qui serait absolument fantastique sur un hypothétique PC à 3 GHz, puis concevoir des processus internes visant à l'adapter aux supports actuels. A mesure que le hardware se développe, votre jeu ne cessera de devenir plus beau. Évidemment, il y a des problèmes de mémoire ou autres qui ne concernent pas le temps de calcul disponible, mais c'est une technique qui vaut la peine d'être étudiée.

Le premier jeu que j'ai programmé professionnellement devait tourner sur un système allant d'un 4.77 MHz 8086 à un 25 MHz 80386 – et parce qu'il s'agissait d'un jeu multijoueur, il devait tourner à la même vitesse sur l'ensemble des systèmes compris dans cette gamme. J'ai réussi à réaliser ceci en basant tout mon travail sur des timers. La boucle principale tournait aussi vite que le permettait la réactivité maximale du clavier et de la souris, mais les processus d'affichages étaient liés à des timers et ne se lançaient que lorsque le timer le permettait. Je n'ai pas essayé mais je subodore qu'il tournerait encore à la même vitesse sur mon Pentium II 400 MHz.

Pause et sauvegarde

Le problème de la sauvegarde constitue un sujet de débat parmi les game designers depuis le tout premier jour, mais je sais comment je me positionne vis-à-vis de ce problème : du côté du joueur. De nombreux designers n'aiment pas la possibilité de sauvegarder à loisir, parce que cela permet aux joueurs de recommencer chaque fois que de petites choses ne se passent pas comme prévu, ce qui signifie qu'ils avancent plus rapidement dans le jeu et perdent moins souvent. Pour faire court, ils ne subissent pas toute la souffrance et la déception que le designer avait prévues pour eux. Dommage. Recommencer tout un niveau parce que vous avez commis une erreur juste avant la fin est frustrant et ennuyeux. En tant que designer, est-ce vraiment votre but ?

Pour moi, l'argument principal est qu'il s'agit de la machine du joueur. Il n'est pas juste de le pénaliser parce qu'il doit aller aux toilettes ou répondre au téléphone. Les joueurs devraient pouvoir mettre en pause quand et où ils le désirent, sans perdre tout ce qu'ils ont accompli jusqu'à présent. Si mettre le jeu en pause affecte le gameplay de façon substantielle, comme dans *Tetris* par exemple, alors il suffit d'afficher un écran totalement noir lorsque le jeu est en pause. Évidemment, ceci ne s'applique pas aux jeux multijoueurs.

Deadlocks

Une deadlock, ou dans une terminologie française assez romantique, une “étréinte mortelle », se produit quand deux processus attendent chacun que l'autre ait fini quelque chose et ainsi, aucun des deux ne fait plus rien. Empêcher les deadlocks constitue un problème classique en informatique pour les systèmes multitâches : le processus A fournit la ressource X mais a tout de même besoin de la ressource Y ; le processus B fournit la ressource Y mais a besoin de la ressource X, si bien que chacun attend indéfiniment que l'autre ait fini pour fournir la ressource nécessaire.

En termes de game design, une deadlock survient lorsque vous avez besoin d'une ressource pour construire un mécanisme de production pour obtenir plus de cette même ressource. J'ai remarqué ce problème pour la première fois en jouant à *Settlers 3*. Je n'avais pas assez de pierre pour construire une hutte de tailleur de pierre, mais sans une telle hutte, je ne pouvais pas récupérer plus de pierre non plus. Ce n'était pas vraiment la faute du jeu, car celui-ci m'avait fourni assez de pierre pour commencer, mais je l'avais allouée entièrement à d'autres choses. Bien sûr, dans certains cas, on peut rechercher la deadlock comme une condition de défaite : dans un jeu de stratégie par exemple, si le joueur perd toutes ses unités de construction ainsi que le seul bâtiment qui permette de produire ces unités de construction, le système peut le détecter et lui signaler qu'il a perdu.



Plus votre modèle économique devient compliqué, plus il est probable que des deadlocks surviennent. *Settlers 3* possède un modèle économique inhabituellement complexe. Quand j'ai créé ce deadlock, ma communauté prospérait et je n'étais pas assailli par l'ennemi à ce moment du jeu, on ne pouvait donc pas vraiment dire que j'étais en train de perdre ; j'ai simplement commis une erreur économique. Heureusement, le jeu fournissait le moyen de contourner le problème : je pouvais démolir d'autres bâtiments afin de récupérer une partie des matériaux de construction. Finalement, j'en ai démolis suffisamment pour récupérer la pierre nécessaire à la construction d'une hutte de tailleur de pierre, et j'étais de retour dans le jeu. Une façon d'éviter les deadlocks consiste à fournir une source alternative, même si sa valeur est minimale. C'est le but de la case départ au Monopoly, qui permet au joueur de récupérer de l'argent à chaque tour de plateau. Même si vous ne possédez aucune propriété permettant de vous assurer une rente, vous êtes quand même sûr de récupérer cette somme. Les deadlocks ne constituent pas toujours des erreurs de designs, mais vous devez fournir un moyen soit de briser la boucle (passer par la case départ, démolir d'autres bâtiments, etc.), soit de la détecter et de mettre fin au jeu. Les jeux qui aboutissent à des deadlocks et y restent indéfiniment (et j'y ai joué) sont pénibles.

Bouger d'improbablement gros blocs de pierre

Me voici donc, en train de jouer à la démo de *Indiana Jones and the Infernal Machine*, et je n'arrive pas à sortir de la toute première pièce du jeu. Je n'ai pas beaucoup joué à ce genre de jeu, parce qu'ils demandent généralement trop de réflexes, mais je pense que je peux faire confiance à LucasArts pour ne pas exiger une trop grande coordination œil-main. J'essaye absolument tout ce à quoi je peux penser, sans succès. Je ne peux même pas sortir de la première pièce du jeu... Décidément, quel nul ! Finalement, je craque et je vais chercher une solution.



Oh, voici la réponse ! Il y a un bloc de pierre, d'environ 1m50 de haut, que je suis censé déplacer quelque part... HEIN ? Voyons voir, la masse volumique de la roche est de l'ordre de 2600 kilogrammes par mètre cube, le bloc fait environ 2,25 mètres cubes, ce qui nous fait une masse de l'ordre de 6 tonnes. C'est à peu près autant que la masse d'un des monolithes de Stonehenge. Et Indy est supposé le déplacer juste en tirant dessus, c'est bien ça ? Décidément, je me demande pourquoi je n'y ai pas pensé.

Ok, je sais que la gestion de la physique dans les jeux est ridicule la plupart du temps – la première fois que j'ai vu Sonic le hérisson changer de direction en plein milieu de sa chute, j'ai entendu un étrange bruit : c'était Isaac Newton qui se retournait dans tombe. Peut être que ce concept de tirer-des-blocs-immenses-aux-alentours est une convention tellement bien acceptée que je devrais peut-être juste essayer de m'y faire. Néanmoins, j'ai toujours pensé que les jeux Indiana Jones étaient conçus pour ressembler un tant soit peu au

monde réel. Les puzzle games impliquent une forme de pensée latérale, mais la pensée latérale elle-même est tout de même soumise à certaines limitations qu'il semble raisonnable de prendre en compte – sans quoi vous n'arrêteriez jamais de vous prendre la tête. Personnellement, j'ai tendance à écarter immédiatement les impossibilités physiques évidentes. Je suppose que cela dépend dans une certaine mesure du genre, mais dans ce cas particulier, cela m'a beaucoup irrité. Indiana Jones est un être humain, pas un Superman.

Énormes poitrines et autres puérités

Il y a quelques années, EA avait demandé à plusieurs artistes de concevoir des animations de pom-pom-girls pour une édition cartouche de Madden NFL Football. Ils nous ont rendu une misérable animation à quatre temps de femmes avec des poitrines complètement improbables en train de sauter de haut en bas. Nous les avons virés et avons engagé quelqu'un autre pour ce travail. Nous ne les avons pas renvoyés uniquement parce qu'ils avaient fait un travail de piètre qualité ; nous les avons renvoyés parce qu'il s'agissait d'abrutis plus occupés à satisfaire leur propre voyeurisme ricanant qu'à livrer à l'entreprise ce dont elle avait besoin, qui nous ont fait perdre notre temps et notre argent. J'ai un message pour les balourds dépourvus de talent qui insistent pour mettre de l'humour puéril et des femmes aux poitrines délirantes dans les jeux vidéo. Comment tourner cela de façon diplomatique ? Ah, ça y est, je l'ai :

Grandissez.

Vous êtes une honte et une disgrâce. C'est à cause de l'exemple que vous donnez que le reste d'entre nous devons expliquer à nos proches, nos amis et, parce que c'est important, aux députés, que nous ne sommes pas tous des rustres sans goût et avides d'argents, que vous seuls l'êtes. J'ai passé beaucoup de temps à essayer de convaincre des non-joueurs (mais des gens qui votent) que vous, les habitués dégoulinant de bave des peep-shows, n'étiez en fait qu'une minorité dont les produits imbéciles et le manque le plus total de discernement salissaient injustement le reste de l'industrie. Bien sûr, si vous vous inquiétiez des effets que votre déballage de lubricité adolescente a sur la réputation du milieu ou sur les conséquences possibles de celui-ci sur les débats sur la régulation des jeux vidéo, vous ne le feriez pas, mais non, vous être trop engoncés dans vos fantasmes infantiles pour faire attention à autre chose que le bout de vos phallus obscènes.

Ce serait encore tolérable si vous étiez un tant soit peu bons dans ce domaine, mais vos produits ne constituent même pas des produits érotiques décents. Ils sont stupides, blessants, ils ne sont ni sexy, ni drôles. J'aime à penser qu'à mesure que le marché mûrit, vous serez poussés dehors, ou au moins considérés comme hors de propos. Finalement, les rois du porno ont arrêté de prétendre qu'ils étaient de vrais cinéastes et sont partis créer leur propre industrie, avec vos propres standards de qualité (très bas). Pourquoi ne faites-vous pas la même chose ? Vous pourriez alors avoir vos propres réunions d'affaires, vos propres récompenses et traîner avec les gens de votre espèce : les mains moites, le souffle rauque, bas de plafond et des gonades à la place du cerveau. Grandissez ou sortez.

[Lire la suite →](#)

Simplification

Par [Pierre Guyot](#) | le 30 octobre 2009 23:23:23 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juillet 1999).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

L'un de mes jeux favoris — du moins quand il s'agit de discuter les principes du game design — est le Monopoly. C'est un jeu où interviennent à la fois l'expérience et la chance ; il est facile de comprendre comment les ressources y circulent ; et il a l'avantage d'être connu par presque tout le monde dans le monde occidental. Quand je me sers du Monopoly comme exemple, je suis presque sûr que les personnes qui me lisent sauront de quoi je parle. Les noms des rues diffèrent selon le pays où vous y jouez, mais les principes restent les mêmes.

Le Monopoly est une version simplifiée du marché immobilier — simplifiée, parce que toutes les subtilités du vrai marché immobilier n'y ont pas été incluses. Il n'y a pas d'inspections, d'assurances ou d'impôts. Monopoly ne garde que les aspects amusants du commerce immobilier, et met de côté ceux qui ne le sont pas.

De même que la symétrie et l'équilibre, la simplification est un principe fondamental du Game Design. La simplification constitue la différence essentielle entre un jeu et une simulation. Le but d'une simulation sur ordinateur est de modéliser un vrai système de façon aussi fine que possible pour étudier son comportement dans différentes conditions. Le but d'un jeu vidéo est de divertir et toute autre considération y est secondaire. La réalité est complexe et difficile à comprendre. Les jeux sont supposés être simples et funs. Pour obtenir l'un à partir de l'autre, vous devez simplifier.

Les joueurs exigent un degré variable de réalisme, qui dépend beaucoup du genre de jeux auxquels ils jouent. Dans la plupart des shoot-em-ups, par exemple, les joueurs se moquent des effets de la gravité ou de l'impact du vent sur leurs projectiles — ils veulent pouvoir choisir une cible et la toucher. D'un autre côté, les gens qui jouent à des jeux militaires basés sur un équipement réel veulent que l'équipement se comporte exactement comme il le ferait dans la réalité. Pour eux, une grande partie du plaisir vient du fait qu'ils savent que ce qu'ils font est « exactement comme » dans la réalité — même si c'est quelque chose de difficile et de complexe à maîtriser. Essayez de jouer à un simulateur d'avion de combat avec les réglages de réalisme poussés au maximum et vous comprendrez vite pourquoi seules très peu de personnes arrivent à piloter et pourquoi cela prend des mois d'entraînement pour y arriver.

De nombreux jeux vidéo aujourd'hui contiennent des éléments de narration : une mise en situation, un conflit ou un problème, et peut-être quelques personnages. La fiction est une simplification du monde réel, le récit d'une histoire qui n'inclut pas tous les détails. Vous pouvez facilement voir les effets de ceci à la télévision, en particulier dans les soap operas. Deux personnages ont une dispute et l'un d'entre eux révèle un fait choquant que l'autre ignorait. La caméra s'avance pour un gros plan dramatique. On entend le son soudain d'un violon et on voit l'autre personne, l'air stupéfait ou horrifié. Tout nous indique qu'il s'agit d'un Moment Important. Coupure publicité.

Bien sûr, ce n'est pas de cette façon que cela se produit dans la réalité. Dans la réalité, quand quelqu'un révèle quelque chose de choquant lors d'une dispute, l'autre parti demande une explication et la dispute continue. Les discussions ne s'interrompent pas juste en plein milieu d'un Moment Important : elles sont soit résolues, abandonnées ou interrompues. Les scénaristes de télévision, contraints pour le temps, sont obligés de laisser de côté tout le côté brouillon des conversations humaines. C'est aussi pourquoi les gens à la télévision ne disent jamais au revoir avant de raccrocher le téléphone. La narration n'en a pas vraiment besoin, et cela prend du temps.

De la même façon, l'art figuratif simplifie le monde réel. Il inclut ce que l'artiste considère comme important et omet ce que l'artiste juge comme secondaire. Le but n'est pas de présenter la scène exactement comme elle se déroulerait si le spectateur était là ; le but est de la présenter comme l'artiste choisit d'en faire le portrait. Même une photographie constitue d'une certaine façon une simplification : le fait de mettre un cadre autour d'une image constitue un acte de simplification, on choisit de montrer une chose et pas une autre. Et avec les graphismes générés par ordinateur, la simplification est particulièrement visible à moins que l'on y prenne un soin tout particulier. Dans la plupart des scènes générées par ordinateur, la lumière est trop régulière, les surfaces trop propres et les arêtes des objets trop nettement définies. Le monde réel est sale, mal rangé et imparfait, irrégulier, cassé et poussiéreux. Les images générées par ordinateur ont rarement ce grain rugueux, parce que cela nécessite plus d'efforts. La réalité est toujours beaucoup plus complexe que sa représentation.

Il y a d'autres raisons qui poussent à la simplification, en dehors d'une complexité qui n'ajoute pas de fun. La plupart des jeux vidéo sont soumis à des limitations techniques qui empêchent un trop grand réalisme. Une fois, j'ai dû travailler avec quelqu'un qui tenait à mettre en place une physique absolument réaliste dans un jeu de football. Le problème était que la machine sur laquelle nous travaillions n'avait pas la puissance nécessaire pour simuler le modèle physique en temps réel. Le jeu n'arrêtait pas de planter et ce n'était pas particulièrement fun. Nous avons simplifié le modèle physique et tout d'un coup, le jeu est redevenu agréable à jouer. Tous les programmeurs de jeu font ces simplifications par habitude, mais en tant que designer, cela aide à comprendre comment et pourquoi elles sont faites. Les simplifications faites pour des raisons techniques ont presque toujours des conséquences sur le gameplay, et parfois introduisent des bugs ou des problèmes.

Une autre raison de simplifier est de rendre un jeu plus accessible. Les jeux, en particulier les jeux de plateau, tendent à être symboliques. Ceci les rend plus simples à comprendre : les échecs sont une guerre réduite au noir et blanc. Les jeux vidéo utilisent souvent la même symbolique. La première édition du jeu *Balance of Power*, qui tournait autour de la géopolitique, considérait qu'il n'y avait que deux superpuissances dans le monde, les Etats-Unis et l'URSS. Toutes les nations représentées (et ceci aussi était simplifié) était alliées à divers degrés avec l'une des deux superpuissances. Ceci rendait l'équilibre du jeu immédiatement compréhensible et accessible, en plaçant le joueur dans une situation du type "moi contre lui" des plus simples. *Balance of Power* était un énorme hit à son époque et a même été utilisé par le U.S. State Department comme outil d'entraînement, malgré ces divisions artificielles.

La simplification n'est pas toujours une bonne chose. Personnellement, je ne pense pas que le jeu vidéo atteindra son plein potentiel avant d'avoir transcendé le concept de « jeu », la morale simplifiée, en noir et blanc, qu'il a emprunté aux jeux de plateau. Néanmoins, il y a de la place pour les deux sur le marché. Il est possible d'avoir des romans d'espionnages et des films d'action avec des gentils et des méchants et des explorations subtiles et riches du cœur humain comme celles de Dostoïevski et de Truffaut. Je pense qu'il est possible d'avoir des divertissements informatiques riches et subtils aussi, bien que je ne pense pas qu'on les appellerait des jeux.

Quand vous concevez un jeu, il est bon de ne pas simplifier trop tôt. Vous pourriez vous fermer trop tôt des pistes de recherches, des aires d'exploration, en décidant trop que « ce jeu ne traitera ni de ci ni de ça. ». Il est mieux d'inclure telle ou telle fonctionnalité dans le design et de l'enlever plus tard que d'essayer de l'implémenter tardivement.

D'un autre côté, l'une des pires erreurs d'un designer inexpérimenté est de croire que les choses peuvent rester éternellement vagues et indéfinies. On a beaucoup vu ceci quand Hollywood a essayé de se lancer dans ce domaine – les gens du milieu du cinéma ne sont pas habitués au développement d'un moteur et ils ignorent donc la catastrophe que cela peut provoquer d'essayer de changer les choses tardivement dans un processus de développement. Il est temps de simplifier lorsque vous vous asseyez avec vos développeurs et que vous dégagez un programme. C'est à ce moment que le brainstorming et les idées folles ont besoin d'être rassemblés en un seul produit, que les gens comprendront et pourront développer.

La simplification n'est pas véritablement un outil qu'on utilise ; il s'agit plus d'un processus par lequel on passe. C'est un processus dans lequel on ne devrait pas se lancer aveuglément. Quand vous tentez de simplifier un game design, demandez vous où se trouvent les compromis : ce que vous y gagnez et ce que cela vous coûte. Plus vous comprenez clairement les décisions que vous prenez, mieux vous saurez les expliquer aux équipes de marketing et de développement.

Tolkien, Beethoven, Vision

Par [Pierre Guyot](#) | le 30 octobre 2009 23:25:12 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juin 1999).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Selon le mythe de la création de l'univers fantastique de J.R.R. Tolkien, la terre a été dessinée par la musique. Eru, l'Elu, créa plusieurs esprits appelés les Ainur, qui chantaient devant lui. Mais il y eut des conflits entre les Ainur au sujet de quel thème devait dominer et on en essaya plusieurs. Finalement, Eru arrêta la musique d'une seule et vibrante note, et il révéla aux Ainur ce que leur chant avait fait en réalité – il avait formé une vision de la terre et de toute son histoire. Puis, avec un mot, Eru fit de la vision une réalité.



Le mythe de la création de Tolkien fait inévitablement penser à la Neuvième Symphonie de Beethoven, à tel point que je demande parfois si Tolkien n'a pas été inspiré par celle-ci. La Neuvième Symphonie est composée de cinq mouvements. Dans les trois premiers, l'orchestre explore trois thèmes très différents. Mais dans le quatrième mouvement, une conversation prend place entre les violoncelles, qui sont clairement un professeur, et les violons, qui sont les élèves. L'un après l'autre, les violons commencent à jouer chacun les thèmes des trois premiers mouvements. A chaque fois, les violoncelles les interrompent et désapprouvent vivement. Finalement, les violons introduisent timidement un nouveau thème. Les violoncelles les encouragent et le thème prend de l'ampleur jusqu'à devenir l'éclatant Hymne à la Joie qu'est le cinquième mouvement.

Beethoven était inspiré par une vision (si on peut appeler vision une expérience purement sonore). En fait, il n'a jamais entendu la Neuvième Symphonie, si ce n'est dans sa propre tête. Le temps de l'écrire, il était complètement sourd. Beethoven dirigea lui même l'orchestre lors de la première, sans en entendre la moindre note.

Le travail de Tolkien était aussi l'expression de sa propre vision, qu'il avait en lui depuis qu'il était un jeune homme. Il l'a raffinée avec les années, mais une grande partie des éléments fondamentaux de celle-ci sont restés identiques tout au long de sa vie. A la fin, Tolkien ne se voyait du tout comme « créateur » de ce monde, mais plutôt comme s'il mettait au jour quelque chose qui existait déjà. Quand quelqu'un lui posait une question sur la Terre du Milieu à laquelle il n'avait pas de réponse, il mettait une note dans son journal : « Doit découvrir comment. ». Pas « Doit penser à quelque chose. » ou « Doit créer quelque chose. », mais « Doit découvrir comment. ». Tolkien n'était pas un romancier en train de créer un monde, mais un historien qui se documentait sur celui qu'il pouvait voir dans sa tête.

Cette qualité de la vision est l'une de celles qu'en tant que créatifs nous ignorons à nos risques et périls. Non, nous ne sommes ni des Beethoven ni des Tolkien ; la plupart d'entre nous ne sont pas des génies ; nous produisons un divertissement de masse pour le marché, pas d'extraordinaires chefs-d'oeuvre de la musique ou de la narration. Nous concevons par groupes et non individuellement ; nous travaillons sous la contrainte de délais ; notre but premier est de faire un jeu qui se vende et qui rapporte de l'argent. Les conditions dans

lesquelles nous travaillons n'ont rien à voir avec celles dans lesquelles Tolkien ou Beethoven ont travaillé. Mais toute personne créative a besoin d'une vision, une image du travail achevé dans sa tête.

Il est facile de repérer les jeux qui manquent d'une vision. Ils semblent pâles, sans substance, sans l'harmonie dont j'ai parlée dans un article précédent. Parfois, ceci arrive quand il y a deux visions du jeu en compétition parmi les développeurs et que le producteur, de façon à rétablir la paix dans l'équipe, tranche par un mauvais compromis entre celles-ci. Parfois, c'est parce que personne n'a réellement cette vision : l'équipe patauge en essayant de produire quelque chose qu'elle ne comprend pas – ou pire, en laquelle elle ne croit pas.

J'ai souvent entendu le rôle de producteur décrit comme "celui qui garde la vision". Le producteur peut être le garant de la vision, s'il est aussi le designer, mais souvent le producteur est tellement coincé par des détails administratifs que la vision disparaît. La vision a besoin d'un garant, néanmoins, une personne qui y croit et qui la comprend, qui l'aime et qui peut l'expliquer et la défendre. Dans beaucoup d'entreprises de nos jours, ce n'est pas suffisant pour concevoir un jeu ; il est aussi nécessaire de mettre sur pied une campagne de propagande pour réussir à intéresser les commerciaux et le management. La vision ne doit pas seulement être gardée et nourrie, mais aussi défendue et répandue comme la divine parole. C'est l'un des rôles du game designer de nos jours.

Les visions, néanmoins, ne sont pas créées en tapant sur un clavier d'ordinateur. Les visions naissent de concepts et de connaissances. Elles naissent d'un croisement fertile d'idées. Elles sont le produit de la pensée elle-même. C'est l'une des raisons pour lesquelles ce serait une erreur pour les développeurs en formation de se concentrer uniquement sur l'apprentissage de la programmation ou du graphisme ou de la musique. La programmation, la musique ou la modélisation 3D sont des moyens permettant d'atteindre une fin, mais ce sont seulement des moyens. Avant eux, il doit y avoir une vision, et la vision émerge de la littérature et de l'art, de l'histoire et de l'architecture, de l'observation, et de l'émotion, et de l'expérience, le tout en interaction dans un esprit qui s'intéresse à de telles choses.

Une importante partie du Game Design consiste à être assis et à regarder le mur. C'est la partie la plus difficile, mais aussi la plus gratifiante. C'est aussi le moment où les managers de projets, obsédés par la productivité, sont le moins aptes à comprendre. Oui, vous devez faire des sessions de brainstorming avec l'équipe. Oui, vous devez consulter les gens du marketing. Oui, vous devez écrire des documents de game design. Mais aucune de ces activités ne peut accomplir la partie du processus de design la plus essentielle, c'est la pensée qui le peut. Parfois, il est nécessaire de passer du temps à rester assis, seul, dans un endroit calme, dans une fauteuil confortable, à regarder le mur et à penser. Peut-être avec un carnet et un crayon – mais pas un ordinateur. Un ordinateur empêche de se concentrer, il demande trop d'attention. Il se tient là, en vous présentant son curseur et en faisant tourner son engin 450 MHz, dans l'attente de pouvoir exécuter la moindre tâche que vous pourriez lui donner en quelques millisecondes. Vous lui donnez quelque chose à faire et BOUM ! Il l'exécute en un éclair, avant de revenir à sa position de départ, avec son curseur qui clignote comme le halètement d'un chien trop enthousiaste.

Les visions ne naissent pas de données rapidement traitées. Elles naissent d'une connaissance lentement assimilée. D'une réflexion et de considérations. Récemment, j'ai eu pour tâche de travailler sur un design pour lequel une équipe prototype avait été rassemblée. Trois programmeurs, un artiste, un level designer et un manager de projet étaient là, à attendre ce que j'avais à dire et ils voulaient que je lance n'importe quelle idée folle que je pouvais avoir pour qu'ils puissent se mettre au travail. J'ai trouvé ça assez bizarre. Je savais qu'ils étaient toujours dans l'attente de matière pour pouvoir travailler, je n'avais donc jamais le temps de simplement m'asseoir et de penser au design, de le laisser fermenter dans mon cerveau. C'est un processus qui ne saurait être brusqué.

Je n'ai jamais vraiment cru que la créativité requérait une sorte de qualité spéciale et inexplicable de l'esprit, qui ne serait pas sujet à l'analyse rationnelle et que seuls les artistes possèderaient. C'est une prétention du vingtième siècle en Occident. La plupart des cultures tribales n'ont de mot ni pour « art », ni pour « artiste ». Les gens fabriquent juste des choses. Ce qu'ils fabriquent peut être décoratif, cérémonial, spirituel ou servir à d'autres fins non utilitaires, mais ils ne le placent pas sur un piédestal en appelant ça de l'art.

Les Balinais sont un bon exemple de ceci. A Bali, tout le monde crée des choses tout le temps. Le chauffeur de taxi qui vous emmène de votre hôtel à votre aéroport crée probablement chez lui des choses que nous classifierions comme étant de l'art, mais ils n'ont pas de mot pour « art » parce qu'ils n'en font pas tout un plat. Le concept d'art et d'artiste n'est pas universel. Le désir de créer des choses est universel, mais placer ces choses et leurs créateurs dans une catégorie spéciale appelée art est une caractéristique de cultures extrêmement complexes comme les nôtres.

J'écris ceci car je ne veux pas que l'on pense que je défends cette période de calme dans le processus de design comme une sorte de chose mystique, ou parce que je veux placer la créativité dans le Game Design sur un piédestal. Quand je parle de la vision d'un designer, je ne parle pas d'une vision religieuse ou spirituelle, mais de la simple vision qui se produit dans la tête de toute personne créative, la vision qui est la réponse à la question « A quoi cela ressemblera-t-il, comment cela fonctionnera-t-il, qu'est-ce que cela fera ressentir quand ce sera terminé ? ». Tous les créatifs ont ceci en commun, de Picasso à un type qui fabrique des meubles dans son garage.

Le développement de jeux est un business avec des délais à respecter et de l'argent à rapporter. La plupart du temps, nos actions sont guidées par des considérations purement pratiques. Mais il s'agit quand même d'un processus créatif qui nécessite imagination et inspiration, et celles-ci se font rarement jour dans les réunions. Il y a une période dans le processus de développement où la chose dont un designer a le plus besoin, c'est la solitude.

Tais-toi et bosse ton jeu !

Par [Pierre Guyot](#) | le 30 octobre 2009 23:34:58 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (avril 1999).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Je suis vraiment fatigué du battage médiatique.

L'une des façons dont un développeur de jeux passe de « jeune et innocent » à « cynique et blasé » est une exposition prolongée au déluge irréflecti de superlatifs qui sort de la machine des relations publiques. Au bout d'un moment, tout sonne pareil et vous réalisez que vous ne pouvez plus rien croire.

Je suppose que dans une certaine mesure, le battage médiatique est inévitable dans notre industrie et plus généralement dans toute industrie du divertissement. Comme il y a peu de critères objectifs pour tester les jeux, le fait d'aimer ou de ne pas aimer un jeu se réduit à un problème d'opinion. Le battage médiatique est notre façon d'influencer l'opinion de nos clients. Et certaines de ces déclarations qui semblent complètement extraordinaires sont justifiées. Les jeux sont beaucoup plus rapides, beaucoup plus fluides et beaucoup plus beaux qu'ils ne l'étaient. Avec le développement du hardware à l'heure actuelle, il est difficile de trouver une façon de décrire ces changements qui ne sonne pas comme une succession d'hyperboles. Néanmoins, je pense que le battage médiatique représente un véritable danger pour le processus de développement.

Il y a quelques années, j'assistais à une réunion avec une tripotée d'autres programmeurs et j'écoutais un nabab de l'industrie Dont-Je-Ne-Prononcerai-Pas-Le-Nom promouvoir un nouveau produit. Il nous a présenté un transparent qui nous montrait que le nombre de couleurs que l'on pouvait afficher avec divers appareils avait changé au cours des ans. La carte EGA pouvait afficher 16 couleurs ; la VGA, 256, et ainsi de suite. Son produit, bien sûr, était hors du graphe – il pouvait afficher 16 millions de couleurs – plus, affirmait-il, qu'une télévision couleur, qui était limitée à 10 millions.

A cet instant précis, il nous a perdus. Son produit utilisait une télé couleurs en sortie ! Quel est l'intérêt d'avoir un appareil qui peut afficher 16 millions de couleurs si la télé couleurs peut seulement afficher 10 millions de couleurs en sortie de toute façon ? Son appareil gérait 6 millions de couleurs de trop, que l'utilisateur ne pourrait jamais voir (je dois aussi signaler qu'une télé couleurs est un appareil analogique et non digital, ce qui la rend théoriquement capable d'afficher une infinité de couleurs, mais il n'était apparemment pas au courant de ça).

Ceci est un exemple classique de battage médiatique décérébré. Le nabab de l'industrie en question a mis plein de gros chiffres sur l'écran et s'attendait à ce que tout le monde soit impressionné. Mais il s'adressait à des gens compétents sur le plan technique, dont le travail consiste à travailler avec les nombres et à raisonner sur leur signification véritable. En affichant ce transparent, il a fait insulte à notre intelligence, ce qui nous a rendus hostiles et suspicieux. Nous étions venus à la réunion en nous attendant à être éblouis. Quand nous sommes repartis, nous étions offensés et déçus. Ses mots avaient eu l'effet inverse de ce qu'il escomptait.

Plus tard, lors d'une présentation, j'ai vu le même appareil présenté par un professionnel avec une mâchoire taillée au couteau, un costume cintré et une quantité extraordinaire de gel. Il prétendait qu'il contrôlait un simulateur de vol au dessus de Yosemite Valley et que chaque branche de chaque arbre était affichée en temps réel. Il se trouve que je savais que c'était un mensonge éhonté, parce que j'avais vu la même démonstration ailleurs. On nous montrait un film prérendu fait avec Vistapro. Le présentateur a eu la présence d'esprit de disparaître par une porte à la fin de la présentation, de façon à ce qu'on ne puisse pas lui poser de questions.

L'engin en question atterrit sur le marché, brassa de l'air, suffoqua puis coula. Il n'est pas mort à cause du battage médiatique ; il est mort parce qu'il coûtait trop cher, réalisait trop peu et qu'il était épouvantable de développer sur ce support. Mais le battage médiatique a été incapable de le sauver. C'était un produit lamentable et la réclame ne lui attirait pas d'avis positifs dans la communauté des développeurs, à un moment où il en aurait eu vraiment besoin.

Au début de mars, cette année, deux entreprises très différentes, mais disposant toutes les deux d'énormes budgets pour les relations publiques, ont annoncé le même jour qu'elles étaient en grande difficulté. L'une était Purple Moon, les éditeurs de *Rockett's New School* et d'autres jeux développés pour les filles. L'autre était Ripcord Games, éditeur de *Postal* (premier coup) et du « fameux » *Space Bunnies Must Die* (deuxième coup).



Purple Moon a été fondée avec les meilleures intentions du monde. La compagnie voulait briser le moule dont semblait sortir la plupart des jeux vidéo. Elle allait développer des jeux ciblant les centres d'intérêt des filles et ce qui était important dans leurs vies. Purple Moon a fait une énorme quantité de recherches pour déterminer ce qui importait pour les filles. Puis ils ont mis en branle la machine des relations publiques pour aller clamer sur tous les toits de façon absolument outrageuse qu'ils allaient révolutionner l'industrie et rendre la technologie accessible aux filles (ce qui est juste un peu condescendant vis-à-vis des millions de filles qui utilisaient déjà joyeusement les ordinateurs). Mais ça a marché. Purple Moon a eu droit à une énorme attention de la part de la presse.

Le problème c'est que les jeux étaient en réalité de misérables aventures réalisées sous Macromedia Director. Ils étaient trop courts et offraient un intérêt faible en comparaison du prix de vente. Les artworks étaient de simples crayonnés. L'utilisation des voix et les effets sonores étaient obsolètes. Pire que tout, les thèmes des jeux étaient insipides : donner des conseils à ses amis, rentrer dans la bonne bande à l'école. Dans la série de jeux de football qui était prévue, le premier jeu était intégralement dédié à décider qui serait dans l'équipe – on ne jouait même pas au football. Mon Dieu, qui voudrait revivre un tel cauchemar ?

Les filles sont de toutes sortes, de celles qui aiment jouer à Quake et raconter des conneries à celles qui semblent être nées avec un nécessaire de maquillage dans les mains. Mais, comme les garçons, elles ont des rêves qui les emmènent hors de leur monde. Le problème des jeux de Purple Moon, si l'on exclut la nullité de la production, c'est qu'ils ne réalisaient aucun rêve.

Purple Moon a prétendu qu'elle ne pouvait gagner la compétition avec Barbie, la licence principale de Mattel. Bon, je suis extrêmement dubitatif quant à Barbie – je pense que son corps représente un idéal inaccessible que les filles tentent désespérément d'imiter, au détriment de leur propre santé, physique et mentale. Mais les jeux Barbie traitent d'aventures sans aucun doute plus intéressantes que celles de Purple Moon. Je suis sûr que Mattel a plus de budget pour le marketing et un meilleur système de distribution que Purple Moon. Mais Mattel a aussi de meilleurs jeux.

La véritable ironie dans tout ça est que Mattel semble sauver Purple Moon en la rachetant, sans doute pour la valeur de son service de relations publiques. Il y a quelque chose de profondément incongru à voir le très « non-PC » Mattel posséder le très « PC » Purple Moon.

L'autre entreprise qui apparut dans les informations le même jour était Ripcord Games. L'entreprise mère Matsushita a annoncé qu'il allait soit vendre, soit fermer Ripcord. Les choix de Ripcord n'auraient pas pu être plus différents de ceux de Purple Moon – *Space Bunnies Must Die* était un jeu sexexploitation de pacotille, avec une héroïne à la poitrine énorme qui dansait, changeait de vêtements et tuait des lapins géants. Mais Ripcord a aussi mis le bouton de réglage du volume de la publicité au maximum. A l'E3 1998, le logo de *Space Bunnies Must Die* – l'héroïne avec un gun – était imprimé sur chaque parcelle de surface disponible.

Matsushita a même fait construire un tunnel *Space Bunnies Must Die* dans l'un des halls, de telle sorte qu'il était impossible d'aller d'un bout de l'exposition à un autre sans passer par le tunnel.



Et pourtant, le jeu de Ripcord n'était rien de plus qu'une copie bas de gamme de Tomb Raider. Grosse poitrine ou non, Alison, le personnage principal, était mal modélisée et mal animée. L'histoire était nulle. Les décors étaient tout sauf impressionnants. La bande-son country n'allait absolument pas avec l'ambiance exploration spatiale. Tout le monde s'en moquait. Si *Space Bunnies Must Die* réalisait un rêve, il était bien trop tordu pour que la plupart des gens s'y reconnaissent.

Sans cesse, nous voyons des produits hypermédiatisés qui ne vont nulle part. Vous vous souvenez de Microsoft Bob ? Le logo Microsoft Bob était placardé absolument partout au Consumer Electronics Show ; il y avait même un avion qui tirait une bannière de façon à ce que vous puissiez le voir lorsque vous arriviez depuis votre place de parking. Mais c'était un produit lamentable et personne ne l'a acheté.

Le message ici est simple : tout le battage médiatique du monde ne saurait sauver un mauvais jeu. En tant que développeur de jeu, c'est votre travail de faire un bon jeu, et pour cela, vous devez avoir une certaine objectivité et un certain détachement. Vous avez besoin d'avoir l'œil vif, d'être capable de vous remettre en question et de vous demander en permanence : est-ce assez bon ? Comment l'améliorer ? Quels défauts ai-je oubliés ou laissés passer ? Vous devez devenir votre critique le plus acerbe.

C'est là que le danger du processus de développement intervient. Restez aussi loin que possible des départements des relations publiques et du marketing, sauf s'il est nécessaire de les informer au sujet de votre jeu. Si vous traînez là-bas trop longtemps, vous commencerez à croire ce que dit votre propre communiqué de presse, et quand cela arrivera, vous vous relâcherez, en croyant que le travail est déjà fait. Je suis sûr que c'est ce qui est arrivé à Purple Moon, Ripcord Games et Microsoft Bob. Leurs développeurs et, pire, leurs producteurs, ont été pris dans la frénésie du service des relations publiques et ont perdu leur détachement. Le département marketing est l'endroit où on vend le jeu, pas là où on vous fournit des critiques objectives. Si vous voulez savoir ce que les gens pensent réellement de votre jeu, demandez-le aux testeurs, qui y ont joué jusqu'à en être malades. Vous pouvez généralement compter sur eux pour vous révéler la cruelle vérité.

Bien sûr, on veut être fier de son propre travail. Mais cette fierté devrait venir d'un constat objectif de vos propres réussites, et non de la prose enflée qui apparaît sur les publicités. Il est évident qu'on ne peut pas se contenter de publicités qui disent que votre jeu est « le deuxième meilleur RTS sur le marché », mais si la publicité dit que c'est le meilleur et si vous savez que non, restez en conscient – autrement, vous ne faites que vous leurrer. Vous devez conserver votre capacité à juger votre propre travail.

L'autre chose dont il faut se souvenir est que si vous avez vraiment un produit incroyablement bon, d'autres personnes se chargeront d'assurer sa couverture médiatique pour vous. Id Software n'avait pas énormément d'argent dédié au marketing ; ils ont juste mis en ligne les premiers niveaux de *Doom* comme shareware et laissé Usenet faire le reste. La simple acclamation du public est de loin plus précieuse que n'importe quelle publicité que vous pouvez acheter.

Travaillez sans relâche et gardez un regard clair. Faîtes le meilleur jeu possible. Ne vous vantez pas et n'exagérez pas dans la presse – cela irrite les gens et sape votre crédibilité si vous ne pouvez pas répondre aux attentes. Votre travail, en tant que designer, est de penser et d'écrire, non de parler. Alors tais-toi et bosse ton jeu !

Vilain Game Designer ! Pas de biscuit pour toi ! III

Par [Pierre Guyot](#) | le 30 octobre 2009 23:40:35 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (février 2002).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Bon, cela fait presque deux ans que le dernier article « Vilain Game Designer ! Pas de biscuit pour toi ! » est paru, je pense donc que l'heure est venue pour un nouvel article. Je tiens à jour une liste de mauvaises fonctionnalités, d'erreurs de conception et de choses qui m'irritent personnellement dans les jeux vidéo auxquels je joue, et elle maintenant assez longue pour être publiée. Certaines d'entre elles sont des erreurs de level design ou même des faiblesses dans la programmation, mais il s'agit au moins toujours de choses sur lesquelles le game designer a une influence.



Armageddon adolescent

« Partez à la conquête du monde ! », « Le sort de l'humanité est en jeu ! » et « Sauvez la galaxie ! » crient les boîtes de jeux dans les rayons du magasin de jeux. « Non ! » suis-je tenté de répondre. « Je ne veux pas ! La galaxie n'a qu'à se débrouiller toute seule ! ».

Trop de jeux vidéo consistent à réaliser les fantasmes de puissance d'un adolescent et une apocalypse sans queue ni tête en est un symptôme classique. Cela fait un moment que je ne suis plus adolescent et je n'y crois plus. Peut-être que cela veut dire que je suis un adulte ennuyeux, qui n'est plus capable de visions grandioses... Mais voyons les choses en face, les types qui traînent en parlant de conquérir le monde sont complètement frappés. Je pense qu'il serait plus juste de dire que je m'en moque, tout simplement. Je ne veux pas diriger le monde. Je ne suis pas profondément intéressé par le sauvetage de la galaxie. C'est une tâche trop immense et trop impersonnelle, et il n'est absolument pas crédible qu'un individu réussisse à lui tout seule une telle chose. Ne me demandez pas de la faire. Je ne m'en sens pas l'envie.

Toute histoire requiert une tension dramatique, et la tension dramatique est créée par une situation qui met en danger quelqu'un ou quelque chose auquel le lecteur tient. De la même façon, tout jeu requiert un but, quelque chose que le joueur espère accomplir, ce qui crée ce que l'on peut appeler une « tension dans le gameplay ». La similitude entre la tension dramatique et la tension dans le gameplay est la raison pour laquelle les jeux vidéo font si souvent intervenir des éléments de narration. Mais vous vous penchez sur les grandes histoires dans la littérature, ce qui est en danger est rarement quelque chose de vaste et d'incalculable, comme le destin ou le monde. C'est plutôt la vie et le bonheur d'individus. Il y a plus de tension dans un roman de Charles Dickens – David Copperfield arrivera-t-il à survivre aux infâmes machinations de Uriah Heep ? – que dans n'importe quel film au sujet de la destruction de la Terre par des astéroïdes. Et même ces films n'essayent pas vraiment de faire jouer notre sympathie pour la Terre comme un tout. Ils essaient plutôt de faire jouer notre sympathie pour les personnages principaux du film et leurs destins personnels. Prenez *Le choc des Mondes*, par exemple. Ils détruisent la Terre et tout le monde dessus, mais – ouf ! – nos héros arrivent à fuir à l'abri. Dieu merci ! Happy ending !

« Mais attendez ! », vous entend-je crier avec irritation. « Pourtant vous êtes un des ces fanatiques de Tolkien ? Et *Le Seigneur des Anneaux* n'est-il pas aussi apocalyptique que l'on puisse l'être ? ». Oui, effectivement, je le suis et oui, effectivement, il l'est. Mais ce qui distingue *Le Seigneur des Anneaux* de la plupart de ses pâles imitations, c'est qu'il ne s'agit pas de montrer ô combien il est merveilleux de sauver le monde. Il s'agit de tout ce qui est irrémédiablement perdu même lorsque on y parvient. C'est un livre sur la tragédie de sauver le monde, le prix qui doit être payé pour y arriver.

Je pense que le succès des *Sims* démontre assez clairement qu'il n'est pas nécessaire de régner sur le monde et que beaucoup de monde n'en a même pas envie. Ils sont occupés simplement à nettoyer la vaisselle et à aller chercher le journal. Des millions d'entre eux sont parfaitement heureux de le faire, et Maxis se fait une fortune grâce à ce rêve très particulier. Il n'est pas nécessaire pour les jeux de tourner autour d'un Armageddon adolescent. Nous avons seulement besoin qu'ils tournent autour de personne auxquelles nous tenons et en fait, ceci nous permet une bien plus grande variété de jeux qu'un simple « Sauvez le monde ! » ne le permet.

Devoir se tenir (ou sélectionner quelque chose) exactement au bon endroit

Il n'y a pas beaucoup de choses à dire à ce sujet. Si le designer a choisi une région de l'écran à sélectionner extrêmement réduite à dessein, il s'agit juste d'un puzzle ennuyeux, d'une perte de temps du type *trial and error*. Si le designer l'a fait par accident, c'est un problème qui aurait dû être repéré durant les tests. Il y a un problème avec les testeurs : ce sont des joueurs si expérimentés – et après quelques centaines d'heures à un jeu, si expérimentés à ce jeu en particulier – qu'ils peuvent ne pas voir des erreurs de design qui pourrait irriter les casual gamers. A mesure que nous développons de plus en plus de jeux pour le marché des non hardcore gamers, nous avons besoin de testeurs qui peuvent penser comme des casual gamers.

Mauvais pathfinding

Le pathfinding est le processus qui permet de trouver comment faire passer une unité au sol d'un endroit à un autre en évitant les obstacles sur le chemin. Le pathfinding peut échouer de nombreuses façons, mais la plus frustrante est quand une unité reste coincée derrière quelque chose et reste incapable de comprendre comment en faire le tour. Le premier *Age of Empires* était connu pour son mauvais pathfinding, jusqu'à ce qu'un patch sorte pour corriger ce problème. Vous disiez à un groupe de gens de se rendre quelque part et ils restaient bloqués et erraient stupidement aux alentours jusqu'à ce que vous leur donniez de nouveaux ordres ou que vous ôtiez quelque obstacle si trivial qu'un enfant de deux ans aurait pu comprendre comment le contourner. Non seulement ceci est frustrant, mais de plus, ceci détruit la suspension d'incrédulité chez le joueur ainsi que son respect pour le jeu.

Le pathfinding n'est pas un problème simple, quelque soit l'approche considérée – je programmais pour gagner ma vie, alors j'en sais quelque chose. Le pathfinding dans les jeux est plus simple à certains niveaux parce que les soldats ne provoquent pas de courts-circuits lorsqu'ils se rencontrent sur le champ de bataille. Néanmoins, on ne peut pas non plus le laisser tourner toute la nuit. Quand le joueur dit à un soldat d'aller quelque part, le soldat doit partir immédiatement, sans s'arrêter de façon visible pour réfléchir à la façon de s'y rendre.

Voici quelques règles au sujet du pathfinding :

- Il ne s'agit pas de ce que les troupes peuvent voir, mais de ce que le joueur peut voir. Typiquement, le joueur a une vue aérienne sur la région et peut clairement voir le passage qu'il veut que ses troupes prennent. Même si ces troupes ne « connaissent » pas le terrain et ne peuvent pas « voir » la meilleure route depuis le sol, ils devraient utiliser le niveau de connaissance du joueur et non le leur

pour définir une route à suivre. Sinon, le joueur se demandera : « Pourquoi est-ce que vous allez par là ? ».

- Les soldats à pied ne devraient pas être bloqués par de l'équipement de leur propre camp. Dans le monde réel, si un groupe de fantassins essaye de passer derrière un groupe de tanks alliés, ils peuvent le faire, même si les tanks sont alignés essieu contre essieu. Ils grimperont dessus, ramperont en dessous ou bien quelque chose d'autre. Cela les ralentira peut-être, mais cela ne les arrêtera pas. C'est l'une des meilleures fonctionnalités du soldat de base – il n'a peut-être pas beaucoup de résistance ou de puissance de feu, mais il s'adapte mieux que n'importe quelle autre unité. Ne le privez pas de ses atouts en bridant son pathfinding.
- Les groupes d'unités devraient filtrer à travers les obstacles qui leur sont similaires en taille, mais devraient rester groupés quand ils se déplacent autour de gros obstacles. En règle générale, les groupes devraient rester ensemble et suivre grosso modo le même chemin, mais pas au point de passer tous du côté droit d'un même arbre. Et combien de fois avez sélectionné un groupe de soldats, leur avez dit d'aller quelque part et vous êtes rendu compte que deux parmi les vingt ont suivi un chemin bizarre de leur côté ? C'est ce qui se passe quand les deux traitent les dix-huit autres comme un obstacle plutôt que comme un groupe dont ils sont censés faire partie. Ils ont un peu trop d'esprit d'indépendance dans leur IA. Vous devez équilibrer leur liberté d'improviser des chemins individuels pour eux-mêmes (filtrer à travers des arbres ou des éboulements) et leur obligation de rester groupés (prendre la même route autour d'une colline ou d'un building).
- Permettre au joueur de placer facilement des points de contrôle. C'est votre « porte de sortie » si jamais votre pathfinding présente des bugs. En plaçant des points de contrôle, les joueurs peuvent éviter les problèmes de pathfindings. Évidemment, il est préférable de réussir du premier coup, mais résoudre le problème avec des points de contrôle permet au joueur d'avancer, plutôt que le faire abandonner de frustration. Et les points de contrôle sont généralement utiles de toute façon.

Des livres entiers sont consacrés au pathfinding, je m'arrêterai donc ici. Il s'agit essentiellement d'une question de tests et de réglages. Mais essayez vraiment de le faire bien ; un mauvais pathfinding est le meilleur moyen pour que le joueur abandonne le jeu en le considérant comme « stupide ».

Les arbres low poly (et les autres modèles 3D aussi)

Oooh, vous avez un moteur 3D. Nous sommes tous très impressionnés. Le problème, c'est que vous avez trop d'objets à afficher avec, alors vous avez décidé de tous les modéliser avec très peu de polygones. Tout dans l'univers de votre jeu semble extrêmement grossier, avec des bords bizarres et ne ressemble en rien à son équivalent dans le monde réel. Les arbres, par exemple, ressemblent à d'affreux parapluies, avec toutes leurs branches à la même hauteur, et des choses dérangeantes vont se produire dès que la caméra passera au-delà de leur feuillage.

Ne faites pas ça. C'est affreusement laid. Demandez plutôt à vos *pixel artists* de réaliser de beaux sprites et collez les sur un seul rectangle, si vous n'avez pas assez de polygones. Oui, cela va pixelliser quand vous vous approcherez, à moins de les MIP-mapper, mais ce sera la même chose pour les textures des murs : on y est habitué. Vous vous rappelez que les créatures de *Doom* avaient seulement un sprite lorsqu'elles reposaient sur le sol, mortes ? Et quand vous alliez de l'autre côté et qu'elles vous faisaient encore face de la même façon, en vous suivant comme les yeux dans ces terrifiants tableaux ? Et vous vous rappelez que ça passait très bien et que ça ne vous dérangeait pas plus que ça ? C'est aussi vrai pour les arbres – et même plus, en fait. A moins que ce ne soit d'une façon ou d'une autre significatif pour le gameplay, ce n'est pas vraiment important si l'orientation de l'arbre reste la même quelle que soit la position du joueur. C'est toujours mieux d'avoir un magnifique sprite d'arbre qu'une espèce de parapluie verdâtre.

Trop peu d'effets sonores pour une situation donnée

Je déteste entendre encore et encore le même effet sonore quand une situation particulière revient dans le jeu. Ça n'est pas un problème si c'est juste un bip de confirmation – dans ce cas, ça devrait toujours être le même sens, de façon à renvoyer un signal au joueur – mais s'il s'agit d'une personne qui parle, ceci devient très rapidement pénible. J'ai été le producteur audio/video pour *Madden NFL Football* pendant de nombreuses années et j'ai donc été coupable de ceci à l'occasion. Nous avions un temps d'enregistrement avec M. Madden limité chaque année et nous ne pouvions donc pas enregistrer tout ce que nous voulions. Le script audio pour *Madden NFL Football* était typiquement long d'environ 75 pages, et j'en aurais écrit le double si j'avais pu.

Si vous voulez associer des effets sonores à des situations particulières (« Je suis touché ! » et ainsi de suite), alors enregistrez en beaucoup. Ma propre règle est qu'il ne devrait jamais y avoir moins de cinq effets sonores différents pour n'importe quelle situation, même la plus rare ; et pour les événements courants, il devrait y en avoir au moins deux douzaines ou quelque chose comme ça. Vous n'avez pas toujours besoin d'enregistrer des phrases complètement différentes ; parfois, la même phrase dite avec une diction à peine différente suffira. Dans le jeu, faites en sorte que le jeu établisse une liste d'effets sonores et en choisisse un au hasard quand la situation le justifie, puis le coche dans la liste. La prochaine fois que la situation survient, choisissez un effet sonore dans le reste de la liste, et ainsi de suite. Quand vous les avez tous joués, réinitialisez la liste, à l'exception des effets qui ont été joués en dernier. De cette façon, le joueur n'entendra jamais deux fois de suite le même son.

Les oiseaux qui transportent des épées

Arg ! Notre équipe subit une attaque de poussins maléfiques de la mort du vil poulailler de Kafoozalum ! Nous risquons de nous faire picorer à mort, façon Tippi Hedren (NdT : l'actrice principale du film *Les Oiseaux*). Nous tranchons. Nous tailladons. Nous invoquons des sorts de Rôtisserie + 3. Certains d'entre nous sont blessés, d'une étrange façon numérique qui n'implique ni sang, ni douleur. Finalement, nous tuons le dernier des poulets (aucune créature maléfique n'est assez intelligente pour s'enfuir, même quand elle est complètement dépassée en nombre : un admirable sens du devoir pour un oiseau). En fouillant les corps, nous découvrons que comme toutes les créatures maléfiques, même le slime aveugle qui rôde dans les caves, ils transportent de l'argent et de l'équipement pour humains sur eux. Quel hasard ! Le poussin maléfique de la mort numéro 3 (le deuxième à partir de la gauche, à part ça, complètement indiscernable des poussins maléfiques de la mort numéro 1, 2 et 4) avait sur lui une Grosse Grande Méchante Epée de Gros Dégâts + 5. C'est marrant, je n'avais remarqué aucune épée sur sa personne pleine de plumes lorsqu'il était encore vivant. S'il était si lourdement armé, pourquoi n'a-t-il pas utilisé cette arme pendant le combat ? Et puisqu'on en parle, où gardait-il tout cet argent ? Dans son gésier ? Beeerk !

Vous avez compris.

Conclusion

Bien, voici ma liste de plaintes pour cette année (ou quelque chose comme ça). Si vous êtes responsable de l'une de ces erreurs, vous êtes un vilain game designer ! Pas de biscuit pour vous ! Et s'il y a des choses qui vous mettent personnellement en rogne ou des erreurs de game design que vous avez repérées, envoyez moi un e-mail et parlez m'en. Il est temps de commencer une nouvelle liste.

[Lire la suite →](#)

Vilain Game Designer ! Pas de biscuit pour toi ! IV

Par [Pierre Guyot](#) | le 30 octobre 2009 23:44:14 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mars 2005).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Je n'écirai pas l'article de ce mois-ci. Vous allez le faire. Ou plutôt, vous l'avez déjà fait. Quelques années après l'article « Vilain Game Designer ! Pas de biscuit pour toi ! », j'avais demandé aux lecteurs de m'écrire au sujet de ce qui les irritait dans les jeux, et hiiiiiiii ! qu'est-ce que j'ai reçu comme lettres ! Voici donc une liste des erreurs de conception et des causes d'énervement envoyées par de nombreux lecteurs. J'essayerai de rendre à César ce qui est à César quand je le peux, mais certains n'utilisent pas leur vrai nom et je ne peux dans ce cas malheureusement rien faire.



Les méchants avec des armes qui s'envolent.

Beaucoup de gens ont écrit pour se plaindre au sujet de celui-ci. C'est le corollaire du « Les oiseaux qui transportent des épées » du dernier article « Vilain Game Designer ! Pas de biscuit pour toi ! ». Vous passez une éternité à abattre un quelconque méchant important de l'histoire et quand vous y arrivez finalement, son arme à disparu. Evan McClanahan disait : « si j'arrive à passer discrètement derrière quelqu'un et à le poignarder dans le dos avec l'un de mes personnages puisqu'il immobilise le reste de mon équipe avec son Inépuisable Mitrailluse de Perforation (+2), je m'attends à pouvoir la récupérer, et non pas un cadavre sans rien, si ce n'est trois unités de monnaie du futur et un chewing-gum. »

Ceci est compréhensible dans les jeux online, dans une certaine mesure. Les monstres réapparaissent en permanence et s'ils laissaient une grosse arme chaque fois qu'ils mourraient, l'univers serait bientôt envahi par de telles armes. Mais dans les jeux à un seul joueur, vous avez la possibilité d'équilibrer le jeu correctement, et vous pouvez compenser en conséquence pour éviter ce genre de problème. Si cela déséquilibre trop le jeu, vous pouvez limiter la capacité du joueur à utiliser cette arme – Ok, vous avez tué le troll et récupéré sa masse, mais en fait, vous ne pouvez pas manier ce tronc d'arbre de trente kilos si facilement. Ou peut-être que vous avez obtenu l'incroyable gun du grand méchant, mais les seules munitions disponibles sont celles qu'il avait sur lui à ce moment. Si vous insistez pour vous trimballer cette chose partout dans l'espoir de trouver plus de munitions, libre à vous.

Un autre corollaire de ceci, envoyé par Chris Oates, est la démone dans un micro-bikini de cuir qui laisse derrière elle un set d'armure complet quand elle meurt. Quoi, elle avait ça sur elle mais ne se sentait pas de la mettre pour le combat ? Il y avait besoin de l'envoyer chez le teinturier ou quoi ?

J'ai joué à des jeux qui ont réussi ce point. Vous tuez un petit kobold, vous obtenez un petit couteau. Vous tuez cinquante petits kobolds, vous obtenez cinquante petits couteaux, dont aucun n'a vraiment de valeur, donc vous les laissez là où vous les avez trouvés, comme vous le feriez dans le monde réel.

Impossible de jouer au jeu avec différents niveaux de difficulté

Celui-ci est un peu délicat, parce que je sais ce que cela coûte à implémenter. Equilibrer un jeu pour des niveaux d'expériences différents prend plus de temps que de l'équilibrer pour un seul type de joueur, et dans le développement d'un jeu, le temps, c'est de l'argent. Néanmoins, tous les joueurs n'ont pas le même niveau et tous ne cherchent pas des challenges de la même difficulté (créer des jeux épouvantablement difficiles juste parce que vous le pouvez est un signe d'auto-indulgence, chez le designer, comme j'en parlais dans un précédent article, « Quel type de designer êtes-vous ? »). En donnant aux joueurs différents niveaux de jeu, vous augmentez la durée de vie du jeu, parce que les joueurs peuvent le finir et y rejouer de nouveau. Plus important, vous agrandissez le marché. Si le jeu est trop facile, il aura une très mauvaise réputation auprès des hardcore gamers. S'il est trop dur, c'est chez les casual gamers qu'il aurait mauvaise réputation. Si vous offrez au joueur la possibilité de jouer avec différents niveaux de difficulté, vous pouvez attirer ces deux catégories de joueurs.

Obscurité excessive

Trent Lucier a écrit pour se plaindre des jeux auxquels vous devez jouer volets fermés et la luminosité au maximum de façon à y voir quelque chose. Ok, pour les cinq premières minutes, c'est très effrayant et ça donne une atmosphère, mais après ça, c'est simplement pénible. Et comment se fait-il que tous les méchants peuvent voir dans le noir bien mieux que moi ? Je suis complètement d'accord avec ceci – je n'ai rien contre le côté « trouble », mais je ne m'amuse pas beaucoup avec le côté « sombre ». Scruter comme une taupe et rentrer dans les objets tout le temps ne rentre pas dans le cadre de ma vision d'un voleur/assassin/ninja furtif et cool. Le truc avec les assassins furtifs, c'est qu'ils voient très bien quand il fait sombre.

Charlie Byers pointe aussi du doigt le problème avec de nombreux portages de jeux PC sur Mac : les écrans Macintosh n'ont pas le même gammut de couleurs et d'après lui, la correction fournie avec l'Unreal Engine ne fonctionne pas correctement sur les Mac. Programmeurs, prenez en note !

Menus bâclés

De nombreuses personnes se plaignent au sujet des interfaces minables dans « l'enveloppe » du jeu – les menus de configuration par lesquels on doit passer avant de pouvoir jouer. Ceux-ci ne sont jamais très intéressants à concevoir ; ce sont juste des écrans de configuration, de sauvegarde et de chargement, etc. En conséquence, ils tendent à souffrir de l'un de ces deux problèmes : soit ils sont faits à la va-vite, avec un mauvais rendu et une organisation bizarre, soit ils sont beaucoup plus enluminés que nécessaire. Afin de dissimuler le fait qu'il s'agit juste de fonctions basiques et de mieux les faire coller au thème du jeu, tous les menus sont faits en lettres de feu tournoyantes, avec des effets de particules sur le nom sur lequel vous avez laissé le curseur.

La meilleure façon de réussir « l'enveloppe » du jeu du point de vue du joueur n'est pas de la rafistoler à grand coup d'effets sans intérêt, mais de lui donner un design efficace et la rendre simple d'utilisation, de façon à ce que le joueur puisse jouer aussi rapidement que possible. Créez une police qui soit en adéquation

avec le thème du jeu par tous les moyens, mais assurez vous qu'elle soit très lisible aussi. Essayez de ne pas avoir de menus avec plus de deux sous-menus à l'intérieur. Arrangez vous pour avoir des paramètres par défaut raisonnables, de façon à ce que le joueur puisse jouer tout de suite la première fois qu'il joue (étant donné la diversité des PC, je sais que cela peut être délicat.). La première page du manuel des vieux compilateurs Borland Turbo Pascal – avant même le titre – était intitulée « Comment commencer immédiatement », parce qu'ils savaient que tout le monde voulait y plonger la tête la première. C'est encore plus vrai pour les joueurs.

N'oubliez pas : si l'enveloppe est la partie la moins intéressante d'un jeu, c'est aussi la première chose que le joueur verra, avant d'entrer véritablement dans le jeu. Comme ils disent dans les guides pour réussir un entretien d'embauche, vous n'avez qu'une seule chance de faire une bonne première impression.

Rencontres aléatoires qui font perdre du temps

Alors vous avez entièrement nettoyé la ville de Patauge-les-Marais des affreux rats qui rendaient la vie triste, gagné quelques points d'expérience, et maintenant vous êtes parti pour accomplir de nouveau hauts-faits ailleurs. Mais, bien sûr, vous devez retourner à Patauge-les-Marais régulièrement pour vendre le fruit de vos rapines et renouveler votre stock de Carreaux d'Arbalète Extra-Pointus +3. Le problème est que, chaque fois que vous revenez, vous vous faites de nouveau attaquer par des rats lors de rencontres aléatoires. A ce moment, vous avez la force de massacrer des furets ou même de jeunes blaireaux, et c'est donc une complète perte de temps. Plutôt que de combattre les rats pour les quelques misérables points d'expérience qu'ils rapportent, vous vous retrouvez à les fuir, juste pour éviter le problème !

Une variante de cette nuisance est quand vous revenez à Patauge-les-Marais juste pour découvrir que la ville a été mystérieusement repeuplée par les rats, qui sont simplement cinq fois plus forts qu'avant, et que tout est à refaire. Ils ont la même apparence et le même comportement que la dernière fournée de rats, il n'y a donc rien de nouveau au niveau du gameplay, juste un challenge vide de sens.

Ca, ça veut clairement dire pas de biscuit. Un instant de réflexion va nous montrer quelle est la bonne approche. D'une part, toute région qui a été nettoyée de la présence d'une espèce de créature doit rester nettoyée pendant un certain temps (c'est un principe écologique de base autant qu'un bon principe de game design). D'autre part, les rencontres aléatoires avec des créatures trop faibles pour représenter une menace devrait faire fuir de terreur les dites créatures (c'est un comportement animal de base). De cette façon, vous ne gaspillez pas le temps du joueur avec des combats sans intérêt (NdT : *Persona 4* et *Persona 3* implémentent ceci de façon remarquable).

Impossibilité de sauvegarder après avoir soigneusement réglé son équipement

Les lecteurs assidus de mes articles sauront déjà où je me positionne dans le débat sur la sauvegarde : je pense que les joueurs devraient pouvoir le faire quand ils le désirent et s'il rechargent tout le temps pour réussir un passage difficile, c'est leur droit. Néanmoins, je suis prêt à accepter qu'il y ait d'autres points de vue sur le sujet.

Mais c'est vraiment très énervant, comme le fait remarquer Brendan Sechter, quand le jeu requiert un réglage soigneux d'unités/d'armes/de quoi que ce soit d'autre au début d'un niveau de façon à faire face aux challenges qui attendent le joueur, mais ne vous donne pas la possibilité de sauvegarder ces réglages. Régler les forces à disposition peut être un élément très intéressant du gameplay, mais pas quand vous devez le refaire de façon répétée chaque fois que vous recommencez le niveau – après un moment, il s'agit juste de par cœur ennuyeux.

Contrôle de la caméra lamentable (ou inexistant)

Dave Wilson a porté à mon attention le fait que certains jeux 3D à la troisième personne ne donnent au joueur aucun contrôle sur la caméra. Vilain game designer ! C'est pour cela que les environnements 3D sont faits : cela ne coûte rien, que dalle, nada de laisser une liberté de perspective. Regardez : le champ de vision normal pour un être humain est large d'environ 120 degrés. Le champ de vision varie considérablement selon les jeux et a tendance à devenir de plus en plus large dans la mesure où les télévisions HD deviennent monnaie courante, mais on est très loin des 120 degrés dans tous les cas. Pour compenser ceci, le joueur a besoin d'un contrôle décent sur la caméra. Ceci peut être bien fait. Jetez un œil à *Spyro the Dragon* ou à *Toy Story 2* sur Playstation et vous verrez d'excellents exemples de caméras fluides, intuitives et contrôlées par le joueur.

Les créatures qui peuvent redonner vie aux cadavres



J'ai vu ceci dans plusieurs jeux et cela donne l'impression d'être un prolongement fun et naturel du gameplay d'un jeu type fantasy. Le sombre nécromant a le pouvoir de ressusciter les corps de ses ennemis morts et de les faire combattre à ses côtés en tant que zombies ou autre. Riche idée... ou du moins ça en donne l'impression. Hélas, si cet aspect n'est pas correctement pris en main, cela peut déséquilibrer de façon radicale le jeu, parce que cela crée une rétroaction positive hors de contrôle : plus vous perdez d'unités, plus l'ennemi en gagne. Comme je l'ai dit ailleurs, imaginez ce qui se passerait si aux échecs, vous captiriez les pièces de l'ennemi pour les utiliser comme les vôtres. Le jeu serait bien plus court. Tout mécanisme qui permet à l'un des camps de récupérer les unités adverses nécessite un très grand soin dans la façon de l'équilibrer.

Dungeon Keeper avait en fait pas moins de quatre mécanismes pour transformer des créatures ennemies en « alliés », et elles étaient toutes équilibrées de façon différente.

1. Vous pouviez capturer les unités ennemies, les mettre en prison et les laisser mourir de faim (d'un certain point de vue, *Dungeon Keeper* ferait passer *Grand Theft Auto* pour un modèle de dignité humaine, mais nous ne nous lancerons pas dans ce débat pour l'instant). Dans ce cas, ils se transformaient en squelette de bas niveau, de piètres guerriers dans le meilleur des cas. Ceci prenait aussi un bon moment. Entre le temps nécessaire et la diminution de la force, ce n'était pas une technique qui déséquilibrait fortement le jeu.
2. Vous pouviez capturer des créatures ennemies et les faire torturer dans vos chambres de torture (oui, oui, je sais), ce qui amenait la plupart d'entre eux à changer de camp. Dans ce cas, la créature restait aussi forte pour vous qu'elle l'était pour l'ennemi. L'inconvénient était que cela prenait un très long

moment et que vous deviez constamment invoquer de coûteux sorts de soin afin que la créature ne meure pas sous la torture. De plus, plus la créature était forte, plus ce processus était long, ce qui naturellement tendait à en contrebalancer le bénéfice.

3. Vous pouviez traîner les cadavres de vos ennemis jusqu'à un cimetière et quand suffisamment de corps y avaient été enterrés, le cimetière produisait un vampire. Les facteurs d'équilibre était que cette méthode nécessitait plusieurs cadavres (huit, si je me souviens bien) pour produire un seul vampire, et le cimetière était particulièrement cher à construire. D'un autre côté, les vampires faisaient partie des créatures les plus puissantes du jeu une fois entraînés, ce qui en faisait une technique très efficace.
4. Le *Dungeon Keeper original* proposait une pièce spéciale appelée salle des charognards, dans laquelle vos créatures persuadaient vos adversaires de changer de camp – une sorte de lieu de recrutement pour traîtres. La contrepartie était que la salle des charognards était extrêmement chère, comme le cimetière, et pendant que vos créatures y travaillaient, elles n'étaient pas disponibles pour l'entraînement ou le combat. De plus, l'adversaire pouvait aussi avoir sa propre salle des charognards. Néanmoins, cette fonctionnalité déséquilibrait trop le jeu – en mode deux joueurs, le premier à avoir réussi à construire une salle des charognards gagnait presque automatiquement – et fut donc éliminé dans *Dungeon Keeper 2*.

Pour faire court, *Dungeon Keeper* est un exemple de jeu qui a réussi sur ce point et dont on peut beaucoup apprendre.

L'autre problème avec la résurrection des cadavres est que, souvent, l'unité qui ressuscite est très puissante : un puissant mage ou quelque chose dans ce goût-là. En conséquence, n'importe quel groupe avec celui est quasiment invincible. Si vous comptez donner à une unité particulière le pouvoir de procéder à de telles résurrections, pensez à la rendre assez faible et vulnérable pour compenser.

Conclusion

J'ai encore beaucoup de conditions pour refuser un biscuit à un game designer que je n'avais pas la place de mettre ici... Je crois que je vais m'en servir pour constituer une base de données ! Mais je suis toujours intéressé par de nouveaux cas. Ecrivez moi (*NdT : en anglais*) à notwinkie@designersnotebook.com au sujet des erreurs de game design qui vous ont vraiment fait enrager (regardez les précédents articles, au cas où celles-ci seraient déjà mentionnées).

[Lire la suite →](#)

Les achievements : le su-sucre des geeks

Par [Thomas Chassin](#) | le 13 juin 2010 00:05:25 | Catégories : [Game Design](#)



ACHIEVEMENT UNLOCKED
Vous lisez Les Forges

I. Introduction

1. Achievement, késako ?

En voilà une bonne question: qu'est ce qu'un achievement ? Certains le savent peut-être déjà, d'autres en ont peut-être entendu parler sans le savoir et, enfin, d'autres n'en ont jamais entendu parler. On verra petit à petit de quoi il s'agit, mais pour que la suite de l'article soit compréhensible, il convient de donner une courte définition de l'achievement.

Il s'agit d'une distinction reçue par le joueur pour avoir accompli un certain fait. En français, on peut parler de hauts faits, d'exploits, de défis et plein d'autres petits noms doux. On les retrouve un peu partout en y regardant à deux fois ! Dans des MMO comme *World of Warcraft* ou même dans des jeux de shoot comme *Call of Duty Modern Warfare 2*, ils sont présents. Pour ceux qui n'auraient pas tout à fait compris, voilà un exemple plus concret : on pourrait imaginer un achievement nommé « Au voleur ! » que le joueur obtient en volant une pomme dans chaque maison du village. C'est parfois aussi simple que cela.

2. Super... Et à quoi ça sert ?



Voilà une autre bonne question. Pour commencer, faisons un petit voyage en arrière dans le temps. Nous sommes en dans les années 70/80. C'est la Guerre Froide, les hippies dominant le monde et un peu partout, des salles d'arcades fleurissent. C'est super génial, il y a du pixel art à tous les coins de rues là-dedans. Mais

le mieux, c'est pas seulement de jouer au jeux, c'est d'avoir son nom dans les highscores (scores élevés pour les anglophobes).

Sauf que les jeux se sont tellement perfectionnés et complexifiés avec le temps que la notion de score est devenue obsolète. En dehors de quelques jeux online ou d'applications sur iPod, ils ont quasiment disparus.

Alors, pour récompenser les joueurs les plus assidus, ceux qui veulent tout finir, on a mis en place les achievements. Ils remplacent les highscores et permettent de montrer que vous êtes le plus fort et que wouah ! Vous avez terrassé le dragon maléfique, le boss final, armé d'une cuillère en bois vermoulu.

3. Donc c'est un truc pour les drogués du jeu, les achievements ?

Et bien, non, pas forcément ! Car il existe de multiples façons de présenter les achievements et rien ne vous oblige à demander au joueur de survivre à l'Armageddon de fin du jeu en caleçon. De plus, il est souvent amusant pour un joueur de voir qu'il a relevé un défi proposé par les game designers. Les achievements peuvent également pousser le joueur à découvrir plus en profondeur le contenu du jeu, par exemple en lui proposant d'explorer un lieu difficile d'accès.

Obtenir un achievement est souvent un moment agréable pour le joueur, puisqu'il ressentira probablement une certaine satisfaction, et verra ses efforts récompensés. Il faut également imaginer l'éventualité où le joueur joue avec des amis à un même jeu, cela leur permet de comparer ce qu'ils ont fait et de se lancer des défis. Il faut également penser immersion et roleplay.

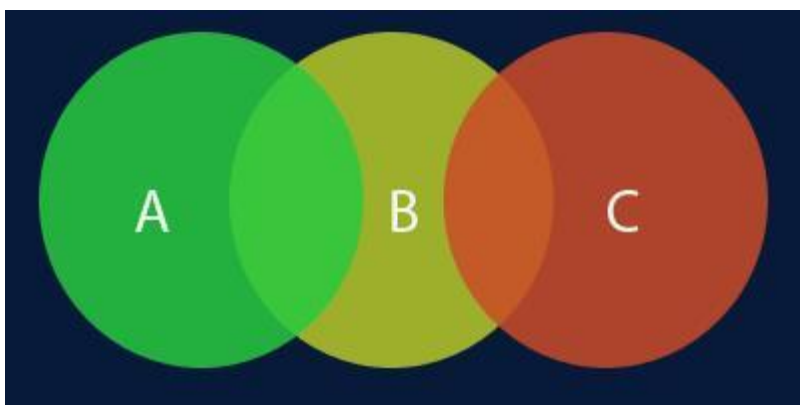
II. Étoffer son jeu grâce aux achievements

1. J'aimerais mettre des achievements dans mon jeu, mais je ne sais pas lesquels...

Il y a trois sortes d'achievements dans ce bas monde, comme indiqué sur le schéma ci-dessous :

- les super achievements du game-designer malin (A)
- les achievements tordants du game-designer facétieux (B)
- les achievements tout pourris du game-designer lourdaud (C)

Les premiers sont une valeur sûre, les seconds sont à utiliser avec précaution et on abordera rapidement les derniers dans les bêtises à ne pas faire. En fait, pour faire simple, vous visez tout ce qui se trouve dans l'ensemble) et vivez dans la hantise de C. Pour ceux qui se posent la question, Si c'est ni A, ni C mais en B, ça reste acceptable, ça fera peut être rire quelqu'un (je vous le souhaite en tout cas).



A. Les achievements du game-designer malin

Qu'est-ce qu'un bon achievement ? C'est un achievement qui met en avant les atouts du jeu, qui pousse le joueur à en découvrir tout le contenu, à tout essayer, qui l'immerge dans l'univers, bref, un achievement qui parachève le jeu et permet au joueur de s'éclater. Ils peuvent faire la différence entre un bon jeu et un excellent jeu. Vous l'aurez deviné, un bon achievement est un achievement d'exploration, de collecte, de roleplay.

Cependant, il faut que cela ne soit pas trop pesant. Un achievement d'exploration, un bon, ne consiste pas pour le joueur assoiffé de gloire d'arpenter chaque pixel de l'univers que vous avez créé, il faut plutôt qu'il amène le joueur à se rendre à certains endroits particuliers que vous ne souhaitez pas qu'il rate, comme le sommet de la montagne magique, les ruines englouties au fond du lagon, ou encore ce magnifique lac soufré couleur émeraude au cœur du volcan ardent. C'est ce genre de détail que vous avez bichonné et que vous voulez qu'il voie, même si le passage n'y est pas obligatoire.

Il peut encore s'agir de collectionner les sept lames ultimes sur lequel votre infographiste aura travaillé jusqu'à mourir étouffé sous une montagne de croquis ou je ne sais quelle gemme qu'on obtient après une longue série de quêtes annexes. Oui, encouragez le joueur à faire les quêtes annexes, vous les avez créées pour ça, et en plus ça rallonge la durée de vie.

Enfin, encouragez-le à améliorer son (ses) personnage(s) autant que possible, que ce soit en atteignant le niveau 100 ou en ramassant les 128 cristaux de vie cachés un peu partout. S'il y a autant de niveaux ou de cristaux, c'est bien pour que le joueur puisse aller au bout. Et en plus, ceux-là sont passe-partout, à moins que votre jeu ne soit fait que d'un seul personnage impossible à améliorer combattant un seul et unique monstre... Et encore, on peut toujours en trouver.

B. les achievements du game-designer rigolo

Je préviens tout de suite, cette catégorie est à manier avec prudence et à utiliser avec parcimonie, à moins que votre jeu ne soit purement comique. Je rappelle qu'ils chevauchent les deux autres catégories d'achievements (voir schéma), donc peuvent être aussi bien tordants que lourds...

Ceux-ci peuvent viser à faire essayer toutes sortes d'action totalement débiles et (théoriquement) désopilantes. Par exemple, trouver les 28 façons stupides de se suicider à travers le jeu ou tuer la marmotte albinos mutante qui apparaît une fois sur 20 au sommet de telle montagne (attention il vaut mieux éviter les événements qui arrivent une fois tous les 23 ans et qui sont quasiment infaisables de surcroît). On retrouve donc également des achievements d'exploration extravagants, incongrus et inattendus. À titre d'exemple, si vous avez inventé un lieu en total décalage avec le reste de votre univers, poussez le joueur à aller le voir. On peut également avoir d'autres objectifs (pour le moins) surprenants, comme par exemple visiter les latrines publiques de chaque hameau traversé, pousser à se saouler un certain nombre de personnes, etc. De la même façon, vous pouvez proposer au joueur de collecter les cinq armes les plus nulles du jeu, ou bien un set d'armures en dentelle rose parfaitement ridicule. Les références à des jeux ou films d'anthologies seront également toujours très bien perçues par ceux qui les comprendront. En clair, soyez imaginatifs et un peu fantaisistes.

C. Les achievements à éviter

Je ne m'étendrai pas sur le sujet. Il s'agit des achievements sans intérêt précis autre que de marquer une progression du joueur ou ce genre de chose... On peut compter les « Mourrez x fois », les « Jouez 10 heures », « Tuez tel boss que vous ne pouvez pas éviter de toute manière » et les « Atteignez le niveau y ». Pas de contresens ici : y est un nombre peu élevé, que vous passerez forcément. Par exemple, dans un jeu où le niveau maximum est 100, il est peu intéressant de récompenser le joueur pour avoir atteint le niveau 5. Si votre jeu est travaillé, il restera de lui-même suffisamment longtemps pour passer ce niveau.

Cela permet au joueur de voir ce qu'il a accompli, mais en contrepartie, il sait quand la fin approche, sans surprise aucune...

De plus, les achievements ne sont pas des marqueurs de progression pour le joueur. En fait, de façon générale, les achievements ne doivent pas pleuvoir sur le joueur lorsqu'il joue, dans la mesure où les obtenir est annexe et requiert d'accomplir des tâches spécifiques, donc logiquement le joueur ne devrait en obtenir que très occasionnellement lorsqu'il ne cherche pas à les accomplir. Gardez aussi à l'esprit que si la diversité de vos achievements et leur nombre peut être un plus pour votre jeu, la valeur de l'achievement tient à la spécialité de l'acte à accomplir. S'il y a trop d'achievements, la spécialité devient une norme, et les actions censées être mises en valeur sont banalisées... Tout est donc une question d'équilibre dans le nombre d'achievements.

2. Mais le vilain flemmard de joueur, il veut pas les obtenir mes achievements !

Haha, le méchant ! Il faut l'appâter le bougre. Il faut bien présenter les achievements, pour le décider à les obtenir.

A. Présenter ses achievements

Pour bien les présenter, il convient de leur donner un titre sympathique. Par exemple, si un achievement consiste à tuer cent trolls, « Tuer cent trolls » est un titre adéquat, mais assez peu alléchant. Choisissez plutôt un titre glorieux « Pourfendeur de trolls » ou amusant « Ceux-là ne pourront plus faire peur aux gosses ». Si vous optez pour ce genre de titre, il est bienvenu de donner une aide au joueur concernant ce qu'il doit faire. Cette aide peut aller du simple indice à une description exhaustive de la marche à suivre. Une petite image ou vignette en rapport avec le fait à accomplir rend également l'achievement plus attirant. Attirer l'attention du joueur est également crucial : signalez-lui lorsqu'il obtient un achievement. On reparlera du signal plus tard.

Voici un exemple de hauts faits bien présentés, tiré d'un MMO dont le nom commence par *World of* et finit par *Warcraft*.



Une image, un titre plus ou moins sympa et une description de l'objectif... Je ne vous ai pas menti, vous voyez ? En plus, les achievements sont classés, et le joueur peut savoir à l'avance ceux qu'il n'a pas encore obtenus et ce qu'il doit faire pour les avoir. Les points de hauts faits renforcent les interactions entre joueurs car ils permettent de jouer à « qui en a le plus ? ».

B. Vendre son bazar au joueur de passage

Bien présenter ses achievements, c'est bien. Mais il y en a qui n'en ont rien à faire du tout. Alors, si vous voulez les avoir, il faut être prêt à donner une récompense autre qu'une vignette et un titre...

C'est là qu'intervient un bas instinct auquel AUCUN être humain ne résiste (je rappelle que les trolls ne sont pas humains, ils n'ont pas d'âme) : l'appât du gain ! Oui, ça peut paraître petit, ça peut paraître bas, mais c'est implacablement efficace. Je rassure ceux qui hurlent à la mort en serrant leur porte-monnaie : non, vous n'allez pas payer le joueur. Pas en argent du moins. Donnez-lui du contenu parfaitement idiot, drôle, obsolète ou attrayant, il sautera dessus. Il existe plusieurs possibilités pour cela :

- Vous donnez au joueur un truc débile ou rigolo en fonction des achievements obtenus et vous regroupez ceux trop facile à obtenir pour donner une récompense pour l'obtention de tous les achievements du groupe (indiquez bien les groupes). Par exemple, si le joueur arrive à trouver les 28 façons idiotes de se suicider (dans le jeu, hein), il pourra par exemple invoquer un petit spectre qui le suit partout et qui ne sert à RIEN. Le joueur appréciera.
- Vous pouvez aussi choisir un système de points de récompense. C'est vieux comme les jeux vidéo, mais ça marche, et ça reste d'actualité : regardez l'exemple précédent. Le joueur reçoit donc des points en fonctions de la difficulté des achievements obtenus et peut ensuite acheter tout un tas de choses inutiles mais jouissives. Si la quête menée par votre quelconque personnage l'amène à croiser de fort jolies créatures, quelques artworks ou costumes à débloquent seront les bienvenus. Vraiment les bienvenus.

- Vous pouvez également faire un mélange des deux : le joueur gagne des points, mais certains objets restent à débloquent en obtenant des achievements spécifiques, ce qui est un bon compromis.

III. Vilain pas beau, toi pas faire ça !

1. Déroulement du jeu

A. Le joueur avec tes bêtises tu ne bloqueras point

Ça me paraît assez clair comme titre. N'oubliez pas : à moins que les achievements soient le pilier central de votre jeu et que tout votre gameplay s'appuie dessus, ils sont SE-CON-DAI-RES ! Concrètement, le joueur aura peut-être envie de finir le jeu, puis de revenir obtenir ses petits achievements, ou bien il les ignorera totalement. Dans les deux cas, il est impératif que les achievements n'interfèrent pas avec le déroulement de l'intrigue. On oublie immédiatement les « Pour débloquent le boss chef orc, tue 14364 orcs en maniant le canif rouillé à cloche-pied les yeux bandés ». Niet. Nada. Pas de ça. La réaction de la plupart des joueurs est Alt+F4 ou bien la désinstallation immédiate du jeu.

B. Au joueur, Excalibur tu ne donneras point

Si j'ai dit plus tôt qu'il fallait donner uniquement des objets inutiles au joueur, c'est bien parce qu'il ne faut pas lui en donner d'utiles. Les achievements, je le répète, sont secondaires. Ils ne doivent pas permettre au joueur d'acheter Excalibur parce qu'il a passé la nuit à tuer des slimes des bois à coup de dents... Le joueur ne doit pas non plus compter sur les achievements pour se débloquent. Ils n'interviennent JAMAIS dans le déroulement du jeu ou le gameplay ou alors, le jeu ne se repose que sur eux. Il faut choisir.

2. Des feux d'artifice dans tous les sens tu ne tireras point

Des feux d'artifice ? Gné ? Je vais clarifier ça au moyen d'un exemple. Prenons un joueur normal : Bob (c'est court à écrire). Bob s'est longuement préparé avant d'avant d'affronter Nécros, le Chevalier noir et son armée de squelettes. Il a tué plein de squelettes pour s'entraîner et maintenant, il est fin près. Ce qu'il ne sait pas, c'est qu'il a tué 198 squelettes. Le combat commence. Bob tape Nécros, le combat est serré, puis deux squelettes apparaissent. Bob les occis prestement et là, c'est le drame. Taratata ! Trompettes et confettis, une ENORME fenêtre apparaît en plein milieu de l'écran dans une gerbe de feux d'artifice pour lui annoncer joyeusement qu'il a enfin tué 200 squelettes. C'est si gros que Bob ne voit presque pas la tête de son héros rouler sur le plancher. Résultat des courses : Bob se pend après avoir dûment défenestré son matériel informatique (Bob était très émotif). Je me suis fait bien fait comprendre : s'il faut annoncer l'obtention des achievements, il faut le faire discrètement.

A titre d'information, la dernière pensée de Bob fut : « Pourquoi diantre des feux d'artifices et des confettis accompagnaient cette immonde fenêtre ??? ». Il faut encore penser roleplay et immersion. Vos achievements ne doivent pas tomber comme un cheveu sur la soupe. Trouvez un moyen ingénieux d'annoncer l'heureuse nouvelle au joueur, tout en restant discret. Par exemple, un personnage pourrait dire « Je crois que nous avons fait quelque chose d'épique, notons le dans le journal de bord. ». Pour peu que les achievements soient présentés comme des notes dans ledit journal, vous avez une présentation cohérente de vos achievements qui respecte l'immersion du joueur et accentue le roleplay : c'est bien. La fenêtre moche, c'est mal.

3. Et la catégorie C, elle devient quoi ?

La catégorie (C), parlons en. C'est tout pourri. Elle regroupe l'ensemble des pseudo blagues des game designer, et les achievements lourdingues qui ne servent à rien. En somme, s'ils ne bloquent pas le joueur, ils ne sont pas totalement néfastes. Mais si le joueur doit décapiter les Sept Dragons Maudits de la Prophétie de la Fin du Monde en caleçon à cœur et armé d'une brosse à dent usagée pour débloquent l'ultime artwork

de son héroïne préférée en petite tenue, il risque de tomber en dépression (et je vous interdit de dire bien fait parce que c'est vous qui l'avez mis cet artwork et parce que vous êtes bien content qu'il joue à votre jeu !). Vous pouvez toujours mettre une catégorie d'achievements pour les longs après midi pluvieux des hardcores gamers qui larvent devant leur ordi, mais ne pénalisez pas les pauvres joueurs en quête d'artworks.

Le délicat mariage de la narration et du gameplay

Par [Diamonds](#) | le 21 novembre 2010 00:24:22 | Catégories : [Game Design](#), [Scénario](#)

Cet article est une traduction de [Diamonds](#).

L'[article original](#) est écrit par Ara Shirinian publié par Gamasutra (janvier 2010).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Le jeu vidéo est un médium incroyablement puissant et complexe. Au vu de toute son histoire (tous ces Pokémons et ces fusils de la Seconde Guerre mondiale ne sont absolument pas à négliger), on peut penser que le jeu vidéo est l'unique médium qui puisse rassembler tous les autres média qui l'ont précédé. Rappelez-vous que dans les années 70, les jeux vidéo n'étaient généralement que des applications à la jouabilité rigide. Lorsque vous achetiez un jeu pour votre Atari 2600, il allait sans dire que ce que vous possédiez était, à de rares exceptions près, un système de règles, un objectif à atteindre et une interface au sein de laquelle jouer.



Les jeux vidéo en 2009 utilisent encore largement ces mêmes ingrédients, mais le développement technologique de ces quatre dernières décennies a permis à nos jeux d'utiliser une myriade d'autres moyens d'expression que la plupart d'entre nous tiennent pour acquis.

Les changements n'ont pas été quantitatifs mais qualitatifs, et ont tellement étendu la signification de l'expression « jeu vidéo » que son sens originel semble avoir dans certains cas complètement disparu. Les jeux des années 70 ne pouvaient pas exprimer l'étendue du paysage qui se déploie devant vous dans un *DiRT 2*, le mystère des premiers pas dans l'utopie en ruine de *Bioshock*, ou ce pur concentré de tradition tolkienesque raconté à travers les livres d'*Oblivion*. Ils ne pouvaient pas exprimer les subtilités auditives de *Batman : Arkham Asylum*, le mélange de douceur et d'agressivité de votre hôte dans *Portal* ou la tension brûlante entre Snake et Ocelot dans la série des *Metal Gear Solid*.

Silencieusement et sans l'assumer, pour le meilleur ou le pire, le jeu vidéo d'aujourd'hui a évolué au-delà de la simple abstraction du gameplay, vers un médium de divertissement général pouvant contenir des images, des sons et des textes de tous types. En effet, nous avons déjà dépassé le stade au-delà duquel le type et la qualité d'exposition ne sont plus limités par une quelconque contrainte technologique mais uniquement par nos choix et notre ingéniosité. A proprement parler, le jeu vidéo est à même de donner forme à une narration (comme le cinéma) et à un gameplay (qui est quant à lui spécifique de l'interaction entre humain et machine). La majorité des gens pense aujourd'hui que le cinéma est plus adapté que le jeu à l'expression de la narration. A l'inverse, le gameplay est une qualité propre au jeu vidéo. Pourtant, les jeux vidéo sont aussi aptes à construire une narration : techniquement parlant, ils ont la plupart des capacités du cinéma. Le jeu vidéo est ainsi le seul médium en lice si vous voulez du gameplay, et il constitue un bon support pour exprimer la plupart des choses que nous avons faites dans le domaine du cinéma. Il n'est donc pas surprenant que nombre des plus brillants développeurs de jeux aient tenté de les combiner de manière élégante – d'unifier les deux média, si vous préférez.

Il y a quelques années, il y a eu une explosion médiatique autour de la perspective d'inciter les joueurs à pleurer. Selon Neil Young, alors General Manager de EA Los Angeles :

« L'une des choses les plus importantes pour nous, c'est de répondre à la question sur laquelle notre entreprise s'est construite : « Un jeu vidéo peut-il vous faire pleurer ? ». C'est une question à laquelle Steven Spielberg peut aider EA à répondre. »

Peu après, le designer David Jaffe révéla qu'il était en train de travailler sur le même problème avec l'un de ses concepts de jeu : « L'un d'eux est d'être le jeu vidéo le plus émouvant jamais créé. L'objectif final est que les joueurs, à la fin du jeu, soient réellement choqués – s'ils ne pleurent pas – parce que nous avons très bien fait notre boulot. »

Hideo Kojima, qui s'est fortement concentré sur la narration avec ses *Metal Gear Solid*, a également exprimé son désir d'intégrer ces éléments dans un gameplay plus efficace que ce que lui et d'autres ont été capables d'accomplir :

« *Halo*, *BioShock* – je vois leur approche et je pense qu'ils sont brillants d'une certaine manière, mais je sens qu'il manque encore une sorte de profondeur dans le scénario, ou dans l'expression des sentiments des personnages. J'ai des idées sur la manière d'essayer d'aborder et de contourner ça. »

« Dans *Metal Gear Solid 4*, oui, j'ai tout donné dans les scènes cinématiques. Ce que je regrette dans une certaine mesure, parce qu'il y a peut-être une nouvelle approche à laquelle je devrais penser. J'y pense toujours – les rendre interactives mais, en même temps, raconter la partie scénarisée et les événements dramatiques plus sensiblement. J'aimerais réussir une telle approche, et j'y travaille encore. »

Selon lui, il s'agit une progression et une combinaison logiques. Vous ne faites que regarder un film, mais vous jouez au jeu et tout ce qui est exprimé dans le film peut aussi être contenu dans un jeu. « Une narration au sein laquelle on jouerait » semble alors être le Saint-Graal du jeu, non ?

Mais pourquoi n'avons-nous pas encore accompli cette parfaite synthèse du gameplay et de la narration ? Pourquoi y a-t-il toujours eu des compromis et des combinaisons guindées entre les deux ? Sommes-nous trop naïfs, ou juste insuffisamment intelligents en tant que créateurs de jeu pour le découvrir ? Ou y a-t-il quelque chose d'autre ?

Pour le savoir, nous devons tout d'abord évaluer ce que nous avons déjà accompli dans ce domaine, avant d'examiner chaque medium, pour déterminer s'il y a un élément qui fait de ce problème quelque chose de plus épineux encore que de réussir le mélange du beurre de cacahuète et du chocolat.

Il y a du gameplay dans ma narration... Ou de la narration dans mon gameplay ?

Narration et gameplay sont sans aucun doute les deux formes d'expression les plus importantes dans le jeu vidéo actuel. Mais avant de nous jeter à l'eau, commençons par préciser ce que nous entendons respectivement par narration et gameplay.

Le gameplay est relativement simple à définir pour nos besoins. Toute action, dans le jeu, produite par le biais de l'interaction entre le joueur et le jeu, est appelée gameplay.

Techniquement parlant, la narration est juste la description d'une série d'événements. A son niveau le plus atomique, pratiquement n'importe quel élément virtuel du jeu peut porter une sorte de narration en lui, en vous racontant quelque chose à propos du contexte et de l'histoire.

Habituellement, quand on parle de narration dans le domaine du jeu vidéo, on pense souvent aux cinématiques, à l'intrigue, aux relations entre les personnages et aux histoires racontées dans le cadre d'un jeu. Toutefois, l'image d'une lame abîmée couverte de sang constitue un élément de narration au même titre qu'une scène entre deux personnages en plein débat sur leur prochaine destination. La principale différence est celle de la complexité, de la sophistication. Dans le premier exemple, la narration est exprimée en un seul visuel : je vois une lame, elle a récemment provoqué le sang et a heurté un objet très dur. Dans le second exemple, la narration est portée par des éléments visuels, comme le choix du cadre ou des costumes, mais aussi par la performance des acteurs, la gestuelle, la voix, etc.

Il s'agit d'un point essentiel : la dimension graphique du jeu crée toujours une certaine forme narration, d'histoire, qui est racontée au joueur (ou du moins est interprétée par celui-ci). Le joueur aura une tendance naturelle à inventer une histoire autour de cette image de lame ensanglantée, même si ce n'était pas proposé ou attendu par le concepteur. Même les jeux d'action pure sans scénario incluent une part de narration, parce que le joueur va toujours essayer d'extraire un sens, et par extension une histoire, des graphismes.

Le fait est qu'il existe des techniques narratives extrêmement variées, qui ne nécessitent pas les mêmes moyens. Vous n'avez qu'à regarder l'épée pendant quelques secondes pour comprendre qu'elle vient de servir à tuer quelqu'un et qu'elle a passé quelques moments difficiles.

Mais considérez maintenant notre exemple des deux personnages débattant sur leur prochaine destination. Avant que nous ne puissions comprendre comment, pourquoi et ce dont ils discutent, ou même la nature de leur relation, nous devons les observer pendant un plus long moment et ne pouvons interrompre leur échange prématurément. Le type de narration utilisé impose certaines contraintes. Si ces exigences ne sont pas satisfaites, l'élément narratif ne peut pas être correctement transmis. C'est pourquoi la plupart des jeux ne vous laissent pas tuer les personnages essentiels au développement de certaines intrigues. Si nous tuons l'un des deux personnages qui était juste en train de commencer le débat, nous n'apprendrons jamais pourquoi ils étaient en train de discuter. Nous venons de détruire le mode de communication de la narration au joueur qu'était la conversation de tiers.

Puisque l'idée de narration est tellement vaste, nous devons, pour les desseins de cette discussion, être très précis sur le genre de narration dont nous parlons.

Il est clair que le jeu vidéo en est resté longtemps au stade d'une combinaison rudimentaire de gameplay et de narration visuelle basique. Quand je joue à *Gradius*, il y a une narration même s'il n'y pas d'histoire à proprement parler. Je contrôle un vaisseau, je suis dans l'espace et je tire sur des méchants. Je vais tirer sur tous les méchants que je vois jusqu'à que j'arrive au plus méchant d'entre eux, puis je vais tirer sur lui aussi. Ce n'est pas l'intrigue la plus intéressante qui soit, mais elle existe. Ainsi, le projet des concepteurs n'est pas de combiner simplement gameplay et narration visuelle : on le fait depuis qu'il est possible de dessiner quelque chose qui ressemble plus à une lame aliasée qu'à une ligne de pixels blancs.

Ce projet consiste en réalité à combiner élégamment et sans compromis les évolutions les plus riches du gameplay et de la narration. Dans la suite, quand nous parlerons de narration, nous entendrons par ce terme le genre de narration plus substantiel qu'un sprite ou un background graphique, plus lent à digérer – le fil d'une conversation, une histoire qui implique des personnages, leurs relations et leurs dispositions les uns à l'égard des autres, etc.

Intéressons nous maintenant jusqu'à où nous ont permis d'aller certaines des meilleures expériences de combinaison entre gameplay et narration.

Comment une narration sans restriction affecte-t-elle le gameplay ?

En 1983, *Dragon's Lair* était l'une des premières et plus célèbres tentatives de réconciliation entre narration cinématique et gameplay. Grâce à l'utilisation d'un lecteur à disque laser, ce jeu d'arcade proposait des graphismes incroyables, qui sont restés inégalés pendant près d'une décennie. Malheureusement, les limites de la technologie ont ici imposé que le gameplay soit en contrepartie réduit à un niveau beaucoup plus bas que celui des autres jeux de l'époque : une séquence cinématique non interactive se joue et, à la fin, le joueur doit presser quelques boutons – que ce soit une direction ou une « attaque ». S'il presse le bon bouton, le joueur assiste à une animation de mort. S'il presse le mauvais bouton, le joueur assiste à la séquence cinématique suivante. Comme le jeu était conçu avec peu de considération pour la lisibilité de l'action par le joueur, ce dernier était très souvent obligé de faire des suppositions : réussir le jeu passait par un nombre d'essais et de tâtonnements épouvantables.

Seize ans plus tard, *Shenmue* sort sur Dreamcast et propose un système de jeu spécial appelé « Quick Time Event » ou QTE. Pendant un QTE, les boutons apparaissent à l'image à différents intervalles. Si vous pressez le bouton correspondant suffisamment rapidement sur la manette, votre personnage exécute l'action correspondante. C'est structurellement identique à *Dragon's Lair*, excepté ces signaux explicites apparaissant à l'écran. Qu'y a-t-il à gagner en créant un jeu de cette manière ? Selon les principes d'un « bon » créateur, une attention particulière devrait être portée à la présentation de l'action, de façon à ce que le joueur puisse voir ce qu'il fait et ce que l'ennemi fait et ainsi à appréhender précisément les relations spatiales entre tous les éléments participant au gameplay dans la scène.

Avec cette abstraction extrême du gameplay, le créateur prend la liberté d'exposer impunément l'action aussi cinématiquement que possible, parce que l'action est devenue tellement abstraite que le joueur n'a pas besoin de savoir exactement ce que son avatar est en train de faire, ce que l'ennemi est en train de faire, ou comment un changement dans les règles dans cette séquence pourrait avoir un impact sur l'action dans une autre partie du jeu. En fait, nous avons rejeté toutes les difficiles considérations de création que nous aurions eu à gérer avec un gameplay conventionnel.

Dans le même temps, on sacrifie ici quelque chose de précieux en contrepartie cette ultime abstraction du contrôle. Comme nous nous sommes débarrassé de toutes les difficultés liées au gameplay, nous avons, nécessairement, tellement découplé le gameplay et l'action que le joueur n'a en réalité pas besoin de faire attention à quoique ce soit d'autre qu'à l'icône lui indiquant le bouton à presser. Si on mettait simplement un écran noir, à la place de la scène cinématique, la nature de l'interaction resterait inchangée. *God of War*, sorti six ans après *Shenmue*, emploie un système virtuellement identique lors de certaines phases de jeu. Alors que la qualité cinématique de ces séquences est stupéfiante, et est sans aucun doute très loin au dessus de ce que notre précédent exemple pouvait proposer, il s'agit exactement du même type de dissociation de l'action et de la cinématique.

Gitaroo Man, sorti en 2002, offrait une narration cinématique de haut vol en tant que toile de fond, opérant de concert avec un gameplay rythme/action sophistiqué au premier plan. C'est un exemple intéressant car même si le gameplay est très connecté à la narration, les deux étaient vraiment dissociés, dans le sens où la narration ne contenait pas d'information qui vous aurait pu vous aider à jouer au jeu efficacement – comme dans le précédent exemple où l'interaction n'était pas affectée par quoique ce soit provenant de l'arrière plan. Cependant, l'arrière plan était affecté par le gameplay, et cela résulte en une très belle illusion qui créa deux couches indépendantes perçues comme une seule. Le modèle de gameplay était scripté pour correspondre à la musique, et l'arrière plan narratif était aussi scripté pour coïncider avec la piste sonore. Le personnage du joueur et les ennemis posaient et agissaient selon vos commandes, bien que vous n'aviez pas besoin du tout de les remarquer ou de les observer pour jouer au jeu. Le créateur du jeu *DS Elite Beat Agents* a utilisé beaucoup de techniques identiques.

Ce que nous pouvons apprendre de ces exemples, au moins avec cette technique, est que le créateur gagne la liberté de faire tout ce qu'il veut avec la caméra de son jeu, et ainsi l'action d'arrière plan, au prix de sévèrement confiner la qualité du gameplay aux interactions du type « Appuie là, vite ! ». En d'autres termes, quand nous autorisons l'expression narrative sous forme de cinématiques illimitées durant le gameplay, ce dernier doit lui-même être sévèrement limité ou découplé de la narration.

Comment la narration restreinte affecte-t-elle le gameplay ?

Supposons que nous avons une narration que nous voulons transmettre, et nous voulons le faire pendant le gameplay, mais nous décidons de restreindre les moyens que nous avons pour l'exprimer – par exemple en la limitant à un seul type de médium. Examinons maintenant différents chemins que les créateurs de jeux ont empruntés dans ce type de situation.

Dans *Bioshock*, le support narratif est fréquemment limité à un canal sonore, sous la forme de journaux audio que le joueur prend et écoute s'il le souhaite tout au long du jeu. Quand la narration est limitée à l'audio, cela signifie que toutes les autres canaux disponibles restent ouverts pour participer au gameplay. Comme le langage visuel est sans conteste la plus importante méthode pour faciliter le gameplay, cela signifie que le joueur peut appréhender la narration auditive sans grande interférence ou limitation sur l'action du jeu. Aussi longtemps que l'audio de la narration n'interfère pas avec n'importe quel signal audio pouvant être crucial pour le gameplay, narration et gameplay peuvent vivre assez bien côte à côte. Cependant, nous trouvons un important compromis même dans cet échange. Vous ne pouvez exprimer les mêmes choses de la même manière, purement auditive, comme vous le pourriez si vous aviez aussi le contrôle de la dimension visuelle. Vous avez seulement le volume, la durée et le ton pour exprimer la qualité d'une explosion. Vous ne pouvez pas décrire un personnage escalader la tête d'une hydre, excepté par la narration littérale (« Je suis en train d'escalader la tête de l'hydre ! »). Tous ces types de détails ne peuvent pas être adéquatement exprimés quand la narration se limite à l'audio.

Dans quelques uns des jeux *Ace Combat*, la narration est transmise au joueur via les communications radio provenant d'un radar spécial d'aéronef qui est essentiellement détaché des actions de gameplay, ainsi qu'il l'est des avions amis et ennemis engageant directement le joueur.

Ce cas est intéressant car le jeu assouplit aussi l'idée de la manière avec laquelle les radios communications travaillent pour présenter une narration plus convaincante. En dehors des amis, ennemis ou sources radios, vous avez à entendre les communications de tous les personnages, tout le temps. Contrairement à *Bioshock*, la narration auditive prend place dans le présent au lieu du passé. Dans *Call of Duty 4 : Modern Warfare*, la narration prend la forme d'un PNJ soldat qui opère à vos côtés, vous donne des instructions, ou vous avertit des obstacles auxquels vous allez faire face. En raison de cela, la narration est reliée au gameplay plus fortement que dans *Bioshock* ou *Ace Combat*, étant donné que nous avons maintenant un visuel, une source physique de narration à proximité s'exécutant de concert avec le gameplay.

Même dans cette situation, notre contrôle sur la narration est limité. Parce que le joueur a un contrôle total sur la caméra, nous ne pouvons voir le moindre travail de caméra comme une partie de l'expression narrative. Quand vous êtes en train de charger vos ennemis, vous n'avez pas beaucoup de temps pour regarder ce que fait votre collègue soldat. *Call of Duty 4* met l'accent sur ce que dit le partenaire quand il y a des ennemis face à vous, et met l'accent sur le canal visuel lorsqu'il n'y en a pas.

De même, *Half-Life 2* permet au joueur de rester dans la perspective et le contrôle du jeu pendant les scènes cinématiques, au lieu d'arrêter la caméra. Il y a deux compromis non évidents ici. Premièrement, le développeur doit précautionneusement amener le joueur à regarder vers la direction attendue pour qu'il puisse voir toute l'action. Cela peut être accompli en limitant la vue par l'intérêt du design inhérent à l'espace physique (si vous êtes à une entrée, il n'y a que deux directions intéressantes auquel regarder, et le joueur va avoir tendance à favoriser la direction dans laquelle il est déjà). D'autre part, le jeu doit établir un « code de confiance » : on ne jette pas un ennemi fou dans le dos du joueur pour le poignarder, alors qu'il regardait deux autres personnages converser. Même si l'interface est inchangée des scènes de gameplay aux scènes narratives, le joueur a besoin d'une indication ainsi que de cette cohérence pour « savoir » que son attention doit se porter sur la conversation au lieu d'aller aux quatre coins de la pièce pour chercher objets et ennemis. Dans chaque cas, vous ne pouvez jamais garantir que le joueur va vraiment regarder ce qu'il est supposé voir.

Dans ces exemples, on peut remarquer que, quand l'expression narrative est restreinte d'une certaine manière, le gameplay peut se développer de manière plus riche.

Dans quelle mesure narration et gameplay sont-ils en concurrence ?

Reprenons le dernier exemple, où je décrivais l'intérêt d'établir un code de confiance dans les situations narratives lorsque le joueur y a un certain contrôle. Il y a aussi un intérêt à la duplicité et la surprise narrative. Imaginons à présent que nous voulions vraiment que le joueur expérimente l'effet de regarder confortablement deux personnages discuter l'un avec l'autre, avant de sentir une décharge de surprise quand quelque chose l'attaque soudainement par derrière. Dans les films, c'est une technique totalement valide et fréquemment utilisée. Dans les jeux, nous avons besoin de procéder plus prudemment. Mais pourquoi ? Dans un film, le public n'a aucun contrôle sur ce qui se passe. Le réalisateur du film, s'il le souhaite, peut savourer la liberté de poignarder le spectateur dans le dos encore et encore, impunément. Le spectateur est entraîné à craindre et s'attend à la possibilité d'un coup en traître, et il ne peut rien faire pour l'empêcher.

Considérons maintenant ce qui se passe dans un jeu. Nous plaçons le joueur dans une scène cinématique où il est totalement en contrôle, où il semble n'y avoir aucun danger et où le seul élément intéressant est deux personnages en train de parler. Le joueur a sa garde baissée, et assiste à la conversation. Sans avertissement, nous le surprenons en le plaçant sans prévenir dans une situation possiblement mortelle.

Mission accomplie pour le développeur, mais nous avons maintenant un joueur entraîné à rester vigilant, et cette attente a des conséquences désastreuses, parce que le joueur garde le contrôle pendant la narration. La prochaine fois que le joueur assistera à une scène cinématique, son attention sera focalisée sur l'éventuel ennemi caché dans le décor et non sur la narration.

Tout espoir que le joueur fasse attention à la narration s'est évaporé. Le prix de cet effet de surprise sur le joueur est la perte de sa confiance, et cette crainte diminuera pas à moins que le joueur assiste à une longue succession de scènes cinématiques sans incident.

La plupart des jeux manient ce type de scénario en travaillant sur la duplicité de telle manière que c'en est inévitable – sans conséquence en termes de gameplay – et donc restreint pour la narration. De cette manière, le joueur est surpris, mais ensuite réalise : « Oh, j'étais censé mourir. ». Il place alors l'action dans la poubelle mentale des « Choses qui arrivent dans la scène cinématique » ou celle des « Trucs qui arrivent dans le jeu ». Il est intéressant de voir que cet effet travaille complètement contre l'intention de brouiller ou de supprimer la frontière entre interaction et narration. Mais c'est ici moins notre « inaptitude » en tant que créateurs qui est en jeu qu'un effet indissociable de « l'entraînement » que subit le joueur.

Supposons à présent que vous n'aimez pas être frappé au visage. Si vous êtes sur le point de voir l'hypothétique film *John Frappe Daniel dans la Figure 80 Fois*, le réalisateur peut dire que Daniel ne va pas se défendre, même contre la volonté du spectateur, parce que l'intention est de faire en sorte que le spectateur sache ce que ça fait de se faire frapper au visage 80 fois successivement. Mais dans le jeu *John Frappe Daniel dans la Figure 80 Fois*, où vous jouez Daniel et où vous avez une barre de vie limitée (et où vous pourriez être amené à subir quelques désagréments en termes de jeu si vous recevez tout ces coups), vous allez sans aucun doute faire tout ce que vous pouvez pour éviter de vous faire frapper. La seule manière pour nous de vous faire sentir la succession de 80 poings est de vous « geler », d'une certaine manière. Mais si vous êtes gelés, nous sommes alors en train de regarder un film, et non de jouer à un jeu.

Il existe des actions que vous ne pouvez faire subir à une personne quand elle a le contrôle de la situation. Dans notre contexte, cela veut dire qu'il y a des choses que vous pouvez faire dans la narration mais pas dans le gameplay. La prochaine question logique serait donc : « Y a-t-il des choses qui fonctionnent dans le gameplay et sont impossibles à accomplir dans la narration ? »

Sans surprise, après une petite enquête, nous apprenons que la réponse à cette question est « Oui ».

Dans *Why We Play Games*, Nicole Lazzaro décrit huit catégories d'émotion qui sont communément expérimentées dans les jeux. Quelques unes, comme le Dégoût ou la Peur, peuvent être expérimentées à intensité égale si le médium est le gameplay ou le cinéma. Cependant, il y en a quelques unes qui sont vraiment exclusives au domaine du gameplay, car elles impliquent un effort comme pré-requis nécessaire.

Nous savons déjà que la narration stricte impose que l'observateur n'ait rien à dire sur le résultat des événements. Comme nous l'avons montré, cette limitation n'est pas négative par nature, vu que le thriller *John Frappe Daniel dans la Figure 80 Fois* ne pourrait exister sans ça. Ce seul élément distinctif a ainsi d'importantes implications – pas seulement que vous ne puissiez contrôler le résultat, mais aussi que vous n'êtes pas autorisés à exercer un effort significatif, dans n'importe quelle mesure. En effet, vous n'avez pas le moindre moyen d'exercer un effort dans le monde de la narration, même si vous le vouliez. Qu'est-ce que cela signifie ? Deux émotions propres au gameplay, « Kvell » et « Fiero », peuvent seulement exister comme possible conséquence d'un effort de la part du joueur – effort qui ne peut jamais être accompli dans une narration pure.

Kvell est un verbe, mais aussi un sentiment. C'est un sentiment de fierté, mais pas celui que vous pourriez ressentir quand vos enfants réussissent quelque chose. Vous ne ressentez pas Kvell quand vos enfants jouent aux jeux vidéo, mais quand vous êtes fier d'avoir appris à votre meilleur ami un combo secret de *Street Fighter* qu'il a utilisé pour remporter un tournoi, c'est Kvell.

Après avoir vu M. Miyagi former Daniel durant *The Karate Kid*, nous nous sentons bien lorsque celui-ci gagne finalement le combat contre le Cobra Kai au tournoi de karaté, mais nous ne pouvons sentir Kvell au-delà de ça, car nous ne l'avons pas formé nous-même. Kvell peut ainsi être vécu dans les jeux où le joueur a le contrôle sur l'entraînement, l'instruction et la culture de ses subalternes – comme dans *Oddworld* et *Pokémon*.

Fiero, d'un autre côté, est le sentiment spécifique au succès après que quelqu'un ait entrepris un effort significatif pour accomplir quelque chose. Quand vous battez un boss difficile après plusieurs défaites ou comprenez finalement un puzzle provocateur, ce sentiment de victoire que vous ressentez est Fiero. Les drogués de Fiero adorent jouer à des jeux difficiles et extrêmement fastidieux, parce que tout est question de récompense pour eux. La satisfaction intense gagnée à surmonter les challenges dans ce genre de jeux l'emporte de loin sur n'importe quelle torture que le game designer déchaîne (involontairement) contre eux.

Pour revenir à l'exemple de *Karate Kid*, nous ressentons certainement de la joie ou du soulagement quand Daniel gagne le tournoi. Mais nous n'avons fait aucun effort pour nous-mêmes gagner un tournoi, donc la signification de victoire n'est pas la même pour nous que pour Daniel-san. La narration peut décrire l'expression de Fiero de Daniel-san, mais ne peut pas nous faire ressentir cette émotion de la même manière. La seule chance que vous avez de faire ressentir au joueur un tel sentiment est à travers le gameplay.

Comment le cinéma perturbe-t-il la continuité pour améliorer la narration et dans quelle mesure la continuité est-elle essentielle au gameplay ?

Un autre terrain où gameplay et narration entrent régulièrement en conflit est celui de la continuité et de la précision de l'expression de l'action. Le rapport à la continuité est un élément essentiel pour comprendre ce qui sépare narration et gameplay.

Dans le gameplay, la continuité des événements est très importante, alors que dans la narration, ce n'est pas le cas. En fait, dans la narration, la continuité est sacrifiée au profit de l'expression de l'histoire. C'est pour

ça que les monteurs de film créent régulièrement des séquences d'événements impossibles afin de rendre l'histoire plus fascinante.

Nous ne sommes pas en train de parler « d'impossibles » comme d'irréalisables ou de fantastiques, comme esquiver une balle, mais « d'impossibles » dans le sens de la continuité et de la compression temporelle, comme nous allons le voir rapidement. En tant que spectateurs, nous acceptons systématiquement ceci comme valeur logique parce qu'après des années et des années d'exposition aux films, nous avons acquis une compréhension intuitive de la grammaire de la caméra comme support de la narration, de même que sa manière de remonter et sauter dans le temps et l'espace.

A l'inverse, dans le gameplay, si la continuité n'est pas exprimée d'une manière parfaitement précise, cela interfère avec la capacité du joueur à prendre des décisions ou à comprendre une situation.

Analysons l'exemple d'un combat en un contre un. Dans *Rocky IV*, le combat entre Rocky et Drago dure quinze longs et éprouvants rounds. Pendant le déroulement du combat, nous voyons d'abord Drago dominer totalement Rocky, fidèle à sa réputation de machine invincible. Petit à petit, Rocky revient, et le match devient de plus en plus tendu, comme la vague s'abat d'arrière en avant, encore et encore, jusqu'à ce que Drago cède. Il y a là quelques éléments intéressants, et cela vaut la peine d'étudier la séquence entière si vous y avez accès (essayez de chercher « Rocky vs Drago » sur Youtube).

Premièrement, examinons l'action de la caméra. Le match commence avec une caméra en quasi « live » pour les premiers rounds, où l'attention est principalement focalisée sur Rocky et Drago. Pendant les dix rounds du milieu au minimum, la séquence devient un montage. Le temps est compressé, alors que nous voyons les images de chaque round voler à travers l'écran non pas par minutes, mais plutôt par secondes. Nous voyons la caméra se focaliser et faire ainsi disparaître les personnages dans l'audience, on voit quelques enfants regarder la télé, juste au moment où nous pouvons voir leurs réactions. La caméra semble réduire son angle de vue (même pendant les moments non montés) pour se concentrer uniquement sur le combat jusqu'à l'instant exact où le poing de Rocky riposte. Nous voyons également des plans de caméra doubles qui sont séparés par le temps et l'espace où les boxeurs se battent sur la moitié droite de l'écran et où Rocky ou Drago est assis dans le coin du ring entre deux rounds.

Deuxièmement, examinons l'action des boxeurs. Nous n'avons pas une perception précise de leur état physique actuel ou de leur performance, en dehors de ce que le film nous raconte explicitement pour les effets dramatiques. Nous ne savons pas si les combattants sont proches de l'épuisement et l'histoire prend en fait l'avantage sur nous en suggérant que le match est sur le point de se finir pour de bon, juste avant de montrer Rocky revenir au combat avec une force surhumaine.

Il n'y a pas d'intérêt à savoir combien de directs se prend Rocky dans la figure et à quel point la blessure de Drago peut être douloureuse. Ce qui importe, c'est le contenu émotionnel de ces éléments, et c'est pourquoi ils sont utilisés dans le film. Considérez que beaucoup de plans de caméra montrant des poings échangés d'avant en arrière pouvaient être complètement remplacés par d'autres plans sans changer d'un poil la perception de la séquence.

Toutes ces techniques rendent la narration plus convaincante, et elles impliquent toutes de trancher et de couper l'ordre actuel des événements (s'ils ont jamais existé) au profit du rythme. Il n'y a là rien d'extraordinaire en terme de cinéma, mais la clé est que toutes ces techniques sont possibles seulement parce que le « joueur » n'a pas le contrôle sur ce qui se passe.

Voir la caméra coupée pendant quelques instants et perdre le contact visuel avec un opposant lors d'un combat est inacceptable dans un contexte interactif (Rocky et Drago ne se reposent certainement pas furtivement quand la caméra n'est pas sur eux).

Remplacer une série de poings par une autre fonctionne parce que personne ne garde exactement l'œil sur la manière dont les combattants sont blessés ou sur la façon dont cela affecte leur résistance au cours du match. Le public ne connaît pas (ou ne prête pas attention à) ce niveau atomique de détail, ce qui permet au

réalisateur d'en profiter. Mais dans le gameplay, le joueur fait attention, parce que chaque événement atomique a une conséquence définie en accord avec les règles du jeu.

Reprenons cet exemple du match de boxe, mais dans un contexte de gameplay. Examinons maintenant un match typique dans un jeu en un contre un comme *Street Fighter IV*. Nous ne pouvons couper vers une foule au milieu d'un match pour accroître l'excitation, parce que le joueur a un contrôle total sur son personnage. Au mieux, une telle tentative serait perturbatrice. Nous ne pouvons pas stylistiquement faire une « avance rapide » jusqu'à la fin du match, parce que le joueur crée littéralement la continuité du match à mesure qu'il se déroule. Et plus que tout, nous faisons attention à garder la trace du déroulement de chaque réussite et échec, même ceux qui n'ont pas d'importance dramatique, car les relations spatiales précises entre les personnages qui déterminent qui a frappé et car chaque personnage ne peut supporter qu'un certain nombre de coups avant de succomber.

Paradoxalement, les techniques qui rendent le cinéma fascinant brisent le gameplay, et les éléments essentiels au gameplay sont sans importance dans le cinéma.

Conclusion

L'objectif de cet article n'était pas d'encourager les puritains du gameplay qui semblent détester les histoires dans leurs jeux, ni de donner totalement raison à ceux qui voient le jeu vidéo comme un magnifique médium pour narrer des chroniques. Si vous vous trouvez dans l'un de ces camps, et même si vous ne vous y trouvez pas, j'espère avoir été capable de montrer que narration et gameplay ne sont pas comme des ingrédients qu'on cherche à mélanger en trouvant la température et le dosage idéaux, mais comme des outils différents qu'ils convient d'utiliser au bon moment.

Il y a certaines choses que les deux médias peuvent accomplir, mais il y a aussi beaucoup de choses qui sont spécifiques de la narration ou du gameplay. Comme pour n'importe quel outil spécialisé, c'est l'instrument adapté qui devrait être sélectionné pour le travail. Choisir le mauvais outil peut être maladroit au mieux, catastrophique au pire.

N'utilisez pas seulement la narration et le gameplay dans vos productions en suivant les conventions typiques. L'idée d'utiliser les deux simultanément avec un espoir aveugle que le résultat sera meilleur d'une certaine manière que de n'avoir utilisé que l'un ou l'autre est tout aussi absurde. Soyez élégants dans la manière avec laquelle vous sélectionnez et articulez le médium qui convient le mieux pour transmettre ce que vous voulez au joueur.

Dans cette perspective, l'idée de faire pleurer un joueur n'est pas une sorte de Saint-Graal mystique : cela se fait de la même manière que les films (tout comme les jeux narratifs) l'ont fait pendant toutes ces années. D'un autre côté, si vous essayez de choquer le joueur uniquement à travers le gameplay, vous aurez certainement autant de difficulté que d'essayer de faire ressentir Fiero à un joueur en lui faisant regarder un film.

Introduction à Photoshop

Par [Pierre Guyot](#) | le 11 juin 2011 20:02:25 | Catégories : [Cours](#), [Graphisme](#), [Photoshop](#)

L'article qui va suivre est une introduction à Photoshop et présente un certain nombre d'outils, dont les calques (leur fonctionnement est similaire dans des logiciels comme Painter ou PaintShopPro). En espérant qu'il sera utile à ceux qui se lancent dans le graphisme :]



Photoshop

Quand vous ouvrez Photoshop, à quoi ça ressemble ?

Un grand fond gris, une barre d'options et plein de fenêtres mobiles. Je vous conseille de les fermer et de ne garder que celles qui vous intéressent (pour ouvrir une fenêtre, il suffit de cocher son nom dans le menu Fenêtres). Je vous conseille de garder uniquement la boîte à outils et la fenêtre des calques, du moins pour commencer.

La boîte à outils

La boîte à outils contient les habituels pinceaux, pot de peinture, sélection, etc. ainsi que deux petites cases pour choisir vos couleurs. Faites attention cependant, chaque icône correspond à plusieurs outils ! Pour changer l'outil sur une icône, faites Alt+Clic dessus ou utilisez le clic droit. Ainsi, le pot de peinture et le dégradé ont la même icône. Pour sélectionner une couleur, cliquer sur la case où vous voulez la mettre pour faire apparaître un menu de réglage.

Quand un outil est sélectionné, les options de réglage s'affichent en haut. Le clic gauche permet de l'utiliser. Le clic droit ouvre un menu qui dépend de l'outil. Le plus utile : pour le pinceau, il ouvre la liste des brushes, c'est-à-dire des types de pinceaux et propose une jauge pour régler la taille du pinceau.

Pour régler plus précisément les brushes, vous pouvez utiliser le menu approprié en appuyant sur F5. Vous aurez alors accès à toutes les options de réglage possibles et imaginables : opacité, taille, dispersion, texture, etc. Je ne les présenterai pas ici mais la plupart des options sont explicites. Les autres nécessiteront un peu d'expérimentation de votre part :]

Petite astuce pour tracer des traits : maintenir Maj permet de tracer des traits horizontaux ou verticaux. Si vous placez un point, puis maintenez la touche Maj enfoncée et cliquez à un deuxième endroit, Photoshop trace automatiquement le segment qui relie les deux points. Très utile pour les bâtiments ou le mecha design par exemple.

Il existe des raccourcis pour sélectionner un outil sans passer par la boîte à outils. Pour les voir, laissez votre curseur sur l'icône d'un outil. S'il y en a un, il sera indiqué entre parenthèse. En voici quelques uns :

- **B (Brush)** : le pinceau ou le crayon de base
- **E (Eraser)** : la gomme
- **M** : la sélection
- **Z (Zoom)** : la loupe (vous pouvez utiliser Ctrl++ et Ctrl+- me souffle Fil)
- **P** : la plume
- **W (Wand)** : la baguette magique
- **L (Lasso)** : le lasso

La fenêtre des calques

La fenêtre des calques permet, comme son nom l'indique, de gérer les calques. Mais qu'est-ce qu'un calque ?

L'intérêt principal d'un logiciel comme Photoshop par rapport à Paint, c'est qu'il permet d'utiliser ce qu'on appelle des calques : on peut en effet séparer le dessin en plusieurs « couches » à sa guise, les fameux calques, qu'on peut gérer simplement à l'aide du petit menu Calques. Par exemple, pour plus de commodité, je peux choisir de séparer le lineart et la couleur pour travailler sur une illustration.

La fenêtre des calques montre la liste des calques par ordre d'affichage (le premier de la liste est le calque le plus « au-dessus » du dessin. A gauche de chaque calque, une petite case à cocher avec un œil dessus permet d'afficher ou de masquer les calques. En haut, on peut régler l'opacité du calque et choisir son mode — nous verrons ce dont il s'agit dans la suite de l'article. Pour changer l'ordre des calques, il suffit de faire un cliquer-glisser et de déposer le calque là où vous le souhaitez.

En bas de la fenêtre, plusieurs icônes sont disponibles. Nous allons en présenter trois.

- La poubelle permet de supprimer un calque : il suffit de faire glisser le calque dessus ou alors de cliquer dessus après avoir sélectionné le calque.
- La petite feuille de calque pliée permet de créer un nouveau calque.
- Le petit dossier permet de créer un groupe. C'est un dossier dans lequel vous pouvez ranger vos calques. Ceci est très utile quand vous travaillez sur des illustrations avec beaucoup de calques. Par exemple, vous pouvez créer un groupe Décor où vous mettrez tous les calques liés au fond de l'image. Il vous sera alors possible de réduire le groupe ou même de le masquer.

Dans les versions récentes de Photoshop, on peut sélectionner plusieurs calques en même temps, par exemple pour les déplacer ou leur appliquer une symétrie. Ctrl+Clic permet d'ajouter un calque à la liste des calques déjà sélectionnés.

On peut afficher et masquer la fenêtre des calques avec la touche F7.

Le fonctionnement des calques.

Une particularité des calques est ce qu'on appelle les modes : en modifiant le mode d'un calque, on va pouvoir modifier la façon dont il va se superposer avec les autres. Par exemple, le mode **Normal** est celui de base, où les couleurs se superposent simplement. Il en existe beaucoup et il vous faudra les tester pour comprendre leurs effets. Par exemple, les modes **Densité Linéaire** – et Incrustation sont extrêmement pratiques pour les effets lumineux. Un calque noir en mode **Couleur** au dessus d'une illustration vous permettra de la voir en niveaux de gris sans toucher à l'image elle-même.

Parlons maintenant du verrouillage. Mais d'abord, qu'est-ce ? Le verrouillage est une option pratique pour travailler proprement. Vous avez sans doute remarqué qu'après avoir posé vos à-plats de couleur, il est

difficile de peindre correctement dessus sans déborder, ce qui nécessitera plus tard un fastidieux détourage. Pour éviter ceci, sélectionnez le calque où se trouve vos à-plats dans le menu Calques et cochez la petite case représentant un damier (si vous laissez la souris dessus, vous pourrez voir le mot Verrouillage apparaître). Un petit cadenas apparaît à côté du nom du cadre. Ainsi, lorsque vous peindrez sur ce calque, la couleur ne sera modifiée que sur la zone déjà colorée et pas ailleurs. Si vous voulez par la suite peindre hors de l'aplaf, il vous suffit de déverrouillez le calque en re-cliquant sur l'icône en forme de damier.

Il existe d'autres formes de verrouillage : ce sont les autres petites icônes à côté du damier. A vous de les tester ! Le calque le plus bas (l'arrière-plan) est verrouillé par défaut

[Image récapitulative](#)

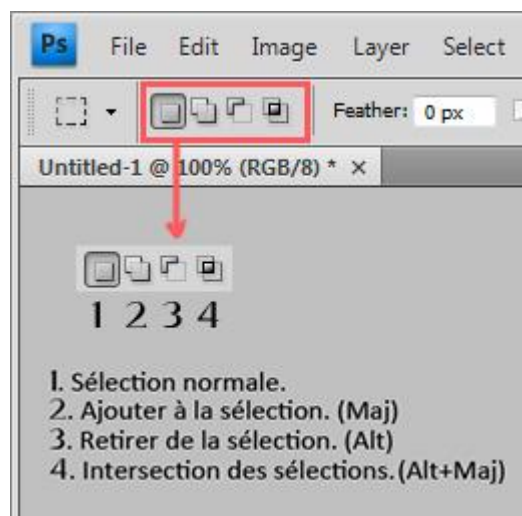
Masque et sélection

Un autre avantage de Photoshop est l'existence de sélections « intelligentes ». Contrairement à Paint, lorsque vous sélectionnez une zone de l'image, vous pouvez retravailler la zone de sélection.

La sélection (M) permet de sélectionner des formes simples : rectangles, ellipses, verticales, horizontales. Maintenir la touche Maj enfoncée permet de tracer des carrés si vous avez choisi une sélection rectangulaire, des cercles si vous avez choisi une sélection elliptique.

Le lasso (L) permet de sélectionner un tracé dont vous dessinez le contour. Le lasso magnétique repère les bords dans une image et vous permet de détourer facilement.

En haut à gauche, dans les options de la sélection ou du lasso, il y a quatre petites icônes :



- En maintenant Maj, vous pourrez ajouter de nouveaux éléments à la sélection.
- En maintenant Alt, vous pourrez retrancher de nouveaux éléments à la sélection.
- En maintenant Maj+Alt, vous pourrez récupérer l'intersection des deux sélections.

Avec ceci, vous devriez pouvoir récupérer des sélections même très complexes. Mais on peut faire encore plus élaboré : parlons maintenant des masques.

Regardez en bas de la boîte à outils, sous la sélection de couleurs : vous pouvez voir côte à côte deux rectangles avec un cercle au milieu, dont l'un est coché. Celui qui est coché correspond au mode Normal, l'autre au mode Masque. En mode normal, vous peignez comme d'habitude avec toute votre gamme de couleurs. En mode Masque, vous n'avez que du noir et du blanc et quand vous peignez, le noir apparaît en rouge : étrange me direz vous.

Mais à quoi cela sert-il donc ? En repassant du mode Masque au mode Normal, l'image que vous aviez dessinée en noir et blanc en mode masque va être transformée en une sélection. Mais pas n'importe laquelle : les pixels sont « plus ou moins sélectionnés » en fonction de la luminosité de votre dessin. Pour être plus clair, si le pixel était peint en blanc (ou rien du tout) en mode Masque, il est sélectionné à 100%, s'il était peint en noir (ce qui apparaîtra donc rouge) en mode Masque, il sera sélectionné à 0%. Et c'est là que ça devient intéressant : entre les deux, vous disposez d'une foultitude de nuances de gris. Si un pixel est gris à 50%, il sera sélectionné à 50% et donc si vous lui appliquez un effet, il le subira à 50%. Par exemple, si vous faites une sélection dans le mode Masque puis que vous faites Suppr dans le mode Normal, les pixels marqués de rouge pur (noir) ne seront pas touchés, les pixels sans marque (blanc) seront totalement supprimés et les pixels marqués de rouge intermédiaire (gris) seront plus ou moins supprimés. Ça peut paraître très compliqué dit comme ça, mais essayez directement dans Photoshop, c'est très simple.

Quelques raccourcis

Les classiques

- **Ctrl + N** : créer un nouveau fichier .psd
- **Ctrl + O** : ouvrir un fichier
- **Ctrl + S** : enregistrer
- **Ctrl + Maj + S** : enregistrer sous
- **Ctrl + X** : couper
- **Ctrl + C** : copier
- **Ctrl + V** : coller

L'affichage

- **Ctrl + U** : menu de réglage des couleurs
- **Ctrl + R** : afficher/masquer les repères sur les côtés
- **Ctrl + "** : activer/désactiver le magnétisme
- **Ctrl + '** : afficher/masquer la grille (vous pouvez la paramétrer en allant dans Editions, puis Préférences)
- **Tab** : afficher/masquer les fenêtres
- **F** : change le format de la fenêtre (en utilisant Tab et F, on peut réussir à ne faire apparaître que le dessin sans rien d'autre à l'écran :D)

Pour travailler

- **Ctrl + Maj + U** : désaturation de l'image
- **Ctrl + D** : désélectionner (pensez y si tout d'un coup vous n'arrivez plus à peindre, c'est peut-être tout simplement parce que vous avez laissé une petite sélection quelque part)
- **Ctrl + E** : fusionner le calque sélectionné avec le calque de dessous
- **Ctrl + Maj + E** : fusionner tous les calques visibles
- **Ctrl + T** : mode transformation manuelle (pour étirer l'image, la pivoter, la réduire, l'agrandir, etc.)
- **Ctrl + Maj + C** : copier en une image tout ce qui est visible à l'écran (si vous vous contentez d'un classique Ctrl+C, Photoshop copiera seulement le contenu du calque sur lequel vous êtes)
- **Alt + clic** : utiliser la deuxième fonction de l'outil (extrêmement utile pour le shading avec le pinceau et le crayon, car il permet d'utiliser très rapidement la pipette et de sélectionner les teintes intermédiaires d'un dégradé)

Mention spéciale, l'historique

- **Ctrl + Z** : switcher entre l'état actuel et l'état antérieur.
Attention, ce n'est pas le Ctrl+Z habituel. Imaginons que vous êtes à l'état n . Faire Ctrl+Z une fois vous enverra à l'état $n-1$, faire Ctrl+Z une deuxième fois ne vous emmènera pas à l'état $n-2$ mais vous ramènera à

l'état n . Une troisième fois vous renvoie sur l'état $n-1$, et ainsi de suite. Une fois de plus, c'est difficile à expliquer, mais très simple à comprendre : essayez par vous-mêmes :]

- **Ctrl + Alt + Z** : retour à l'étape précédente (le classique « Undo »)
- **Ctrl + Maj + Z** : rétablir une action

Ceci étant dit, je vous souhaite bon courage pour la suite :]

Costumes et character design

Par [Pierre Guyot](#) | le 11 juin 2011 23:22:52 | Catégories : [Character Design](#), [Cours](#), [Dessin](#), [Graphisme](#)

Le costume est un élément particulièrement important du character design. Plus encore qu'à son visage, c'est à ses vêtements et son allure qu'on reconnaît un personnage. Dans un jeu en pixel art par exemple, on ne peut pas, la plupart du temps, reconnaître un visage, pour des raisons de résolution.



L'article qui suit vise à développer quelques points concernant le design de costumes. Je tiens à préciser qu'il ne s'agit pas d'une « méthode » à suivre, mais juste d'une réflexion.

Qu'est-ce qu'un *design* ?

Avant de se lancer dans un design de costume, essayons de comprendre quel est le but que nous visons. Un design, étymologiquement, est à la fois un *dessin* et un *dessein*, c'est-à-dire une intention, un projet. Il ne s'agit donc pas ici de simplement dessiner un vêtement, mais de faire passer des informations au joueur ou au spectateur : le vêtement a pour fonction de *décrire* le personnage.

La première chose à faire lorsqu'on réfléchit au costume d'un personnage est de se rappeler qu'un costume fait passer un message : il définit l'origine géographique et sociale du personnage et il peut même informer — ou induire en erreur — le joueur sur sa personnalité.

Prenons un exemple connu. Ganondorf, le grand méchant de la série des *Zelda*, est un méchant et une brute épaisse, tout le monde le sait. Mais surtout, cela se *voit* immédiatement à ses vêtements : énorme armure de plaques métalliques, piques, lambeaux de tissus déchirés, chaussures à coques. C'est un Gerudo, comme en témoignent les symboles rouges et bleus sur sa tenue — en cohérence avec le reste de l'univers — et c'est un personnage d'un certain rang — à savoir un roi Gerudo — comme en témoignent les bijoux qu'il porte. Le choix des couleurs ne présage rien de bon non plus : du noir et du brun sombre, entrecoupés d'un blanc terne — auxquels s'ajoutent par ailleurs des cheveux rouge vif.



Au premier coup d'œil et sans même connaître le personnage, le joueur a déjà une grande quantité d'informations le concernant. C'est cela la fonction même du costume dans un jeu : il conditionne la façon dont le joueur va appréhender le personnage.

C'est pour cette raison qu'il vaut mieux commencer en se posant la question suivante : « Quelle impression ai-je envie que ce personnage donne au joueur ? » Pour réussir à communiquer autant d'informations, il faut commencer par se donner les éléments que l'on veut voir apparaître dans le design ou pour le dire autrement, établir le cahier des charges du costume.

Une fois les contraintes définies, il devient beaucoup plus simple de dessiner, car un costume se construit au moins autant dans sa tête qu'au bout de son crayon.

Messages

Intéressons nous maintenant aux types de messages que peut transmettre un costume.

Un costume permet de situer un personnage géographiquement et culturellement dans l'univers. Dans beaucoup de jeux et plus particulièrement les RPG, les vêtements varient d'une région à l'autre et certains lieux ont une culture d'autant plus marquante qu'elle rappelle une culture existante : Nasrad, dans *Skies of Arcadia* est conçue pour évoquer le Proche-Orient. En choisissant et en réservant certains motifs, certaines couleurs, certains tissus à une région déterminée du jeu, vous marquez plus fortement son identité culturelle. Ceci permet par exemple de reconnaître immédiatement un voyageur qui a quitté sa ville d'origine et d'approfondir l'univers du jeu.

Un costume permet de donner une idée du métier ou du « job » d'un personnage. Un soldat et un mage n'auront pas les mêmes vêtements. Il permet de plus de situer le personnage dans une hiérarchie : les vêtements d'un paysan n'auront pas le luxe d'étoffes et de bijoux de ceux d'un riche marchand. Du point de vue du gameplay, ceci permet au joueur de s'y retrouver plus facilement : s'il cherche un garde de la ville, il

le repère immédiatement à son armure, alors que s'il cherche un clochard, ce sont ses haillons qui permettront de le voir tout de suite.

Un costume exprime aussi l'importance d'un personnage. Les protagonistes de l'histoire méritent, en raison de leur importance dans le scénario, un design plus élaboré, plus complexe que celui des simples PNJ. C'est sur eux qu'il faut fournir le maximum du travail et c'est à eux qu'il faut réserver les accessoires et les détails. C'est le même principe qui fait qu'on réserve toujours plus de polygones pour les modèles 3D des personnages principaux.

Pour finir, un costume situe un personnage dans une époque. Des combinaisons cybernétiques agrémentées de lumières bleues caractérisent immédiatement un jeu situé dans le futur ou dans un univers de science-fiction, alors que des armures de métal ouvragées vous placent dès le départ dans un univers médiéval. A cet égard, il peut être intéressant d'étudier les vêtements de différentes époques, pour comprendre ce qui fait leur spécificité.

Références

Pour élaborer un costume, il est très important d'aller chercher des références de costumes existants et ce pour plusieurs raisons. Cela permet :

- de comprendre comment un vêtement est conçu et pourquoi on l'utilise.
- de dessiner de façon plus juste.
- d'enrichir son « vocabulaire graphique » et de donner plus de variété à ses personnages.
- de savoir quels vêtements étaient utilisés dans tel lieu ou à telle époque.

N'hésitez pas à puiser partout, même là où vous n'en auriez pas l'idée au premier abord : un bon designer se doit de ratisser très large. Allez voir le travail d'artistes que vous appréciez (ou même que vous n'appréciez pas, pour essayer de voir les écueils à éviter), fouiller les expositions ou les livres d'histoire, utilisez des références photographiques, regardez comment sont fabriqués vos propres vêtements, renseignez vous sur les usages et les techniques qui régissent la fabrication d'un vêtement, intéressez vous au domaine de la mode (au sens le plus large qui soit).

Je vous encourage personnellement à aller voir du côté de la mode, il y a du bon à prendre — après, tout ne l'est pas, mais un bon designer doit être capable de faire le tri. Ainsi, un designer comme Tetsuya Nomura — qu'on l'aime ou pas — a objectivement renouvelé le design du RPG en y introduisant des éléments tirés du monde de la mode : le blouson à col de fourrure de Squall, les bijoux comme Chronos, les ceintures de la robe de Lulu, la coupe à pics de Cloud, le design urbain de *The World Ends With You* inspiré du domaine du graffiti, etc.



Il s'agit d'un point que je ne soulignerai jamais assez, mais c'est ce qui permet de ne pas tourner en rond et de refaire tout le temps la même chose. Il y a trop de jeux ou de films où le design des costumes ont été vus et revus, en particulier dans les domaines de l'heroic fantasy et de la science-fiction.

Et je me permets d'ajouter au passage un [lien](#) vers un remarquable cours de [klindicative](#), ainsi qu'un [lien](#) vers le site My Armoury qui référence une foultitude d'épées et autres armes anciennes.

Cadre de travail

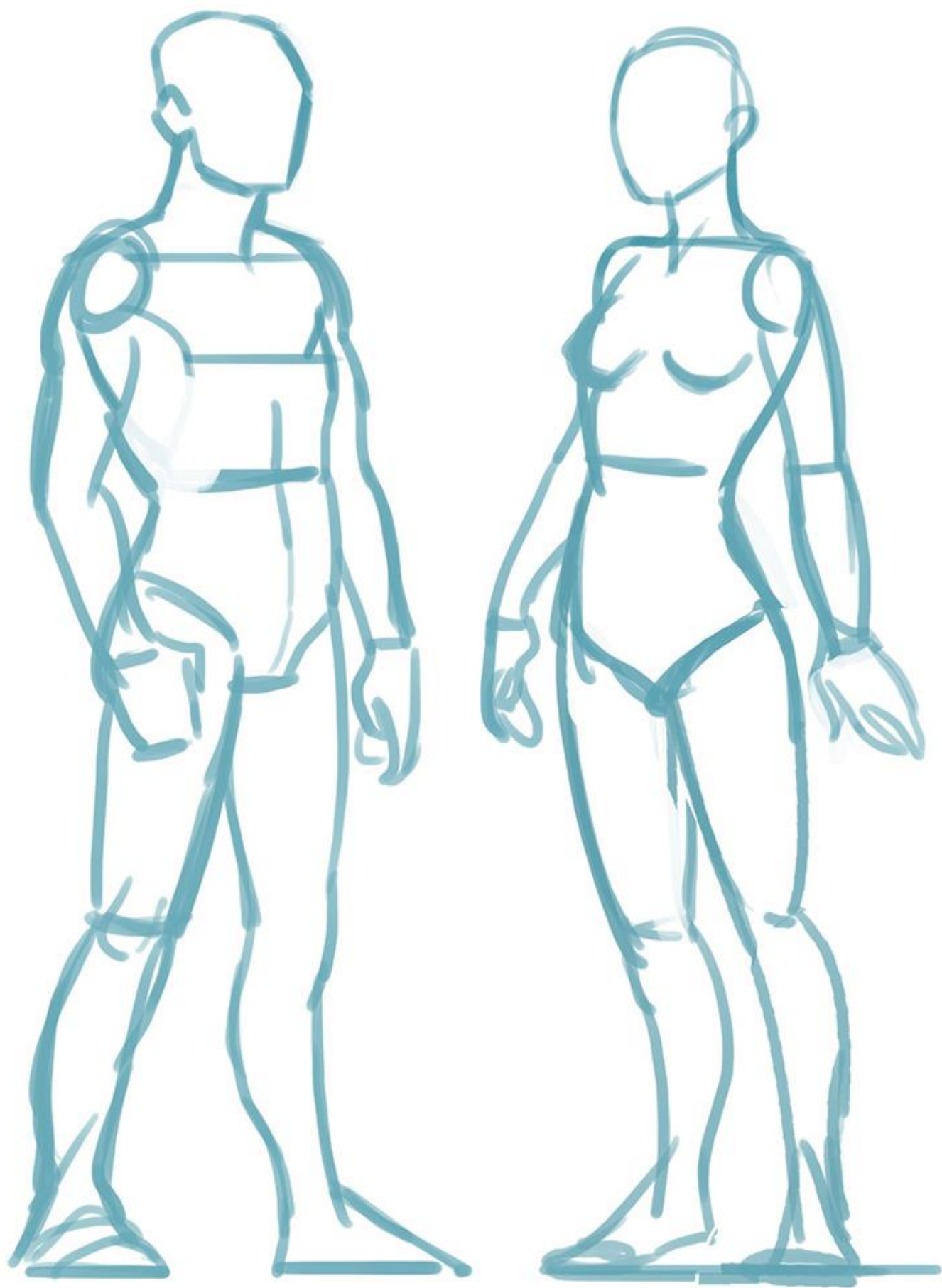
Il est maintenant essentiel de savoir comment vous comptez travailler.

Après avoir défini les concepts et les idées qui doivent apparaître chez le personnage à travers son costume, on peut commencer à faire des recherches. Pour commencer, il n'est pas nécessaire de se soucier d'anatomie ou de structure, il faut essayer de chercher le plus largement possible. Une fois des idées mises sur le papier, il est temps d'essayer de mettre ça au propre.

J'ai remarqué que beaucoup de gens dessinent directement les vêtements du personnage, sans dessiner son corps au préalable. Ceci peut être utile pour les premières recherches, afin d'aller vite et de ne pas trop se

restreindre tout de suite. Néanmoins, pour la suite, cela peut devenir très problématique. En matière de dessin pur, poser les vêtements dans le vide est une mauvaise technique, qui aboutit à des postures bancales ou des « tricheries » dans le dessin — des tissus immenses et mous qui masquent l'anatomie ou des mains cachées dans le dos. Généralement, ceci révèle immédiatement le (mauvais) niveau du dessinateur au spectateur.

Aussi, pour construire vos costumes, il vaut mieux systématiquement dessiner le corps, même très simplifié, puis dessiner les vêtements par-dessus. Ceci permet de mieux poser les tissus, d'avoir des personnages plus stables et de construire des drapés plus élégants. Voici des exemples de « mannequins » qu'on peut utiliser.



Une fois le mannequin d'un personnage dessiné, il suffit de le copier coller autant de fois que voulu et de tester différents costumes dessus, à partir des recherches préliminaires. Ceci évite d'avoir à redessiner le

personnage à chaque fois et permet de mieux comparer les différents costumes, sans être influencé par la pose : on ne s'intéresse ici qu'à la variété des vêtements et pas à celle des corps. En plus, ça vous fera travailler l'anatomie : que demande le peuple ? Ces [recherches](#) de [Matt Rhodes](#), character designer sur *Mass Effect 2* constituent un très bel exemple.

Faites attention à ce que le mannequin utilisé ait la carrure du personnage dont vous dessinez le costume : le dessin des vêtements n'est pas le même pour un homme et pour une femme, pour un adulte et pour un enfant, pour un nain et pour un géant.

Principes de design

Je ne prétends pas dresser ici une liste de lois immuables qui permettraient de parfaitement réussir un design. Il s'agit juste de remarques que j'ai pu me faire en dessinant et en discutant avec d'autres joueurs.

Less is more. Trois petits mots prononcés par l'architecte Mies Van der Rohe et qui ont bouleversé l'architecture et le design. Il est inutile d'en faire des tonnes et de surcharger un design. Un design extrêmement simple peut fonctionner à la perfection — les personnages d'Hayao Miyazaki en sont la plus parfaite illustration. Ce n'est pas en rajoutant des accessoires sur un mauvais costume qu'on le rend meilleur. Je rappellerai à cet effet cette citation de Chanel : « Le luxe, c'est ce qui ne se voit pas. »



Un bon design n'a pas besoin d'un bon dessin. Un bon design, une fois résumé en formes simples — par exemple en SD ou en chibi — doit encore fonctionner. Il n'y a pas besoin que le dessin soit particulièrement léché pour que le design marche. A l'inverse, un design catastrophique peut être particulièrement bien dessiné sans que cela ne lui apporte quoi que ce soit. Évitez la surcharge, car n'est pas Yoshitaka Amano qui veut. Ne vous souciez pas de bien dessiner : un bon design ne doit pas être bien dessiné, il doit être efficace et compréhensible (pour les gens qui réaliseront les sprites, les modèles 3D ou les animations plus tard).

Détailler à bon escient. Il est inutile de vouloir absolument tout détailler. Le détail n'est pas une qualité en soi : il sert à mettre en valeur certains éléments plutôt que d'autre, par contraste. Rappelez vous toujours qu'on ne met jamais de détail à un endroit gratuitement : si on détaille quelque chose, c'est qu'on veut lui donner de l'importance — c'est l'idée de « dessein », d'intention qui sous-tend la notion de design. Détailler minutieusement tous les éléments d'un design ne sert à rien. Ceci revient à donner la même importance à tous les éléments, à les mettre sur le même niveau : détailler de la même façon tous les éléments d'un design revient à ne pas détailler du tout. Les multiples bijoux et perles utilisés pour le costume de Jack Sparrow constituent un très bon exemple de « détails » utilisés à bon escient.

Jouer sur la symétrie. Dans l'ensemble, les vêtements sont symétriques par rapport à l'axe du corps, qu'il s'agisse de vrais vêtements ou d'un costume de personnage. Introduire un élément de dissymétrie va permettre de focaliser l'attention du joueur sur un détail particulier. Dans beaucoup de cas, les seuls personnages à posséder des éléments de dissymétrie dans leurs costumes sont les protagonistes de l'histoire.

Ceux-ci peuvent être très discrets (des bagues sur une main et pas sur l'autre) ou très voyants (par exemple les jambes du « pantalon » de Tidus dans *Final Fantasy X*). Il faut utiliser la dissymétrie avec soin pour donner de l'importance à ce que vous souhaitez, de la même façon qu'un accessoire.

Homogénéité. Dans l'ensemble, essayez de vous rappeler qu'il faut une certaine harmonie entre les personnages d'un même univers. On ne peut pas se contenter de juxtaposer des personnages complètement hétéroclites en espérant que le résultat sera harmonieux. Essayer de choisir des teintes dominantes, un style d'accessoires, des matières, des motifs qui reviennent régulièrement, afin de construire une impression d'ensemble. Celle-ci sera plus forte au sein d'un même groupe : par exemple, les habitants d'un même royaume devraient avoir un design assez proche les uns des autres. Au sein d'un même costume, on peut utiliser des « rappels » de motifs, de façon à ce que celui ne jure pas avec le reste — ou au contraire, le laisser seul pour le mettre bien en valeur. Dans l'ensemble, essayez d'avoir une palette de couleurs et de motifs restreinte, pour ne pas tomber dans la surcharge. Il faut veiller à toujours maintenir un équilibre entre variété et homogénéité.

Couleurs et valeurs. Essayez de regarder les couleurs du personnage sur une palette et de regarder le personnage en noir et blanc. La palette de couleurs doit pouvoir fonctionner toute seule, ainsi que le personnage en noir et blanc. Il peut aussi être utile de faire une version chibi ou SD du personnage pour voir s'il fonctionne sans détails. La couleur permet par ailleurs de jouer sur des « rappels », de façon à fondre un élément dans la masse du costume ou au contraire à le faire ressortir en ne l'assortissant d'aucun rappel. Essayez de ne pas trop utiliser de couleurs : on conseille souvent de se limiter à trois couleurs pour ses choix d'habillement, de façon à limiter les chances que deux couleurs jurent entre elles.

Le drapé

Les costumes incluant le plus souvent du tissu, voici quelques remarques sur la question des drapés. Il y aurait matière à écrire des articles entiers sur le sujet, sans compter ceux sur la façon de représenter les autres matières qui peuvent intervenir dans un costume : métal, perles, cuir, plumes, verre, etc. Voyez ici plus une ébauche qu'un véritable cours sur le sujet.

- Comme pour tout le reste, entraînez vous à observer et à dessiner des drapés d'après photo ou d'après nature, afin de comprendre la façon dont les plis se forment.
- Ne dessinez jamais les drapés directement : il faut toujours dessiner, même de façon très simple, l'objet qui soutient le tissu. Il est très important de comprendre qu'un élément invisible au spectateur peut avoir une grande influence sur le dessin : sous ces vêtements, il y a un corps caché qui détermine la forme et les plis du tissu.
- Réfléchissez aux propriétés des matières que vous dessinez : il existe une multitude de tissus, extrêmement différents les uns des autres. Ils peuvent être fins, épais, rugueux, lisses, brillants, mats, moirés, etc. Un tissu très léger aura tendance à faire beaucoup de plis, alors qu'un tissu épais fera des plis plus rares, plus gros et plus « cassés ».
- Pensez le tissu en termes de forces : les plis sont le résultat de la concurrence entre le poids, qui entraîne le tissu vers le bas, et les forces de tension, qui tirent le tissu vers le point d'accroche.
- Les plis sont plus nombreux là où le tissu est resserré.
- Les lignes dessinées par les plis convergent vers un point où le tissu subit une force de tension.
- Essayez de penser les plis en termes de masses d'ombre et de lumière plus qu'en termes de traits.
- Ne dessinez **jamais** des plis appris par cœur, que ce soit par habitude ou par mimétisme : le procédural, c'est mal.

Comme pour toute chose, il n'y a pas de secret : TRAVAILLEZ.

Cet article n'est qu'une ébauche sur un domaine extrêmement vaste. Il aurait fallu ici parler de bien d'autres choses comme la théorie des couleurs, le rendu des matières, les principes de la théorie du Gestalt et de bien d'autres choses encore. Néanmoins, je n'ai ni la place ni les compétences d'en discuter ici et j'espère que j'aurais au moins réussi à attiser votre curiosité.

Rappelez vous surtout qu'un bon design n'est pas forcément le fait d'un bon dessinateur, mais de quelqu'un qui a bien réfléchi et qui entretient sa culture, et évitez une complexité gratuite : pour être capable de faire bien et compliqué, il faut être capable de faire bien et simple d'abord. J'espère que cet article vous sera utile et je suis bien évidemment ouvert à tout commentaire, remarque ou contradiction : ne vous en privez surtout pas :D

La difficulté dans les jeux vidéo

Par [Pierre Guyot](#) | le 18 juin 2011 18:33:03 | Catégories : [Game Design](#)

L'un des thèmes qui revient de façon récurrente dans les débats sur le game design est celui de la difficulté. On entend beaucoup de choses : si certains critiquent les joueurs assistés et appellent à un retour au « oldschool », d'autres tentent de diminuer le challenge afin d'élargir leur public au maximum (c'est l'un des piliers de la politique de Nintendo, qui a pourtant donné son nom à l'expression « Nintendo Hard »).



Le but de cet article est de proposer une ébauche de réflexion sur la difficulté dans le jeu, en essayant de la définir et de dégager quelques principes pour réussir à éviter certains écueils.

Commençons tout d'abord par essayer de définir les termes — ou plutôt le terme — de notre sujet : la difficulté. Pour faire simple, on peut dire qu'il s'agit juste d'une mesure de l'effort qu'un joueur va devoir fournir pour surmonter un obstacle dans un jeu. Mais si l'on y réfléchit bien, il apparaît qu'il n'y a pas une, mais plusieurs formes de difficultés. Essayons d'en dégager quelques aspects.

I. Vrai challenge et fausse difficulté

Une difficulté bien dosée peut constituer l'un des éléments essentiels d'un gameplay. Pour le joueur, réussir une épreuve qui a nécessité un effort conséquent sans que cela tourne à l'arrachage de cheveux peut constituer une gratification immense. Le game design a beaucoup évolué depuis les années 80 et l'on est passé de jeux conçus *contre* le joueur à des jeux conçus *pour* le joueur. Néanmoins, de nombreux jeux continuent de présenter ce que j'appellerai une *fausse* difficulté. Souvent, il s'agit de rallonger la durée de vie de façon artificielle (bien qu'on puisse discuter s'il s'agisse réellement de « durée de vie ») ou d'échapper à une étiquette « trop facile » qu'on épingle sur beaucoup de jeux récents.

Une discontinuité dans la progression : le joueur fait littéralement face à un mur, le plus souvent sous la forme d'un boss, dont la difficulté est très nettement surélevée par rapport à ce qu'il a pu affronter avant.

Un petit pic de difficulté peut très bien apporter à un combat un tour plus dramatique : l'exemple type est le combat « de fin de CD1 » qu'on trouve dans moult RPG, comme le premier combat contre Dhoulmagus dans *Dragon Quest VIII*. Mais un changement de difficulté mal dosé se solde dans la plupart des cas par un abandon du jeu ou par une phase pénible de level grinding.

Un exemple particulièrement marquant est celui de *Star Ocean III*, où le niveau de tous les monstres du jeu se voit relevé d'un seul coup de façon scandaleuse. Pour le joueur, il y a un sentiment d'injustice, d'un *manque de fair-play* de la part des développeurs qui changent tout un coup les règles du jeu sans prévenir.



Une mauvaise maniabilité : un travers qu'on rencontre dans de nombreux jeux qui prétendent ainsi donner une touche rétro à leur gameplay, où un personnage peu maniable rend le jeu beaucoup plus difficile qu'il ne devrait l'être.

Proposer un saut à la limite des compétences du héros et avec un timing serré peut constituer une vraie difficulté. Proposer un héros dont la physique est contre-intuitive, des boîtes de collision abusives ou de mauvais contrôles de caméra constitue une fausse difficulté. Un bon jeu de plate-forme rétro nécessite des sauts millimétrés, mais propose une maniabilité impeccable en contrepartie — l'exemple le plus frappant qui vienne à l'esprit est celui de *Super Meat Boy*.

Nombre de vieux jeux de plate-formes souffrent grandement de ce travers. Un exemple assez remarquable est celui de la série des *Resident Evil* où la maniabilité du héros (la touche gauche ne sert pas à aller vers la gauche mais à pivoter sur la gauche, par exemple) est totalement contre-intuitive. Néanmoins, si cela peut être vu comme un défaut, on peut aussi considérer que cela participe du stress ambiant et contribue d'une certaine façon à l'ambiance du titre. A discuter, donc.

Des chiffres trop élevés : un travers qui consiste à multiplier les grandeurs comme les points de vie des ennemis ou des boss ou à copier-coller des passages entiers dans un niveau, pour créer une durée de vie artificielle.

Il arrive que les développeurs trouvent un combat trop court, même s'il est parfaitement dosé. L'idée est donc, sans modifier la difficulté réelle du combat, de multiplier les points de vie d'un boss (ou à dupliquer les différentes parties d'un niveau). Ceci oblige le joueur à répéter outre mesure un pattern, ce qui n'augmente pas la difficulté mais seulement le temps de jeu et tend à donner une impression de répétitivité.

Un travers de Square-Enix dans ses jeux récents. *Kingdom Hearts II*, malgré des combats très bien pensés, propose un certain nombre de boss à la jauge de vie invraisemblablement longue, ce qui nécessite parfois de répéter plus d'une dizaine de fois un pattern très long.

De même pour *Final Fantasy XII* où certains boss pas particulièrement difficiles ont un nombre de HP dépassant la dizaine de millions : on passe plus d'une heure à attendre qu'un boss peu menaçant finisse par bien vouloir s'effondrer, ce qui détruit par ailleurs toute la tension dramatique du combat.

Rétention d'information : le joueur ne sait pas que ce qu'il a pu manquer au début du jeu peut rendre le jeu beaucoup plus difficile, voire impossible à finir par la suite, ou que ceci l'empêchera d'accéder à certains éléments du jeu.

Très connu et donc souvent évité : peu de choses ont autant de chances de provoquer une dépression nerveuse chez un joueur que de savoir qu'il ne pourra jamais terminer le jeu parce qu'il a manqué, douze heures plus tôt, un ridicule petit bout de papier dans un placard secret. Néanmoins, une version plus soft est toujours à l'œuvre.

Un exemple dans *Valkyrie Profile II* : vers la moitié du jeu, le joueur voit deux personnages quitter son équipe et en échange, reçoit des équipements. La puissance de ces équipements dépend du niveau des personnages qui quittent l'équipe. A moins d'être au courant du fonctionnement de cette récompense (ce qui est impossible sans une solution) et d'avoir passé quelque chose comme trois heures à faire du level grinding pour passer plus d'une dizaine de niveaux, la suite du jeu devient très difficile. A l'inverse, si le joueur fait tout ceci dans l'idée d'obtenir une arme digne de ce nom et d'avoir accès une difficulté normale pour la suite du jeu, tout est déséquilibré dans l'autre sens et devient trop facile.

Un exemple un peu différent est celui de *Valkyrie Profile* où l'obtention de la vraie fin nécessite d'aller farfouiller dans une solution, sans quoi vous passerez très largement à côté. *Persona 4* reprend ce même principe, même s'il est vrai que c'est avec plus de brio (je n'en dis pas plus pour ne pas trop spoiler, mais posez vous la question de savoir si vous avez vraiment obtenu la véritable fin du jeu :]).



Roue de la fortune : l'issue de l'épreuve ne dépend pas dans une mesure raisonnable des actions du joueur et est principalement déterminée par le hasard.

Épouvantable. Cela peut nécessiter de répéter un passage du jeu (souvent pénible) à de très nombreuses reprises et provoque chez le joueur un sentiment d'injustice, puisqu'il ne peut lui-même rien y faire.

Très présent dans les RPG dans le principe des drops. Réaliser certaines quêtes nécessite de tuer un type de monstre qui libère parfois un objet. Si le taux d'apparition de l'objet est trop bas, cela peut tourner au monster bashing bête et méchant. Particulièrement surnois si combiné avec des chiffres trop élevés. L'obtention des objets ultimes pour le craft est particulièrement infâme pour cette raison dans *Final Fantasy XIII*.

Un exemple un peu différent est la roue utilisée pour les combats du fort mal réputé *Unlimited Saga*, qui rend difficile l'élaboration de tactiques. A l'inverse, c'est en proposant un timing extrêmement précis — et donc qui ne repose pas sur la chance — que *Shadow Hearts II* a pu réussir à faire de la roue le point crucial d'un excellent gameplay.

Une difficulté trop réglée : un problème plus narratif, qu'on retrouve dans la célèbre [liste des clichés du RPG](#). Étrangement, dans beaucoup de jeux et aux dépens de l'immersion, les monstres ont toujours un niveau comparable au vôtre : au début du jeu, vous combattez les slimes de base et dans le donjon final, ce sont les démons des dernières strates infernales qui vous attaquent. C'est moins un problème de gameplay qu'un problème de récit et d'univers.

Un travers qu'on retrouve dans presque tous les RPG classiques, mais que des jeux comme *SaGa Frontier II* ou *Final Fantasy XII* ont su éviter, en rajoutant une difficulté optionnelle : dans une plaine ou un donjon, on trouve globalement des ennemis d'un niveau acceptable, mais aussi des ennemis plus faibles et plus forts — parfois *beaucoup* plus forts. Néanmoins, ces derniers peuvent être évités, de façon à ne pas entraver la progression du joueur, et les battre conservent un intérêt, pour les récompenses substantielles qu'ils offrent au joueur.



II. Difficulté annoncée

Lorsque l'on cherche à doser la difficulté d'un jeu, l'un des problèmes à cerner est de déterminer ce qu'attend le joueur. Il faut donc dans une certaine mesure annoncer au joueur ce qui l'attend, de façon à ce que celui ne se sente pas lésé.

Beaucoup de jeux proposent pour cela plusieurs niveaux de difficultés. Ceci permet d'une part d'essayer de satisfaire une plus large tranche de joueurs, d'autre part de permettre au joueur de trouver quel est le créneau qui lui convient. Un joueur occasionnel qui n'a pas beaucoup de temps pour lui et qui voudrait quand même découvrir l'histoire ou le gameplay du jeu cherchera plutôt un mode facile, alors qu'un hardcore gamer avide de challenge se ruera sur les modes de difficultés extrêmes aux intitulés plus colorés les uns que les autres (citons parmi d'autres « Dante Must Die », « 4D Mode » et « European Mode »).

Il faut pour autant faire attention à ce que les annonces correspondent à peu près à ce que le joueur trouvera en y jouant. Il n'est pas évident que le mode « Normal » d'un *Star Ocean III* ne soit pas un mode difficile, au vu de l'arrachage de cheveux potentiel qui attend le joueur par moments. Par ailleurs, dans un souci d'immersion, on peut essayer de faire rentrer ce choix dans le contexte du jeu. *Resident Evil* et *Kingdom Hearts* le font très bien sous la forme d'une sorte de test psychologique.

Une autre façon d'annoncer la difficulté passe par la réputation et par le genre du jeu. Ainsi, la plupart des joueurs savent à quoi s'attendre lorsqu'ils se lancent dans un shmup. De même, quand on commence un jeu Capcom, on s'attend plutôt à quelque chose de difficile, à l'inverse d'un jeu Nintendo (actuel du moins, ce n'a pas toujours été le cas :]). Il faut donc faire attention à l'image que l'on véhicule et à ce qu'elle ne trahisse pas le joueur en lui donnant une fausse impression.

III. XP versus Skill

J'aimerais ici parler d'une distinction qui me semble fondamentale et qui permet de distinguer les jeux sur le plan de la difficulté en les plaçant sur une échelle. La difficulté à laquelle le joueur va faire face est principalement définie par la façon dont le joueur va devoir réagir.

D'un côté, ce sont les personnages du jeu qui vont gagner de l'expérience. Face à un boss trop difficile ou à un niveau trop dur, le joueur va reprendre les niveaux précédents, les refaire pour que son avatar gagne de l'expérience, ce qui matérialisera par des HP plus élevés ou de nouvelles compétences. C'est le cas le plus classique dans les RPG classiques et dans beaucoup de jeux où l'on peut acheter des compétences et c'est le fondement du level grinding. C'est ce que j'aurais tendance à appeler l'XP.

A l'opposé du spectre, c'est le joueur qui va gagner en expérience. En réessayant un niveau ou en recommençant à plusieurs reprises un combat contre un boss, le joueur va maîtriser de plus en plus son personnage et comprendre les patterns qui régissent le fonctionnement du boss. C'est le principe de beaucoup de jeux de plate-formes où, face à un boss difficile, on recommence le combat jusqu'à en connaître parfaitement les principes selon une mécanique de *trial and error*, sans possibilité que le personnage devienne plus fort. C'est que j'aurais tendance à appeler le skill.



Entre les deux, on peut observer une immense diversité de jeux qui les combinent. Beaucoup de MMORPG ne reposent par exemple que sur l'XP. A l'inverse, un jeu de plate-forme comme *Mario* ne repose que sur le skill. Un *Final Fantasy XIII* se place plutôt du côté du skill, puisqu'il bride l'expérience et propose un game over non pénalisant, qui permet de réessayer un combat presque immédiatement. Un *Odin Sphere* équilibre très bien les deux en proposant des boss à pattern et un système d'expérience qui permet de ne pas tomber dans le game breaking.

Les deux possibilités ont leurs avantages et leurs inconvénients. L'XP permet de s'assurer que le joueur pourra aller jusqu'au bout du jeu, même s'il faut passer par une phase de level grinding. En contrepartie, cette phase peut s'avérer répétitive, voire pénible : un exemple est la quasi-intégralité de l'infâme série des *SaGa* — à l'exception notable du remarquable *SaGa Frontier 2* — qui nécessite un level grinding absolument infernal. A l'inverse, le skill permet de s'assurer que le joueur n'aura rien d'autre à faire que jouer véritablement pour avancer, mais au risque de le voir définitivement bloqué devant un ennemi qui dépasse ses compétences. L'exemple le plus extrême — mais tout à fait volontaire — est *Super Kaizo Mario*, une sorte de version ultra difficile des aventures du plombier moustachu.

IV. Principes

Je ne prétends pas dresser ici une liste de lois immuables qui permettraient de parfaitement régler la difficulté d'un jeu. Il s'agit juste de remarques que j'ai pu me faire en jouant et en discutant avec d'autres joueurs.

Le jeu est conçu pour le joueur, pas contre lui. Le game designer ne doit pas voir le joueur comme un ennemi qu'il doit embêter, mais comme quelqu'un à qui il est censé procurer du fun. Un jeu vidéo, comme son nom l'indique, n'a pas vocation à être pénible — même s'il peut dépasser sa dimension de simple jeu — et victimiser le joueur est loin d'être la preuve d'un game design intelligent. Il ne faut pas donner au joueur l'impression que le jeu n'est pas fair-play avec lui — ce qui arrive souvent lorsqu'il se rend compte que l'ordinateur ne joue pas selon les mêmes règles que lui. La majorité des épisodes du *Joueur du Grenier* illustre très bien ce principe.

Quand on annonce un niveau de difficulté, on s'y tient. Si le joueur a choisi un mode de difficulté, facile ou difficile, il a ses raisons et le game designer doit les respecter. Ce n'est pas parce qu'on joue en facile qu'on ne mérite pas de jouer. Aussi, relever soudainement la difficulté d'un mode facile (ou inversement), en particulier en toute fin de jeu, est un motif de frustration particulièrement efficace chez un joueur.

La difficulté doit varier continûment, c'est-à-dire sans à-coup et sans pic. S'il y a une seule chose à retenir de tout cet article, c'est peut-être celle-là. Rien de plus rageant que de jouer pendant vingt heures sans problème et de se voir tout à coup bloqué sans raison par un monstre dix fois plus fort que les autres. Quand on traverse un donjon, le niveau du boss doit être corrélé à celui des ennemis. Un joueur qui traverse le donjon sans fuir aucun combat devrait avoir un niveau du bon ordre de grandeur pour aller battre le boss. Un peu moins si vraiment c'est un jeu dur, mais pas beaucoup moins, sans quoi cela débouche sur des aller-retours incessants et du level grinding stupide. Beaucoup de jeux perdent *énormément* à être joués sur ce point.

Les jeux Tri-Ace et les jeux de la série des *SaGa* en sont des représentants tout particulièrement infâmes. Prenons l'exemple de *Star Ocean III*, qui a pourtant un certain nombre de qualités par ailleurs (dont un système de combat jouissif). Les passages trop faciles et trop difficiles alternent sans aucune logique ni continuité, plaçant le joueur face à des boss quasiment invincibles à la sortie de donjons aussi inoffensifs qu'une promenade ou à l'inverse, proposant un boss aussi faible qu'un poussin asthmatique à la sortie d'une grotte infernale remplie de monstres dont le niveau excède celui du joueur d'une dizaine d'unités. L'augmentation drastique de la difficulté en fin de DVD 1 — même si elle est tempérée d'une justification scénaristique — est absolument lamentable en termes de gameplay. Je ne m'appesantirai pas plus avant sur le cas des *SaGa*, qui sont absolument remarquables pour leur nullité crasse sur ce point, à l'exception de *SaGa Frontier 2*.

Un autre exemple, plus ponctuel, est celui du début du CD 2 de *Baten Kaitos II*. Celui-ci commence par un boss beaucoup trop dur, après avoir fait sauvegarder le joueur et sans lui donner la possibilité de faire de l'expérience. A moins d'avoir consulté une solution pour trouver une faille dans la façon dont celui fonctionne — pour ainsi dire un bug de conception — vous êtes condamnés à reprendre la partie depuis le début, soit la bagatelle de trente heures.



Un début de jeu n'est jamais trop facile. Le meilleur moyen de faire qu'un joueur arrête de jouer et balance le CD et sa boîte par la fenêtre, c'est de lui proposer un début de jeu trop difficile. C'est par exemple ce qui fait que vous éteignez un jeu amateur après seulement cinq minutes de jeu. Non seulement, cela peut-être pénible si cela oblige le joueur à repasser une longue — voire interminable — introduction, mais en plus, c'est presque insultant pour sa fierté de mourir si tôt en début de partie. De plus, ça n'est souvent pas cohérent avec la montée en puissance qui a classiquement lieu dans un jeu.

Pas de final trop facile. A l'inverse, si la fin du jeu est trop facile, ceci risque de gâcher le plaisir du joueur en donnant l'impression qu'on sous-estime ses compétences et en provoquant une retombée de la tension. Ceci peut ruiner un final du point de vue scénaristique, par exemple en privant un grand méchant d'un combat final qui soit à la hauteur de sa puissance, telle qu'elle est décrite par le scénario. De très bons jeux se cassent les dents sur ce point, dont les excellents *Final Fantasy X* et *Kingdom Hearts*, dont les boss de fin ne sont, en termes de difficulté, pas à la hauteur de leur design et de leur importance dans le scénario.

Ne jamais laisser de gap, c'est-à-dire de situation où le joueur ne peut faire de level up nulle part. A la fin du jeu, les ennemis les plus forts doivent donner une expérience suffisante pour qu'en faisant du level grinding, il soit possible d'atteindre en un temps raisonnable les derniers niveaux. Ceci peut passer par des donjons secrets, des boss optionnels, etc. L'idée est que le joueur ne doit pas se retrouver dans une situation où les monstres qu'il combat actuellement sont simples à battre et donnent donc peu d'expérience et où le prochain niveau le met face à des monstres complètement hors de sa portée. C'est le problème de *Persona 4* par exemple, où les ennemis du donjon de fin donnent une quantité d'expérience très faible, au regard des quantités exigées pour monter de niveau à cet endroit du jeu. Bien que cela permette d'avoir un boss de fin suffisamment difficile, ceci oblige le joueur à level grinder pendant un temps absolument hallucinant s'il compte pouvoir utiliser les plus puissantes personas et ruine ainsi l'intérêt de ces dernières.

Si cela est faisable, donner au joueur la possibilité de moduler un peu la difficulté. *God of War* proposait ainsi, si le joueur recommençait trop de fois une même séquence, d'abaisser temporairement la difficulté du jeu.

La difficulté doit toujours être récompensée. Ce qui est important dans la gestion de la difficulté, c'est que les efforts du joueur doivent être récompensés. A l'inverse, il ne faut pas distribuer les récompenses à tout va, sans quoi celles-ci perdent de leur valeur. C'est un peu le principe des *achievements* et ce qui fait qu'il est extrêmement gratifiant de terminer un manic shooter.

J'espère que cet article aura permis de dégager quelques pistes de réflexion sur cette question ardemment débattue de la difficulté et qu'il pourra être utile. Si vous avez des remarques, des commentaires ou des contradictions à apporter, n'hésitez absolument pas :)

Game Design & MOBA

Par [Antoine Gersant](#) | le 26 juin 2011 12:44:34 | Catégories : [Analyse](#), [Game Design](#)

Je ne sais pas pour vous mais depuis que je fais des jeux (et depuis que je ne suis plus au lycée, ce qui commence à dater un peu), j'ai l'impression que je passe de moins en moins de temps à jouer. En tant que passionné, je suis au courant de l'actualité, des annonces et des jeux qui sortent ; je lis la presse, je regarde des screenshots, je m'arrête dans le salon regarder à quoi jouent mes chers colocataires, mais je ne joue pas très souvent moi-même.



Un peu largué dans les conversations de gamers, je me demande bien où réside l'attrait de bon nombre de succès récents. Par exemple la série des *Halo*. Je suis sûr qu'il doit y avoir quelque chose à sauver et qu'autant de gamers ne peuvent pas s'être tous trompés, mais vu de loin il s'en dégage pour moi l'image d'un FPS qui tourne au ralenti où les personnages sont des cosmonautes neurasthéniques dans une série B avec des aliens (et des épées lasers).

Heureusement, mon instinct de joueur n'a pas disparu et il m'arrive de redevenir un *hardcore gamer* pour quelques semaines ou quelques mois. C'est ce qui s'est produit l'an dernier quand j'ai eu le plaisir (ou le malheur ? à vous de trancher) de découvrir *League of Legends*, l'un des MOBA à la mode ces derniers temps.

Qu'est ce qu'un MOBA ?

Historique rapide

Si vous êtes déjà familier du genre, vous pouvez vous épargner la lecture de quelques paragraphes et passer directement au cœur de l'article (*Pourquoi ça marche ?*). Pour les autres, le MOBA (pour *Multiplayer Online Battle Arena*) est un genre récent à la croisée du RPG et du RTS. Le jeu à l'origine du genre est le célèbre mod de *Warcraft III* baptisé DOTA (pour *Defense of the Ancients*) qui sert de modèles à tous les autres — c'est à dire *League of Legends*, *Demigod*, *Heroes of Newerth*, le futur *Dota 2* développé par Valve et quelques autres jeux plus obscurs.

Mon expérience se limite à DOTA auquel j'ai peu joué, et à *League of Legends* qui a englouti plusieurs mois de ma vie. Mes explications et mes exemples reposeront donc sur ce dernier mais je vais faire de mon mieux pour faire abstraction des spécificités du jeu et ne m'intéresser qu'au genre. Les puristes me pardonneront de ne pas rentrer dans les détails, de ne pas mentionner des variantes de règles, ou d'oblitérer des subtilités qui ne font pas avancer mon propos.

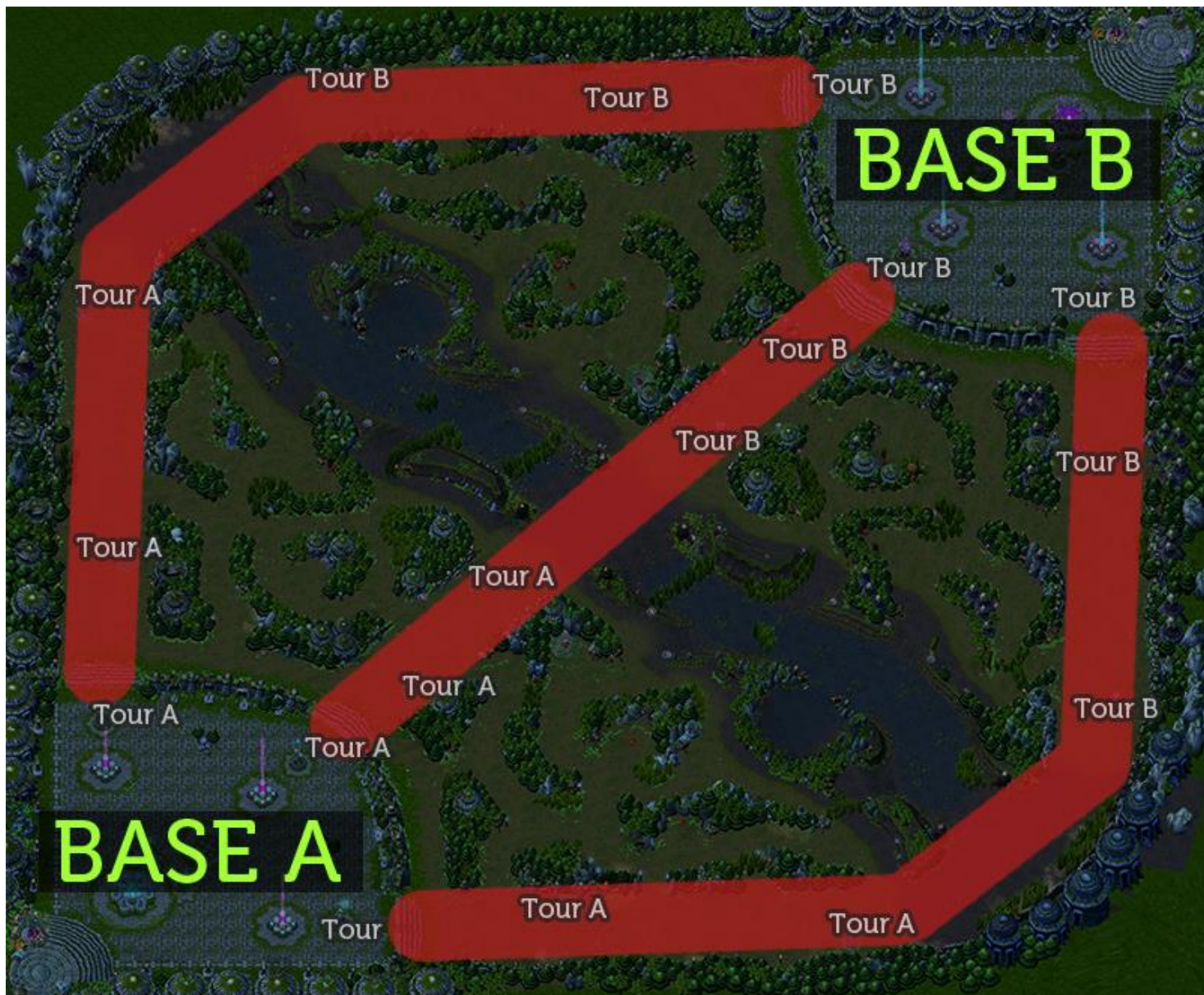
Mode d'emploi d'un genre à part

La première chose à vous mettre dans la tête est que les MOBA sont un genre à part entière. Ce ne sont pas des RTS ni des RPG mais bien un enfant bâtard qu'aucun des deux genres ne pourrait jamais reconnaître comme le sien. C'est ce fragile équilibre qui rend le genre si pointu. Vous allez le voir par vous même, à

entendre le concept d'un MOBA on se demande comment on peut tirer plusieurs jeux différents d'un genre aux caractéristiques si précises.



Une partie dure entre vingt minutes et une heure et met face à face deux équipes de 5 joueurs humains. La caméra a un point de vue semblable à celui d'un RTS, mais chaque joueur ne contrôle qu'un seul personnage, avec sa souris (comme dans un RTS ou un Hack'N Slash). Il n'y a pas de bâtiments à construire ni d'unités auxquelles donner des ordres. Ce qui semble très étrange au premier abord, c'est que l'on joue toujours sur la même carte. Voici une version schématisée de la carte en question (commune au moins à *DOTA*, *LoL* et *HoN*).



Les équipes ont chacune une base (coins haut-gauche et bas-droit de la carte) où les joueurs respawnt quand ils meurent. L'objectif de la partie est de détruire le cœur de la base ennemie, qui est défendu non seulement par les joueurs mais aussi par de puissantes tours de défense automatiques. Il y a 3 chemins (appelés *lanes*, en rouge sur ma carte) qui relient les deux bases et qui sont tous pourvus de tours de défenses (indiquées sur la carte). La dynamique du jeu repose sur le combat à 5v5 sur ces 3 champs de batailles différents entre lesquels il faut sans cesse alterner, car une tour sans joueur(s) pour la défendre peut être rapidement abattue (et une tour en moins, c'est un pas en plus vers la base).

Les 10 joueurs ne sont pas seuls sur la carte pour autant. Chaque base génère en continue des unités très faibles (les *creeps*) qui avancent de manière robotique le long des 3 lanes pour aller combattre les *creeps* de l'équipe adverse. Ces *creeps* ont plusieurs intérêt pour le Game Design du jeu :

- Ils ont un champ de vision dont le joueur profite car toutes les informations accessibles à son équipe sont représentées sur sa minimap.
- Ils permettent de savoir grosso modo jusqu'où une équipe a le contrôle stratégique de la carte (les *creeps* se font arrêter par les tours et les *creeps* adverses, ou par des joueurs ennemis qui les massacrent).
- Ils servent de chair à canon pendant que les joueurs abattent une tour (la tour tue les *creeps* — et non les joueurs — pendant que les joueurs sapent la tour. En l'absence de *creeps*, la tour s'attaquerait aux joueurs).
- Ils rapportent de l'or et de l'expérience (voir ci-dessous)

De l'expérience et de l'or, c'est là que l'aspect RPG entre en jeu. A mesure qu'ils tuent des creeps, des joueurs ennemis et des tours, les joueurs sont récompensés par de l'or et de l'expérience. L'or sert à acheter des

objets au magasin (dans la base de l'équipe) pour rendre son personnage plus puissant (un peu comme le magasin dans *Counter Strike*), et l'expérience permet d'améliorer ses caractéristiques (force, défense etc.) et ses compétences (les *skills*) comme dans un RPG traditionnel.

Là où le système s'éloigne d'un RPG traditionnel, c'est que les caractéristiques de base et les compétences d'un joueur sont essentiellement déterminées avant le début de la partie par le choix d'un personnage. Chaque MOBA propose plusieurs dizaines de personnages (104 dans *DOTA*) qui ont chacun un design propre, des caractéristiques de base et surtout 4 *skills* qui seront leurs seules actions possibles pendant la partie (en plus d'une attaque standard commune à tous les personnages).

On retrouve comme personnages les archétypes développés dans les RPG et MMORPG ces dernières années : *healers* (personnages avec des sorts de soin), *tanks* (personnage capable d'encaisser beaucoup de dégâts pour son équipe), *DPS* (personnage capable d'infliger des dégâts conséquents sur la durée), *nuker* (personnage capable de faire beaucoup de dégâts en peu de temps) etc., avec toutes les variantes possibles et imaginables (personnage corps à corps ou à distance, dégâts physiques ou magiques...).

La gestion des objets (achetés pendant la partie avec de l'or, si vous avez tout suivi) est elle aussi différente d'un inventaire traditionnel de RPG. Chaque joueur a 6 *slots* d'inventaire pour son personnage qu'il peut remplir comme il le souhaite. Je veux dire par là que vous pouvez équiper votre personnage avec 6 épées différentes, c'est tout à fait l'état d'esprit. La subtilité du système réside dans le fait que :

- Les objets du magasin sont très nombreux (mais les joueurs un peu entraînés les connaissent sur le bout des doigts).
- La plupart des objets s'obtiennent en combinant entre eux des objets de coût inférieur.
- Les objets ont des effets souvent complexes qui agissent sur plusieurs caractéristiques, et qui peuvent (ou non) se cumuler entre eux.



Il en résulte des wikis, des guides, et des débats de *theory crafting* (sur les forums ou avec vos colocataires) pour tenter d'établir quels sont pour chaque personnage les objets les plus adaptés. Il n'y en a généralement pas de solution exacte mathématique au problème, et tout dépend de la composition de l'équipe adverse, du déroulement de la partie et des changements d'équilibrage des derniers patches. Équilibrer un tel jeu est en effet un exercice périlleux en raison du nombre colossal de personnages, d'objets et de *skills* mis en jeu (et c'est sans doute l'une des raisons pour lesquelles les MOBA ne se jouent que sur une seule carte).

Enfin, dernier élément de briefing avant que l'on ne s'intéresse aux astuces de Game Design qui rendent les MOBA si addictifs (et si bien conçus) : la carte ne se limite pas aux trois *lanes* où sont plantées les tours. Il est possible de circuler entre les *lanes* en passant par la « jungle » qui est faite de petits chemins tortueux (parfaits pour tendre des embuscades) et où l'on trouve à emplacement fixe des monstres neutres (les *neutral creeps*) qui ne s'en prennent aux joueurs que si on les attaque et qui pour certains donnent des récompenses considérables — ce qui fait d'eux des endroits stratégiques de la carte à contrôler.

Pourquoi ça marche

Bon, c'est bien gentil tout ça mais mon intention n'est pas de faire un guide d'initiation aux MOBA. Passons donc aux choses sérieuses si vous le voulez bien, tentons d'isoler quelques-unes des spécificités du genre qui le rendent si prenants, et qui ont poussé plus d'un joueur raisonnable à veiller jusqu'à l'aube.

Avatars et héros

Comme je vous l'ai expliqué, chaque partie commence par un écran de sélection de personnage où chaque joueur choisit le héros qu'il incarnera pendant la partie. Là où les choses deviennent intéressantes, c'est que le choix du héros donne l'impression au joueur non pas qu'il incarne le héros mais que le héros le *représente*. La nuance est très importante ; considérez la différence d'affection et d'attachement éprouvés envers Lara Croft (quand vous jouez à *Tomb Raider*) et votre personnage dans *Guild Wars* ou *World of Warcraft*.

Ce qui différencie fondamentalement Lara Croft et Belin le nain forgeron (votre personnage de *World of Warcraft*), c'est que Belin est un personnage qui a une apparence physique et des caractéristiques (en terme de règles du jeu) qui sont uniques et le fruit de vos décisions. Lara Croft a le même physique pour tous les joueurs de *Tomb Raider* et toujours le même fusil à pompe. Vous l'avez compris, c'est la *customisation* qui fait la différence : en étant amené à prendre des décisions le concernant, le joueur se sent plus responsable et proche de son personnage.

Dans un MOBA, on ne personnalise pas son héros. L'identification est pourtant possible grâce à plusieurs astuces :

- Le nom du joueur est toujours associé au héros qui le représente dans la partie en cours. Dans *League of Legends*, le nom du joueur est affiché sur le portrait du héros dès l'écran de chargement (en dessous du nom du héros). Pendant la partie, ce sont les noms des joueurs qui sont affichés au dessus des personnages. Dans *HoN*, les noms des joueurs ne sont même pas affichés à l'écran de chargement et seul un portrait de leur héros apparaît.



- Le très grand nombre de héros disponibles rend le choix d'un héros finalement peu différent d'un procédé de *customisation*. Avoir une centaine de personnages procure autant de diversité (voire plus puisqu'ils sont créés à la main) qu'une *customisation* où l'on aurait 20 options à choisir avec 5 valeurs possibles pour chacune. C'est de l'ordre de ce que proposent certains MMORPG. La recherche d'un héros qui corresponde au joueur est facilitée (et encouragée) par la présence d'informations détaillées (*artwork, background, skills, style de jeu, etc.*) sur chaque personnage accessibles depuis le jeu et de manière très ordonnée. Ci-dessous, l'écran d'information sur les personnages dans *League of Legends*. C'est cet écran aperçu sur le PC d'un colocataire qui a attisé ma curiosité envers le jeu, comment résister à l'envie de cliquer sur ces petits tiles carrés bien agencés ?



- Le caractère très compétitif et technique du jeu fait que la plupart des joueurs passent un très grand nombre d'heures à jouer avec une petite poignée de héros qu'ils veulent bien maîtriser. En ce qui me concerne, j'ai joué mes 200 premières parties de *League of Legends* — une centaine d'heures — avec un seul et unique héros. Le même phénomène a lieu avec les jeux de combats un peu techniques (*Guilty Gear*, *Street Fighter*, etc.) où les bons joueurs sont connus en tant que spécialistes de tel ou tel personnage, auquel on les associe ensuite naturellement. A force de jouer toujours les mêmes héros, on a l'impression d'établir un lien avec eux qui perdure entre les parties (qui sont pourtant indépendantes).

Ce sont là les techniques que j'ai pu identifier pour permettre aux joueurs de considérer les personnages qu'ils incarnent comme leurs avatars, ce qui renforce leur lien avec le jeu.

Le feedback immédiat

Une fois la partie démarrée, les DOTA-like usent et abusent de *feedback immédiat*. Je sais que le nom est proche mais cela n'a rien à voir avec la [rétroaction positive](#) dont Ernest Adams parle si bien. Le *feedback immédiat* est une notion que j'aime beaucoup parce que je trouve qu'elle renvoie très fortement au concept de jeu vidéo dans sa forme la plus simple, à savoir :

- Je regarde l'écran
- J'appuie sur un bouton
- Il se passe un truc à l'écran

Ce qui rend en grande partie les jeux vidéos si addictifs, c'est le savant dosage de ces mécaniques d'action/réaction qui donnent l'impression au joueur qu'il fait quelque chose de plus qu'appuyer sur des boutons. Quelques exemples de *feedback immédiats* particulièrement mémorables sont le **gling gling** rythmé des pièces que l'on accumule dans *Super Mario Bros.* (ou le **dôôi dôôi** de la caisse de pommes que l'on vide avec un rythme de métronome dans *Crash Bandicoot*), la ligne complétée qui clignote frénétiquement avant de disparaître dans *Tetris*, le *barrel roll* d'un *Starwing* ou d'un *Wipeout HD*, ou encore le *stealth kill* millimétré de *Tenchu*. La liste pourrait être beaucoup beaucoup plus longue car il s'agit d'une notion assez simple à appréhender et que beaucoup de Game Designers semblent maîtriser (parfois inconsciemment).

L'absence totale de *feedback immédiat* associé à une action produit un effet très désagréable, qui donne au joueur l'impression que le jeu n'est pas réactif voire carrément mal programmé. Essayez d'imaginer comment ce serait de jouer à *Zelda* si les rubis ne faisaient pas de bruit quand on les ramasse ; on aurait l'impression qu'ils disparaissent quand on passe à proximité (alors qu'avec le son on a l'impression de les ramasser).

Vous aurez remarqué que beaucoup de mes exemples de *feedback immédiat* sont associés à un son caractéristique et pas juste à un effet visuel à l'écran, c'est dû au fait que plus le feedback est important, plus le joueur ressent de plaisir immédiat. Chez Blizzard North, c'est une chose que l'on a compris depuis bien longtemps, et c'est pour cela que des générations de joueurs ont tremblé au son cristallin (**ding**) des anneaux magiques qui tombent au sol dans *Diablo*, ou au comparable **ding** de Level-up de *World of Warcraft*. D'ailleurs les Game Designers de chez Blizzard ont décidé d'aller encore plus loin. Voici ce qu'on pouvait lire en 2008 dans une preview de *Diablo 3* sur [planetdiablo.com](#) :

Leveling up is probably the greatest thing I have seen yet. Imagine yourself surrounded by a horde of zombies, you are clicking away, casting or swinging weapons, and all of a sudden you level up; Big deal right? Ok, let's take that one step further. When you level up in *Diablo III*, it does a massive damage area-of-effect skill. I was grinning from ear to ear.

Et là tout est dit. Vous vous battez normalement, vous cliquez sur un monstre et là **CRAC** c'est l'explosion de lumière, le déluge d'effets de particules lumineuses dans l'écran, et j'imagine qu'une montagne d'effets sonores (chœur angélique qui délivre un saint « Aaaaah », craquement gras de l'explosion, cris et bruits percussifs des monstres qui meurent) vient accompagner ce festival d'orgasmes de poche. Autant dire que vous n'êtes pas déçus d'avoir cliqué et que vous ne risquez pas de passer à côté d'un level up.

Pour en revenir aux MOBA, sachez que non seulement ils font un usage très bien dosé du *feedback immédiat* (il y en a même quand vous cliquez pour déplacer votre personnage) avec toutes les ficelles classiques dont on vient de parler, mais à cela s'ajoute un aspect important. Rappelez vous que nous sommes en multijoueur à 5 contre 5, et que tout le monde profite de vos *feedbacks* (pour peu que tout le monde ne soit pas à l'autre bout de la carte), et cela implique que le joueur les apprécie d'autant plus qu'il les impose à tout le monde. C'est un autre niveau par rapport au concept de base de feedback immédiat. Ici le schéma est le suivant :

- Je regarde l'écran
- J'appuie sur un bouton
- Il se passe un truc à l'écran
- Il se passe un truc sur les écrans de 9 autres personnes qui y prêtent attention

Je vous ai déjà fait remarquer que le fun ressenti était lié à l'intensité du *feedback*, il est intéressant de constater que dans le cadre d'un jeu multijoueur le *feedback* en question devient un pouvoir exercé sur les autres joueurs — autant dire quelque chose de bien plus intense qu'un effet sonore. Le travail du Game Designer est de doser ce pouvoir donné au joueur pour maximiser son plaisir individuel sans pour autant briser la cohérence du jeu ou créer de la frustration chez les autres joueurs.

Si vous avez déjà joué à *Quake III* en activant le *mod* qui fait que le *Plasma Gun* tire des grenades à fragmentation au lieu de boules d'énergies (avec une cadence de tire de mitrailleuse), vous aurez remarqué que le premier joueur à s'en saisir s'amuse comme un fou au milieu de morceaux de cadavres en charpie qui volent sur fond d'explosion, pendant que les autres joueurs se font tuer souvent moins d'une seconde après leur *respawn*. Voilà un exemple de *feedback immédiat* si important qu'il en dérègle le jeu. A l'inverse, un exemple particulièrement bien dosé dans *League of Legends* est le *skill* Paranoïa du personnage Nocturne ; entre autres effets, ce *skill* assombrit pendant quelques secondes l'écran de tous les joueurs, joue un effet sonore (une voix grave et fantomatique qui murmure *darkness* avec plaisir) et réduit le champ de vision des adversaires. Le jeu n'est pas dérégulé, mais le joueur qui incarne Nocturne a eu son heure de gloire en déclenchant tout un petit festival de *feedbacks immédiats* en appuyant sur une touche de son clavier.

La compression du RPG

Passons à un atout vraiment spécifique aux MOBA. Le choix du personnage et le feedback immédiat, c'est bien gentil tout ça, mais dans une certaine mesure on en trouve presque autant dans *Street Fighter II* . Aussi excellent soit-il, il n'est pas aussi addictif qu'un MOBA.

Dans un RPG classique, et c'est surtout vrai pour les RPG dits à l'occidentale (pensez *Wizardry*, *Baldur's Gate* ou encore une fois *Diablo*), une grande partie de l'amusement du joueur repose sur la progression et la montée en puissance de son personnage. Au début du jeu, vous peinez pour éradiquer le grenier d'une horde de rats. Quatre-vingt dix heures plus tard, à la fin du jeu, vous êtes récompensés pour vos choix tactiques (équipements, compétences etc.) et vous êtes en mesure de sauver des royaumes et vaincre des dragons.

Maintenant imaginez exactement *la même chose* mais compressée en une demi-heure de temps. Vous voyez où je veux en venir ? Un MOBA propose le temps d'une partie la même structure ludique qu'un RPG entier, et je vous laisse deviner l'intensité du gameplay qui en résulte et le caractère crucial que prend chaque choix d'équipement (ce qui explique le phénomène du *theory crafting* dont je vous parlais plus tôt). C'est un tour de force de Game Design qui demande bien plus d'ajustements que diviser tous les chiffres d'un RPG classique par 100. Proposer au joueur un tel champ d'évolution en un laps de temps aussi court expose l'équilibrage des mécaniques de jeu puisqu'il devient très facile pour les joueurs de les tester en profondeur et de voir les effets à « long terme » (c'est à dire une heure tout au plus) de leurs décisions.

Le résultat est à la hauteur du concept. En début de partie les personnages sont de puissance comparable et peuvent se faire tuer d'un seul coup par certains monstres neutres de la carte. A la fin, un joueur qui a réussi sa partie peut sortir vainqueur d'une rencontre à 1 contre 2 (voire plus selon le jeu et le personnage dont on parle).

De la coopération à la dépendance

Dernière spécificité de Game Design des MOBA dont nous allons parler, la relation entre les joueurs d'une même équipe. Je suis très loin d'être un connaisseur en matière de jeux en lignes (rassurez vous, j'ai tout de même fait mon quota d'heures sur *Unreal Tournament* ou *Neverwinter Nights*), mais je pense que les MOBA apportent là aussi quelque chose d'intéressant.

La notion d'équipe dans un MOBA dépasse celle que l'on trouve dans un *Team Deathmatch* classique. Dans les FPS en ligne, les joueurs sont subordonnés à un objectif commun mais ils sont capables d'évaluer leurs propres performances (par exemple avec leur ratio *ennemis tués/morts*) indépendamment du reste de leur équipe et surtout de la tournure générale que prend l'affrontement. Dans un MOBA, chaque mort donne un avantage en or très considérable à l'équipe adverse et un court répit pour avancer vers les objectifs de la partie. En plus de cela, l'équilibrage de ces jeux est si fin qu'un seul joueur moins bon que les autres peut mener une équipe droit vers la défaite, et c'est d'ailleurs ce qui se passe dans la plupart des parties.

Un système de jeu qui intègre une telle dépendance entre les joueurs (un peu comme dans *Left4Dead* d'ailleurs) est une arme à double tranchant. C'est la porte ouverte à beaucoup de frustrations et à beaucoup de parties gâchées, mais c'est aussi une route toute tracée pour un jeu très stratégique et très compétitif à haut niveau — sans parler de toutes les émotions qui naissent du lien de confiance et de respect mutuel qui s'établit naturellement entre les joueurs d'une bonne partie. Voir votre personnage marcher au coude à coude avec un autre joueur de votre équipe, tout en sachant de manière tacite quelle stratégie complexe et coordonnée employer quand un ennemi apparaîtra à l'écran — c'est un instant d'une qualité rare que peu de jeux peuvent offrir.

Une ficelle de Game Design à utiliser avec précaution donc.

Je crois être arrivé au bout de ma réflexion, merci de l'avoir suivie jusqu'à son terme. J'espère qu'elle vous aura donné quelques perspectives de Game Design ou au moins la curiosité de jouer à un MOBA par vous-même pour vous rendre compte du travail d'orfèvre que représente leur conception (prévoyez beaucoup de temps libre ou une volonté de fer). Soyez bien conscient que cet article n'avait pas pour but de donner un avis objectif sur les MOBA mais bien d'analyser les points forts de leur Game Design, sans m'attarder trop sur leurs défauts (interdépendance excessive des joueurs, la courbe d'apprentissage trop raide, etc.). Comme toujours, vos remarques et commentaires sont les bienvenus, je suis convaincu que le genre a bien d'autres trouvailles de Game Design à considérer :).

Merci à Widowan pour ses screenshot de Heroes of Newerth et à Deodred pour ceux de League of Legends.

Vilain Game Designer ! Pas de biscuit pour toi ! V

Par [Pierre Guyot](#) | le 3 juillet 2011 15:30:00 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juin 2004).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



L'heure est venue une fois de plus de publier dans le Designer's Notebook le recueil annuel des bugs, des gâchis et des erreurs de jugement ou de goût – ces éléments qui font qu'un jeu est moins bon qu'il ne devrait l'être. Ces Twinkie Denial Conditions ou TDC (*NDT : littéralement, condition pour refuser un biscuit, à savoir à un mauvais game designer*) ne se limitent pas uniquement aux mauvais jeux ; parfois, il ne s'agit que d'une petite imperfection dans un jeu qui est tout à fait bon dans son ensemble.

Presque toutes ces remarques m'ont été envoyées par des lecteurs (et si vous avez des remarques à faire sur les choix de designers qui ne mériteraient pas leur biscuit, j'aimerais le savoir, envoyez moi un mail [*NDT : en anglais*] à notwinkie@designersnotebook.com). Malheureusement, je ne suis pas suffisamment riche pour me payer à la fois une PS2, une XBox, une Gamecube, une Dreamcast, une GBA et un PC, je n'ai pas donc pu vérifier tous les exemples mentionnés. Mais même si un détail ou deux est incorrect, c'est le principe qui compte.

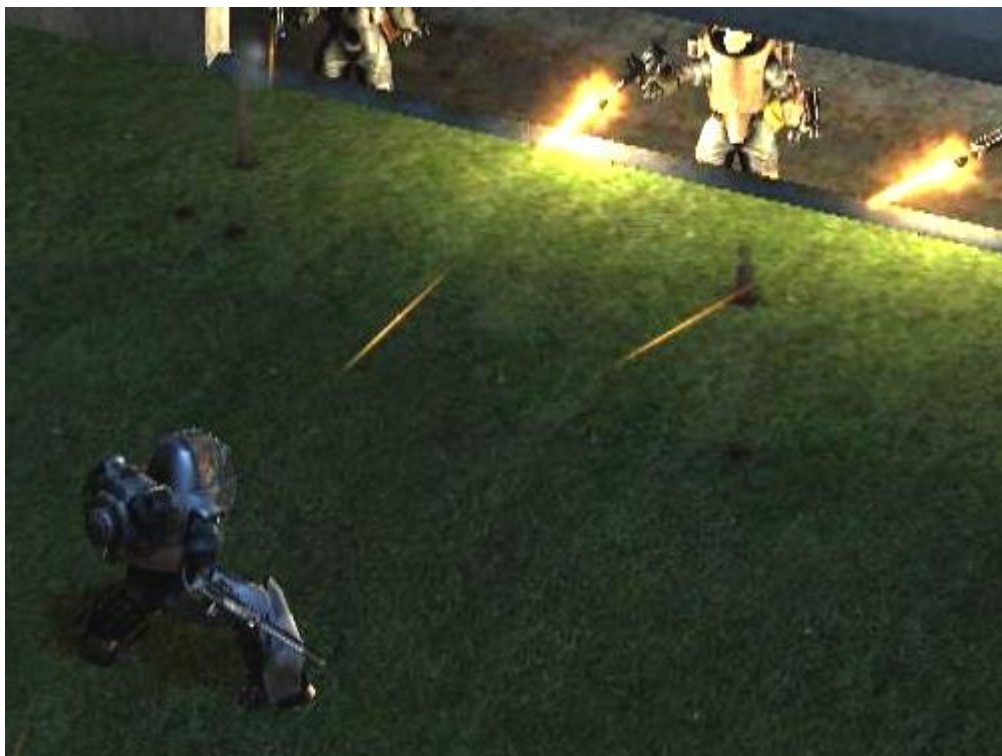
Briser l'équilibre d'un jeu, tout ça pour une fonctionnalité « trop cool »

Bryan Buschmann nous écrit :

La plus récente des perles que j'ai pu voir est l'œuvre des réalisateurs de *Silent Storm*. Ils vous proposent un jeu d'action-infiltration au tour par tour dans un contexte de Seconde Guerre mondiale. Et au cours d'une mission, ils balancent de véritables tanks sur pattes, qui sont littéralement invulnérables à toutes les armes, à l'exception du nouvel équipement, qui existe uniquement sous la forme d'un gigantesque canon laser impossible à manier et que vous n'avez jamais vu auparavant — ou sous celle d'un laser monté sur l'une de ces fameuses monstruosité sur pattes. Une fois que vous avez trouvé et utilisé ces armes lasers pour tuer la personne qui se trouve à l'intérieur du tank (et non le tank lui-même), vous pouvez rentrer dedans et en substance transformer vos compagnons en une légion de monstres invincibles, détruisant ainsi totalement l'équilibre du jeu. C'était vraiment triste à voir, une telle réussite de style et de technologie ruinée lors des niveaux suivants par un game design ridicule. Heureusement, ils ont écouté les joueurs et décidé de revoir entièrement leur implémentation dans l'extension du jeu.

Il y avait des gens sur les forums qui postaient des répliques aussi brillantes que « Bon, si on ne tient pas compte de la stratégie précédente qu'on utilisait avant et qu'on ne tient compte que de ce qu'on est en train

de faire, ça va ! »...Je n'aime pas qu'on me dise en plein milieu d'un jeu que mon précédent style de jeu est maintenant obsolète.



Je pense que Bryan a bien résumé la situation. Beaucoup de designers succombent à la tentation. Leur désir d'incorporer, disons, des armes absolument trop cools prend le dessus sur leur désir d'équilibrer le jeu correctement.

Personnellement, je ne vois pas de toute façon ce que de gigantesques tanks sur pattes peuvent faire dans un jeu qui traite de la Seconde Guerre mondiale ; nous nous trouvons soixante ans après le Jour J et nous ne disposons toujours pas de tels tanks – et je doute que cela arrive un jour, car ceux-ci ne sont absolument pas pratiques. Science-fiction, OK – simulation historique, pas OK.

Les ennemis qui ressuscitent lorsque vous dormez...

...ou quand vous sauvegardez le jeu. Anita Ray (ainsi que plusieurs autres personnes) m'a écrit pour se plaindre de ce problème. Elle tourne les choses de la façon suivante :

Vous venez de réussir à tuer le Monumental et Titanesque Démon de Krall et ses 18 000 sbires. Votre santé et votre mana sont épuisés, mais maintenant qu'ils sont tous morts, le jeu vous permet de vous reposer et de continuer ainsi votre aventure. Vous vous reposez donc... Et vous découvrez que, comme par magie, le Monumental Démon Titanesque et ses sbires ont été ressuscités (probablement avant même que vous n'ayez pu vous soigner, tant qu'on y est). Vous mourez d'une mort atroce, ignoble. Ce problème a l'air de se produire souvent aussi après une sauvegarde.

C'est une programmation paresseuse qui ruine tout ce que le joueur a accompli. Ceci transforme le repos et la sauvegarde en rencontres potentiellement mortelles qui frustrant au dernier degré les joueurs (ou au moins moi). Et pourtant, cela ne semble pas inhabituel dans les RPG. J'ai eu ce problème dans des jeux qui étaient par ailleurs tout à fait corrects (*Morrowind* et *Baldur's Gate* me viennent à l'esprit). Ceci a provoqué chez moi une utilisation de la langue anglaise encore plus créative que celle provoquée par l'administration politique locale, c'est dire.

Ah, si seulement ce problème pouvait être attribué aux programmeurs ! Désolé Anita, ce coup-ci, la faute est entièrement la nôtre, celle des game designers. Le problème de l'apparition et de la résurrection des monstres est très compliqué et j'en parlerai dans un prochain article, mais je suis sûr d'une chose : quand un boss meurt, il devrait rester mort à moins que vous ne rechargez le jeu à un stade où vous ne l'aviez pas encore rencontré. Je ne suis pas un fan des bosses de toute façon, mais je dois reconnaître qu'il s'agit d'une convention standard, nous avons donc besoin de règles pour les implémenter. Un boss ne devrait pas être ressuscité à moins qu'il y ait une raison claire à cela (et une façon de l'en empêcher). Et un boss ne devrait certainement jamais réapparaître de la même façon que l'ennemi de base.

Commandes impossibles à configurer

Quelqu'un qui signe du nom de « Deathbliss » fulmine ainsi :

Je viens juste de pouvoir jouer à *Brute Force* sur Xbox et, devinez quoi ? Il n'est pas possible de configurer les commandes de la façon que je veux ! Ceci devient un problème de plus en plus important sur Xbox, et j'ai entendu dire que c'était la même chose pour la PS2. Le meilleur système de configuration des commandes que j'ai vu était celui de *Quake Arena 3* sur Dreamcast – c'est comme ça que cela devrait être fait. Dans un autre genre, sur la scène PC, nous avons droit à des jeux comme *Silent Hill 2* SANS UTILISATION DE LA SOURIS ! Ou bien comme *Gothic* où il est impossible de configurer ou d'utiliser la souris comme vous le voudriez.

Je ne pourrais pas être plus d'accord. C'est pire qu'une TDC ; n'importe quel gros jeu commercial qui ne permet pas au joueur de reconfigurer les commandes mérite de finir au pain sec et à l'eau. Ceci est particulièrement vrai s'il s'agit d'un jeu de réflexes nécessitant une grande dextérité, où un dixième de seconde compte. La possibilité de configurer les commandes est complètement triviale à implémenter, ne coûte presque rien en termes de RAM ou d'espace de stockage et rend votre jeu beaucoup plus accessible. Quand les game designers réaliseront-ils que la prise en compte des facteurs humains n'est pas juste réservée aux avions de combat ou aux centrales nucléaires ?

Ceci doit néanmoins être fait correctement. R. Alan Monroe nous a écrit pour se plaindre des...

Mauvais systèmes de configuration

M. Monroe écrit :

Quand je modifie les touches pour un jeu, comme pour un FPS, je peux cliquer sur action et appuyer sur une touche, mais comment puis-je savoir quelles touches sont déjà utilisées ? Il y a trois mauvaises façons de gérer cette situation : un message d'erreur qui indique que la touche est déjà utilisée ; la nouvelle action pique la touche à l'action à laquelle elle était affectée ; ou je finis par devoir lire une liste sans fin dans une minuscule fenêtre pour voir si la touche est déjà utilisée.

Pourquoi ne pas montrer une liste de toutes les touches non affectées ou bien afficher l'ensemble des touches en quatre ou cinq colonnes de façon à ce que je puisse toutes les voir d'un seul coup ?

Une fois de plus, le faire correctement est juste aussi simple que de le faire mal et ne vous coûte rien, alors, pourquoi ne pas le faire correctement ?

Checkpoints hors de propos

Celui-ci est un peu difficile à décrire de façon générique, je laisserai donc la parole à Jason Seip qui nous présente ceci dans un exemple particulier. Il explique que ceci lui est arrivé dans *Splinter Cell* sur Game Cube et que c'était « un mauvais moment dans un bon jeu. »

Je me trouvais dans un bâtiment qui était sur le point d'être détruit par une bombe. J'ai mal interprété mes ordres de mission, car je croyais que j'étais censé fuir alors qu'en réalité j'étais censé désamorcer la bombe. La sauvegarde A est intervenue au moment où on me parlait de la bombe et où on m'indiquait le temps qu'il me restait avant l'explosion. J'ai couru aussi vite que j'ai pu à travers le niveau, dépassant la pièce avec la bombe et me frayant un chemin malgré les ennemis. Juste avant que le compte à rebours arrive à zéro, je suis passé au niveau d'un checkpoint B et j'ai pensé avoir réussi, avant de voir le jeu se terminer cinq secondes plus tard parce que la bombe venait d'exploser. Quand j'ai rechargé ma partie, je me suis retrouvé au checkpoint B, de telle sorte que la bombe a re-explosé au bout de cinq secondes. Ceci ne cessait de se répéter, me laissant sans possibilité de revenir en arrière ou de résoudre le problème.

La morale, d'après lui, est :

Ne laissez pas le joueur atteindre le checkpoint B s'il n'a pas complètement rempli les objectifs au niveau du checkpoint A. Le checkpoint B aurait dû se trouver dans une zone complètement inaccessible au joueur tant que la bombe n'avait pas été désamorcée.

Jason s'est en fait retrouvé face à l'un des problèmes fondamentaux de la narration interactive linéaire : le problème du flot narratif. Comment s'assurer que le joueur a fait tout ce dont il a besoin au moment où l'histoire atteint son point culminant, son climax, quand vous ne contrôlez pas ses actions ? Il y a de nombreuses solutions, dont l'une des plus simples est la suivante : ne débloquent les portes vers le climax dramatique qu'une par une, de sorte que le joueur ne puisse pas s'y rendre tant qu'il n'est pas prêt à y aller. Jason s'est en fait retrouvé en avance sur l'histoire en sautant une étape, de sorte que l'histoire a pris un mauvais tournant et a abouti à une impasse.

Encore d'autres éléments qui détruisent la magie

Dans mon tout premier article « Vilain Game Designer ! Pas de biscuit pour toi ! », j'ai mentionné des éléments qui détruisaient la magie : des éléments du jeu qui sont ridicules ou inappropriés et qui détruisent la sensation d'immersion. Comme vous pouvez l'imaginer, ceci couvre un large panel d'erreurs. Gregg Tavares nous en propose quelques uns :

Une musique qui n'est pas adaptée au jeu

Dans de nombreux jeux, il est important d'établir un lien émotionnel avec le jeu. La musique aide à créer cette émotion quand elle est utilisée de façon adéquate (enregistrée comme dans un film, par exemple) et la détruit quand elle est mal utilisée (une musique « de jeu » générique et énergique, une musique connue, une musique choisie par l'utilisateur, etc.). [Je voudrais ajouter que la pratique si populaire qui consiste à utiliser des musiques complètement inappropriées pour les jeux est un échec esthétique aussi dramatique que pour les films. M.C. Hammer pour *La famille Addams* ? Pardon ? Le département marketing est devenu fou. – Ernest]

Des dialogues in-game qui font référence à des éléments du monde réel

Dans *Metal Gear Solid*, le Commandant appelle Snake sur le codec et dit quelque chose comme « Snake, pour ouvrir la vanne, appuie sur le bouton A de la manette. ». Hein ? Dans le monde réel, personne ne ferait référence à la manette, parce que celle-ci n'existe qu'à l'extérieur de l'univers du jeu. *Grand Theft Auto* proposait une meilleure implémentation de ceci. Dans GTA 3, les personnages du jeu restaient dans leur personnage en permanence, et les explications liées au gameplay étaient affichées à l'écran, indépendamment des personnages.

Des situations extrêmement ridicules

Ceci arrive bien sûr dans de nombreux films hollywoodiens aussi. Par exemple, vous disposez de 18 heures pour arrêter une bombe atomique sur le point d'exploser. Oh, je sais, gaspillons de précieuses minutes à flirter avec un personnage in-game (*Metal Gear Solid*). Le monde sera détruit demain, alors allons au parc d'attractions (*Final Fantasy VII*). Je suis dans un coin avec un garde à à peine un mètre de moi, c'est le moment idéal pour une conversation avec le Commandant sur le codec (*Metal Gear Solid* encore).

Les cinématiques qu'il est impossible d'interrompre

Jerry Strand et « loganb » ont écrit tous les deux pour se plaindre des cinématiques qu'il est impossible d'interrompre. Parfois, il y a d'incroyablement longues introductions que vous devez subir chaque fois que vous recommencez le jeu ; parfois, des cut-scenes juste après un point de sauvegarde. Si vous vous faites tuer et devez recharger depuis ce point de sauvegarde, vous devrez revoir le film. J'y ajouterai ceci : les cinématiques de 15 secondes impossibles à interrompre qui surviennent chaque fois que vous lancez une invocation dans *Final Fantasy 8* (je n'ai pas joué aux autres jeux de la série, je ne sais donc pas s'ils en sont tous coupables [NDT : *ils le sont.*]). Elles sont impressionnantes la première fois, intéressantes les trois ou quatre fois suivantes, puis excessivement fatigantes au-delà. C'est à vous décourager de lancer un sort, voire de jouer au bout d'un moment.

Bon, en fait, j'aime les cinématiques dans les jeux (oui, je sais, ça ne respecte pas le Dogma 2001), tant qu'elles sont bien réalisées, courtes et intéressantes, compatibles avec le reste du jeu et qu'elles apportent véritablement quelque chose à l'intrigue. Hélas, la réunion harmonieuse de toutes ces qualités ne se produit que dans environ 25 % des cas. Le reste du temps, elles sont bêtes, inappropriées, voire les deux, et je veux un bouton pour les passer et aller directement jouer. Une fois de plus, la solution est triviale – alors, faites le !

Conclusion

Voici pour cette édition de Vilain Game Designer ! Pas de Biscuit pour toi ! Merci à tous ceux qui ont écrit et mes excuses pour toutes les suggestions pour lesquelles je n'avais plus de place. Mais continuez d'envoyer des e-mails (NDT : *en anglais*) à notwinkie@designersnotebook.com, parce qu'en un an, vous pouvez tous jouer à beaucoup plus de jeux et repérer beaucoup plus de TDC que je ne le peux.

[Lire la suite →](#)

DLCs et achats par micro-paiement

Par [Thomas Chassin](#) | le 10 juillet 2011 14:37:37 | Catégories : [Business](#), [Game Design](#)

Certains, en lisant ce titre quelque peu opaque, ont pu se demander « Mais de quoi il parle cet article ? ». Avant de m'étendre sur le sujet, je commencerai donc par clarifier ce point. Les DLCs, ou *downloadable contents* dans la langue de Shakespeare, sont des contenus additionnels téléchargeables pour un jeu qui reprennent le principe de l'extension, à la différence près que l'absence de support physique fait que ces contenus peuvent être gratuits (ou pas) et sont généralement téléchargeables depuis le jeu ou un support fourni avec (Steam, Xbox Live, PSN, etc.).



Derrière le terme barbare d'achats par micro-paiement, se cache l'acquisition de contenu utilisable dans le jeu, comme de l'or, des objets ou un déguisement de bisounours pour votre barbare, bref tout ce qui peut passer par la tête des responsables du projet, par le truchement d'une transaction réelle (comprenez « en balançant l'oseille »).

I. A quoi ça sert ?

Aha ! Je vous vois venir : « Ouais, mais je l'ai payé ce jeu, je vois pas pourquoi je paierai plus maintenant que je l'ai ! ». C'est le point de vue d'un joueur. Maintenant, imaginez que vous créez un jeu vidéo (si vous traînez sur le site, il est fort possible que l'idée vous ait titillé). Vous avez lancé votre projet amateur et aimeriez mettre un petit pécule de côté pour en réaliser un de plus grande envergure plus tard, ou bien vous avez lancé un projet pro et cherchez un financement qui ne freine pas pour autant la diffusion de votre jeu. Les donations ne suffiront pas nécessairement, et il est possible que vous ayez envie de vous autofinancer. Alors comment faire ? Je vous donne un indice: proposez du contenu payant optionnel au joueur, comme ça ceux qui apprécient votre jeu peuvent vous apporter un support financier et profiter de contenu supplémentaire, tandis que le joueur lambda qui n'a pas envie ou pas les moyens de payer joue quand même à votre jeu. Attendez, c'est pas ce dont on parlait il y a 5 minutes ? Mais si ! Voilà l'utilité de ces contenus payant: permettre au joueur qui apprécie le jeu d'avoir un petit plus, et permettre au studio de toucher de l'argent tout en lançant un jeu essentiellement gratuit.

II. Ah, mais c'est cool en fait

Eh bien, en théorie, on a très envie de dire oui. Malheureusement, ces contenus ne sont pas toujours bien implémentés dans le jeu, et on a parfois l'impression que le jeu est conçu pour pousser à l'achat. Il faut donc trouver un équilibre raisonnable pour mettre le plus grand nombre d'accord.

1. Les pièges des DLCs à éviter

A. Le jeu composé à 90% de DLCs

Il vous est déjà arrivé de boucler un jeu en peu de temps pour vous rendre compte que les 90% du jeu étaient composés de DLCs ? C'est un principe qui s'est développé avec le principe des versions d'essai, à la différence que la version d'essai est par définition même un échantillon de test d'un jeu plus vaste. Ce n'est pas un vrai jeu. Un vrai jeu doit fournir une quantité de contenu conséquente, ce qui signifie plus que les DLCs. Bien sûr, au bout de 24 DLCs, le jeu d'origine n'est qu'une petite partie d'un ensemble, mais c'est tout de même la plus importante des parties. C'est d'autant plus vrai si le jeu de base est payant, auquel cas le joueur se sentira floué, et risque de ne pas investir dans vos productions futures. Soyez honnêtes avec le joueur. Le DLCs a pour vocation d'être le carré de chocolat en plus qu'on donne au joueur qui à fini la tablette (comment ça elles sont boiteuses mes métaphores ?). C'est un plus qui vient enrichir l'expérience de base, pas une expérience de base qui vient enrichir plus le studio.

B. Les DLCs radins en contenu

Proposer du contenu additionnel payant au joueur est un droit légitime. Certains seront même heureux de pouvoir prolonger leur expérience contre quelques espèces sonnantes et trébuchantes. Cependant, à partir du moment où le jour paye pour du contenu, il est en mesure d'attendre une certaine qualité de celui-ci, de même qu'une augmentation raisonnable de la durée de vie du jeu. Ainsi, lors de la conception d'un DLC, il est important de prendre en compte la durée de vie ajoutée par le contenu avant de fixer un prix. Dans le cas d'un jeu payant, la durée de vie du DLC devrait être relativement élevée par rapport au prix, dans la mesure où le joueur a déjà payé pour le jeu, et accepte de payer plus. Un exemple à ne pas suivre est celui de *Call of Duty: Black Ops*. Sur console HD, le jeu coûte environ 60/70€, ce qui est un prix assez conséquent, et le DLC de 4 cartes multijoueur coûte la modique somme de 15€ supplémentaires. Bien sûr pour un joueur qui passe ses nuits sur le jeu, cela peut sembler raisonnable, mais pour le joueur lambda, c'est un prix scandaleux. Bilan des courses: les joueurs crient au scandale et ne se sentent pas respectés, ce qui fait que les studios ont mauvaise presse et que certains joueurs refusent de payer lesdits DLCs (qui se vendent tout de même comme des petits pains). L'autre abus est de bâcler la réalisation d'un DLC sous prétexte qu'il s'agit de contenu en plus. Le joueur paye, mais il en veut pour son argent. Le DLC doit donc théoriquement proposer du contenu de qualité supérieure ou égale au reste du jeu. Attention cependant à ne pas faire l'inverse en bâclant le jeu pour peaufiner le contenu payant.

2. Micro-paiement et équilibrage du jeu

A. Mais, non ce n'est pas obligatoire tousse

Les achats internes au jeu sont un contenu optionnel, et le jeu peut tout à fait être joué sans. En tout cas en théorie. Il existe des concepteurs vicieux, qui pour placer leurs petits coup de pouces payants, s'amusent à les rendre quasi indispensables. Vous ne voulez pas acheter de bonus d'expérience pour votre personnage ? Rien ne vous y oblige, après tout, vous pouvez très bien tuer 2000 monstres pour passer au niveau suivant, c'est gratuit. Vous ne trouvez jamais d'argent ou d'objets de valeurs sur les monstres ? C'est dommage que vous ne vouliez pas en acheter ! Dans le même genre, on peut compter les « retournez au dernier de checkpoint, avec moult pénalités si vous n'avez pas acheté d'ankhs (ou le quelconque objet qui vous sert de billet retour une fois envoyé du mauvais côté du Styx) », ou encore les jeux PvP où les joueurs ont tous payé pour acheter des bonus qui les rendent invincibles.



Ce n'est en aucun cas parce que l'achat de contenu peut faciliter le jeu que le gameplay doit être pensé contre le joueur. Idéalement, le jeu devrait être conçu et équilibré AVANT de penser à intégrer du contenu payant, autrement vous frustrerez les joueurs qui ne peuvent/veulent pas payer pour ledit contenu. De plus, si le joueur doit payer constamment pour avancer, il ne jouera pas longtemps et ne rejouera probablement jamais à votre jeu, et vous aurez mauvaise presse.

B. Achète Excalibur+5 et terrasse tout le monde !

L'ajout de contenu payant dans un jeu ne doit pas déséquilibrer le jeu en défaveur du joueur qui ne paie pas, cela a été dit. Mais alors si je vends des armes et des bonus au joueur qui paie, le jeu sera trop facile pour lui ? Eh ben non, pas forcément. Il faut proposer pour cela au joueur du contenu raisonnable. Ainsi, si vous vendez des armes au joueur, ne lui vendez pas des armes surpuissantes. Pensez également aux possibles combinaisons pernicieuses auquel le joueur pourrait penser: si vous proposez en parallèle des armes de haut niveau et des tomes d'expérience, le joueur qui craque son porte-monnaie traversera votre jeu comme un sabre traverse du fromage blanc. Mais rien ne vous empêche de lui imposer des restriction sur ses achats: pas de tome d'expérience avant d'avoir fini le jeu une fois par exemple.

Pensez aussi que le joueur voudra pouvoir adapter ses achats à ses besoins. Par exemple, le joueur de bas niveau préférera surement acheter plusieurs ankhs de base, plutôt qu'une ankh de réincarnation divine +4. Il peut pour cela être préférable de proposer au joueur d'acheter une monnaie spéciale, plutôt que d'acheter directement les objets. Pensez cependant à proposer des objets simples à bas prix pour que le joueur puisse dépenser l'intégralité de son argent: rien de plus frustrant que d'avoir 99 crédits quand le premier prix est à 100 crédits.

III. Quoi !? Je dois vous faire une conclusion !?

1. Récap et morale de l'histoire.

Bon, va pour la conclusion. Les DLCs et autres contenus payants ne sont ni bons ni mauvais de manière objective. Ce sont des éléments du jeu qui deviennent ce que vous choisissez d'en faire et influent sur le jeu en fonction des choix d'implémentation que vous faites. De manière générale, avant de proposer du contenu supplémentaire, finissez le contenu basique, et finalisez le. Le contenu additionnel vient toujours en dernier, un peu comme la cerise sur le gâteau, ou la fiente en plein milieu de la tarte, selon ce que vous en faites. Comme pour tout, et je dis bien TOUT, ce contenu doit être soigneusement dosé, et dispensé avec parcimonie, faute de quoi vous aboutirez à un résultat néfaste pour le joueur, donc pour votre image, donc

pour vous (à moins que vous soyez un névrosé aigri désireux de frustrer autant de joueur innocents que possible).

2. Maintenant que les gosses sont couchés, parlons business

Les chiffres de ce paragraphe sont ceux donnés par l'[AFJV](#).

Concrètement, quel place y a-t-il sur le marché pour les DLCs et contenus achetables par micro-paiement ? Une chose est sûre, les jeux gratuits, de par leur accessibilité, ont une portée immense et peuvent facilement devenir populaires, pour peu que la qualité soit au rendez-vous. Sur l'ensemble des joueurs de jeux gratuits, la tranche des joueurs payant se situe entre 5 et 10% de la population totale. Les autres joueurs permettent une promotion gratuite et participent à leur façon à la réussite de votre jeu.

Les revenus générés par les jeux gratuits proviennent à 90% de la vente de contenu (objets, bonus, etc.) ou d'unité monétaire et atteignent en moyenne 0,70 à 1,50€ par joueur et par an, mais sont de l'ordre de 20 à 27€ par joueur payant par an. Pour garantir la rentabilité optimale d'un jeu, il est conseillé de mettre à disposition des utilisateurs une multitude de moyens de paiement, et de l'adapter à votre population cible (paiement par SMS, carte prépayée, appel téléphonique pour les mineurs par exemple).

Il faut également garder en tête que les dépenses varient en fonction du type de jeu proposé. Par exemple, dans le cas d'un MMO F2P (*Free-to-Play* ou MMO gratuit), les dépenses sont émulées par les interactions entre joueurs, que ce soit pour être mieux armé que le voisin, ou pour frimer avec le familier que tout le monde veut.

De plus, les jeux vidéos sur smartphone sont un secteur à part, mais prometteur car adapté aux micro-transactions en permettant des transactions rapides et simple (via un quelconque *Application Store* généralement) rendant l'accès au contenu simple pour le joueur. Rappelez vous que plus le paiement est complexe à effectuer, moins le joueur sera prêt à mettre la main au port monnaie.

Qui plus est, les jeux gratuits financés par la vente de contenu sont un marché en pleine explosion. Grâce à la gratuité de la majorité du contenu, ces jeux disposent d'une capacité de diffusion largement supérieure aux jeux payants. En témoigne l'explosion constatée dans le secteur des MMO F2P, qui peuvent facilement dépasser le million d'abonnés, à tel point que certains jeux anciennement payant ont changé leur modèle économique pour le F2P tout en annonçant une hausse spectaculaire des revenus : *Lord of the Ring Online* annonce avoir doublé ses revenus et *Donjon & Dragons Online* avoir multiplié par 5 ses revenus. Le système des micro-paiements apparaît donc comme prometteur et plein d'avenir, grâce à sa souplesse.

Remerciements :

- à Antoine Gersant, pour ses relectures et son aide.
- à Christophe Kopler, pour m'avoir aidé dans mes recherches.
- à Pascal Luban, pour ses articles riches d'enseignements.

Pourquoi les fins de jeu craignent

Par [Thomas Chassin](#) | le 17 juillet 2011 13:12:47 | Catégories : [Game Design](#), [Scénario](#)

Cet article est une traduction de [Thomas Chassin](#).

L'[article original](#) est écrit par Tom Jubert publié par Plot is Gameplay's Bitch (avril 2011).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

La raison pour laquelle les fins de jeu craignent (pour la plupart, naturellement) a déjà été traitée, mais je voulais en parler dans une perspective spécifiquement narrative.

Donc. Les fins de jeux craignent parce qu'elles sont le dernier élément développé, qu'il n'y a plus de budget, que tout le monde est crevé et parce que la plupart des joueurs ne les regardent jamais de toute façon. Il y a deux choses qui me frappent particulièrement : pourquoi les joueurs ne regardent-ils jamais la fin, et pourquoi personne ne parle-t-il de l'histoire ?



Premièrement, je suis un fervent partisan des jeux courts. Les critiques passent beaucoup de temps à estimer si un jeu vous en donne pour votre argent et je peux comprendre cela, mais en réduisant l'équation à « 10h de jeu = 20€ », on s'expose au risque de la promotion du design bâclé. Je l'ai déjà dit, mais je préfère avoir 3h de *Portal* pour 35€ plutôt que de payer une dizaine d'euros pour une plus d'une centaine d'heure de grinding dans un quelconque RPG, et je prendrai 20h de *Mass Effect 2* plutôt que 40h de *Dragon Age*.

Prenez par exemple *Uncharted 2*. Il a une des *production value* les plus élevées que j'ai jamais vues, et pourtant les développeurs ont ressenti le besoin d'allonger sa durée de vie avec des méchants en carton dont les points de vie augmentent sans fin. Le résultat est un dénouement auquel manquent le spectaculaire, le script et le fun du reste du jeu – au nom d'une ou deux heures de bêtises supplémentaires. C'est un cercle vicieux : les joueurs demandent plus de contenu en milieu de jeu, ce contenu est alors dilué, moins de joueurs prennent la peine d'aller jusqu'à la fin, et donc plus d'attention est portée au contenu de milieu de jeu.

Mais à part en supprimant tous les bouche-trous, comment peut-on apporter une conclusion satisfaisante à un jeu dont la majorité du public n'atteint jamais la fin ?

- Supprimez la fin. Certaines des expériences narrative les plus fortes sont générées dynamiquement dans des jeux comme *SimCity* ou *Les Sims*.
- Multipliez les fins possibles. Des jeux comme *Weird Worlds* ou *Atom Zombie Smasher* encouragent de courtes sessions de jeu d'une heure, mais arrivent à maintenir un déroulement dramatique à mesure que différentes routes sont empruntées par le joueur.
- Forcez la fin du jeu. A moins de lâcher la manette, il est plus ou moins impossible de jouer plus de 7 ou 8 heures à *Heavy Rain* sans en voir la fin.



Enfin, je pense que la clé du problème réside dans la façon dont on approche la fin du jeu : on l'approche du point de vue du gameplay, et non d'un point de vue narratif.

Quel rapport une « fin » à-t-elle avec le concept de gameplay ? Ce n'est pas une chose absolue – la plupart des jeux encouragent à continuer à jouer après avoir été officiellement terminés en proposant de nouveaux challenges, du multijoueur et bien d'autres choses. Peut-être doit-on le voir comme une question d'échelle, de montée en puissance ; mais évidemment les jeux surenchérisent toujours en termes de taille et en d'intensité — ils semblent capables de faire monter la tension tout au cours du jeu, mais font souvent un bide lorsqu'il n'est plus question de faire monter la tension mais d'atteindre un pic. C'est ce pic que le développement tardif peine souvent à atteindre.

Et ce pic est aussi responsable de l'embarrassant boss de fin de jeu dont-on-doit-encore-discuter. Cela m'abasourdit que tant de jeux intelligents et bien faits se contentent de réduire la conclusion de leur histoire dramatique à cribler de balles un gros bonhomme bleu. Lorsqu'on s'intéresse à la conclusion dans un contexte de gameplay, quel choix a-t-on ? On ne peut pas juste envoyer quelques méchants supplémentaires au joueur, mais on ne peut pas non plus en envoyer, il faut donc offrir quelque chose de différent. Quelque chose comme, je ne sais pas, un énorme type orange avec un point faible qui clignote.

Si on jette un autre regard à tout ceci d'un point de vue narratif en revanche, peut-être doit-on se demander « Pourquoi avoir une fin apparentée au gameplay ? ». Je fais mon possible pour lier ensemble histoire et gameplay, mais si le faire signifie abandonner toutes les fantastiques mécaniques polies par les développeurs et qui ont marché tout au long du jeu au profit de grosses boules bleues à 1000 PV, quel intérêt ?

Une fin est un problème d'histoire. Je pense que si nous-autres les développeurs avons appris à faire confiance au gameplay et à la narration pour accomplir leurs tâches respectives, nous pourrions faire une fin

de jeu satisfaisante en laissant le gameplay sur sa lancée et en utilisant notre temps et notre budget pour lui donner de l'ampleur avec la narration. Le boss final de *Portal* – du point de vue du gameplay – était nul. Mais l'histoire l'a transformé en une conclusion satisfaisante. Les fins des deux premiers jeux *Penumbra* étaient les éléments dont j'étais le plus fier, mais elles n'exploitaient le gameplay que de façon superficielle.



Une de mes astuces favorites quand j'apprenais à écrire était « Commence au début, et continue d'écrire jusqu'à ce que tu aies fini. » Cela peut sembler parfaitement stupide, mais ce que cela signifie, c'est qu'il est inutile de construire artificiellement des tenants et des aboutissants à votre histoire. Si vous ne vous préoccupez que de l'écriture, les choses se feront d'elles mêmes.

Vilain Game Designer ! Pas de biscuit pour toi ! VI

Par [Pierre Guyot](#) | le 24 juillet 2011 17:28:39 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juin 2005).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



Et c'est reparti ! Les messages sont arrivés, les gens ont parlé et j'ai maintenant une demi-douzaine d'autres erreurs de game design à vous présenter, pour que les évitiez. Je suis toujours très intéressé par le sujet, donc si vous pensez à certaines, prenez soin de m'envoyer un message à notwinkie@designersnotebook.com (NDT : en anglais).

Comme toujours, une mise en garde : même les bons jeux peuvent contenir de mauvais choix de game design. Ce n'est pas parce qu'un jeu est cité ici qu'il est mauvais. Parfois, c'est juste un bon jeu avec une erreur à un moment.

Et passons maintenant aux TDC.

Le Mode Facile est suppose être Facile, bordel !

Dans l'un de mes précédents articles sur les erreurs de game design, je m'étais plaint des jeux qui n'offrent pas plusieurs modes de difficulté. Maintenant, je réalise que cela n'est pas approprié pour tous les genres de jeux ; mais c'est habituellement possible pour les jeux d'action ou de stratégie. Offrir plusieurs niveaux de difficulté, disais-je, élargit le marché et rend votre jeu accessible à des gens qui n'y auraient pas joué sinon.

Ce TDC en est un corollaire, envoyé par Christopher Kempke. Il écrit :

« J'ai remarqué que dans beaucoup de jeux récents (et dans presque tous les shooters), il semble y avoir une volonté profonde de vous empêcher de finir le jeu. Vous pouvez arriver jusqu'à la dernière pièce, le dernier niveau, le dernier boss, mais cette pièce, ce niveau, ce boss doivent représenter « un challenge ». »

« J'ai tendance à jouer aux jeux dans le mode de difficulté le plus bas, parce que je ne peux pas passer ma vie sur un jeu. Mais je voudrais le terminer et j'ai choisi le Mode Facile pour une certaine raison. Si je suis capable de finir les quinze premiers niveaux avec une demi-douzaine de morts, il semble raisonnable que je puisse finir le dernier niveau en 2 ou 3, ou au pire 5 ou 6 tentatives. Mais non ! Généralement, la dernière pièce, le dernier niveau ou le dernier boss sont grosso modo vingt fois plus forts que tout ce à quoi vous avez pu être confrontés jusqu'à présent, nécessitent des techniques de combat complètement différentes ou se situent tout simplement hors de portée de mes capacités. »

« *Far Cry* est le pire exemple que j'ai vu récemment : on vous donne un plein stock d'armes et de boucliers, une porte se ferme derrière vous et vous êtes maintenant face à une douzaine des ennemis les plus puissants du jeu. A aucun moment du jeu auparavant on ne fait face à plus de trois ennemis à la fois (en mourant presque à l'occasion). Et là, il y en a huit, sans compter les trois ou quatre ennemis supplémentaires qui vous tirent dessus dans l'ombre. Les tuer tous (probablement au prix de toutes vos munitions) ne fait qu'ouvrir

une nouvelle porte qui en relâche encore plus. Et bien sûr, il s'agit d'un *gantlet* (NDT : un mot tiré du français "gantelet" qui désigne une forme de châtiment au cours duquel une personne est force de courir entre deux rangées d'hommes se faisant face et armés de bâtons ou de fouets pour frapper la victime. Décidément, certains mots sont charmants :3) ; vous ne pouvez rien faire d'autre qu'ouvrir les portes une par une, en sachant que chacune cache une armée que vous n'avez plus les moyens de combattre.

« Je crois que je fais en moyenne deux jeux par an où je peux faire 99% du jeu sans aide, mais où j'ai besoin d'un code pour la toute dernière partie. Ça me rend dingue, et ce n'est pas amusant. Si je voulais subir une telle expérience, j'aurais sélectionné un mode de difficulté plus élevé. »



En fait, je n'ai pas de problème avec le fait de devoir utiliser des techniques différentes contre les boss – c'est une convention du jeu vidéo qui est vieille d'au moins 25 ans et elle ajoute du challenge et de l'intérêt au gameplay. Mais je pense que Christopher a tout à fait raison en ce qui concerne le problème de la difficulté. Ce TDC est un bon exemple d'une erreur classique de design : ne pas réussir à donner au joueur ce qu'il veut réellement. Si le joueur choisit le Mode Facile, c'est qu'il veut un jeu qui soit facile, à jouer et à terminer.

« Mais », entend-on s'écrier une armée de game designers à l'ancienne, « ça veut dire qu'il va finir le jeu trop vite ! ». Ah oui ? Trop vite par rapport à quoi ? Vous voulez dire trop vite pour votre ego, parce que vous travaillez encore avec cette notion dépassée selon laquelle le joueur est votre adversaire. Il ne l'est pas. Il est votre public, la personne que vous essayez de distraire et à qui vous essayez de procurer du plaisir. Et si cela veut dire que cela doit lui prendre deux heures de terminer le jeu, soit ; c'était son choix. Il peut toujours revenir en arrière et rejouer dans un mode de difficulté plus élevé.

Vibrations ou retour de force mal implémentés

Celle-ci m'a été envoyée par Adam Doyle :

« Je déteste quand la manette vibre à la moindre action. Est-ce qu'une vibration à chaque pas est nécessaire ? C'est tout particulièrement un problème dans les jeux de combat et les FPS, où chaque impact ou tir fait vibrer la manette. Quand c'est bien fait, ça peut contribuer à une meilleure immersion, mais si les vibrations sont trop fréquentes, elles perdent de leur impact parce que les joueurs s'y habituent. Un bon exemple serait le nouveau *Serious Sam* sur console ; mais beaucoup de jeux de tir ou de combat (comme *Tekken 4*) utilisent beaucoup trop la vibration. »

« Il y a aussi le problème d'une mauvaise gestion de la force de la vibration. Une petite vitre se brise et le contrôle vous saute des mains ? Ou un entrepôt explose avec votre personnage à l'intérieur et tout ce que

vous sentez, c'est un léger mouvement ? Une fois de plus, ceci brise la connexion entre le joueur et le jeu. Avec des volants utilisant le retour de force, la roue peut vibrer si violemment qu'elle force le joueur à la lâcher et à attendre que ce soit fini pour reprendre le jeu. Un exemple en est la nouvelle simulation *Enthusia Racer*. Le volant vous saute littéralement des mains. »



« Non seulement une mauvaise utilisation de la vibration fait sortir le joueur du jeu, mais cela peut aussi leur faire perdre le contrôle en lui faisant rater les boutons ou les actions, ou faire glisser ses doigts sur le stick analogique dans le feu de l'action. Il y a aussi des problèmes de santé dans les cas extrêmes : de longues périodes de forte vibration provoquent un engourdissement des doigts et des mains. Les développeurs doivent savoir que ce n'est pas parce que la manette peut vibrer qu'il est nécessaire d'implémenter une fonction vibration. Selon moi, il est préférable de pas implémenter la vibration ou le retour de force que de mal les implémenter. »

En fait, on peut généraliser cette dernière phrase à à peu près tout dans les jeux : une mauvaise fonctionnalité, c'est pire que pas de fonctionnalité du tout. Comme je l'ai déjà dit à de nombreuses reprises, si vous ne pouvez pas le faire bien, ne le faites pas.

Bien vu, Adam. La vibration n'est pas juste un gadget que vous mettez dans le jeu juste parce que vous le pouvez. Ce devrait être une rétroaction intelligente pour le joueur, qui fait partir de l'interface utilisateur et qui est donc de votre ressort en tant que game designer. Prenez le temps d'y penser soigneusement et de le faire bien.

Les angles de caméra qui rendent le jeu impossible

Adam m'a aussi proposé une seconde TDC, les angles de caméra qui rendent le jeu injouable – généralement dans l'espoir de produire une « expérience cinématique ». Il écrit :

« Le problème avec beaucoup de survival horrors (et d'autres genres aussi) est que les développeurs, à la recherche de cette « expérience cinématique » choisissent de placer la caméra de façon à bien cadrer la scène, mais en rendant le jeu complètement injouable. Si votre personnage combat des tonnes et des tonnes d'ennemis, pourquoi donc placer la caméra au ras du sol avec un angle dynamique ? Effectivement, ça peut être joli, mais les joueurs se moquent complètement de savoir si c'est joli s'ils n'arrivent même pas à jouer. »

En fait, j'aime beaucoup les angles de camera inhabituels et tout ce qui crée un intérêt visuel et brise la monotonie de la vue à la troisième personne. Mais Adam a raison : le jeu doit rester jouable ! Des angles de caméra recherchés sont intéressants pour des genres de jeux plus lents, comme les jeux d'aventure ou d'action-aventure s'ils ne sont pas trop difficiles, comme *ICO*. Mais si vous vous battez pour survivre, il est nécessaire de voir clairement ce qui passe.

Mauvais manuels et mauvais tutoriels

T.C. Fox a écrit pour dire :

« Je suis l'un de ces joueurs qui lit toujours le manuel et qui aime les tutoriels complets et informatifs avant de se lancer dans le jeu. J'ai beaucoup apprécié les tutoriels dans *Gran Turismo 4*. Néanmoins, après une longue absence du monde du golf, j'ai décidé de tester *Tiger Woods 2004* et j'ai été consterné par le manque d'information au sujet des points plus avancés du jeu (que ce soit le jeu de golf ou le jeu vidéo lui-même). Le tutoriel auquel on vous fait jouer avant toute chose est hâtif et superficiel. Le manuel est épouvantablement fin. Ceci a constitué une première expérience de jeu incroyablement frustrante, qui m'a presque poussé à ramener le jeu au magasin. »

Les manuels semblent un peu dépassés aujourd'hui, dans la mesure où ils ont été pour la plupart remplacés par des niveaux tutoriels. Même si c'est le cas, le manuel offre une fonctionnalité que le tutoriel ne peut pas proposer : vous pouvez le consulter lorsque le jeu est en pause. Ceci est aussi valable pour un joueur qui aime savoir ce à quoi il doit s'attendre avant de se plonger dans le jeu, plutôt que de passer par la case essaye-échoue-recommence. Cette phase est caractéristique d'un certain type de gameplay hardcore (« tire et vois ce qui se passe ») avec lequel beaucoup de casual gamers ne se sentent pas à l'aise et, plus important, pour lequel ils n'ont pas le temps.

Tiger Woods est un jeu qu'un casual gamer ou même un non-joueur pourrait essayer. Quelqu'un pourrait l'offrir comme cadeau d'anniversaire à un aficionado du golf qui n'aurait jamais touché à un jeu vidéo. Ce genre de joueur a besoin – et attend – une introduction décente à cette expérience. Si vous le frustrez à ce niveau, vous mettez en l'air votre chance de lui vendre l'édition de l'année prochaine.

Je sais que les manuels coûtent de l'argent et prennent de la place dans la boîte ; c'est pourquoi nous les avons réduits à presque rien, si ce n'est une double page, de nos jours. Mais si vous voulez atteindre un marché autre que celui des traditionnels hardcore gamers, qui ont tout le temps et toute la patience du monde, vous lui devez un manuel et des tutoriaux décents.

Mauvaise conception de l'écran splitté

C'est un problème majeur dans les jeux de course console, où l'écran doit être divisé entre les joueurs. Mat Lamarche écrit :

« En mode Un Joueur, les écrans sont grands et beaux, vous avez une vue immense sur la piste et tout ce qu'il y a sur la route... Mais j'attends de voir un jeu qui gère correctement la division de l'écran en mode Multijoueur. Ces stupides indicateurs de temps (de combien vous gagnez ou perdez) apparaissent soudain au milieu de l'écran et vous ne pouvez quasiment plus rien voir sur la route ! »

Il s'agit juste d'un cas particulier d'un problème plus général : le design d'interface avec un écran splitté est difficile. Et c'est sur le point de devenir plus difficile, à mesure que nous passons à une génération de télévisions HD. Avec une plus grande résolution, on pourrait s'attendre à ce que cela soit plus facile, mais hélas, pour les cinq prochaines années ou quelque chose comme ça, il y aura des millions de joueurs qui n'auront pas de télévision HD. Vous devrez concevoir des jeux qui soient aussi jouables sur leurs télévisions, ou perdre ce marché.

Tant que vous y êtes, gardez les pop-ups hors de la route ! Mettez sur le côté ou quelque chose dans ce goût-là.

Des caisses sans palettes et sans chariots élévateurs (des objets dans des endroits incongrus)

Les caisses constituent déjà une blague récurrente dans le monde des jeux vidéo, parce que nous en voyons tout le temps. Le site [Old Man Murray](#) comporte un passage en revue hilarant d'un certain nombre de jeux, basé sur le temps que vous devez passer sur le jeu avant de voir une caisse – à ce moment précis, déclare l'auteur, c'est que le game designer n'avait plus d'idées. Plus vous pouvez jouer longtemps sans voir une caisse ou un tonneau, meilleur est le jeu (NDT : assez incroyablement, le temps caractéristique avant d'apercevoir une caisse est, dans les jeux PC, de l'ordre de la seconde !).



Le même article pointe du doigt le fait qu'il n'est pas possible de déplacer des caisses sans chariots élévateurs et sans palettes pour poser les caisses dessus. S'il y a des caisses dans un endroit, il vaudrait mieux qu'il y ait des palettes en dessous et au moins un chariot élévateur aussi. En fait, quelqu'un m'a écrit (malheureusement, j'ai perdu son nom dans un crash de ma boîte e-mail) et m'a fait remarquer que les caisses en bois sont de toute façon complètement dépassées de nos jours. Le transport par bateaux se fait actuellement avec des piles de cartons maintenus ensemble par un film plastique industriel très résistant. Le bois est lourd et cher, alors que le carton est léger, économique et recyclable. Pourtant, nos FPS nous montrent encore cette technologie vieille de 40 ans, mais dans des jeux de science-fiction futuristes.

Je ne peux pas prétendre que des caisses sans palettes constituent une TDC, parce que les types de Old Man Murray y ont pensé en premier, mais je peux généraliser ce cas particulier au problème d'objets se trouvant dans des endroits incongrus, un sujet dont plusieurs personnes se sont plaintes à moi. Maintenant, bien évidemment, ceci ne s'applique pas à des jeux de plates-formes pour lesquels il est courant de voir des caisses (parmi beaucoup d'autres objets) flotter au milieu du vide. Les caisses en lévitation font partie des conventions du jeu de plate-forme ; je comprends ceci et voilà peut-être pourquoi ils n'ont pas besoin de palettes ou de chariots élévateurs. Mais si votre jeu veut parler du monde réel, si vous voulez recevoir des tonnes de retours sur ô combien excitant et crédible est votre scénario, alors essayer de ne pas créer de pièces sans porte, pleines de caisses qui apparemment sont arrivées par un conduit d'aération. Essayez de ne pas mettre de barils dans un système d'égout où tous les passages sont plus petits que les barils. Essayez de ne pas mettre une caisse contenant une demi-tonne d'armes et d'armures dans une petite grotte nichée dans le flanc d'une falaise de 150 mètres de haut. Et essayez de penser à quelque chose de plus intéressant que des caisses ou des barils pour servir de containers pour les kits de soin et les munitions.

Conclusion

C'est tout pour cette année : six choses stupides de plus à ne pas faire dans votre jeu. Certaines d'entre elles, comme les mauvais angles de caméra, tuent la jouabilité. D'autres, comme une vibration mal réglée, vont juste irriter le joueur sans raison. Dans tous les cas, prenez garde à ces avertissements de peur d'être, vous aussi, privés de biscuit. Merci à toutes les personnes qui ont contribué à cet article et encore une fois, continuez d'en envoyer !

[Lire la suite →](#)

Blending et Shading

Par [Pierre Guyot](#) | le 31 juillet 2011 15:48:22 | Catégories : [Couleur](#), [Cours](#), [Dessin](#), [Graphisme](#), [Photoshop](#)



Que de noms barbares pour un simple titre d'article ! Mais sachez que derrière ce jargon improbable sa cache la base de beaucoup d'illustrations sur ordinateur. Le **blending**, c'est pour ainsi dire la façon de mélanger ses couleurs et le **shading**, celle de poser les ombres et les lumières de façon à créer le volume. Dans cet article, nous tenterons de donner une introduction très simple à la mise en couleur et à l'illustration numériques.

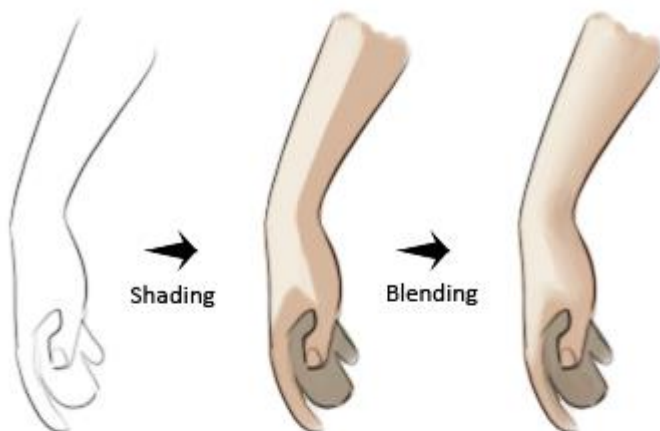
Quelques définitions

Tout d'abord, précisons de quoi nous allons parler.

Le **shading** consiste à poser les ombres et les lumières sur un objet de façon à rendre le volume et le relief de celui-ci, en tenant compte de la direction des différentes sources de lumière.

Le **blending** consiste à réaliser un dégradé entre deux couleurs, à les mélanger d'une certaine façon pour que la transition entre les deux ne soient pas trop brutale, si possible en reflétant la texture et le relief de l'objet.

Ça paraît tout bête et pourtant, ce sont sans doute deux des actions qu'on effectue le plus souvent lorsqu'on met en couleurs ou qu'on réalise une illustration sur ordinateur (ces principes s'appliquent aussi aux médiums traditionnels, mais d'une façon différente dont je ne traiterai pas ici). Une image sera sans doute plus parlante qu'un long discours :



Ici, j'utilise un **line** ou **lineart**, c'est-à-dire un dessin mis au propre traditionnellement (à l'encre par exemple) ou numériquement sous forme de traits bien nets. Le shading consiste à déterminer comment vont être

placées les différentes zones d'ombre et de lumière sur l'objet, en fonction de la position, de la couleur et de l'intensité de la source de lumière. Des à-plats de couleur façon celluloïd peuvent très bien suffire. Le *blending* consiste ensuite à raffiner les transitions entre les différentes zones pour donner à l'ensemble un relief plus fin.

Le shading

Maintenant, comment procéder ?

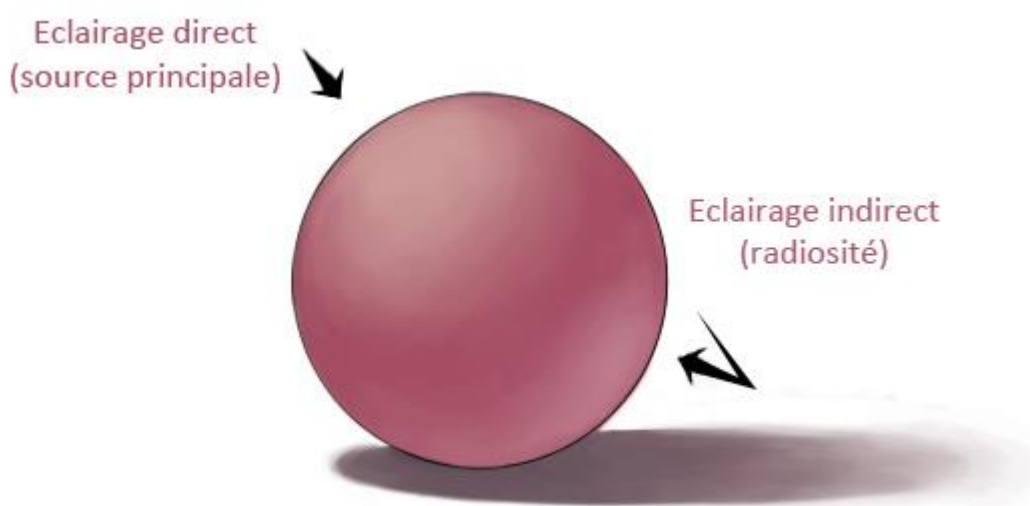
Tout d'abord, pour pouvoir poser ombres et lumières, il convient avant toute chose de choisir une ou plusieurs sources de lumière. Si vous ne savez pas où se trouve la source lumineuse, vous placerez vos ombres et vos lumières au hasard, ce qui donnera un très mauvais volume à l'ensemble. De plus, le positionnement et la couleur de la lumière conditionnent grandement l'ambiance et le « feeling » d'une illustration.

Ensuite, une fois choisie votre source lumineuse et votre palette de couleurs (je vous renvoie plutôt du côté de Fil (NDLR : Antoine Gersant) pour choisir cette dernière :]), vous pouvez commencer à poser vos à-plats. Essayez de réfléchir physiquement à l'endroit où vient frapper la lumière et penser les différents volumes comme des reliefs simples. N'oubliez pas de traiter certains éléments : les yeux, les cheveux, les ongles sont aussi affectés par la lumière, il ne faut pas l'oublier.

Parlons maintenant de quelques phénomènes physiques particuliers liés à la lumière et qui ont une influence sur la façon dont on applique le shading.

La radiosité

La **radiosité** est le phénomène qui se produit quand de la lumière rebondit sur une surface pour venir en éclairer une autre. En général, cet éclairage indirect est plus faible que l'éclairage principal et on ne le voit donc apparaître que dans les zones laissées dans l'ombre par l'éclairage principal. Il donne ainsi l'impression que les ombres sont plus claires. Pour ceux qui lisent les tutoriaux de Fil sur le pixel art, c'est un cas particulier d'**ambient lighting**.

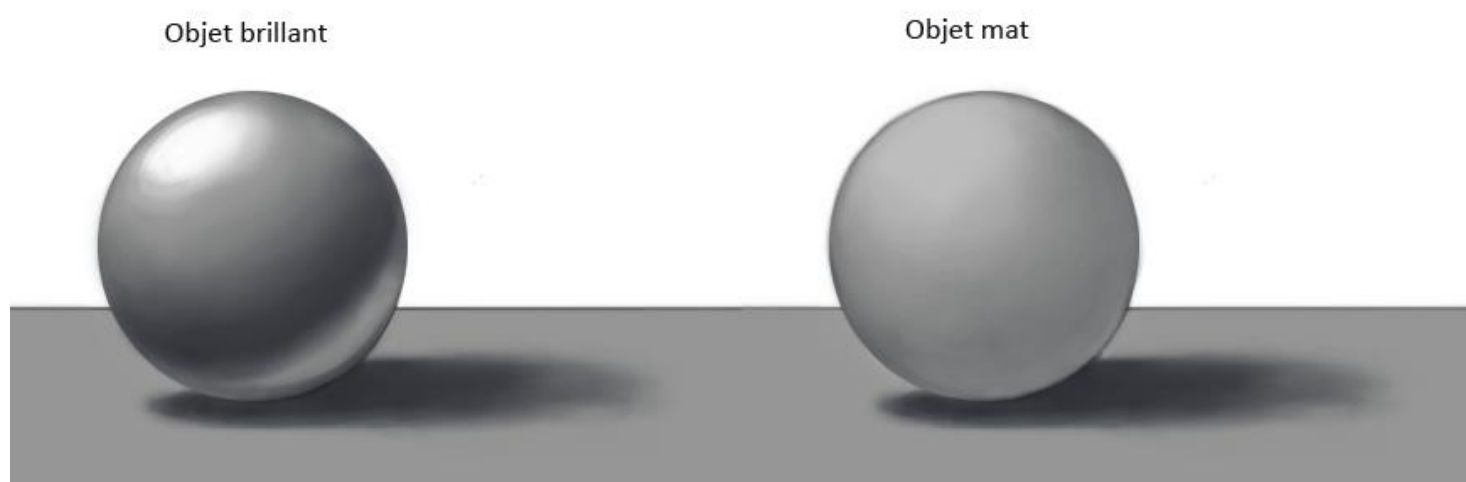


Regardons cette petite sphère. L'éclairage direct, en haut à gauche, laisse la partie en bas à droite de la sphère dans l'ombre. La lumière ambiante rebondit sur le support blanc et vient éclairer le bas de la sphère qui devient plus claire. La sphère reflète donc le support à la manière d'un miroir, mais beaucoup plus faiblement. Une petite remarque sur le choix des couleurs : j'ai observé qu'en général, quand il y a une

radiosité « ton sur ton » comme une sphère rose sur un support rose, la couleur de l'éclairage indirect est plus saturée.

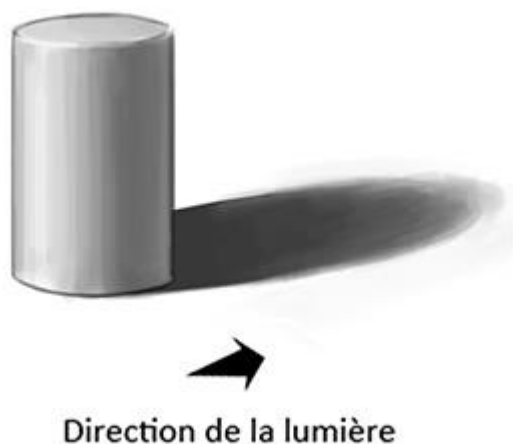
Mat ou brillant ?

Le caractère **brillant** ou **mat** d'un objet dépend de la façon dont il reflète la lumière. Pour un objet mat, la couleur la plus claire, le **highlight**, ne sera pas très claire et la palette sera peu contrastée. A l'inverse, un objet brillant aura un highlight très clair et une palette très contrastée. Pour les objets mats, les effets de radiosité sont faibles, alors que pour les objets brillants, ils sont très prononcés, au point d'en faire de véritables miroirs. Il est donc très utile, pour un objet brillant, de savoir dans quel environnement il se trouve, dans la mesure où il va refléter très fortement les couleurs de cet environnement. Essayez d'observer les différences entre objets mat et brillant sur l'exemple qui suit.



La diffraction

La **diffraction** est un phénomène dont vous avez peut-être entendu parler dans le cadre d'un cours de physique. Pour faire simple, disons que c'est le phénomène qui fait que lorsqu'on s'éloigne de la source de lumière, les limites entre ombres et lumières sont moins bien définies et deviennent plus floues. Ceci permet de suggérer le fait que l'on s'éloigne de la source de lumière, par exemple dans le cadre d'une ombre portée.



Si on observe l'ombre du cylindre sur l'image précédente, on se rend compte qu'elle se « dégrade » au fur et à mesure qu'elle s'éloigne de la source lumineuse, les « bords » de la zone d'ombre deviennent de plus en plus flous, jusqu'à ce que l'ombre disparaisse complètement. Selon la source de la lumière et la situation,

ceci se fait plus ou moins vite et je vous conseille d'essayer d'observer le phénomène en vrai pour vous en rendre compte.

Les objets transparents

Le verre et les matières transparentes sont particulièrement difficiles à peindre. Pour y arriver, quelques principes peuvent être utiles. Tout d'abord, un objet transparent laisse apparaître le fond derrière lui, il n'a donc souvent pas de couleur propre — ou une légère teinte — et c'est surtout la façon dont il déforme l'environnement qui permet de le repérer. Par exemple, pour peindre un verre, on ne verra pratiquement que les bords du verre et son pied, qui renvoient des highlights et déforment fortement le fond, alors qu'il n'y a presque pas besoin de peindre le ballon (si ce n'est un léger reflet de l'environnement alentour).

Les ombres portées des objets en verre fonctionnent d'une façon un peu particulière. Leur ombre ressemble à une ombre normale, sauf qu'en raison des propriétés optiques du verre, tous les rayons qui traversent l'objet ont tendance à converger vers un même point (comme avec une loupe) qui sera très lumineux et dont la couleur sera très saturée. C'est ce qu'on appelle la **caustique**. Pour un verre de vin par exemple, on aura une ombre portée sombre en forme de verre et vers le bas du ballon, on verra une tache lumineuse rouge très vive et très claire. Voici un exemple.



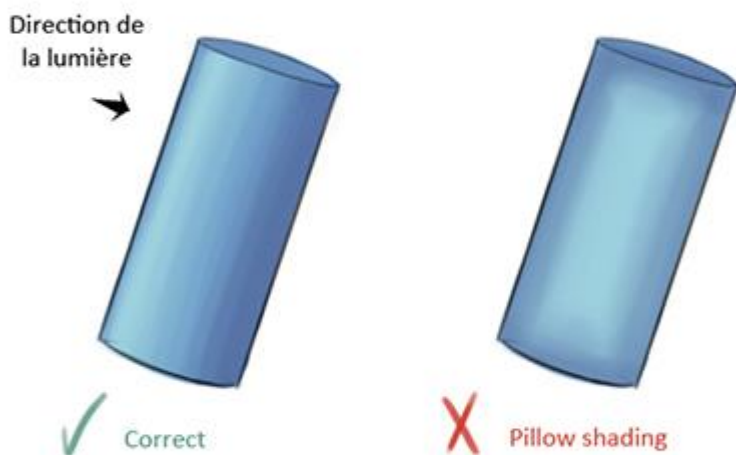
Vous pourrez remarquer que j'ai ici essayé de mettre en pratique la caustique, les reflets lumineux sur le verre et la diffraction de l'ombre.

Le pillow shading

Il faut absolument éviter ce qu'on appelle le **pillow shading**. Si vous avez déjà lu les tutoriels de Fil sur le pixel art, ce nom devrait vous être familier : il s'agit de la tendance à coloriser les objets systématiquement de la même manière sans tenir compte de la lumière, en plaçant les zones claires au centre et les zones sombres sur les bords. C'est une très mauvaise chose, que ce soit en pixel art ou ailleurs, parce que cela tend à tuer tout le volume du dessin, pour deux raisons :

- L'éclairage de l'objet est incohérent puisqu'il ne tient pas compte de l'éclairage ambiant.
- Le pillow shading place les ombres en tenant compte des bords, alors que ces mêmes bords n'ont pas d'existence physique (de la même façon qu'une frontière sur une carte).

Comme toujours, c'est bien plus simple à comprendre avec une image.



N'essayez pas non plus de placer des dégradés partout : une zone plane est souvent uniformément éclairée. C'est justement souvent ce qui permet à l'œil et au cerveau de comprendre que c'est une surface plane.

Le blending

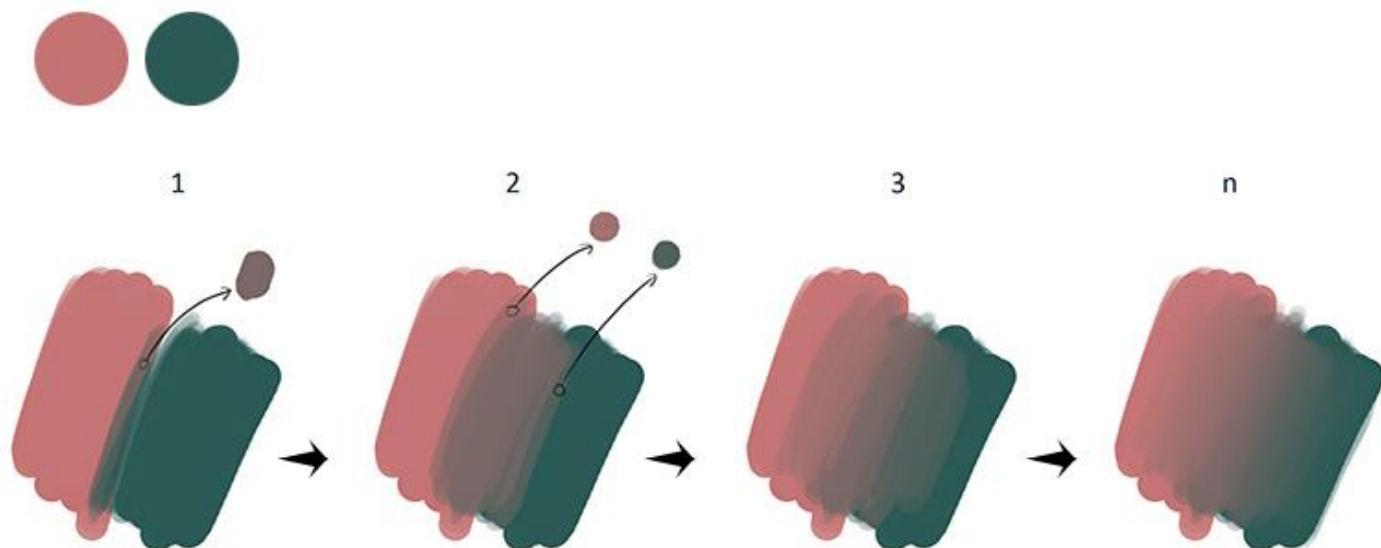
Maintenant que nous nous sommes intéressés à quelques éléments importants pour le shading, intéressons nous au blending. C'est en fait un processus de « mise au propre » de l'image. Il y a plein de façon de procéder, mais j'en présenterai une ici qui a l'avantage d'être simple, efficace et très modulable (je remercie au passage Madocks, bien qu'il semble avoir disparu des forums, de m'avoir montré cette technique). Je présenterai ici ce qu'on peut faire avec Photoshop, mais on peut faire la même chose avec n'importe quel autre logiciel de dessin digne de ce nom, seules les commandes changent, pas le principe.

Si vous avez besoin de quelques rappels sur le fonctionnement de Photoshop, je vous invite à aller jeter un œil [ici](#).

L'idée est la suivante : vous voulez réaliser une transition entre deux zones de couleurs (par exemple, une zone d'ombre et de lumière). Pour avoir un rendu intéressant et naturel, il faut absolument éviter les outils automatisés, comme les dégradés automatiques, le doigt ou le flou. Les brosses floues sont à mon avis à éviter si vous êtes débutants.

Sélectionnez le pinceau et avec le clic droit, choisissez une brosse dure (dureté 100 % ou un peu moins, mais pas au dessous de 70 % pour commencer). Réglez le pinceau pour que l'opacité (et si possible la taille) varie avec la pression du stylet de votre tablette graphique. Si vous n'en avez pas, il faudra régler l'opacité manuellement. En maintenant Alt, il est possible de passer à la pipette. **Ce raccourci est extrêmement utile pour passer de l'un à l'autre, souvenez vous en.** Peignez vos deux grandes zones, sans vous soucier de la transition, mais faites en sorte qu'elles se recouvrent. A l'intersection, grâce à la transparence du pinceau gérée par Photoshop, des couleurs de transition sont apparus (ça ne fonctionne pas en tout ou rien comme avec Paint). Récupérez la couleur de transition avec la pipette (pensez à utiliser Alt !) et peignez sur la zone de transition au milieu. Vous avez maintenant trois zones de couleurs et deux transitions. Récupérez les couleurs créées au niveau des transitions avec la pipette et repeignez dessus. Il suffit d'itérer le procédé jusqu'à le rendu vous semble assez lisse.

Je sais que dit comme ça, ça peut sembler compliqué, mais c'est en fait très simple et je pense qu'il est plus facile de voir comment ça marche sur un dessin :



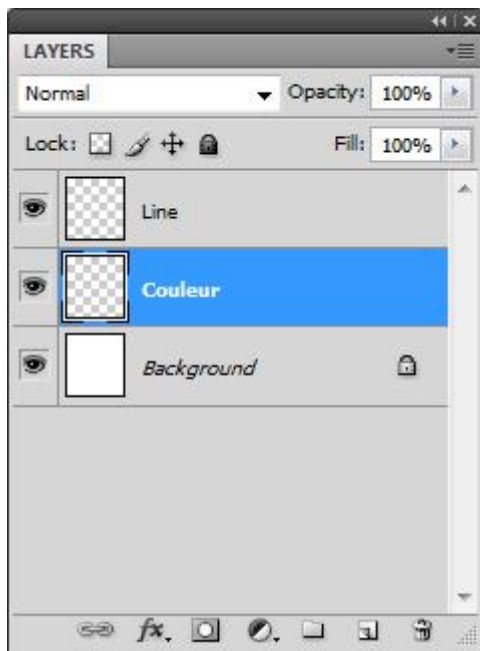
A chaque étape, je récupère les couleurs formées au niveau des nouvelles transitions et je m'en sers pour adoucir ces transitions. L'avantage de cette méthode est qu'elle vous permet de bien contrôler ce que vous faites. Si l'objet est très lisse, vous pouvez réaliser des transitions très lisses, s'il est plus anguleux, vous pouvez réaliser des transitions plus brutales. Cette méthode marche aussi avec d'autres brosses que celles de base et peut donner des textures intéressantes. On peut s'en servir bien sûr pour mélanger plus de deux couleurs mais ceci vient avec la pratique. Vous vous rendrez vite compte qu'il s'agit d'une façon de procéder naturelle et efficace.

Pour pousser les choses un petit peu plus loin, je vous recommanderai, si vous cherchez à obtenir un rendu plus « peinture » que « dessin », d'utiliser des brosses assez grandes (plus grandes que le niveau de détail auquel vous voulez travailler) et de travailler avec une opacité relativement forte avec des brosses dures (plus de 70 % d'opacité et de dureté). Mais c'est là une recommandation plus personnelle, à chacun de voir ce qui lui convient :]

Mettre un lineart en couleurs

Je vais ici présenter une façon assez simple pour commencer à mettre un lineart en couleurs. Les méthodes précédentes ne nécessitaient pas de lineart mais si vous en avez un, il peut être utile de connaître quelques astuces.

L'idée principale est de décomposer notre image en calques. Créez un calque au-dessus de l'arrière-plan, que vous pouvez nommer Couleur et un autre au-dessus, que vous pouvez nommer Line (ça semble assez explicite, non ?). Vous devriez avoir quelque chose de ce genre-ci :



Supposons que votre lineart est prêt à l'emploi. Vous l'avez scanné, nettoyé, vous l'avez peut-être redessiné sous Photoshop ou passé à l'encre et vous avez réglé les niveaux et le contraste de façon à ce qu'il soit bien propre. Vous devez avoir quelque chose qui ressemble à la main de gauche dans la première image de cet article. Maintenant, allez le chercher et copiez le dans le calque Line. S'il se trouve sur un fond blanc, ce n'est pas grave. Maintenant, passez le en mode **Produit** dans la fenêtre des calques. A quoi cela sert-il ? Cela va faire en sorte que lorsque vous allez peindre sur le calque Couleur, en-dessous du calque Line, la couleur va apparaître sous le line, comme si le blanc était transparent. C'est très pratique et cela évite d'avoir à enlever tout le blanc du line, ce qui laisse souvent des pixels extrêmement disgracieux. Maintenant, vous allez pouvoir passer à la couleur.

Comme vous travaillez avec un lineart, vous pouvez remplir le line avec une couleur neutre (si vous voulez celle qu'on retrouvera le plus souvent sur le dessin). Vous devriez avoir une silhouette. Sélectionnez le calque Couleur et cochez la petite icône en forme de damier en haut à gauche : votre calque est maintenant verrouillé. Cela veut dire que si vous peignez, Photoshop fera en sorte que vous ne peignez pas à l'extérieur de la silhouette, même si vous ne faites pas attention à ne pas dépasser. Une image, encore une fois, est plus parlante qu'un gros pavé de texte :



Ceci vous permettra de séparer différentes zones de l'image si vous en avez besoin. Par exemple, vous pouvez créer un calque pour chaque personnage de la scène, peindre la silhouette de chaque personnage et la verrouiller de façon à travailler aussi proprement que possible.

Pour un style lineart, vous pouvez vous contenter de cette approche, mais vous pouvez aussi vous en servir avec des dessins moins propre pour des effets de texture, de grain ou tout simplement si vous voulez juste utiliser le line comme brouillon (pour les traits de constructions, les lignes de fuite, etc.). Vous voudrez alors peut-être créer un calque au-dessus du calque Line, pour pouvoir peindre et recouvrir le tracé. Cela permet de donner un rendu plus proche de la peinture que du dessin.

Pour compléter ces quelques remarques sur la mise en couleur, je vous invite à aller voir les différents cours de pixel art de Fil sur ce site, ainsi que la [leçon de dessin d'Arne](#) qui en dit beaucoup plus long à ce sujet. J'espère que cet article vous sera utile pour vos prochaines illustrations et si vous avez des questions ou des remarques, n'hésitez surtout pas. Normalement, des making of arriveront bientôt pour essayer d'illustrer ceci de façon plus concrète :]

La narration libre

Par [Pierre Guyot](#) | le 8 août 2011 20:00:00 | Catégories : [Ernest Adams](#), [Game Design](#), [Scénario](#), [Univers](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (août 2010).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



En 1995, j'ai donné une conférence à la *Computer Game Developer's Conference* au cours de laquelle j'ai identifié plusieurs des problèmes de la narration interactive. J'ai repris ces idées quelques années plus tard dans l'un des articles du *Designer's Notebook* intitulé « [Trois problèmes de la narration interactive](#) » et j'y ai suggéré qu'au lieu d'essayer de raconter des histoires, nous devrions essayer de construire des univers dans lesquels des histoires peuvent se produire — des univers où les joueurs puissent vivre des histoires de leur propre création. L'industrie ne disposait d'aucun terme spécifique pour ceci à l'époque, et j'ai proposé le nom de *sandbox storytelling*.

Dans le cadre d'un jeu bac à sable, l'idée est de présenter au joueur un grand monde ouvert truffé d'opportunités pour réaliser des interactions. Le joueur n'est pas obligé de suivre une intrigue linéaire comme un rail, mais peut interagir avec le monde dans l'ordre dans lequel il souhaite. Si le monde est correctement construit, une expérience narrative devrait naître spontanément.

Tout le monde ne pense pas que la narration libre est une bonne idée. L'année qui a suivi cette conférence, Bob Bates en a donné une au *CGDC 1996* intitulée « La responsabilité de l'auteur ». L'un des choses qui y a été dite était que « [Les environnements ouverts] sont peut-être amusants à explorer, mais ils ne remplissent les fonctions d'une histoire. Il n'y a ni début, ni milieu, ni fin. Il n'y a pas de *pathos*, ni de drames humains, ni de vérité supérieure qui se révélerait derrière les combats menés par le joueur ».

Bob proposait d'utiliser à la place une série linéaire d'environnements ouverts — ce que nous appelons maintenant une intrigue multilinéaire ou *foldback story*, dans laquelle le joueur est contraint de passer par certains points obligatoires de l'intrigue.

Néanmoins, Bob suppose ici que dans un environnement ouvert, le joueur aurait à aller chercher l'intrigue et que tout ce qu'il obtiendrait, c'est une série d'événements déconnectés les uns des autres. Je pense que Bob imaginait que les différents événements étaient liés à des lieux spécifiques et que si le joueur pouvait les déclencher dans n'importe quel ordre, ils ne pourraient pas être reliés entre eux. Je ne suis pas surpris qu'il ait fait cette hypothèse, surtout à l'époque. Nous sommes totalement habitués à définir les intrigues en fonction des lieux physiques où elles se déroulent — à tel point que c'est notre approche par défaut et que tout autre système nous semble étrange. Depuis *Zork* à *Fallout 3* en passant par *Half-Life*, le mouvement à travers l'espace est synonyme de mouvement à travers l'histoire. Mais pour réaliser une narration dans un jeu bac à sable, nous devons nous débarrasser de cette notion et plutôt réfléchir à comment créer une intrigue qui se déploie — et maintient sa continuité — par d'autres moyens.

Les *Grand Theft Auto* sont célèbres pour leur environnement bac à sable, mais ils ne mettent pas en place une véritable narration libre. À la place, vous avez droit à l'habituelle chaîne linéaire de mission : vous en complétez une, puis une autre, et ainsi de suite. Il se trouve par ailleurs que les missions se déroulent dans un monde ouvert très vaste et que vous pouvez abandonner la mission et vous mettre à vous balader en semant le chaos autour de vous (ou en conduisant un taxi) si vous en avez envie. D'une certaine façon, c'est ce que Bob désignait comme une série linéaire d'environnement ouvert, sauf qu'au lieu d'une série d'environnements différents, les *Grand Theft Auto* vous donnent juste de nouvelles missions dans un nouvel environnement — bien que vous débloquent de nouvelles zones de temps à autres.



Les Sims offrent d'une certaine façon une narration libre. Le jeu vous propose un monde avec beaucoup d'éléments et de personnages avec des personnalités variées. En tant que joueur, vous pouvez les faire interagir et créer une (sorte de) fil narratif. Dans la mesure où les Sims ne parlent pas anglais, la majeure partie de la narration se fait dans votre tête, mais ce n'est pas grave. Vous pouvez vous faire votre propre film, le doubler et l'uploader sur Youtube.

Mais *Les Sims* utilisent une modèle d'interaction dans lequel vous n'avez pas d'avatar particulier au sein de l'univers du jeu. Pour élaborer une histoire à partir des Sims, vous devez manipuler plus d'un personnage, plutôt que d'incarner un unique personnage. Ceci fait de vous un créateur plus qu'un participant. Ce n'est pas la façon dont la plupart des jeux narratifs fonctionnent, et je pense que ce n'est pas ce que la plupart des gens attendent d'un jeu narratif.

Les jeux de rôle PC offrent au joueur un vaste univers ouvert, mais plutôt que fournir une unique intrigue, cet univers est plein de quêtes — fondamentalement, des sous-intrigues déconnectées les unes des autres. J'aime les RPG occidentaux, mais il ne m'apportent pas le même ressenti qu'un jeu avec une histoire à intrigue unique. Ils fonctionnent plus comme les douze travaux d'Hercule ou comme n'importe quel ancien héros qui apparaît dans plusieurs histoires déconnectées les unes des autres.

Alors, comment faire pour réaliser un jeu bac à sable où le joueur puisse à la fois se balader comme bon lui semble et faire partie intégrante de l'histoire ? Tout d'abord, comme je l'ai dit, nous devons abandonner l'idée que le joueur ne pourra faire l'expérience de la narration qu'à travers l'exploration. Dans le même temps, le voyage doit aussi être une composante à part entière de l'histoire, car sinon voyager serait juste pénible. Habituellement, les films se débarrassent de ce genre de passages — quelqu'un sort de sa maison le matin, monte dans sa voiture et au plan suivant, il est train de marcher jusqu'à son lieu de travail — à moins que le film ne se focalise spécifiquement sur ce voyage, comme un film de course-poursuite.

Dans les jeux d'aventure ou les RPG classiques, tous les événements de l'intrigue dépendent du joueur : ils ne se produisent pas tant que le joueur ne vient pas les trouver et les déclenche. En utilisant des environnements confinés, on peut s'assurer que le joueur les trouve dans le bon ordre. Le problème d'une intrigue composée uniquement d'événements dépendants du joueur, comme je l'expliquai lors de la conférence, est qu'elle donne l'impression d'être mécanique : le monde entier se contente d'attendre que le joueur fasse quelque chose.

Si vous rendez l'intrigue complètement indépendante du joueur — c'est-à-dire qu'elle continue d'avancer quoi que le joueur fasse, même s'il ne fait rien du tout — alors le joueur a tendance à perdre beaucoup. Il n'est pas là où il le devait, ou bien il n'a pas fait ce qu'il fallait au moment où le climax se produit.

Le problème dans la narration d'un jeu bac à sable, c'est de construire l'intrigue avec une combinaison d'événements dépendants du joueur et indépendants du joueur. Faîtes avancer l'intrigue en permanence quoi que fasse le joueur pour que l'univers ne semble pas statique, mais pas trop vite pour que le joueur ne se retrouve pas trop en retard et perde (à moins que l'intrigue ne se focalise sur une bombe à retardement). Mettez la pression de façon raisonnable sur le joueur pour qu'il agisse, mais réduisez la pression si le joueur est sur la bonne voie. Dans un jeu bac à sable, ce n'est pas l'exploration qui fait avancer l'intrigue — alors à la place, utilisez une combinaison de l'écoulement du temps (c'est la pression sur le joueur) et d'activités du joueur : rencontrer des gens, résoudre des énigmes, prendre des décisions, surmonter des obstacles. Changer le rythme de temps à autre. Parfois, James Bond explore à son rythme (il est maître de la situation) et parfois, il tente désespérément de fuir les méchants (ce sont eux les maîtres de la situation). Puis il réussit à s'enfuir ou alors il les abat et redevient maître de la situation. Bien sûr, tous les jeux vidéo n'ont pas à mettre une forte pression sur le joueur. Vous pouvez laisser le joueur avoir une expérience très calme si vous voulez.

L'une des question que posent certains designers, c'est : « Et si le joueur se contente de se balader et ne rentre jamais dans l'histoire ? ». La réponse est : cela dépend du type d'expérience que vous voulez qu'il ait. Cela peut être très bien de le laisser se balader tranquillement. J'aimerais beaucoup explorer les paysages des différents *Far Cry* sans me faire tirer dessus à tout bout de champ. D'un autre côté, si vous voulez que le joueur rentre dans l'histoire, alors vous devez vous demander pourquoi il se contente de se balader. S'il se ballade parce qu'il est perdu ou parce qu'il ne sait pas où aller, c'est de votre faute. Le designer Chris Bateman a écrit un chapitre intitulé « Maintenir le joueur sur la bonne route » dans le livre *Game Writing: Narrative Skills for Videogames* qu'il a publié. Dans ce chapitre, il parle des techniques « d'entonnoir » : il s'agit d'outils variés pour aider le joueur à trouver « l'épine dorsale » du jeu. Dans un monde ouvert, vous ne pouvez pas utiliser l'environnement spatial pour forcer le joueur à retourner vers l'histoire, mais vous pouvez laisser de nombreux signes et indices ça et là. Voyez le livre en question pour plus d'information.



D'un autre côté, si le joueur se contente de batifoler et que vous voulez qu'il retourne à l'histoire, vous devez faire lui mettre la pression en utilisant des événements indépendants du joueur et de sa localisation dans le jeu. Les maîtres du jeu des jeux de rôle papier connaissent bien ce problème. Si le joueur ne vient pas à l'intrigue, c'est l'intrigue qui ira à eux (NDT : on appelle ça le *railroading*). S'ils ont été engagés pour se débarrasser d'un parrain local et qu'au lieu de ça, ils se contentent de rester ici et de boire des bières, le dit parrain peut très bien avoir vent de leurs projets et envoyer une bande d'assassins à la taverne pour se débarrasser de l'expédition avant même qu'elle ait commencé. Lors du combat qui s'ensuit, la taverne prend feu. Même si l'équipe survit, elle ne pourra plus continuer à boire ici.

Une autre question que les gens posent souvent : « Dans un monde ouvert, comment empêcher que le joueur voit trop dès le début du jeu des éléments qu'il n'est supposé voir que plus tard ? » Cette question est enracinée dans l'hypothèse qu'une histoire a besoin d'être physiquement présente dès le début au sein d'un monde statique — comme il est en généralement dans les jeux d'aventure et les RPG occidentaux, où l'histoire correspond aux lieux. Mais nous ne faisons pas correspondre l'histoire aux lieux, mais au rythme et à l'activité du joueur. La réponse est simple : ne faites pas apparaître un objet dans l'univers du jeu tant qu'il n'y en a pas besoin. Dans les *Grand Theft Auto*, on ne peut pas détruire une certaine voiture à la mission 1 parce qu'elle sera nécessaire à la mission 3, tout simplement parce qu'elle n'est pas dans l'univers du jeu à la mission 1. Vous ne voulez évidemment pas faire surgir des voitures de nulle part juste devant les yeux du joueur, mais vous pouvez faire une sortir une voiture d'un garage fermé auparavant. Le joueur ne peut pas être à plus de deux endroits à la fois, vous pouvez donc faire plein de choses dans son dos.

Alors, quel genre d'histoire pouvons nous raconter dans des mondes ouverts que le joueur peut explorer dans n'importe quel ordre et dans lesquels le voyage est une facette à part entière de l'expérience ? Et bien, en voici une série :

- **Trouver le trésor enfoui.** C'est un type de jeu où l'urgence est faible, puisque le trésor n'ira nulle part. Le joueur se ballade dans la ville pour récupérer et assembler des indices. Il n'est pas nécessaire de trouver les indices dans le bon ordre : il peut s'agir des parties disséminées d'une carte au trésor.
- **Trouver le trésor enfoui avant que quelqu'un d'autre ne le fasse.** Même histoire, mais il y a déjà plus de pression sur le joueur. Vous pouvez donner à cette histoire la forme d'une simple course ou vous pouvez faire monter et retomber la pression en faisant en sorte que l'équipe ennemie essaye de saboter les tentatives du joueur. S'ils y arrivent, le joueur subit encore plus de pression ; si c'est le joueur qui joue les saboteurs, la pression redescend (faites néanmoins attention à ne pas créer une trop forte rétroaction positive).

- **Trouver une bombe à retardement.** Évidemment, c'est là qu'il y a le plus de pression et c'est assez difficile à développer dans un univers ouvert, mais pas impossible. Une façon de donner un peu plus de contrôle au joueur sur le rythme de l'aventure est d'accorder à son avatar une super-vitesse en quantité limitée, comme Superman, Flash ou Neo dans *Matrix*. Le joueur peut utiliser ce pouvoir pour essayer de gagner un peu temps.
- **Trouver des gens plutôt que des objets.** Bien sûr, c'est tout de suite beaucoup plus compliqué vu que les gens se déplacent. Le joueur peut incarner un chasseur de primes à la poursuite de fugitifs, un détective à la recherche d'une personne disparue, un membre des services de contre-espionnage qui traque un agent double, etc.
- **Une enquête de police.** On en revient à la vieille série des *Police Quest* de Sierra On-Line. Afin de monter un dossier en béton, la police passe beaucoup de temps à se déplacer pour rendre visite aux témoins, questionner les proches des suspects et rechercher des preuves matérielles. Certaines preuves peuvent être trouvées dans n'importe quel ordre, alors que d'autres n'apparaissent qu'en suivant une certaine succession d'indices. Vous aurez peut-être besoin de restreindre la taille de votre ville, car bien que le voyage fasse partie intégrante du métier, ce n'est pas follement palpitant. Vous pourriez épicer ça en laissant les policiers gérer la délinquance et repérer les témoins ou les suspects sur leur lieu de travail. Afin de baisser la pression, laissez le joueur mener son enquête alors qu'un suspect est déjà arrêté. Pour faire remonter la pression, laissez le s'échapper.
- **Infiltrer une large zone depuis n'importe quel endroit.** Trop de shooters se contentent de placer le joueur sur des rails. C'est moins coûteux à développer, mais c'est aussi moins intéressant pour le joueur. Dans l'actuelle guerre en Afghanistan, les alliés de l'OTAN contrôlent l'espace aérien et disposent d'hélicoptères, de façon à pouvoir déposer les troupes n'importe où hors des zones de combat et à les laisser se déplacer comme ils l'entendent. La planification des missions implique un examen minutieux de photographies aériennes afin de choisir une tactique d'approche.
- **S'échapper d'un territoire hostile.** Lors de mes ateliers de game design, je donne souvent comme défi à l'une des équipes de faire un jeu sur l'*Underground Railroad*, le réseau souterrain utilisé par les abolitionnistes avant la guerre de Sécession pour aider les esclaves à s'échapper. Certaines de ces routes sont en réalité longues de plusieurs centaines de kilomètres et donc inadaptées pour un jeu, mais vous avez saisi l'idée.
- **La contrebande,** ce n'est pas juste s'infiltrer ou s'échapper, mais faire les deux — et le plus souvent avec un chargement peu orthodoxe. Durant les longues, longues guerres entre l'Angleterre et la France aux dix-huitième et dix-neuvième siècles, le commerce entre les deux nations était techniquement interrompu. Néanmoins, on trouvait encore une grande quantité d'eau-de-vie en Grande-Bretagne. Les littoraux des deux nations offraient de nombreuses entrées où un bateau de guerre ne pouvait pas rentrer, et les contrebandiers les connaissaient très bien.
- Comme la police, les **reporters** voyagent beaucoup, souvent sous la pression d'une échéance, afin de récolter des informations. Sans oublier les **espions**.
- Les **chasseurs** ont une tendance naturelle à explorer l'univers. La chasse ne permet pas d'ordinaire de construire une véritable histoire à moins que quelque chose d'inhabituel ne se produise, mais parfois cela arrive. Je me rappelle de deux histoires que j'avais lues quand j'étais petit où de riches (et méchants) chasseurs nourrissaient une passion secrète pour la chasse à l'homme — plus précisément, les guides qu'ils engageaient.
- **Démanteler un réseau criminel.** Les tentacules du professeur Moriarty sont partout. Quelque soit l'endroit de Londres où se rend Sherlock Holmes, il y découvre systématiquement des traces des agissements sinistres de Moriarty. Mais où se trouvent les véritables bras droits de Moriarty, où se trouve son quartier général et où se trouve-t-il lui-même ?
- **Résistance.** Une autre idée de jeu que je propose dans mes ateliers consiste à concevoir un jeu sur un mouvement de résistance dans un pays occupé. Ils ne sont pas autorisés à faire de shooter : le jeu doit consister à saboter et à pourrir la vie de l'occupant sans se faire prendre. Ceci aussi peut être développé sur un grand espace de jeu, où les soldats de l'armée d'occupation sont constamment à la recherche des combattants de la résistance et maintiennent le joueur sous pression.

Il y a plein d'autres sortes de métiers ou de loisirs qui nécessitent de voyager : pompiers, électriciens, chasseurs de tornades, critiques de restaurants... Tous ne conviennent pas nécessairement à un jeu vidéo, mais il suffit juste d'un peu d'imagination. Je suis sûr que vous pouvez en trouver d'autres.

Pour donner une dimension narrative à une expérience ludique, vous devez éviter de faire intervenir des événements sans lien entre eux de façon trop aléatoire ou trop répétitive (à ce sujet, voir mon article « Innovations dans la dramaturgie des jeux et des histoires » pour plus d'information). Si vous lisez un thriller qui se passe pendant la Deuxième Guerre mondiale, il ne s'agit pas juste de voir des légions de Nazis tous identiques se faire massacrer : chaque situation doit être unique. Ceci signifie que votre bac à sable doit contenir des éléments très variés, et non pas la simple répétition d'un même événement. Il s'agit probablement du plus fort argument en défaveur d'une narration libre : c'est extrêmement coûteux à développer et nécessite beaucoup de travail. Mais à la différence des jeux linéaires, si vous concevez votre univers avec suffisamment de soin, le jeu pourra être refait un grand nombre de fois. Différents parcours dans l'univers du jeu offriront des expériences de jeu différentes. Il n'est pas nécessaire qu'ils proposent les mêmes objectifs ou la même fin.

Pendant des centaines d'années, la plupart des habitants de Rome ont soutenu l'une des quatre équipes qui participaient aux courses de char. Les conducteurs portaient des vêtements de couleur de façon à ce que les gens puisse les distinguer et les différentes équipes étaient appelées les Rouges, les Blancs, les Bleus et les Verts. Supposons que notre joueur se retrouve tout à coup dans Rome. Il peut rejoindre n'importe lequel des groupes de supporters, de la même façon que dans n'importe quel sport d'équipe aujourd'hui, mais à une différence près : les différents groupes provoquaient souvent des émeutes et se livraient à des combats sanglants dans les rues. Ce qui signifie en pratique qu'un PNJ qui appartient au groupe de supporters que le joueur choisit devient un allié, mais que si le joueur recommence une partie et choisit un autre groupe, ce même PNJ devient alors un ennemi. Pas besoin d'écrire deux histoires ou de concevoir deux personnages : le drame, le conflit émergent naturellement de la situation.

Pour faire court, je pense qu'une narration libre est à la fois possible et amusante. Vous aurez besoin de remplir autant que possible votre bac à sable de façon à proposer autant d'opportunités dramatiques que possible (de nombreux jeux de type bac à sable semblent vides et stériles) et vous devrez décider quelle pression faire peser sur le joueur et comment. Ceci peut faire intervenir des éléments de l'intrigue qui dépendent du temps mais pas du joueur, afin d'obliger celui-ci à aller de l'avant. L'environnement en lui-même est aussi crucial : ce doit être un lieu où le joueur apprécie d'être, parce qu'il va y passer beaucoup de temps.

Il y a un film noir célèbre intitulé *La Cité sans voiles* qui a été par la suite adapté en série télévisée. A la fin du film et de chaque épisode de la série, on entendait le narrateur dire : « Il y a huit millions d'histoires dans la Cité sans voiles. Ceci en était une. ». Essayez de concevoir votre propre Cité sans voiles et de voir combien d'histoires vous pouvez y créer.

Comment se lancer dans l'industrie du jeu

Par [Thomas Chassin](#) | le 14 août 2011 22:33:47 | Catégories : [Business](#), [Ernest Adams](#)

Cet article est une traduction de [Thomas Chassin](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (décembre 1998).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



Il y a quelque chose d'amusant avec les jeux vidéos : y jouer provoque couramment l'envie d'en faire un soi-même. C'est exactement ce que j'ai ressenti en jouant à mon premier jeu sur ordinateur. Après cinq minutes passées dessus, je me suis dit : « Il FAUT que je sache comment on fait ça ! ». Je ne connais aucun média qui provoque cette sensation aussi fortement. Enfant, je regardais la télévision sans vouloir faire des émissions télévisées. J'ai bien eu envie d'écrire après avoir lu des livres, mais les enfants sont toujours encouragés à écrire. Personne ne m'a encouragé à faire des jeux vidéos. C'était simplement quelque chose que je savais devoir faire.

Je reçois beaucoup de lettres de gens qui veulent se lancer dans l'industrie. Beaucoup de ces gens sont des lycéens ou des étudiants qui sont fascinés par les jeux vidéos exactement comme je l'étais. Je leur écris toujours des conseils et des encouragements, mais j'ai écrit la même lettre tant de fois que j'ai réalisé que la meilleure chose à faire était d'écrire un article qui vivra sur le web (NDT : et pour preuve, le voilà aujourd'hui sur les Forges !). De cette façon, je pourrai juste diriger mes futurs correspondants vers Gamasutra (NDT : ou les Forges pour les anglophobes). En revanche, je ne vous dirai pas comment commencer à développer vous-même, ou comment vous faire connaître. C'est là un sujet pour un livre, pas pour un article. Nous parlerons ici d'obtenir un poste dans l'industrie du jeu vidéo.

Si vous êtes au primaire ou au collège

S'il vous plaît, ne vous inquiétez pas pour ça pour le moment. La meilleure chose à faire est d'avoir une éducation solide. Si vous consacrez votre vie entière à apprendre à faire des jeux vidéos dès votre âge, vous finirez par n'être bon qu'à une seule chose, et à ce moment-là vous n'en aurez peut-être plus envie. Vous êtes à l'heure de la généralisation, pas de la spécialisation.

Cependant, il y a certaines choses que vous pouvez faire pour vous mettre sur la bonne voie. Travaillez l'algèbre et la géométrie, et si votre école offre des cours d'informatique ou de programmation, inscrivez-vous y. Même si vous ne souhaitez pas devenir programmeur, il est très utile d'avoir une compréhension générale de ce que font les programmeurs. Si vous avez la chance de pouvoir travailler sur des logiciels

d'infographie, ou sur des des outils musicaux électroniques, essayez les également. Familiarisez-vous avec les ordinateurs. Apprenez à bien taper – si vous pouvez taper vite et avec précision, vous gagnerez beaucoup de temps plus tard.

Si vous êtes au lycée

L'époque où il n'y avait pas besoin de diplômes pour décrocher un poste dans l'industrie du jeu vidéo est bientôt révolue, j'en ai bien peur. Il y a quelques années, il était possible d'obtenir un poste simplement par la force de la programmation que vous aviez fait au lycée, mais maintenant, les entreprises sont suffisamment importantes pour avoir un département des ressources humaines, et les CV sur lesquels ne figurent pas de diplômes post-bac finiront le plus probablement à la poubelle. Suivez les cours nécessaires pour pouvoir pousser vos études jusqu'à une licence, un master ou un diplôme équivalent.

Si vous êtes étudiant

Arrivé à ce point, il est temps de réfléchir aux talents et aux compétences que vous pouvez amener à l'industrie du jeu vidéo. Qu'aimez-vous ? En quoi êtes vous bon ? Pouvez-vous dessiner, peindre, programmer, écrire des histoires ou composer de la musique ? Il est maintenant l'heure de développer ces talents, en étudiant les maîtres et en vous entraînant.

Bien que cela semble niais, je suggère fortement que vous suiviez un enseignement varié. Bruce Sterling, l'auteur de science-fiction, a une fois dit que les gens formés « un peu à tout » étaient lisses et mornes, et qu'il valait mieux être un individu atypique aux connaissances pointues dans un domaine précis. C'est un bon conseil pour un artiste indépendant, mais pas pour quelqu'un qui cherche un travail. La plupart des personnes ne veulent pas être des programmeurs ou des artistes toute leur vie — le stress finit par les détruire (comme il m'a détruit). Pour passer à la production et la conception de jeu, vous devez être un individu instruit qui a quelque chose à offrir à sa société en plus de ses compétences artistiques et informatiques. Assurez-vous de suivre des cours d'anthropologie, d'histoire et de littérature. La saga *Star Wars* n'est pas subitement apparue dans l'esprit de George Lucas ; elle est apparue – et a rencontré un tel succès — parce qu'il a compris le pouvoir de thèmes mythiques comme celui du jeune homme héroïque voyageant pour affronter un passé caché. Lucas avait cette compréhension parce que son enseignement allait au delà de « tenir une caméra et éditer des films ».

Étudiez les jeux !

Quel que soit votre âge, vous devriez jouer à autant de jeux que possible. Formez un groupe avec vos amis pour les acheter et les échanger. Jouez à différents genres : jeux d'aventure, jeux de tir, simulations militaires, simulations sportives, jeux de réflexion, jeux pour enfants, jeux « pour les filles », etc. N'y jouez pas seulement pour le plaisir, pensez-y sérieusement et observez la façon dont ils fonctionnent. La plupart des jeux ont une économie interne – certaines valeurs varient au cours de la partie, et sans elles, vous perdez (ou mourrez). Au Monopoly par exemple, c'est l'argent. Dans *Doom*, ce sont les munitions, l'armure et la santé. Quels sont les flux d'entrée ? De sortie ? Quels sont les rôles respectifs de la chance et de l'habileté du joueur ? Comment le jeu est-il équilibré ? Si les jeux proposent différents niveaux de difficulté, jouez à ces modes et prenez note de ce qui a changé.

Regardez aussi l'interface utilisateur : la façon dont le clavier, la souris ou le joystick sont utilisés, la façon dont l'écran est organisé, la navigation dans les menus. Sont-ils logiques et pratiques ? Vous retrouvez-vous en train de rêver d'une touche ou d'un bouton spécial que le jeu n'a pas ? Quel type de « caméra » est employé : à la première personne, comme *Doom* ou *Quake* ? Troisième personne, comme dans *Tomb Raider* ? Une vue aérienne en déplacement libre, comme dans *Dungeon Keeper* ? Une perspective isométrique comme dans *Starcraft* ? Pouvez-vous changer l'angle de la caméra (comme dans *NFL Football*)

et si oui, comment la jouabilité du jeu change-t-elle quand vous le faites ? Toutes ces choses rentrent dans l'analyse de la conception d'un jeu.

Un game designer, ça n'existe pas

Ok, c'est une redite. Mais à part une ou deux très rares exceptions, il n'existe rien de tel qu'un game designer. [Note de l'auteur : ceci était vrai en 1998, quand cet article a été écrit. Les jeux étant devenus plus importants et plus complexes, il commence à y avoir des game designers à temps plein]. Trip Hawkins, le fondateur d'Electronic Arts et 3DO, avait l'habitude de dire qu'il n'y aurait jamais de poste intitulé « game designer » dans son entreprise car chacun dans le projet participait au design du jeu, et qu'il serait injuste de donner à une personne tout le mérite (et le travail le plus agréable).

Il y a une autre raison, plus pratique à cela. Le design initial d'un jeu prend de un à trois mois, selon son envergure et sa complexité. Durant le reste de la production, il reste toujours des choix de design à faire, mais pas suffisamment pour consacrer une personne à temps complet. En général, le designer du jeu commence à remplir des tâches administratives, ou participe à la production une fois le développement lancé. Une entreprise ne peut pas se permettre d'avoir à payer quelqu'un ne faisant rien d'autre que du game design toute l'année. [Note de l'auteur : Depuis que j'ai écrit cet article en 1998, l'industrie a changé sa façon de concevoir les jeux. On ne procède plus séquentiellement, en faisant tout le travail de design au début et le développement ensuite. De nos jours on utilise des méthodes de développement agiles (comme [Scrum](#)) qui permettent de prendre les décisions de design au long du développement.]

Un cas exceptionnel est celui Sid Meier, qui est si célèbre en tant que designer qu'il a son nom sur la jaquette du jeu. Mais Sid a traîné sa carcasse depuis un bout de temps, ses jeux rencontrent un succès phénoménal, et il a plus que payé son dû. Ne pensez pas que vous allez décrocher d'entrée un poste de game designer. A moins de lancer votre propre entreprise – et de pouvoir vous payer à ne faire que du game design – cela ne risque pas d'arriver.

Tester des jeux n'est pas aussi fun que vous l'imaginez

Tester des jeux, ça doit être un job de rêve, un peu comme être chauffeur d'un type qui roule en Porsche (et qui, inexplicablement, ne veut pas la conduire lui-même). Être payé à jouer ! Que demander de plus ?

Malheureusement, tester des jeux à la recherche de bugs, ce n'est pas comme jouer pour le plaisir. Quand vous testez à la recherche d'erreurs, vous devez tester chaque possibilité et chaque combinaison (ou au moins, autant que vous le pouvez raisonnablement). Vous ne pouvez pas jouer de manière compétitive ; vous ne pouvez pas jouer pour gagner ; vous jouez en suivant un plan de test qui vous dicte quoi faire. Heure après heure, jour après jour, pour des semaines et des semaines. Tester un jeu, c'est un déjà plus comme manger des sundae au caramel toute la journée, tous les jours. Vous les aimerez peut-être au début, mais vous les détesterez certainement à la fin.

Si vous testez une simulation d'un jeu réel, vous devez aussi tester chaque condition imaginable pour vous assurer qu'elle corresponde à ce qui se passerait dans le vrai jeu. Dans un match de football, lors d'un engagement, que se passe-t-il si le ballon sort du terrain ? Qu'arrive-t-il s'il roule hors de la zone des buts ? Si la *kicking team* le touche d'abord ? Si la *receiving team* le touche ? S'ils demandent un *fair catch* ? S'ils demandent un *fair catch* et le rate ? (NDT : je renvoie les plus curieux d'entre vous sur la page [Wikipédia](#) consacrée au football américain))

Ce travail demande beaucoup de patience et d'autodiscipline. Il faut aussi avoir le sens de l'observation. Vous devez savoir exactement ce que vous étiez en train de faire au moment où un bug est survenu, et être capable de l'expliquer clairement et précisément. Ce n'est pas à la portée de tout le monde. Il faut être plus

qu'un joueur : il faut être un joueur capable de comprendre la pensée d'un programmeur, programmeur que vous ne rencontrerez probablement jamais.

Beaucoup de lycéens se demandent s'il est possible d'obtenir un poste de testeur pour l'été. D'après mon expérience, cela n'arrive pas souvent. La plupart des stagiaires que j'ai rencontrés étaient des étudiants. Je ne sais pas si c'est une question de maturité, si c'est en rapport avec la législation concernant le travail des mineurs, ou autre chose, mais je pense que c'est improbable.

Si vous avez déjà un travail dans une autre industrie

La bonne nouvelle est que, oui, il est possible d'obtenir un poste dans le milieu du jeu après avoir travaillé dans une autre industrie et j'en suis la preuve vivante. Avant d'en venir au développement de jeux, j'ai travaillé pendant sept ans dans l'ingénierie assistée par ordinateur à écrire des programmes qui posent des puces sur du silicium. Je savais que je voulais travailler sur des jeux un jour, mais je ne voulais pas le faire sur les petites machines disponibles à l'époque. Quand le 80386 et la carte VGA sont apparus, j'ai supposé que le moment était venu et c'est alors que j'ai changé de secteur.

Voilà comment j'ai fait : j'ai écrit une brève lettre de motivation qui expliquait comment travailler dans l'industrie de l'ingénierie assistée par ordinateur m'avait appris la discipline du génie logiciel et des compétences de travail d'équipe qui avaient jusque là été rares dans l'industrie vidéo-ludique ; et que l'expérience acquise à programmer avec un délai, même dans le mauvais champs d'activité, était toujours précieuse. J'ai aussi introduit une petite démonstration d'un jeu que j'avais conçu pour Kaypro (!) et porté sur PC. À son éternel crédit, Rob Fulop, le fondateur de p.f Magic, m'a offert un poste de programmeur pour un jeu AOL (du temps où ils se faisaient encore appeler PC-Link), et je n'ai même pas subi de diminution de salaire dans l'affaire.

La mauvaise nouvelle est que c'est probablement plus difficile maintenant. À l'époque, l'industrie du jeu vidéo ne connaissait pratiquement rien à la discipline du génie logiciel ; même si c'était déjà un niveau au dessus du « programmeur de garage ». De nos jours, la programmation en équipe est la norme, et la gestion de versions, jadis moquée par ces codeurs de l'extrême, est à présent acceptée comme une pratique standard. Venant d'une communauté d'ingénierie établie, j'apportais quelque chose que l'industrie n'avait pas à l'époque — un avantage que vous n'aurez pas. Mais vous en avez peut être d'autres. Pensez attentivement à ce que vous avez fait et comment votre expérience pourrait s'appliquer au développement de jeu.

Heureusement, il n'est plus aussi vital de savoir programmer les machines directement. Quand je suis arrivé, les personnes capable de hacker les registres VGA et d'écrire des protocoles modem étaient très recherchées, mais maintenant de telles choses sont prises en charge par les librairies système. Du fait que vous n'avez plus besoin d'être si près de la machine, c'est plus facile que ça ne l'était. Je pense cependant que de l'expérience dans la programmation machine est extrêmement valorisante, en particulier si vous allez programmer sur des consoles de jeu.

Au sujet du salaire, les nouvelles sont moins bonnes. Le salaire des programmeurs, comme le reste, est régi par la loi de l'offre et de la demande. Il y a tant de gens désireux de rentrer dans l'industrie du jeu que les salaires sont plus bas en comparaison avec les autres industries. Un programmeur expérimenté peut s'attendre à une réduction de salaire de 10 à 25%. Vous pouvez avoir de la chance – j'en ai eu – mais préparez vous à un choc. Soit dit en passant, méfiez-vous des offres de travail ou vous travaillez gratuitement en échanges de royalties. Les start-ups sont en manque chronique d'argent, donc cette méthode s'impose à elles, mais je ne n'accepterais pas une telle offre à moins de connaître personnellement les gens et d'être un partenaire de l'entreprise. Je ne peux même pas commencer la liste de toutes les façons dont vous pouvez vous faire avoir ainsi – elles vont du jeu jamais lancé à la fraude pure et simple.

Une chose à propos de laquelle je ne peux pas donner beaucoup de conseils est de se lancer avec des compétences artistiques ou musicales venant d'autres industries. Je sais que c'est possible, parce que j'ai rencontré une animatrice expérimentée dans le milieu qui travaillait à Hollywood, mais je ne sais pas vraiment comment elle a fait la transition.

Si vous avez une idée géniale

Beaucoup de gens m'écrivent pour dire qu'ils ont une idée géniale pour un jeu, mais refusent de m'en parler de peur que je leur vole leur idée et gagne une fortune qui leur revient de droit. Ils veulent savoir comment se lancer sans vraiment révéler leur idée. Je suis désolé de le dire, mais les idées géniales il y en a dix à la douzaine. La chance que vous ayez une idée que personne n'ait eue est infinitésimale, surtout maintenant qu'il y a des dizaines de milliers de personnes qui travaillent dans l'industrie.

Un studio ne va pas passer un contrat de développement avec vous sur la base d'une idée géniale. Ce qu'une entreprise veut savoir c'est : pouvez-vous créer un jeu et le rendre génial, dans les temps et avec un budget fixé ? Ce qui compte vraiment n'est pas l'idée mais sa mise en pratique. Comme disait Thomas Edison : « le génie, c'est 1% d'inspiration et 99% de persévérance ». Il ne s'est pas juste assis et a soudainement inventé l'ampoule ; il a fait passer de l'électricité à travers tous les matériaux sur lesquels il pouvait mettre la main jusqu'à ce qu'il en trouve un qui marche – et encore, il n'a même pas touché au but étant donné qu'il s'est arrêté au fil de coton carbonisé. C'est un autre homme nommé Coolidge qui a inventé l'ampoule à filament en tungstène moderne.

Bien sûr, tout le monde connaît les lubies des consommateurs qui ont rendu leurs inventeurs millionnaires – le Rubik's cube, les pogs, et autre trucs du genre. Cependant ce genre de lubies n'arrive pas dans le monde du jeu vidéo. Je suis incapable de citer un seul jeu dont les gens sont devenus fous comme il l'ont été du Rubik's cube. Une des raisons en est le prix : les gens sont beaucoup moins fous lorsqu'il s'agit de dépenser une cinquantaine d'euros que lorsqu'il s'agit de cinq euros. De toute façon, compter sur le lancement d'une mode, c'est un peu jouer à la loterie. Pour chaque gagnant, il y a des dizaines de milliers de perdants. Pour chaque Rubik's cube, il y a des milliers de merdouilles qui ont vécu trois jours avant de sombrer dans l'oubli.

J'avais une idée dont je parlais à tout le monde depuis des années : faire un jeu sur les chevaux, et en vendre des millions aux petites filles. Maintenant Mattel l'a finalement fait avec *Barbie Club d'Équitation*, et il vont se faire un paquet. Je ne suis pas amer, premièrement parce que je suis sûr qu'ils n'ont pas « volé » mon idée – c'est quelque chose d'assez évident quand on y pense – et ensuite, parce qu'il n'y a aucune raison que je ne puisse pas faire un jeu de chevaux aussi. Peut-être que le mien sera meilleur et se vendra plus. C'est une affaire de divertissement, pas de technologie. Les idées géniales sont la clé de voûte d'entreprises comme DuPont et Dow Chemicals (NDT : deux grandes entreprises dans le domaine de la chimie) où la recherche et les brevets sont essentiels à leur succès, mais dans le divertissement interactif, n'importe qui peut faire un jeu sur n'importe quoi — ce qui compte c'est avant tout de faire un bon jeu.

Obtenir un travail

Il y a deux façons très différentes d'obtenir un travail pour faire des jeux vidéos, et chacune a ses avantages et ses inconvénients. Il y a d'autres chemins qui mènent à cette industrie (vente, marketing, législation, comptabilité, etc.) mais ils n'ont pas beaucoup de lien avec la création du jeu, donc je ne les évoquerai pas ici. Ces deux méthodes sont *développer* et *produire*.

Se lancer dans le développement

La première méthode et la plus directe est de développer un talent dont l'industrie a besoin, et de le lui vendre. Il y a une longue liste de ces derniers : programmation, illustration 2D, modélisation 3D, ingénierie du son, écriture, édition vidéo. La plupart ont des sous-disciplines : programmation 3D, programmation sonore, programmation d'IA, programmation d'interfaces utilisateur. Ce sont les professions fondamentales

requis pour créer des jeux vidéos, et c'est ce à quoi pense la plupart des gens quand ils disent « développeur de jeux » : quelqu'un qui crée et assemble les différentes parties du jeu.

L'avantage de ce genre de métiers est qu'il vous amène directement à la création du jeu et paie bien dès le début, surtout si vous êtes bon. Les entreprises sont toujours à la recherche d'artisans compétents qui connaissent vraiment leur métier. La plupart des différentes compétences sont enseignées au cours des études post-bac, et il commence même à y avoir des cours dédiés entièrement dédiés au développement de jeux. En général, il y a deux sorte d'entreprises pour lesquelles travailler : les éditeurs qui ont des studios de développement internes, et les studios de développement qui vendent leurs productions aux éditeurs.

L'inconvénient de ce type de métier est, du moins chez les éditeurs, qu'il y a un plafond hiérarchique – il est difficile d'être promu au delà d'un certain niveau. Il y a à cela deux raisons. Premièrement, si vous êtes excellent techniquement ou artistiquement, une entreprise va vouloir que vous continuiez à faire ce que vous faites. Si vous êtes promu à un poste de management, vous n'aurez plus le temps de faire ce pour quoi vous êtes le meilleur, ce pour quoi ils vous apprécient le plus. À moins de leur démontrer que vous êtes un meilleur manager que programmeur, ils vous feront continuer à programmer. Les éditeurs en expansion recrutent presque toujours leurs managers intermédiaires à l'extérieur plutôt que de promouvoir leurs créatifs.

La deuxième raison pour laquelle développer pour un éditeur présente un plafond hiérarchique est que les postes des échelons les plus élevés sont fondamentalement focalisés sur l'argent. Pour vous élever dans une entreprise, vous devez avoir des compétences en rapport avec l'argent, car gagner de l'argent est le but primordial d'une société. De plus, il est préférable d'être du bord des gens qui rapportent de l'argent (vente et marketing) plutôt que de celui de ceux qui le dépensent (développeurs). Les employés du département vente sont les héros qui « font du chiffres », etc. Les développeurs sont ceux qui dépassent les délais, explosent leur budget, et demandent constamment du nouveau matériel hors de prix. Il y a peut-être des ingénieurs à la tête des entreprises d'ingénierie, mais il n'y en aura pas beaucoup à la tête des entreprises du marché de la consommation de masse, qui est le marché dans lequel nous sommes.

Ce n'est pas nécessairement vrai dans le développement de jeux. Contrairement aux *éditeurs*, un studio de *développement* indépendant est sur le marché de l'ingénierie. Son client est un éditeur, pas le consommateur individuel. Il n'est pas rare pour un expert programmeur d'être partenaire d'un studio de développement. Cependant, les studios de développement ne sont pas très stables. Ils vivent en fonction de leurs contrats de développement, et s'ils ne peuvent pas décrocher un nouveau contrat quand l'ancien est rempli, ils disparaissent. Cela peut ne pas importer si vous débutez, mais si vous avez deux enfants et une hypothèque, vous préférerez peut être travailler dans le studio de développement interne d'un éditeur.

Se lancer dans la production

La second façon de se lancer dans l'industrie est en tant qu'expert joueur. Il y a des gens qui aiment passionnément les jeux vidéos, qui vivent pour eux, qui n'aiment rien au monde plus que d'y jouer, d'en être proche et d'en parler. Ces joueurs obtiennent souvent des postes comme stagiaires d'été dans des maisons d'édition, à faire du test. S'il sont solides et compétents, il peuvent être engagés comme testeurs à temps plein, ou s'ils ont beaucoup de patience et de bonnes compétences explicatives, comme agents du service clientèle. De là, il peuvent être promus dans le service d'assurance qualité (contrairement aux testeurs normaux, ces gens font les tests de dernière seconde avant qu'un jeu soit lancé), puis producteur assistant, producteur associé et producteur. Les producteurs gèrent tout ce qui a trait à la création d'un jeu vidéo : ils négocient un accord avec les développeurs, ils sont responsables du budget et des délais, ils travaillent avec les gens du département vente et le marketing pour trouver comment faire vendre le jeu, ils travaillent avec les départements législatifs sur les affaires de licence, etc. Les producteurs ne travaillent pas réellement sur les parties techniques et artistiques du jeu, mais ils donnent la direction générale à suivre. Les producteurs assistants et associés conçoivent généralement les missions et les niveaux, et aident de mille autres façons en fonction des besoins du projet sur le moment.

Les producteurs ont besoin de nombreuses compétences différentes, la plus importante d'entre elles, très subtile, est appelée « sens de production ». C'est une capacité à dire quand le jeu est amusant, quand il ne l'est pas, et plus important, de savoir ce qu'il manque pour le rendre amusant lorsqu'il ne l'est pas. Cela semble facile, mais après que vous ayez joué à un jeu quelques centaines de fois durant le développement — la plupart du temps avec un code buggé et à moitié terminé — il faut de bonnes capacités de jugement pour savoir si c'est en bonne voie ou pas.

Cette voie d'accès à l'avantage qu'il n'y a pas de plafond hiérarchique — ou s'il y en a, c'est à un niveau beaucoup plus élevé que celui des développeurs. Étant donné qu'un producteur est impliqué dans l'argent — décider du budget de développement, travailler avec les prévisions de ventes — un producteur talentueux peut continuer à s'élever. Un producteur dont le jeu est un succès peut être promu et commencer à gérer toute une ligne de produits similaires. De là, il peut gérer un studio complet, avec plusieurs producteurs travaillant pour lui, etc.

L'inconvénient de cette méthode est qu'il y a une longue période de dur labeur mal payé au début. Les testeurs, agents du service clientèle et producteurs assistants font beaucoup de travail répétitif et ennuyeux. C'est vital pour faire de bons jeux, mais ce n'est pas très glamour. Toute cette expérience sert au développement de ce « sens de production » pour devenir un bon producteur.

Finalement, j'ajouterai qu'il est possible de passer de l'échelle « développement » à l'échelle « production », car c'est ce que j'ai fait. Après onze ans dans le génie logiciel (dont quatre dans l'industrie du jeu), j'ai arrêté de programmer pour devenir un producteur audio/vidéo, et ça fait cinq ans que je fais ça (NDT : nous sommes alors en 1998).

Conclusion

Ceci, d'après mon expérience, est la façon dont les jeux sont créés et dont les gens se lancent dans l'industrie. Si vous voulez un emploi pour créer des jeux, c'est ainsi que vous en décrocherez un, et la plupart des règles sont les mêmes que dans n'importe quelle industrie. Trouver un travail est un travail en soi ; la persévérance est essentielle. Étudiez les jeux. Lisez l'actualité du secteur. Cherchez des entreprises qui vous intéressent. Relancez par téléphone toutes vos candidatures. Soyez poli. Écrivez correctement. Ne soyez pas arrogant ou désinvolte. Prenez vous, et votre travail, au sérieux. Assistez aux conférences du secteur autant que vous le pouvez, et surtout, surtout, à la *Game Developers' Conference* dont le forum pour l'emploi est inégalé dans l'industrie du jeu vidéo. Cela peut paraître être un coup de pub flagrant, mais ne le négligez pas : la GDC est une occasion sans pareille de rencontrer des développeurs et d'en apprendre plus sur le milieu.

Cet article peut avoir semblé un brin cynique, au moins pour ceux qui n'ont jamais travaillé dans l'industrie du jeu vidéo auparavant. Je n'ai pas l'intention d'être cynique du tout ; c'est ainsi que l'industrie fonctionne. Les gens qui sont beaux parleurs et qui serrent des mains deviennent Président de la République ; les gens qui gagnent des prix Nobel non. Cela peut sembler injuste, mais le pays s'en porte probablement mieux.

Je finirai avec une dernière dure vérité : *presque personne ne parvient à faire exactement le jeu qu'il souhaite faire*. Les jeux faits pour être vendus ne sont pas faits par des individus seuls ; ce sont des collaborations, et collaboration signifie compromis. Les développeurs font des compromis avec le producteur, le producteur fait des compromis avec le marketing et le département des ventes. Les seules personnes qui font exactement le jeu qu'elles veulent sont celles qui le font elles-mêmes, sans se soucier de savoir si les gens l'achèteront. C'est merveilleux si vous pouvez le faire, et je pense que nous avons besoin de plus de ce genre de développeurs indépendants, mais ce n'est pas la même chose qu'avoir un emploi rémunéré dans une entreprise florissante.

Si après tout ça, vous avez toujours l'idée en tête, alors allez-y et bonne chance !

Le genre qui ne voulait pas mourir

Par [Pierre Guyot](#) | le 21 août 2011 08:00:00 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (avril 2009).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



Ma vie a été un peu bizarre ces derniers temps. L'autre jour, j'avais décidé d'être chanteur d'opéra et puis finalement je me suis retrouvé coincé dans le château d'un vampire. Quelques heures plus tard, je me suis retrouvé dans la peau d'une adolescente de dix-sept ans un peu stupide et je ne sais comment, j'ai remonté le temps et l'espace pour me retrouver au milieu des Caraïbes à l'époque des pirates et, oh mon Dieu ! Je ne trouve plus mon nécessaire de maquillage !

Et puis je suis devenu un flic du nom de Phoenix Wallis, qui enquête sur un meurtre dans une utopie profondément dérangeante, et maintenant je suis un minuscule tyran qui essaye de récupérer son royaume volé par un usurpateur.

Au cas où vous ne l'auriez pas deviné, je joue à quatre jeux d'aventures en même temps. C'est marrant, même si c'est un peu déstabilisant. Et tout ça en une seule journée. Plus tôt dans la semaine, j'avais essayé de devenir la reine de la promo dans mon lycée et j'ai aussi aidé un fameux archéologue à découvrir les secrets d'une amulette magique.

Les journalistes spécialisés dans le jeu vidéo annoncent souvent avec une aisance déconcertante que les jeux d'aventure sont sur le point de s'éteindre, mais ils ont tort. Les jeux d'aventure ne seront plus jamais le genre dominant qu'ils ont été, mais il y a un marché bien établi pour ce genre de jeux et le nombre de gens qui y jouent est en augmentation, grâce à l'arrivée récente d'un grand nombre de joueuses et de casual gamers sur le marché.

Il y a presque dix ans, j'ai écrit un article pour le Designer's Notebook intitulé « [Il est temps de faire revenir les jeux d'aventure](#) » (NDT : en anglais, « [It's Time to Bring Back Adventure Games](#) »). Puisque près d'une décennie s'est écoulée depuis la parution de l'article original, j'ai décidé de regarder quelques uns de ces jeux pour voir ce qu'il est advenu du genre pendant l'intervalle.

Je ne parle pas de jeux d'action/aventure — les jeux d'action avec une importante composante narrative — mais bien de purs jeux d'aventure, qu'il s'agisse de point-and-click ou de jeux avec des contrôles directs. Voici ce que j'en ai conclu.

Ce qui s'est amélioré

L'animation. Il s'agit de l'amélioration la plus flagrante. La plupart des autres jeux offrent aux joueurs un nombre limité d'activités — un shooter consiste à tirer et une simulation d'élevage à s'occuper d'un animal. Mais dans un jeu d'aventure, vous pouvez faire toutes sortes de choses — charger un canon, huiler une porte, danser une gigue. Ceci implique que votre avatar a besoin d'une grande quantité d'animations.

La plupart des jeux d'aujourd'hui sont encore en 2D, mais les personnages sont souvent affichés avec des animations construites à partir de modèles 3D. Il est plus facile de construire et de rendre une large variété d'animations d'après un modèle bien utilisé que de tout dessiner à la main.

Les mouvements sont plus fluides et plus naturels. On n'en est pas encore à la qualité d'un Pixar, parce que les jeux d'aventure n'ont pas le budget d'un Pixar. Mais ils sont bien meilleurs sur ce point qu'ils ne l'ont été.

Les rôles que le joueur peut incarner. Les jeux d'aventure ont toujours offert une large palette de rôles au joueur, encore une fois parce que ces jeux ne sont pas contraints de se limiter à un certain type d'actions.

Comme on peut le voir dans ceux auxquels je suis en train de jouer, il y a beaucoup de possibilités — probablement plus encore qu'à l'âge d'or des jeux d'aventure. La comédie offre naturellement beaucoup de possibilités pour les rôles que le joueur peut incarner et une bonne part des jeux d'aventure sont des comédies.



Les décors. Comme pour tous les autres genres, les décors dans les jeux d'aventure se sont beaucoup améliorés au cours des dix dernières années. C'est particulièrement important parce que les joueurs du genre aiment à pouvoir profiter du décor.

Malgré l'impact considérable des accélérateurs 3D sur le reste du monde vidéoludique, beaucoup de jeux d'aventure utilisent encore un arrière-plan peint à la main pour chaque scène.

Les environnements 3D peuvent être une bénédiction pour les jeux d'aventure. Ils sont merveilleux si l'équipe arrive à élaborer des environnements aussi riches qu'un décor peint. Mais avec les budgets limités de notre époque, les développeurs n'ont pas souvent les ressources pour tout modéliser et le joueur se retrouve à courir dans de grands espaces vides.

Pourquoi, par exemple, le quartier général de la police où travaille Phoenix Wallis dans *Culpa Innata* ressemble-t-il à une gigantesque cathédrale avec en tout et pour tout six bureaux dedans ? Cela prend un temps non négligeable juste pour marcher du bureau de Phoenix jusqu'à la porte d'entrée. J'aimerais beaucoup voir un jeu d'aventure avec le moteur et le niveau de détail des jeux Crytek (NDT : les développeurs de Crysis, entre autres).

L'harmonie des jeux. Ceux qui suivent mes articles depuis longtemps savent déjà qu'il n'y a rien que je ne déteste plus que de sortir d'un univers de jeu à cause d'une blague bas-de-gamme ou d'un anachronisme. De ce que j'en ai vu, les jeux ont une meilleure cohérence interne qu'auparavant.

L'amélioration des graphismes y contribue, mais je pense aussi que les jeux d'aventure se prennent un peu plus au sérieux — même les comédies — et le résultat en est une expérience plus immersive et plus esthétique.

Le son. Pas de surprise ici. La qualité audio des jeux a globalement augmenté, et donc dans les jeux d'aventure aussi. Ceci ne signifie pas que le contenu audio s'est amélioré, comme nous allons le voir.

Ce qui ne s'est pas amélioré

Malheureusement, les jeux d'aventure ne se sont pas améliorés sur tous les points. Voici quelques éléments sur lesquels il faut que nous continuions à travailler.

Des énigmes obscures et incroyablement compliquées. Les énigmes dans les jeux d'aventure d'aujourd'hui semblent être les mêmes que dans les vieux jeux, pour le meilleur et pour le pire, ce qui veut dire qu'on trouve encore quelques énigmes d'une complexité effroyable.

Si tant est qu'il existe une catégorie de hardcore gamers pour le jeu d'aventure, voilà ce qu'ils aiment. Mais c'est un tout petit segment du marché, et si le genre veut attirer le segment grandissant des casual gamers, il doit devenir plus accessible.

Les *non sequitur* et les énigmes qui requièrent une forme extrême de pensée latérale constituent deux des pires TDC que je connaisse (NDT : voir l'article « [Vilain Game Designer ! Pas de biscuit pour toi !](#) » et ses multiples suites). Je ne m'offusque pas si la solution d'une énigme est inhabituelle ou si elle nécessite la mise au point de gadgets à la McGyver, du moment que le résultat est suffisamment crédible. Si la solution est si bizarre que le joueur peut uniquement la découvrir par *trial and error*, le designer a mal fait son job.



Ron Gilbert, le designer derrière la série des *Monkey Island* se positionne contre les énigmes « à l'envers » — des énigmes dans lesquelles vous trouvez tous les éléments nécessaires à la résolution avant même de tomber sur l'énigme.

Un tel déroulement encourage le joueur à récupérer tout ce qu'il voit, juste au cas où il en aurait besoin plus tard et je suis tout à fait d'accord que cela semble bizarre : vous vous déplacez en trimbalant avec vous une collection hétéroclite d'objets sans aucune raison apparente.

Il estime que le joueur devrait tomber sur une énigme sans la solution en main et ainsi être motivé à explorer plus avant, et je comprends sa position. Néanmoins, je ne me sens pas aussi radicalement concerné que lui par ce problème. A mon sens, une complexité excessive constitue un péché bien plus grave.

De mauvais dialogues. Les jeux d'aventure ont plus de dialogues que n'importe quel autre genre et ceux-ci constituent une partie importante du gameplay et non une simple exposition qu'on peut passer.

Malheureusement, nous faisons des jeux où les gens parlent beaucoup trop longtemps (et trop lentement) et où de nombreuses conversations semblent extrêmement artificielles. Ceci est partiellement imputable à une mauvaise traduction de jeux écrits dans une langue étrangère, mais une bonne part reste due à une mauvaise écriture.

Regardez par exemple des séries télé qui tournent principalement autour du dialogue, comme *New York, police judiciaire*. Les gens ne jacassent pas pendant dix minutes sans interruption : la plupart des scènes durent moins de trois minutes dans leur intégralité. Il faut que ça reste court et percutant.

Un mauvais jeu d'acteur et une mauvaise production audio. Parfois, un mauvais script peut être sauvé par un bon jeu d'acteur, mais un mauvais jeu d'acteur peut faire d'un bon script une catastrophe. Encore une fois, les jeux d'aventure tournent autour des personnages et les interactions entre humains doivent donc sonner vrai.

Engagez des acteurs dignes de ce nom et faites les jouer ensemble. Trop de scènes de jeux d'aventure semblent clairement avoir été enregistrées ligne par ligne, avec un seul acteur présent. On n'a pas l'impression d'assister au jeu d'un acteur, mais d'entendre un narrateur réciter.

La BBC, l'un des derniers diffuseurs dans le monde qui continue à produire des pièces radiophoniques en anglais, fait toujours enregistrer ses acteurs autour d'un seul micro et s'ils sont supposés se trouver dans une cuisine, ils enregistrent dans un décor de cuisine, de façon à ce que l'acoustique soit crédible.

Je réalise bien qu'il est impossible d'enregistrer les dialogues d'un script à embranchements de la même que ceux d'un script linéaire et sans doute n'atteindront nous jamais un dialogue qui sonne parfaitement naturel.

Néanmoins, quand j'étais chargé des doublages pour *Madden NFL Football*, j'ai appris tout l'intérêt qu'il y avait à placer les lignes à enregistrer au milieu d'une phrase plus longue ou d'un paragraphe.

Nous enregistrons l'intégralité du paragraphe, puis coupons les parties dont nous n'avons pas besoin. Le résultat sonnait comme un flot continu de mots, plutôt qu'un mot ou une ligne enregistrés seuls.

La musique. Il s'agit peut-être uniquement des exemples auxquels j'ai joué récemment, mais les musiques dans les jeux d'aventure d'aujourd'hui semblent plus courtes et plus répétitives que celles du Bon Vieux Temps. On dirait qu'on a taillé dans le budget à cet endroit pour économiser de l'argent.

Les poitrines. J'ai l'impression que les artistes (principalement masculins, hélas) qui ont créé les jeux auxquels j'ai joué fantasment beaucoup sur les femmes, mais ils n'ont visiblement pas beaucoup regardé à quoi elles ressemblaient réellement.

Dans *Culpa Innata*, il y a une scène où quelqu'un fait un discours devant une grande assemblée. Toutes les femmes de la pièce, sans exception, ont des poitrines énormes — et absolument identiques.

Il s'agit d'une autre TDC et je suis déçu de voir qu'elle continue d'exister, tout particulièrement aujourd'hui où une grande partie du public est constitué de femmes. Grandissez les gars et renseignez vous un peu sur la diversité de l'anatomie humaine tant que vous y êtes.

Conclusion

En 1989, il y a encore plus longtemps que mon article « Il est temps de faire revenir les jeux d'aventure », Ron Gilbert a écrit un excellent recueil de principes pour game designers intitulé « Pourquoi les jeux d'aventure sont-ils nuls et que faire pour y remédier ? » (NDT : en anglais, « [Why Adventure Games Suck and What to Do About It](#). »). En fait, ce recueil est précieux pour quiconque travaille sur un jeu avec une dimension narrative et je vous le recommande chaudement.

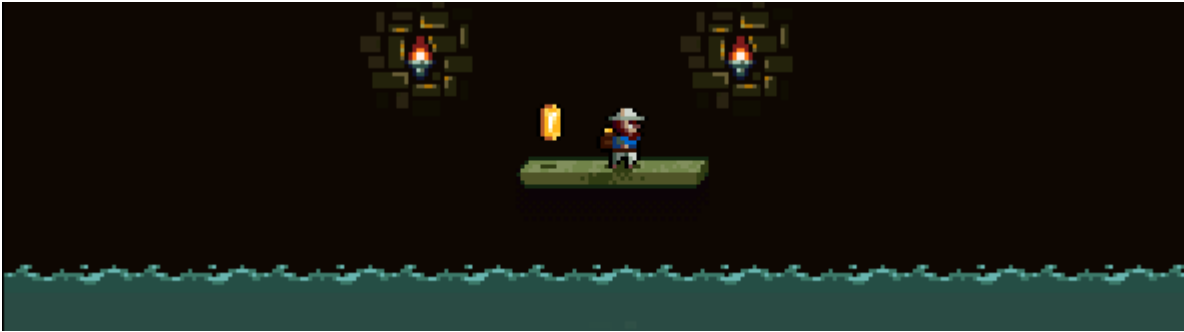
Dans les prochaines années, grâce à la dématérialisation, aux nouveaux outils de développement gratuits et au mouvement indie, nous allons commencer à voir sortir toutes sortes de jeux funky qu'on ne verrait jamais sur les étagères d'un supermarché. Des jeux d'architecture. Des simulations de train. Des jeux d'archéologie. Et, plus classique, des jeux d'aventure avec des énigmes.

On ne peut pas tous réaliser un Mario ou un Halo et tout le monde n'en a pas envie. Celui qui compte entrer en compétition directe avec Mario ou Halo a intérêt à avoir plusieurs dizaines de millions à la banque rien que pour le marketing. Si vous développez pour une niche spécifique du marché, vous ne deviendrez jamais immensément riche, mais au moins vous n'aurez pas à lutter contre la machine des relations publiques de vos gigantesques concurrents.

Les jeux d'aventure ont trouvé leur niche. Allez sur [AdventureGamers.com](#), [the Adventure Shop](#) ou JustAdventure+ et jetez y un œil. Vous serez surpris par leur nombre. Her Interactive continue de sortir les *Nancy Drew* à un rythme constant et il y en a maintenant près de vingt. Tant qu'il y aura des gens qui veulent jouer à des jeux d'aventure, il y aura des gens qui voudront les créer. Et nous trouverons encore de nouvelles façons de les améliorer.

Compte-rendu de Ludum Dare #21

Par [Antoine Gersant](#) | le 30 août 2011 13:57:25 | Catégories : [Game Design](#), [Graphisme](#), [Level Design](#), [Ludum Dare](#), [Ma vie](#), [Pixel art](#)



Il y a tout juste une semaine se terminait la 21ème édition du [Ludum Dare](#), une compétition de développement de jeu en temps limité. Les règles sont simples : tous les outils sont autorisés et les participants doivent travailler seul pour produire *from scratch* un jeu complet en respectant un thème imposé.

Cette année, un total de 599 (!!) jeux ont été réalisés à l'occasion de l'évènement (dont ~90 d'entre eux hors-concours avec un peu plus de personnel ou de temps). Évidemment, je faisais partie de la fête et cet article sera consacré à mon récit des événements. Voici tout d'abord [Ernest, the Plump Adventurer \(in trouble\)](#), ma participation avec laquelle je vais vous bassiner pendant le reste de cet article.

Avant que je ne démarre mon récit, vous remarquerez l'apparition d'une nouvelle catégorie d'articles sur Les Forges. Cet article est en effet le premier taggé « Ma vie ». J'espère que ma petite vie de développeur sera suffisamment intéressante pour justifier d'autres articles.

Cela fait plusieurs années que je suis de loin l'organisation du *Ludum Dare* avec intérêt et je me suis décidé à y participer cette année sur un coup de tête, deux jours avant le début de la compétition. J'étais paisiblement en train de voter pour aider à choisir le thème de cette année, et je me suis rappelé comme je m'étais amusé à participer à une compétition de développement en 48 heures sur Flashkit (un forum aujourd'hui tout mort) il y a deux ans. J'étais curieux de voir l'étendue de ma progression de ces deux dernières années, et c'était aussi une occasion de mettre à l'épreuve *Bacchus*, un moteur de jeux en [HaXe](#) que j'ai développé au fil de l'année passée.

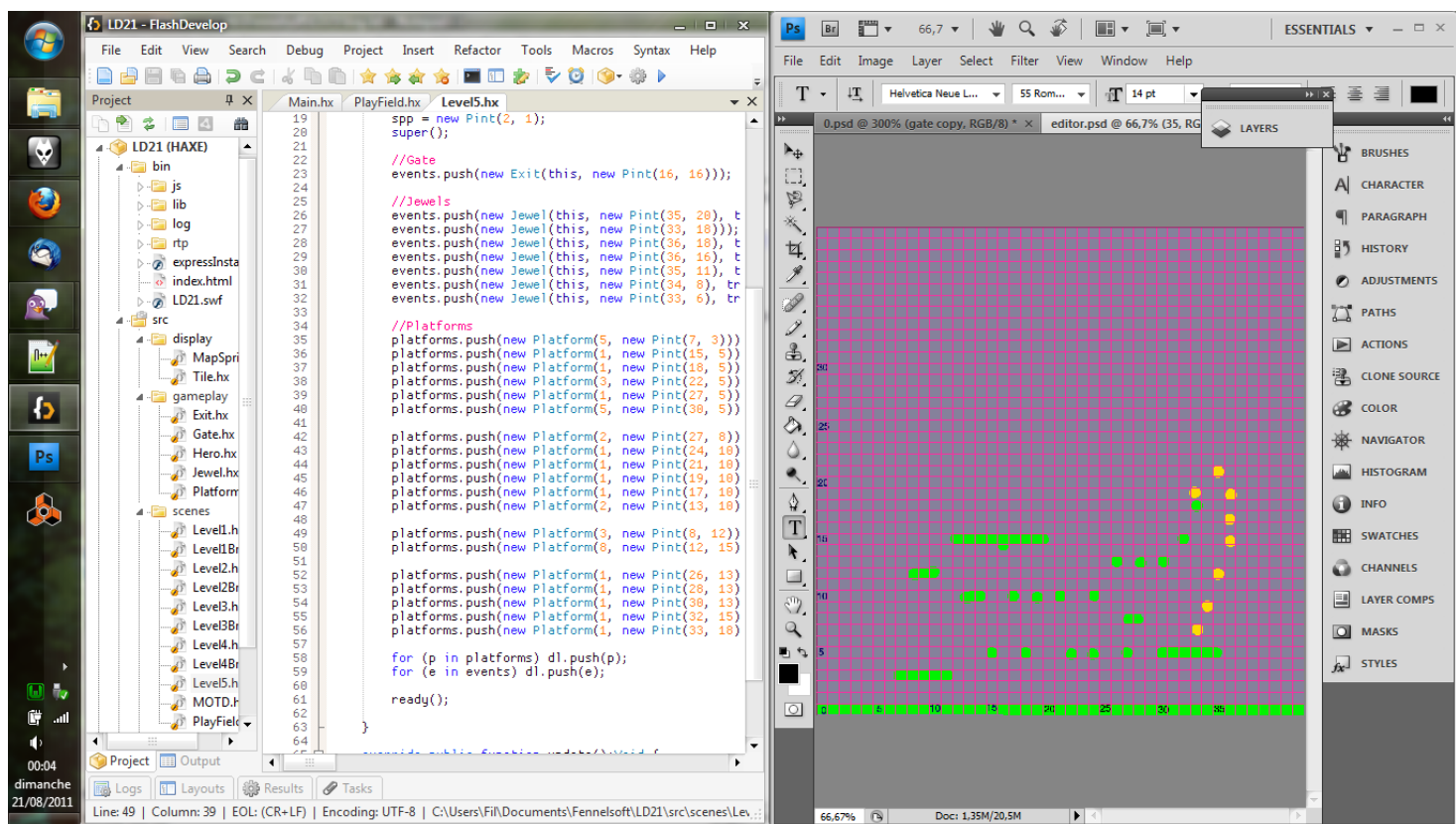
L'aventure démarre plusieurs heures avant le coup d'envoi de la compétition (à 4h00 du matin heure française — ouille) par une indispensable séance de ravitaillement au supermarché pour acheter de quoi survivre un weekend sans voir le jour et sans mourir de faim. J'avais prévu de me coucher tôt pour démarrer la compétition de bon matin mais un dîner avec Pierre (qui traduit et publie plein d'articles sur le site que vous lisez en ce moment) et un bout de nuit passé sur *Limbo* ont quelque peu entamé ma vigueur matinale.

J'ai finalement réussi à basculer hors de mon lit vers 10h00 du matin (soit 6 heures après le début de la compétition. Aïe, plus que 42 heures !) pour aller prendre connaissance du thème de cette édition : « Escape ». Cruelle déception, je m'attendais à un thème plus évocateur comme les excellents « Castle », « Ruins » et « Self-replicating » qui étaient en lice (sans parler du « It's dangerous to go alone, take this ! » de la précédente édition). Après quelques minutes de réflexion à la recherche d'une idée géniale qui avait décidé de ne pas venir, j'ouvrai *Photoshop* avec dans l'intention de dessiner quelques *tiles* pour me donner des idées.

Quelques heures plus tard j'avais devant moi assez de pixel-art rudimentaire pour développer un jeu de plateformes dans lequel le héros, un aventurier barbu ambiance *Dr. Livingstone* (toute ressemblance avec l'apparence et le prénom d'un célèbre *Game Designer* dont les articles sont régulièrement publiés sur les

Forges n'est peut-être pas entièrement fortuite), devrait échapper à une montée d'eau. Pas très original mais au moins ça collait à peu près au thème et c'était réalisable dans les temps. J'étais tout de même un peu naïf d'imaginer que j'aurais le temps de coder un générateur de niveau procédural. La faim commençant à se faire sentir, je me suis offert une petite pause bien méritée.

J'en ai aussi profité pour faire un tour sur le canal IRC dédié à la compétition qui avait près de 400 connectés pendant le weekend. C'est non seulement très impressionnant mais aussi motivant ! J'ai passé mon après-midi à coder le gros des mécaniques de jeu : déplacement du personnage et sauts, plateformes, collisions, merdouilles brillantes qui font **gling gling** à ramasser, fin de niveau et mort du joueur. Aux environs de minuit j'avais de quoi commencer à faire un jeu, et je crois que c'est aussi pendant cet après-midi que j'ai animé le personnage (et l'eau qui monte). Évidemment il n'était pas question de se coucher si tôt, et j'ai passé ma nuit à faire du *Level Design*. Avec le recul, je réalise que c'est une activité où je n'excelle pas et il aurait sans doute été bon de lui consacrer une heure où j'étais plus réveillé (et moins alcoolisé). Comme je n'avais pas d'éditeur de niveaux sous la main, il a fallu ruser pour designer rapidement. J'ai mis en place un fichier *Photoshop* avec une grille et des unités de mesure dans lequel je dessinais mes niveaux « au brouillon » avant de les intégrer « en dur » dans le jeu. Voici à quoi ressemblait mon écran à ce moment :



J'ai fini par aller me coucher vers 5 heures du matin après avoir fait les 10 premiers niveaux du jeu et l'écran titre (alors que je m'endormais sur mon clavier). Cinq heures de sommeil plus tard, branle-bas de combat pour reprendre le développement (après un copieux petit-déjeuner). J'ai repris mes méthodes de la veille pour faire 5 niveaux de plus (pour un total de 15). Je voulais en faire 20 mais j'étais complètement à cours d'idées de niveaux et ça risquait de se voir.

Je me suis donc consacré à une autre fonctionnalité : rajouter des torches (décoratives) dans les niveaux. Problème, je n'avais toujours pas d'éditeur de niveaux et je n'avais pas gardé la plupart de mes brouillons sous *Photoshop*. J'ai une fois de plus eu recours à une solution un peu ridicule mais efficace. J'ai fait une version du jeu dans laquelle le joueur ne risquait pas de mourir noyé et où il pouvait sauter à l'infini. J'ai ajouté une touche pour enregistrer dans les *log* la position actuelle du personnage et je me suis ensuite baladé en volant dans les niveaux en appuyant sur cette touche magique à chaque endroit où je comptais placer une torche. Une demi-heure plus tard l'opération était terminée.

Je me suis offert une nouvelle pause IRC pour faire tester mon prototype et pour en essayer quelques-uns faits par mes « concurrents » (que j'ai ensuite eu le plaisir de retrouver finis à l'issue de la compétition). Les retours étaient plutôt positifs et plein de bonnes idées (comme celle de la jauge d'oxygène, rajoutée quelques heures plus tard). Seulement voilà, il était déjà près de 18 heures et mon jeu était toujours silencieux. J'ai réalisé quelques bruitages avec [BFXR](#) et je me suis lancé dans la composition de musiques avec mon synthétiseur et *Reason*. Le style du jeu et les contraintes de temps m'ont imposé un style « chiptune » dans lequel j'ai assez peu d'expérience mais je suis assez content du résultat, surtout au regard du temps passé (~3 heures pour ~3 minutes de musique, c'est assez rapide par rapport à mes standards). Un peu comme dans tout le reste du jeu, j'ai laissé passer quelques détails plus ou moins importants comme en témoigne la capture d'écran de *Reason* ci-dessous. Bizarrement, le thème de fin de jeu a un petit problème de stéréo :



Grâce à un ami qui a testé le jeu plusieurs fois (le pauvre) pendant que je composais les musiques, j'ai pu corriger un paquet de bugs avant de finalement soumettre mon jeu quelques minutes avant la *deadline* à 4 heures du matin. Une fois le jeu envoyé, j'ai décapsulé ma dernière bière et la tension a commencé à retomber. Sans que je m'en aperçoive, j'avais focalisé toute mon attention des 48 heures précédentes sur le jeu et plus rien d'autre n'existait dans mon esprit. C'est un sentiment assez incroyable, un peu comme s'endormir le soir et se réveiller le matin en découvrant qu'on a fabriqué un jeu entier pendant son sommeil. Rien que pour ce sentiment d'accomplissement inattendu, je recommande à tous ceux qui le pourront de participer à la prochaine édition de *Ludum Dare* (en décembre prochain).

Mon jeu semble plutôt bien reçu par la communauté des participants. Beaucoup ont fait des remarques très justes sur les mécaniques de jeu (mort trop pénalisante, double saut absurde car le saut « simple » ne sert jamais, etc.) et le *Level Design* qui pêche un peu par moment. Les graphismes et la musique semblent être les points les plus appréciés, ce qui est assez ironique vu que je me considère avant tout comme un programmeur.

J'ai aussi tenté d'envoyer le jeu à ma famille pour voir ce qu'ils pouvaient en faire. Hélas, les non-joueurs se sont retrouvés bloqués très vite (niveau 1, 2 ou 3) par manque de dextérité tandis que les joueurs occasionnels ou réguliers sont parvenus au bout du jeu plus ou moins laborieusement. Tout cela ne fait que confirmer que l'on ne rend jamais un jeu trop simple. Quelques heures de tests supplémentaires m'auraient aidées à ajuster la difficulté plus finement je suppose. La tendance naturelle est **toujours** à faire trop difficile.

Au final, mon jeu est loin d'être parfait mais je suis très fier (vous savez que j'aime m'envoyer des fleurs) d'avoir accompli tout ça en 48 heures. J'ai fait un jeu complet, j'ai mis mon moteur à l'épreuve, j'ai participé à un événement fantastique et j'ai foutu en l'air toute chance d'être en forme Lundi matin. Je n'ai qu'une hâte c'est de recommencer !

Je ne vais évidemment pas vous laisser filer sans parler de quelques autres jeux réalisés pour la compétition. Parmi mes préférés pour le moment, citons :

- [Stratus](#), un court jeu d'aventures Steampunk
- [Mouse Trap](#), un jeu où l'on incarne une souris tourmentée dans un labyrinthe
- [Flee Buster](#), 3 jeux en un (avec un petit côté *Wario Ware*)

Bon jeu à tous, en attendant de vous voir participer à la prochaine édition !

[Lire la suite →](#)

Chiffres, émotions et comportement

Par [Pierre Guyot](#) | le 15 septembre 2011 15:22:54 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (janvier 2009).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Les ordinateurs ont été inventés pour manipuler des nombres. La machine à différences de Charles Babbage a été construite pour calculer des polynômes et Lady Lovelace a écrit le premier programme de sa machine analytique pour calculer ce qu'on appelle les nombres de Bernoulli. Ce n'est que bien plus tard que les ordinateurs furent utilisés pour des tâches appliquées comme l'artillerie. Et c'est encore plus tard que les jeux vidéo furent inventés. Nous avons placé une couche de vernis sur une simulation mathématique pour créer un monde imaginaire, mais souvent le vernis n'est pas très épais. C'est particulièrement visible dans les simulations économiques et les jeux de rôles. On gagne en regardant des chiffres sur un écran.

$$\#(\bigcup_{i \in I} X_i) = \sum_{i \in I} \#X_i ;$$

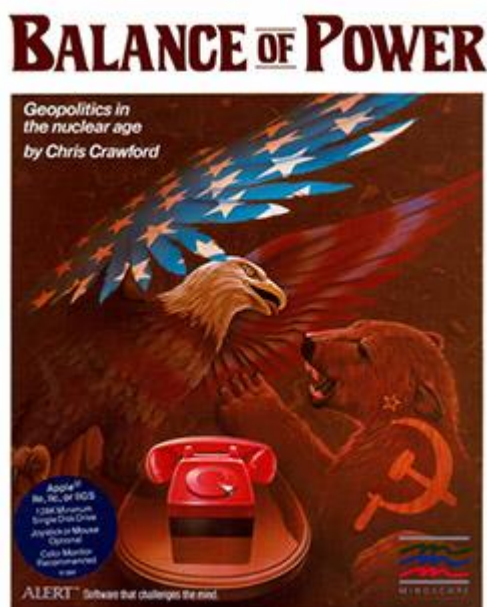
Ceci fait sens dans les simulations économiques, particulièrement celles qui sont à but éducatif. Récemment, j'ai eu la chance de pouvoir jouer à la version beta d'un drôle de petit jeu intitulé *Super Energy Apocalypse* de Brain Juice Games. Le game designer, Lars Doucet, le résume ainsi : « Des énergies renouvelables... et des ZOMBIES ! ». Chaque nuit, votre communauté est attaquée par des zombies (ou plus exactement, des aliens mangeurs d'ordures déguisés en zombies... C'est une longue histoire.). Pour gagner, vous devez développer des sources d'énergie non polluantes afin d'alimenter vos défenses. Les zombies adorent les fumées toxiques et les déchets nucléaires, vous devrez donc garder un œil sur votre taux de production d'énergie et sur la quantité de fumées et de déchets que vous produisez ainsi. C'était un jeu amusant et, puisqu'il se base sur de véritables systèmes pour générer de l'énergie, j'ai appris quelques petites choses.

Dans l'ensemble néanmoins, je commence à me lasser des jeux dans lesquels les mathématiques sont trop apparentes. J'aime jouer à des jeux de rôle pour l'exploration, pour l'histoire et pour la diversité de lieux et de personnages qu'ils proposent, mais je suis moins enthousiaste en ce qui concerne la marchandage et l'upgrading permanent. Cette façon d'insister sur l'équipement me semble particulièrement *geek*. Il n'y a pas tant de différence que ça entre se la raconter avec sa Hallebarde Flamboyante Supérieure de Raclée Majeure et se la jouer avec son PC *Alienware* overclocké, watercoolé et muni du dernier type de DEL à la mode.

Les RPG sont basés sur l'évolution des personnages, mais cette évolution est exprimé en termes numériques. A la fin du jeu, votre personnage est plus rapide, plus fort, plus adroit, etc. Les chiffres le prouvent. Mais dans ce cas-là, il pourrait aussi bien être un robot qu'un être humain. On n'accorde que très peu de place à ce qu'on pourrait appeler une évolution psychologique du personnage — une évolution en tant que personne.

Je commence donc à penser à des jeux où les chiffres seraient remplacés par un comportement. Bien sûr, de nombreux jeux utilisent des automates finis (NDT : pour plus de précisions, je vous renvoie à la page [Wikipédia](#) sur le sujet. Pour faire simple, c'est ici une forme d'IA très simple où l'on peut se retrouver dans un seul état à la fois) pour les IA de leurs personnages et ces automates récupèrent des valeurs numériques en entrée (« Si l'ennemi est à portée, passer en mode attaque »). Mais ces automates ne changent pas — les personnages n'évoluent pas et n'apprennent pas à modifier leur comportement. Et comme l'a fait remarqué le designer et professeur Michael Mateas, les automates finis ne peuvent pas marcher et mâcher du chewing-gum en même temps — c'est-à-dire qu'ils ne peuvent occuper qu'un état à la fois, ce qui leur interdit de mettre en œuvre des actions complexes.

Un exemple ancien de « personnage » qui changeait en apparence d'humeur et de comportement de façon complexe apparaît dans le jeu *Balance of Power* de Chris Crawford. J'ai déjà écrit souvent au sujet de ce jeu parce qu'il est à la fois brillant et unique — une simulation de gestion géopolitique avec deux superpuissances, une idée qui, à ma connaissance, n'a jamais été reprise depuis. Il s'agissait d'un jeu à un seul joueur. Vous jouiez soit les Etats-Unis, soit l'URSS et vous deviez essayer de maximiser le prestige de votre pays aux dépens de l'autre camp, en partie en leur faisant face lors de crises diplomatiques (la crise des missiles de Cuba en est le plus célèbre exemple réel). A tour de rôle, les deux camps entreprenaient des actions diplomatiques variées (comme la formation d'alliances) dans d'autres pays du monde et au tour suivant, chaque camp avait la possibilité de s'opposer aux actions de l'autre camp — une crise. Quand une crise survenait, l'un des deux camps devait finir par céder. Il était essentiel de choisir soigneusement sur quels conflits intervenir, car si vous provoquiez une guerre nucléaire, vous aviez perdu.



Balance of Power a été un tel succès que Crawford a écrit un livre qui expliquait comment le jeu fonctionne, avec toutes les équations utilisées. Le livre s'intitulait *Balance of Power : International Politics as the Ultimate Global Game*. Il est depuis longtemps en rupture de stock, mais Crawford en a fait une version texte disponible sur son site. Je pense qu'il s'agit d'une lecture essentielle pour tout game designer : une explication claire et intelligente de la façon de convertir des formules mathématiques en comportements.

Il n'y avait pas à proprement parler de personnages dans *Balance of Power*, mais les actions de l'IA ennemie donnaient dans une certaine mesure l'impression qu'il s'agissait d'un vrai gouvernement, et Crawford a renforcé ce sentiment avec le langage diplomatique utilisé pour annoncer les actions de chaque camp. Les annonces pouvaient être fermes ou conciliantes selon la façon dont le camp adverse se « sentait ». Crawford a créé une variable appelée la pugnacité pour chaque camp et une autre appelée la tension pour l'ensemble du jeu. La pugnacité était initialisée à une valeur aléatoire proche de la valeur neutre (plus haute pour les Russes que les Américains ; cette vision du monde est aujourd'hui un peu dépassée) et augmentait ou diminuait à mesure que la superpuissance adoptait un comportement agressif ou conciliant respectivement. La tension général du jeu augmentait avec les activités militaires et les crises et ne diminuait qu'avec le temps.

Comme Crawford l'écrit dans son livre, « Ces deux variables ont pour effet de créer une ambiance dans le jeu. Les joueurs qui mènent des stratégies de confrontation augmenteront la pugnacité de leur camp et la tension du jeu. Si c'est fait correctement, une stratégie aussi impitoyable encourage les nations plus faibles à « finlandiser » avec le joueur (NDT : dans le domaine politique, le verbe anglais *to finlandize* signifie ne pas chercher à défier une puissance plus importante tout en conservant la souveraineté nationale. C'est un terme péjoratif souvent associé avec une idée de lâcheté. Ici il s'agit de s'aligner diplomatiquement sur le joueur, plutôt que de chercher à s'opposer à lui.). Mais de petites erreurs peuvent amener la pugnacité de l'ennemi à augmenter et la votre à baisser si vous vous retrouvez à céder trop souvent au cours de crises diplomatiques. »

Si vous jouez suffisamment longtemps au jeu, vous pouvez effectivement ressentir cette ambiance. Des actions agressives vont amener l'autre camp à devenir lui aussi plus agressif, mais si vous vous montrez trop peu sûr de vous, les petits pays auront tendance à penser que vous êtes faibles et à s'allier avec l'autre camp. Vous pouvez tout à fait sentir lorsqu'on ne vous respecte pas. La pugnacité, le respect diplomatique et le prestige géopolitique sont des concepts émotionnels qu'on ne voit pas souvent dans le jeu vidéo, et *a fortiori* en 1985 quand *Balance of Power* est sorti.

Il y a quelque chose comme cinq ans, j'ai écrit un article intitulé « Quel failles peuvent bien se cacher dans le cœur des PNJ ? ». Dans celui-ci, je discute du sujet de la création de personnages véritablement tridimensionnels — des personnages qui se comportent de façon apparemment incohérente (et donc réaliste) parce qu'ils sont confrontés à des conflits internes. Par exemple, au lieu de caractériser les sentiments d'une personne pour une autre avec une seule variable chiffrée qui varie du négatif (je le hais) au positif (je l'aime), imaginez que nous utilisions deux variables, l'une pour l'amour et l'autre pour la haine. S'il y a beaucoup d'amour et pas de haine, les deux personnages vont être amis (ou plus si affinités !). S'il y a beaucoup de haine et pas d'amour, ils seront ennemis. Mais il pourrait y avoir aussi beaucoup des deux, ce qui peut arriver en certaines circonstances. Il faudrait alors élaborer un système de comportement qui permette d'exprimer ce conflit interne.

Je ne sais pas exactement comment fonctionne *Façade*, un jeu expérimental brillant sur le thème d'un mariage qui s'effondre par Michael Mateas et Andrew Stern, mais les personnages éprouvent véritablement des sentiments complexes les uns vis-à-vis des autres. *Les Sims* aussi parviennent à transformer des chiffres en comportements — quoique d'une façon peut-être un peu moins sophistiqué que dans *Façade*. *Façade* tourne autour d'un couple de personnages et de leur ami, le personnage incarné par le joueur.



Les Sims permettent au joueur de créer n'importe quel type de personnage dont il a envie, il est donc nécessaire qu'il dispose d'un modèle de relations et de comportements générique, qui peut fonctionner pour tout le monde. Mais dans les deux cas, ces jeux produisent un résultat sous forme d'interactions humaines. Le joueur ne reçoit pas de chiffres ou de jauges. Il n'est pas censé penser les personnages du jeu en termes de simulation, mais en tant que véritables personnes.

Façade utilise le langage, ce qui introduit toutes sortes de complications — parce qu'une conversation doit pouvoir se dérouler de façon intelligible même si les mots choisis reflètent l'état émotionnel du personnage. *Les Sims* utilisent uniquement le Simlish (NDT : un faux langage inventé pour le jeu, qui n'a pas de sens véritable mais dont la sonorité imite une vraie conversation) mais les personnages expriment les sentiments à travers les symboles qui apparaissent à l'écran et ils doivent prendre certaines décisions : dois-je continuer à parler à cette personne ou arrêter et partir ? Dois-je interagir physiquement avec lui ou elle, le ou la prendre dans mes bras, l'embrasser, le ou la gifler, me battre avec lui ou elle ? Un tel système est extrêmement compliqué à développer et encore plus compliqué à tester, tout particulièrement si les personnages sont capables de ressentir des émotions complexes. Si un personnage se comporte de façon contradictoire, est-ce un bug ou est-ce une expression légitime de ses sentiments ?

Je pense qu'il y a un autre domaine où nous pourrions améliorer la simulation d'émotions et ainsi produire des comportements plus riches. Parfois dans un film, on voit une scène dans laquelle une femme est effrayée ou fâchée par ce que l'homme qu'elle aime a fait. Elle le frappe, elle lui jette des insultes au visage et tout d'un coup elle s'effondre en pleurant dans ses bras. Sa colère se dissipe presque immédiatement et elle lui pardonne — tout particulièrement si le public sait que ce n'est pas vraiment de sa faute — car son amour pour lui et son désir d'être consolée prennent le dessus.

L'homme ne fait rien si ce n'est accepter les injures puis de la réconforter quand elle s'effondre. Marion Ravenwood le fait à Indiana Jones dans *Les Aventuriers de l'Arche Perdue* et j'ai vu ça dans plusieurs autres films, mais jamais dans un jeu vidéo. C'est une sorte d'astuce, dramatiquement parlant — on ne voit pas ça souvent dans la vie réelle. Mais le but d'une simulation d'émotions humaines est justement de produire des effets dramatiques dans un médium qui est, pour l'instant, dramatiquement dépourvu de vie. Si je mentionne cet exemple, c'est parce qu'il illustre bien un mécanisme que nous n'utilisons pas souvent.

La plupart des simulations émotionnels utilisent une simple boucle stimulus/calcul/comportement. Quelqu'un dit ou fait quelque chose à un personnage. Ceci influence son état émotionnel. Il agit en fonction de ses sentiments. Cet état émotionnel peut revenir à un état plus neutre avec le temps (j'étais en colère il y a une demi-heure, mais maintenant, je me suis calmé) ou changer en réponse à une stimulation d'un autre personnage. Si ces systèmes ont l'avantage d'être vraiment simples, ils produisent des résultats absurdes : un personnage est furieux puis une seconde plus tard il est joyeux, comme un personnage de cartoon de Warner Bros. C'est le genre de chose que vous obtenez avec un automate fini.

Cette approche ne tient pas compte du fait que le comportement en lui-même modifie les émotions. Le comportement n'est pas juste une façon de rendre nos émotions visible : il affecte aussi la façon dont nous nous sentons. Il agit sur notre état émotionnel. Quand une femme se dispute avec son amoureux dans un film, l'acte de frapper quelqu'un qu'elle aime fait disparaître sa colère et elle commence à chercher du réconfort parce qu'elle a subi cette expérience traumatique et ces sentiments violents. Nous frappons dans un oreiller ou donnons un coup de pied au chien. Ceci ne correspond pas à un comportement qui vise à satisfaire un besoin, comme dans *Les Sims*, parce que frapper un oreiller ne fait pas disparaître la source du problème. Tout ce que cela fait, c'est d'alléger l'esprit (Mark Twain a écrit : « Quand vous êtes en colère, comptez jusqu'à quatre. Si vous êtes vraiment très en colère, jurez. »). Mais je pense que c'est quelque chose que l'on peut incorporer à un jeu sans trop de difficultés.

En fonction du caractère d'un individu, les comportements peuvent dissiper une émotion ou la renforcer. Dans l'exemple que j'ai donné, la colère de la femme disparaît parce qu'elle n'est pas habituellement une personne violente. En fait, le seul fait d'agir violemment la fait se sentir mal. D'un autre côté, dans le cadre de vraies violences conjugales, l'agresseur devient souvent de plus en plus furieux à mesure qu'il frappe la victime. Ils ne s'arrêtent que lorsque la victime a complètement arrêté de réagir. Leur machinerie émotionnelle est sévèrement abîmée. Ce n'est pas un sujet particulièrement joyeux, mais je pense qu'on peut le simuler dans un jeu s'il y a un vrai besoin dramatique pour cela.

Afin de faire réellement évoluer un personnage dans un sens psychologique (et dans cet exemple, une spirale tragique vers l'auto-destruction), nous devons créer une méta-simulation — une simulation capable d'altérer la base émotionnelle avec laquelle le jeu commence. Ceci pourrait prendre la forme de quelque chose qui interrompt la boucle stimulus/calcul/comportement et redéfinit ses variables. Un événement-clef pourrait transformer une rétroaction négative en une rétroaction positive, ou inversement.

La drogue est un exemple classique de spirale tragique. Supposons qu'un personnage accro à la drogue rencontre quelqu'un — un professeur charismatique par exemple — qui soit capable de lui apporter le réconfort émotionnel qu'il cherchait à l'origine dans la drogue. Cette source alternative de réconfort pourrait briser la dépendance à la drogue, en la remplaçant par une nouvelle dépendance vis-à-vis du professeur, ce qui pourrait à son tour introduire de nouveaux problèmes dans l'histoire.

Je vous dirai tout de suite que je ne suis pas certain de la façon de procéder. Tant que quelqu'un ne me finance pas, je peux écrire des articles, mais pas de code. Mais je suis convaincu que ceci peut être réalisé

suffisamment bien pour créer un jeu à la fois divertissant et crédible. Je ne propose pas que nous visions un réalisme psychologique absolu — après tout, la psychanalyse en est encore au stade de l'enfance. Je suggère plutôt que nous essayions d'introduire plus de richesse dans nos simulations, afin de produire chez l'IA de nos personnages des comportements dramatiquement intéressants.

Les personnages de jeu sont pour l'instant beaucoup trop rationnels et leur comportement est assez proche de celui de Vulcains (NDT : le peuple extraterrestre dont fait partie Monsieur Spock dans *Star Trek* et qui ne réagit que sur des bases purement logiques et non émotionnelles) : ils réagissent à des stimuli pour accomplir leur buts. Les personnes réelles ne sont pas comme ça. Elles agissent contre leur propre intérêt, elles font des choses irrationnelles, elles font même des choses complètement stupides. Il y a beaucoup d'opportunités pour l'expérimentation.

Cercles vicieux

Par [Pierre Guyot](#) | le 29 septembre 2011 16:15:52 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mai 2010).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

« Les riches s'enrichissent et les pauvres s'appauvrissent », avait l'habitude de dire mon père lors des derniers tours d'une partie de Monopoly familiale. Vu qu'il était le seul d'entre nous à l'époque à réellement comprendre les principes stratégiques du jeu, c'était en général lui qui s'enrichissait.



Il y a huit ans, j'ai écrit un article intitulé « [Equilibrer les jeux grâce à la rétroaction positive](#) ». Dans celui-ci, j'explique dans quelle mesure la rétroaction positive peut intervenir dans des mécaniques de jeu : c'est la tendance qu'ont les réussites du joueur à s'auto-entretenir et à renforcer le joueur. La rétroaction positive aide le joueur riche à devenir encore plus riche au Monopoly. Mais que faire pour le joueur pauvre qui devient encore plus pauvre ?

Malheureusement, la terminologie rend les choses plutôt confuses, parce qu'une rétroaction positive qui joue contre le joueur n'est pas une « rétroaction négative ». La rétroaction négative est un mécanisme qui a tendance à faire que les différents paramètres restent en place et à les empêcher d'augmenter ou de diminuer.

Un thermostat est un appareil qui fonctionne sur le principe de la rétroaction négative. Si la température devient trop élevée, le thermostat déclenche l'air conditionné. Si la température devient trop basse, elle déclenche le système de chauffage. Le thermostat essaye en permanence de maintenir la température ambiante à un niveau constant.

Marc LeBlanc illustre les concepts de rétroactions positive et négative dans le game design de façon très intéressante en prenant l'exemple d'une combinaison de jeu de course et de jeu de combat — pensez à *Mario Kart*. Si les voitures peuvent se tirer dessus uniquement avec des armes qui tirent vers l'arrière, le premier de la course peut tirer sur ceux qui sont derrière et ceux-ci ne peuvent rien lui faire. Ceci crée une rétroaction positive en faveur du leader et il a tendance à distancer de plus en plus ses concurrents. Si les armes peuvent uniquement tirer vers l'avant, le premier de la course ne peut toucher personne alors que tous les autres peuvent lui tirer dessus. Ceci crée une rétroaction négative : celui qui prend la tête a toujours tendance à en être écarté.

La rétroaction négative est un outil précieux pour le game design : on peut l'utiliser pour freiner l'effet d'une rétroaction positive (il ne faut cependant pas la rendre trop efficace, ce qui risquerait de provoquer une situation d'impasse). Mais cet article a pour objet une rétroaction positive qui travaille « vers le bas » : un mécanisme qui vise à diminuer et non à augmenter une certaine valeur. Pour éviter toute confusion, je parlerai de cercle vicieux.

Au Monopoly, le cercle vicieux commence quand un joueur doit donner une propriété à un autre joueur pour payer une dette. Pour gagner de l'argent, un joueur doit essentiellement attendre qu'un adversaire atterrisse

sur sa propriété et lui paye un loyer. Quand il doit abandonner une propriété, il se voit privé d'une source de revenus, ce qui le rend moins compétitif et moins à même de résister à un mauvais lancer de dés. Pire, la propriété qu'il possédait est maintenant entre les mains de quelqu'un d'autre et, alors qu'elle constituait un endroit sûr pour s'arrêter, est devenue une nouvelle menace de loyer.

Le cercle vicieux n'est pas complètement mauvais en soi, mais il est déprimant (le game designer uruguayen Gonzalo Frasca l'a utilisé à bon escient dans son excellent *Shockwave*). Personne n'aime sentir qu'il n'a aucune chance de gagner et une longue descente vers la défaite n'est pas particulièrement agréable.

Je vais ici émettre quelques suggestions concrètes pour essayer d'éviter des cercles vicieux et de bien les utiliser si jamais vous devez en inclure dans un de vos jeux.

Faire en sorte que les événements négatifs (comme les dégâts) n'entravent pas les capacités du joueur.

L'exemple le plus évident de cercle vicieux dans le jeu vidéo se produit quand dans un jeu de guerre, les unités endommagées combattent moins efficacement — par exemple en tirant moins souvent ou avec moins de précision. Quand une unité se bat moins efficacement, elle ne détruit pas les ennemis aussi vite, elle subit donc plus de dégâts et ainsi de suite.

En pratique, on implémente rarement cette fonctionnalité. La plupart des jeux de guerre suivent cette recommandation : les unités se battent à pleine puissance jusqu'à leur dernier souffle. Les jeux de combat typés arcade fonctionnent de la même manière. Autrement, le premier à joueur à faire une erreur ou à encaisser un mauvais coup perdrait quasiment à coup sûr.

Bien sûr, ce n'est pas réaliste : une personne blessée combat bien moins efficacement qu'une personne en pleine santé. Il y a eu quelques contre-exemples héroïques — des hommes qui ont continué à combattre malgré des blessures terribles — mais ils étaient exceptionnels et c'est pour ça que ces hommes ont reçu des médailles.

De nos jours, les unités militaires essaient de retirer les soldats blessés du front aussi rapidement que possible. La première raison est humanitaire, évidemment, mais il y a une deuxième raison derrière celle-ci : un soldat blessé ralentit toute son unité. Il vaut mieux que tous les combattants d'une unité soient en pleine forme, même avec un homme en moins, qu'une unité rendue inefficace par la présence d'un blessé. Eux aussi essaient d'éviter un cercle vicieux.

Si vous ne permettez pas aux dégâts d'entraver les capacités du joueur, alors les dégâts ne sont qu'un moyen d'évaluer combien une unité est proche de sa destruction. Il n'y a pas de rétroaction. Cette suggestion s'applique que les capacités soient définies en termes de combat, de production ou de performance d'un véhicule.

Implémenter un mécanisme de réparation.

Supposons que vous vouliez vraiment inclure un cercle vicieux dans un souci de réalisme, mais vous ne voulez pas que cela ruine le jeu. Prenons l'exemple d'un *serious game* sur la famine. Les gens qui souffrent de la famine ont peu d'énergie pour s'occuper des champs ou pour élever des animaux, ils récupèrent donc moins de nourriture, ce qui renforce la famine et diminue leur énergie, et ainsi de suite. Finalement, une personne affamée finit par devenir si faible qu'elle ne peut pas se nourrir elle-même même si la nourriture est disponible et donc, meurt. Dans un *serious game*, on peut vouloir implémenter ce cercle vicieux comme un outil d'apprentissage.

On peut ralentir — voire inverser — un cercle vicieux en élaborant un mécanisme de réparation qui vise à restaurer ce que le joueur a perdu. Dans un jeu sur la famine, ceci pourrait être l'effet de l'aide alimentaire : si vous donnez suffisamment de nourriture pour qu'ils se rétablissent, ils peuvent commencer à produire de la nourriture à nouveau (si vous voulez lancer une controverse politique, vous pouvez aussi développer une

« culture de la dépendance » et créer un mécanisme à travers lequel les bénéficiaires de l'aide décident qu'il est dans leur intérêt de profiter d'une nourriture gratuite plutôt que de travailler pour produire la leur).

Les mécanismes de réparation fonctionnent aussi pour les jeux de guerre. Habituellement, dans les simulations de combats navals (ou spatiaux), les dégâts réduisent les capacités de combat — les armes sont détruites, les moteurs endommagés, etc. En incluant un système de contrôle des dégâts qui permet de réparer pendant le combat, on peut ralentir ou inverse le cercle vicieux.

Au Monopoly, la case départ fonctionne comme un petit mécanisme de réparation. Le joueur est sûr de toucher 20 000 francs chaque fois qu'il passe par cette case, indépendamment de ce qui se passe par ailleurs. Néanmoins, vers les derniers tours de jeu, 20 000 francs est une somme négligeable qui ne suffit pas à réellement à empêcher le cercle vicieux.

Définir la victoire en d'autres termes.

Cette suggestion est identique à celle que j'ai pu écrire dans mon précédent article, mais j'ai pense qu'elle vaut la peine d'être répétée. Même dans un jeu de combat, vous n'avez pas besoin de définir la victoire en termes de dégâts infligés au joueur. Si la victoire dépend d'une quantité ou d'un mécanisme qui se trouve en dehors de la boucle de rétroaction, alors cette boucle n'a pas tellement d'importance.

Perdre une pièce aux échecs crée un (petit) cercle vicieux pour le joueur qui la perd — il n'a plus la pièce en question et se trouve donc dans une situation de désavantage. Néanmoins, la victoire aux échecs n'est pas définie en termes de nombres de pièces restantes sur le plateau. Cela peut même être une bonne chose de perdre une pièce à un moment donné, pour des raisons stratégiques.

Dans un jeu de course sérieux (pas *Mario Kart* !), si le joueur endommage sa voiture, il subit une diminution de ses performances. C'est parfaitement approprié. Mais la victoire est définie par le fait de traverser en premier la ligne d'arrivée. Une voiture peut être terriblement endommagée et gagner quand même. C'est une différence basique entre les jeux de course et les jeux de combat — dans le jeux de course, les dommages entravent les performances, mais il n'y a pas de boucle de rétroaction qui affecte la condition de victoire.

Les dégâts du véhicule dans un jeu de course ne créent pas un véritable cercle vicieux vers l'échec de toute façon, parce qu'ils ne s'auto-entretiennent pas. Ils limitent les performances de la voiture, mais n'augmentent pas la probabilité de subir de nouveaux dégâts. C'est un inconvénient fixe qui ne s'aggrave pas tant que le joueur ne percute pas d'autre objet.

Éviter les jeux à somme nulle.

Dans un jeu de course, les dégâts du véhicule constituent un handicap pour le joueur, mais ne confère pas un avantage direct à ses opposant. Ce n'est pas la même chose que de donner une de vos propriétés à un adversaire au Monopoly. Cela ralentit la voiture du joueur, mais n'accélère pas les voitures des autres joueurs.

C'est parce que la course n'est pas un jeu à somme nulle. Dans un jeu à somme nulle, aucune quantité ne disparaît du jeu : la perte d'un joueur est le gain d'un autre. Mais les dommages subis par la voiture font disparaître entièrement quelque chose du jeu — ici l'aérodynamisme — plutôt que de le donner à un autre joueur.

Pour formuler les choses autrement, quand les mécanismes de jeu prennent des ressources précieuses à un joueur, ne les donnez pas à un autre.

(Monopoly n'est pas un jeu à somme nulle dans l'ensemble, mais une fois que toutes les propriétés ont été réparties entre les joueurs, elles sont à somme nulle. Elle ne retournent jamais à la banque et doivent toujours appartenir à l'un ou l'autre des joueurs.)

Augmenter l'influence de la chance (et d'autres facteurs).

J'avais déjà mentionné ceci dans mon précédent article comme une façon de limiter une croissance due à une rétroaction positive et cela marche bien sûr dans ce cas aussi. Le [Backgammon](#) contient un faible cercle vicieux, parce que les pions places sur la barre centrale ne sont disponibles ni pour entraver ni pour attaquer l'autre joueur.

De plus, un joueur dont les pions sont sur la barre n'est pas autorisé à bouger l'un de ses autres pions tant que tous les pions sur la barre ne sont pas retournés sur le plateau. Dans cette situation, il est légèrement plus probable que son adversaire puisse l'attaquer encore et renvoyer de nouveaux pions sur la barre (si le premier jour a été suffisamment imprudent pour rester en position de vulnérabilité).

Néanmoins, le rôle de la chance au backgammon est tellement grand que ce mécanisme ne garantit pas un échec. Avoir des pions sur la barre est clairement un handicap, mais la probabilité que d'autres pions soient renvoyés sur la barre spécifiquement comme conséquence de cette situation n'est pas très élevée. Quelques bons lancers de dé et un jeu astucieux en viennent facilement à bout.

Voici un exemple plus sophistiqué qui sera familier au joueurs de RPG PC : l'aggro.

Pour les non-initiés parmi vous, « aggro » est un terme de gamer pour désigner la tendance des monstres à se comporter de façon agressive. Le terme est généralement précédé (NDT : en anglais du moins) par le nom du stimulus qui déclenche l'attaque : la vue, le son, l'utilisation de sortilèges ou d'objets, etc. Les monstres qui présentent un *blood aggro* attaquent quand ils repèrent un personnage dont les HP sont bas (supposément lorsqu'ils sentent l'odeur de son sang).

La santé basse de l'avatar attire les ennemis, ce qui aboutit à plus de dégâts et ainsi de suite pour former un cercle vicieux. Pour cette raison, un dénommé Pteryx m'a écrit pour me proposer que le *blood aggro*, du moins tel qu'il est implémenté dans *Final Fantasy XI*, soit considéré comme une *Twinkie Denial Condition* (NDT : je vous renvoie aux articles taggés [« Pas de biscuit »](#) pour plus de précisions). Il explique que le *blood aggro* sanctionne une conséquence d'une façon de jouer normale : la perte de HP. La seule façon de l'éviter est de faire en sorte d'être en pleine santé à tout instant, ce qui encourage une approche du jeu très prudente et complètement anti-héroïque.

Néanmoins, je ne pense pas qu'il s'agisse d'une mécanique de jeu intrinsèquement mauvaise : c'est exactement la façon dont se comportent les requins et autres prédateurs opportunistes dans la vie réelle. Cette fonctionnalité est simplement mal implémentée dans *Final Fantasy XI*.

Dans ce jeu, d'après Pteryx, les monstres qui présentent un *blood aggro* apparaissent généralement par paquets, sont immunisés contre les attaques magiques utilisées classiquement contre les monstres et peuvent sentir le sang à très grande distance. Tous ces facteurs auraient pu être équilibrés différemment : réduire la quantité de monstres, se débarrasser de l'immunité magique, réduire la portée de leur odorat ou une combinaison des trois. On peut aussi mettre en place d'autres facteurs prioritaires sur le *blood aggro*. Un monstre intelligent peut être attiré par le sang, mais peut aussi se rendre compte que le personnage blessé est armé jusqu'aux dents et accompagné de cinq compagnons et réaliser que ce n'est pas une si bonne idée. C'est la façon dont les créatures comme les chacals se comportent : l'odeur du sang les poussent à aller voir, mais leur instant de survie les aident à décider s'ils vont effectivement attaquer ou non.

(C'est l'un des problèmes que j'ai avec les monstres dans les RPG : la plupart d'entre eux sont agressifs au point d'être suicidaires et le joueur a besoin d'être ridiculement bien armé pour compenser. Si les monstres étaient moins bêtes, les joueurs pourraient passer plus de temps à explorer et moins de temps à acheter des armes.)

Conclusion

Le Monopoly est un jeu bien conçu, mais l'une de ses faiblesses est ce cercle vicieux. A partir du point où un joueur réalise que sa situation est désespérée, cela lui prend encore une heure avant d'être réellement en faillite... Et c'est un moment plutôt ennuyeux et déprimant à passer.

Les cercles vicieux ne sont pas intrinsèquement mauvais s'ils sont correctement équilibrés. De façon générale, c'est toujours mieux de faire sans si vous le pouvez, mais comme je l'ai montré, un jeu a parfois besoin de les inclure. Maintenant, il s'agit de suggestions, pas de commandements. Vous pouvez trouver adéquat d'en ignorer une, voire toutes. Mais je pense que vous les trouverez utiles.

Modes de difficulté et ajustement dynamique de la difficulté

Par [Pierre Guyot](#) | le 13 octobre 2011 15:48:46 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (mai 2008).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Je viens de finir un livre intitulé *Interactive Storytelling*, par Andrew Glassner. Bien que les premières centaines de pages constituent une introduction utile à la narration et au game design (pour les débutants, du moins), le livre a de sérieux défauts et je ne peux pas vraiment vous le recommander. Mais en cours de route, Glassner fait des digressions sur divers sujets, dont les modes de difficulté réglables. Il est contre. Il pense qu'ils devraient être bannis.

Je me suis arrêté net quand j'ai lu ça. Mon opinion se situe à l'exact opposé : les modes de difficulté réglables devraient être obligatoires dans la plupart des jeux. En fait, je considère l'absence de modes de difficulté réglables comme une TDC (NDT : voir la rubrique « Pas de biscuit ! »), à l'exception de quelques types de jeux pour lesquels ils ne sont pas adéquats. Dans la mesure où j'y suis tellement attaché, j'étais particulièrement intéressé par les arguments de Glassner à leur sujet. J'y ai jeté un œil et je vais essayer d'y répondre, avant de parler de son alternative favorite, l'ajustement dynamique de la difficulté (aussi connu sous le nom de difficulté adaptative).

Voici les arguments de Glassner contre les modes de difficulté réglables.

Le joueur doit décider trop tôt. La plupart des jeux demandent aux joueurs de choisir un niveau de difficulté dès le début, c'est-à-dire à un moment où il ne connaît pas véritablement la difficulté du jeu puisqu'il n'y a pas encore joué.

Ma réponse : Ceci n'est pas vraiment un argument contre le réglage de la difficulté. Un jeu pourrait très bien donner la possibilité au joueur de faire un niveau d'entraînement réglé à la difficulté moyenne, puis permettre au joueur de décider s'il veut que le reste du jeu soit plus facile, plus dur ou du même acabit. Mais même sans cela, de nombreux joueurs peuvent deviner ce qu'il en est en raisonnant sur leur expérience avec des jeux du même type (je sais que je suis nul aux jeux de plates-formes) ou peuvent choisir un niveau de difficulté pour d'autres raisons. Je commence toujours les jeux en mode facile parce que je suis très occupé et que je veux voir un maximum du jeu en un minimum de temps. D'un autre côté, les *hardcore gamers* qui tiennent à montrer leur testostérone commenceront tous les jeux au niveau le plus difficile ; c'est comme ça qu'on peut se la ramener.



Les options sont trop grossières. Que faire si le mode moyen est trop facile mais le mode difficile est trop dur ? Les catégories sont trop largement espacées.

Ma réponse : il n'y a pas de raison pour qu'une difficulté ajustable par le joueur se limite à trois ou quatre options ; il peut s'agir d'une jauge. En fait, n'importe quel jeu bien conçu fait varier son rythme indépendamment du mode de difficulté et présente des passages faciles et des passages difficiles. Même les jeux d'arcade donnent au joueur la possibilité de souffler de temps à autres. Le réglage de difficulté n'est pas censé déterminer la difficulté d'absolument chaque épreuve, mais seulement les maxima et minima de cette difficulté à n'importe quel moment du jeu. J'estime qu'il ne s'agit pas d'une raison suffisante pour les interdire. J'ai néanmoins une exigence : le mode facile est censé être facile (NDT : bordel). Si facile que vous devez pouvoir gagner le jeu en frappant votre clavier avec votre front et en faisant glisser la souris avec votre coude. Si ce n'est pas le cas, le designer a d'une manière ou d'une autre échouer à comprendre ce que veut dire le mot « facile ».

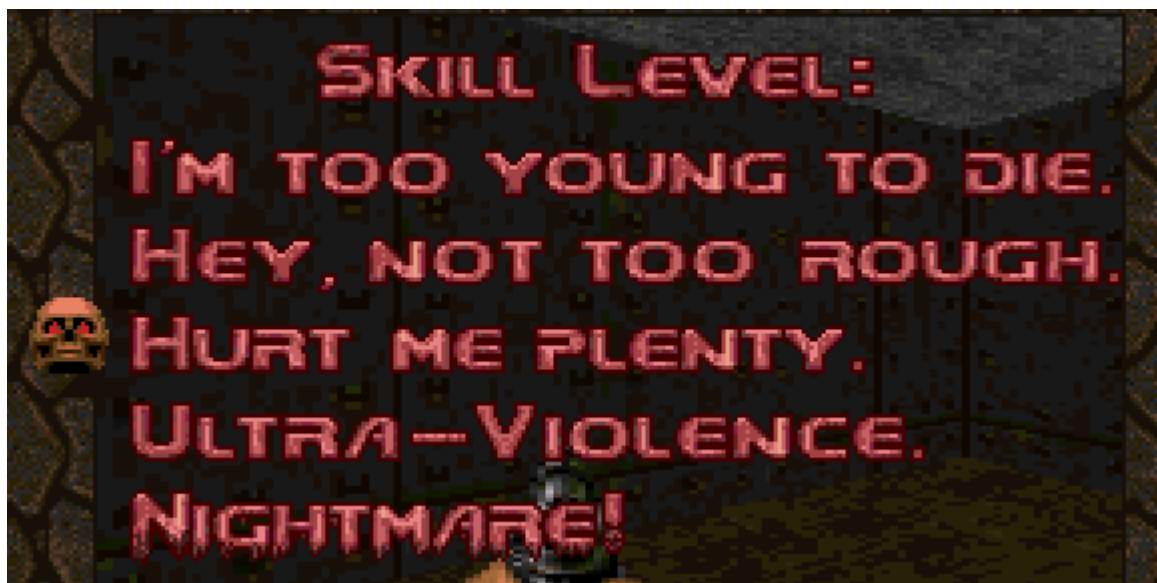
Les modes de difficulté sont trop vastes, ce par quoi Glassner entend que les réglages de difficultés s'appliquent généralement à tous les différents types de challenges dans un jeu, alors qu'un joueur pourrait être un bon dans un domaine, comme le tir, mais mauvais dans un autre, comme la conduite. Quel réglage devrait-il choisir ?

Ma réponse : il s'agit sans aucune doute de l'argument le plus fort de Glassner et il mérite que je m'y attarde. Tout d'abord, son objection ne s'applique pas nécessairement à tous les genres de jeux. De nombreux jeux se concentrent sur un petit nombre de challenges différents — *Tetris*, par exemple. D'un autre côté, dans un jeu complexe comme *Civilization*, le réglage de la difficulté ne s'applique pas à des challenges en particuliers mais à l'ensemble du jeu d'une façon plus générale. Vos adversaires sont juste plus intelligents, plus méchants, plus riches et plus rapides que vous. Dans ce genre de jeux, le fait que les modes de difficultés soient généraux ne pose pas de problème.

Ceci nous laisse les jeux avec des challenges hétérogènes et nettement séparés. Les jeux de plate-formes et les TPS nous fournissent les exemples les plus évidents. Je suis bon en tir, mais quand il s'agit de mouvements acrobatiques, du type sauter au dessus d'un précipice ou se balancer sur des cordes, je suis désespérément nul. Alors comment faire ? Eh bien, tout d'abord, nous pouvons mettre en place des réglages différents pour des types de difficulté différents. *Weird Worlds : Return to infinite Space* comprend deux réglages pour la difficulté, l'un pour le mode stratégique (voyager à travers la galaxie) et l'autre pour le mode tactique (les combats de vaisseaux). Mais même si je ne veux pas me donner toute cette peine, je crois qu'un joueur choisit un niveau de difficulté principalement en pensant au niveau de difficulté maximum qu'il rencontrera sur l'ensemble du jeu. Quand je choisis le mode facile, j'ai besoin de savoir que tout sera facile dans le jeu, que rien ne sera plus dur qu'un certain maximum. De la même façon, le joueurs ne s'attendent

pas à ce qu'à réglage fixé, tous les challenges d'un jeu aient la même difficulté. Un mode cauchemar ne garantit pas qu'absolument que tout passage du jeu sera cauchemardesque et en fait, les joueurs de shoot protesteraient avec véhémence s'ils se rendaient compte qu'en mode cauchemar, ils devaient se traîner sur une canne et prendre dix secondes pour recharger.

Pour faire court, Glassner a raison lorsqu'il dit qu'on ne peut pas garantir qu'à réglage donné, tous les challenges aient la même difficulté pour tous les joueurs. Mais je ne pense pas non plus que ce soit ce que les joueurs attendent.



Les modes de difficulté sont définitifs, c'est-à-dire qu'un réglage de difficulté ne s'adapte pas à la progression du joueur, tout particulièrement s'il n'est pas autorisé à le changer. La courbe de difficulté, quel que soit le réglage, peut se révéler trop plate ou trop raide pour le joueur.

Ma réponse : c'est indéniablement l'une des faiblesses de tous les jeux qui n'implémentent pas un ajustement dynamique de la difficulté. Les niveaux de difficulté réglables ne créent pas ce problème, ils aident à l'alléger un peu. Un mode facile propose une augmentation très lente de la difficulté, alors qu'un mode difficile propose une augmentation brutale de celle-ci (parfois décrit par le terme « steep learning curve »). Les joueurs le savent et choisissent en conséquence. Il est vrai que la difficulté ne s'ajuste pas d'elle-même, mais c'est toujours mieux que rien.

Ils sont trop généraux, ce qui est une reformulation de l'argument sur leur caractère trop « vaste ». Glassner fait remarquer que de nombreux jeux introduisent au fur et à mesure de nouveaux types d'action, ce qui nécessite de nouvelles compétences. Le joueur peut être meilleur pour certaines que pour d'autres. Dans la mesure où le joueur ne sait pas de quelles compétences on va lui demander de faire preuve, il pourrait regretter son choix lors de l'introduction d'un nouveau type d'action.

Ma réponse : encore une fois, j'ai l'impression que la plupart des joueurs veulent une idée globale de la difficulté maximale qu'ils vont rencontrer, et non pas une garantie que tous les challenges du jeu seront d'égale difficulté. Et franchement, c'est pour ça que nous avons des tests et des guides de jeux. Si vous voulez savoir ce à quoi vous devez vous attendre dans la suite du jeu (« Attention, dans le niveau 5, vous aurez à piloter un hélicoptère »), il y a moyen de se renseigner.

Pour faire court, bien que ces remarques méritent notre attention — et que leurs effets devraient être améliorés dans la mesure du possible — je pense qu'exiger le « bannissement » des niveaux de difficulté réglables de tous les jeux, c'est jeter le bébé avec l'eau du bain (NDT : sic). Et plus important, les objections de Glassner font l'impasse sur la plus importante de toutes les règles de game design : faire preuve d'empathie vis-à-vis du joueur, c'est-à-dire lui donner ce qu'il veut. Les joueurs veulent des niveaux de

difficulté réglables et enlever ces derniers sur la base d'un argumentaire purement théorique ne constitue pas une bonne façon de se mettre au service de son public.

Glassner conclut cette discussion en disant : « Les jeux ne devraient pas demander aux joueurs de sélectionner un niveau de difficulté. Les jeux devraient s'adapter au fur et à mesure afin d'offrir au joueur un niveau de challenge cohérent basé sur ses capacités dans différents domaines. » En d'autres termes, ils devraient présenter un ajustement dynamique de la difficulté, ce que j'appellerai ADD à partir de maintenant (parce que l'ADD est aussi appelée difficulté adaptative, ce qu'on confond parfois avec intelligence artificielle adaptative. L'intelligence artificielle adaptative est une façon de mettre en place un ADD, mais c'est un sujet bien trop compliqué pour cet article).

L'ADD existe depuis des décennies. Le plus vieil exemple que je puisse trouver est le jeu *Astrosmash* d'Intellivision, un clone du classique *Asteroids* d'Atari. *Astrosmash* devenait de plus en plus dur dans un style de jeu typiquement arcade, mais il prenait aussi en compte les moments où vous manquiez de vies et redevenait plus facile pour un moment. Même les débutants pouvaient y jouer pour un bon bout de temps, ce qui l'a rendu très populaire. Dans la mesure où ce n'était pas un jeu payant, les designers n'avaient aucune raison d'essayer de faire perdre le joueur rapidement.

J'aime le principe d'un ajustement dynamique de la difficulté, parce que cela me parle en tant que programmeur. Ce serait vraiment intéressant de coder un jeu suffisamment intelligent pour ajuster les challenges qu'il propose en fonction des capacités du joueur, de façon à garantir qu'il passe un bon moment. Mais malgré la confiance que Glassner place dans l'ADD, je dois aussi reconnaître qu'il n'est pas nécessaire de l'implémenter.

Jetons un œil à certains problèmes de l'ADD :

Certains joueurs détestent ça. C'est la première raison de réfléchir à deux fois avant d'implémenter un ADD, tout particulièrement s'il ne peut pas être désactivé. De nombreux joueurs se sentent infantilisés quand ils découvrent que le jeu devient plus facile quand ils ont du mal et ils ne veulent pas que le niveau de difficulté change du tout. Pour eux, c'est dans la difficulté que réside le plaisir, même si cela signifie qu'il leur faudra mourir 500 fois avant de parvenir au succès.

Les joueurs peuvent apprendre à exploiter l'ADD en faisant semblant d'être moins qu'ils ne sont réellement. C'est comme si vous faisiez croire à votre adversaire que vous êtes un novice et que dès que son argent est sur la table vous le massacriez. C'est l'un des arguments contre l'ADD le plus évoqué parmi les joueurs et les designers professionnels. Personnellement, je n'ai pas de gros problème avec ceci parce que je ne tiens pas à rendre les choses difficiles pour les joueurs qui jouent à mes jeux. Mais ils ont raison dans la mesure où ceci transforme le jeu et peut le reprendre gravement déséquilibré en mode multijoueurs.

L'ADD ne fonctionne pas pour tous les types de challenges. Si les challenges dans un jeu sont basés sur des grandeurs numériques — points de vie, argent, vitesse, etc. — alors il est plutôt facile d'ajuster dynamiquement ces valeurs pour changer la difficulté. Mais qu'en est-il d'entités symboliques ? La plupart des énigmes ont une difficulté fixée au départ et il n'est pas facile de la changer en cours de route. Si je dois devenir le bon ordre pour activer les interrupteurs et ouvrir la porte de la salle du trésor, cela prendra une quantité faramineuse de code supplémentaire pour fournir différentes solutions à cette énigme — et à toutes les autres dans le jeu — en fonction du niveau du joueur. Les énigmes exigeant une forme de pensée latérale, comme le pneu-en-caoutchouc-plus-cactus-égale-lance-pierres d'un des *Monkey Island*, ne peuvent avoir qu'une solution : il n'y a pas de façon de rendre l'énigme plus simple à moins de la supprimer purement et simplement. Cette objection s'applique de même aux niveaux de difficulté réglables, ce pourquoi les jeux d'aventure les implémentent rarement. Ceci montre néanmoins que l'ADD n'est pas une solution universelle au problème de l'équilibrage de la difficulté.

L'ADD peut créer des absurdités. Dans certains jeux de course, si vous avez un accident, le jeu ralentit les autres joueurs de façon à vous donner une chance de les rattraper. C'est là qu'intervient le fameux effet « élastique » : si vous êtes trop en avance, vos adversaires finissent toujours par vous rattraper et si vous êtes

trop en retard, ils ralentissent toujours. Cela donne l'impression que votre voiture et celles de vos adversaires sont attachées par des bandes de caoutchouc. C'est absurde et les joueurs le savent. Ce genre de chose convient pour un jeu léger comme *Beetle Adventure Racing*, mais c'est hors de question pour une simulation de course sérieuse. Un mécanisme d'ADD doit intervenir d'une façon logiquement et émotionnellement cohérente avec l'univers du jeu.

L'ADD rend caduc un bon level design. Un hypothétique système d'ADD qui fonctionnerait parfaitement et maintiendrait tous les challenges d'un jeu au même niveau ruinerait le rythme de ce jeu. Ce serait comme écouter une symphonie de Beethoven où toutes les notes seraient jouées exactement au même volume ou vous promenez dans un musée avec des lunettes de couleur. Un niveau bien conçu, avec ses variations émotionnelles, est une œuvre d'art en soi et mérite d'être apprécié comme tel.

Je ne suis pas opposé à l'utilisation de l'ADD en toutes circonstances : comme avec presque tout ce qui a trait au game design, l'ADD a ses forces et ses faibles. Je pense que c'est une idée brillante pour des jeux simples, abstraits et *casual* comme *Astrosplash*, mais plus les différents challenges d'un même jeu sont hétérogènes, plus il est difficile d'élaborer un système d'ADD qui les prennent tous en charge.

Tout système d'ADD nécessite un mécanisme d'évaluation des performances et un mécanisme d'ajustement pour chaque type de challenge sur lequel il opérera. Le mécanisme d'évaluation des performances détermine à quel point un joueur est bon pour un certain type de challenge et le mécanisme d'ajustement rend ce challenge plus ou moins difficile, dans la plupart des cas en modifiant les paramètres qui définissent la difficulté du challenge (comme la force d'un ennemi). Dans *Astrosplash*, l'évaluation de la performance était très simple : le jeu regardait le nombre de vies restantes. Vous pouvez facilement faire la même chose avec des points de vie, de l'argent ou toute autre ressource nécessaire à la survie du joueur. Robin Hunicke a fait des recherches pratiques sur le sujet en utilisant un mod d'*Half-Life* : vous pouvez aller voir son article « [The Case for Dynamic Difficulty Adjustment in Games](#). » (il vous faudra un compte ACM pour télécharger le .pdf).

Si vous voulez absolument utiliser l'ADD, prenez en considération les suggestions suivantes :

Rendez le jeu plus dur en renforçant les adversaires et non en affaiblissant le joueur. Les joueurs ont à juste titre le sentiment que les pouvoirs dont ils disposent dans un jeu, ils les ont mérités. N'enlevez jamais de façon arbitraire un avantage au joueur : s'il perd quelque chose qu'il avait gagné, il doit y avoir une raison pour cela. Il est acceptable de rendre l'intelligence artificielle plus chanceuse, mais pas de rendre le joueur plus malchanceux. S'il s'en sort bien et qu'il se rend soudain compte que son fusil sniper ne tire plus droit, il va vous haïr.

Rendez l'ADD optionnel, comme dans *Lego Star Wars II*. De cette façon, les joueurs qui n'aiment pas l'ADD pourront le désactiver. *Lego Star Wars II proposait à la fois un ADD et des niveaux de difficulté réglables, mais tout le monde ne peut pas se permettre d'y consacrer autant de travail.



Soyez subtils. Je n'insisterai jamais assez là-dessus. Les monstres qui étaient faciles à battre au niveau 1 ne devrait pas soudainement devenir invincibles au niveau deux, ou inversement. Des changements trop brusques brisent l'immersion et créer des pics ou des puits dans la courbe de difficulté telle qu'elle est perçue par le joueur. Si c'est possible, ajustez la difficulté à travers de petits changements fréquents plutôt qu'à travers une seul gros changement, de façon à ce que le joueur ne se rende compte de rien.

Ne montrez pas au joueur le fonctionnement de l'ADD. Ne laissez pas les joueurs comprendre quel type de mécanisme d'évaluation vous utilisez, ni ce que le jeu fera pour rendre les jeux plus faciles. S'ils ne savent pas ça, les joueurs ne pourront pas exploiter le système si facilement.

L'ADD est difficile à implémenter, rend l'équilibrage d'un jeu complexe et il se peut même que votre joueur n'en veuille pas du tout. Réfléchissez soigneusement si vous voulez vous lancer dedans.

Jeu et identité

Par [Thomas Chassin](#) | le 18 octobre 2011 23:00:00 | Catégories : [Game Design](#)

Dans le dur monde du jeu vidéo, tous les titres ne sont pas égaux. Certains deviennent des succès planétaires et d'autre finissent rapidement aux oubliettes. Une différence entre ces deux catégories, c'est ce que j'appellerai l'identité.



Gameplay.

Commençons par le cœur du jeu : le gameplay. Comme écrit dans l'article [Comment aller à l'essentiel et donner aux joueurs ce qu'ils veulent](#), l'ensemble des mécaniques d'un jeu tourne autour d'un « noyau » central et surtout, unique. Dans la mesure où la majeure partie du temps passée sur un jeu va se résumer à un petit nombre d'actions répétées, ce noyau va définir très fortement l'identité du jeu.

Prenons l'exemple d'un monument du jeu vidéo au succès planétaire : *Pac Man*. Le jeu se résume à ramasser des points dans un labyrinthe, un noyau central sur lequel viennent se greffer les fantômes, les pilules, etc. Avec la diversification du jeu vidéo liée à l'évolution des supports de jeu, les jeux se définissent en fonction de l'action centrale (ce qu'on pourrait appeler le « genre », pour faire simple) : dans un shmup, on tire, dans un jeu de plate-formes, on saute, dans un *Pac-Man*, on gobe, dans un *Bomberman*, on fait tout péter.

Dans un deuxième temps, un jeu se définit par la façon dont cette mécanique centrale est déclinée : rien à voir entre un *Kingdom Hearts*, qui vise une action frénétique avec une flopée d'ennemis et des combos acrobatiques, et un *Zelda*, où l'exploration et la résolution d'énigmes ont une grande place, alors que par ailleurs, tous deux sont des A-RPG.

Direction artistique.

L'identité visuelle et musicale d'un jeu est très importante, si ce n'est capitale. En effet, toutes les interactions avec le joueur passent par deux sens : la vue et l'ouïe (on peut compter en plus le toucher depuis l'apparition de la fonction vibration, mais cela reste anecdotique). Peu importent la profondeur de l'univers ou le soin porté aux personnages si les choix visuels (character design, choix des couleurs, etc.) ou musicaux (thèmes, sons, etc.) ne permettent pas au joueur d'en garder le souvenir. Indépendamment de la technique du jeu (graphismes next-gen, orchestration, etc.), la musique et le graphisme permettent de créer des points de repères pour le joueur. Un joueur ne pourra pas tout retenir d'un jeu et ces points de repères lui permettront de garder en mémoire les éléments essentiels du gameplay et du scénario. Quel que soit le jeu Mario auquel vous jouez, il aura toujours une moustache et un casquette rouge, et vous retrouverez le célèbre thème musical.

Par exemple, dans un *Zelda*, l'acquisition d'un rubis ou la résolution d'une énigme s'accompagnent d'un bruit très caractéristique dont les joueurs se souviennent longtemps après avoir fini le jeu. De nombreux personnages de la saga *Final Fantasy* restent en mémoire grâce à leur thème musical : Kefka, Aérís, Edea, etc.

Dans un *Mario*, les joueurs associent immédiatement des éléments de gameplay à des repères visuels, des symboles (un champignon pour une vie, un cube avec un point d'interrogation contre lequel il faut sauter, une étoile qui rend invincible, etc.). L'identité visuelle et musicale de votre jeu tient en grande partie à la présence de tels éléments, récurrents et facilement mémorisables, comme les ennemis (goombas, space invaders, etc.), les symboles (triforce, champignon « 1-up », etc.) ou encore l'utilisation récurrente d'un même thème musical avec différentes variantes (le chant des Priants dans *Final Fantasy 10*). Il faut que ces éléments soient à la fois simples et marquants.

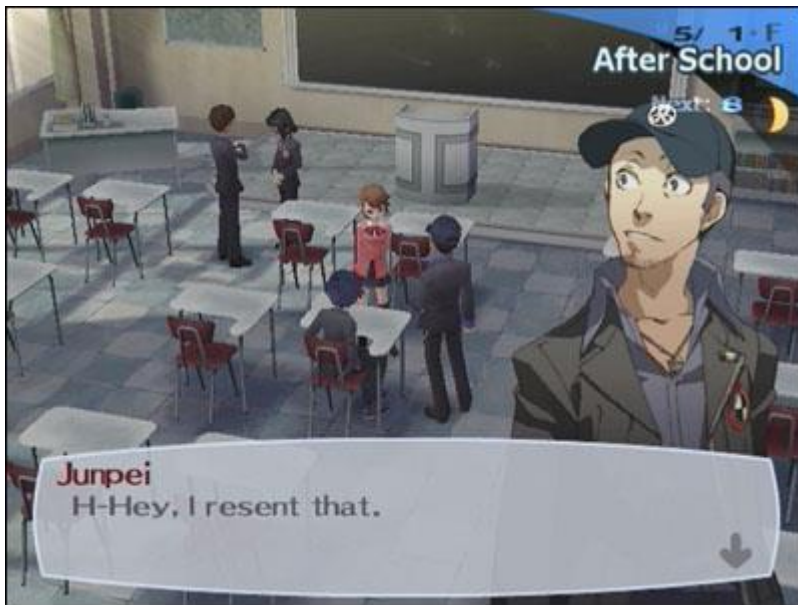


Le choix d'un style visuel est essentiel pour que votre jeu soit reconnaissable : le pixel art haute résolution d'un *Owlboy* saute aux yeux immédiatement et les personnages sont très reconnaissables à leurs teintes brunes et à leurs nez rouges.

Adopter un style visuel ou mettre en place une ambiance musicale ne fonctionne que si vous avez un réel souci d'homogénéité mais aussi si votre style sert votre gameplay. Si vous concevez un jeu d'un type donné, gardez à l'esprit que votre style graphique et votre environnement sonore y sont subordonnés. Si vous faites un beat'em all dans lequel le joueur éclate des zombies à grands coups de démonte-pneu, il serait incohérent d'utiliser des tons pastels chauds et de passer une musique douce (bien sur, cela n'a aucune valeur dans le cas d'un jeu comique ou parodique, puisque l'incohérence peut être source d'humour). De plus, il faut maintenir une cohérence entre ambiance sonore et ambiance visuelle, sans quoi la magie n'opèrera pas.

Personnages.

Les personnages jouent un rôle central dans la façon dont un jeu marque les esprits. Très souvent, on peut résumer l'image d'un jeu simplement à son personnage principal : l'image de Sonic ou de Solid Snake suffit à immédiatement rappeler tous les jeux auxquels ils sont associés. Si un joueur doit s'incarner dans un personnage, il est capital qu'il puisse s'y attacher, même si ce personnage n'est pas sympathique. Si la mort brutale de votre personnage ne provoque qu'un rire méprisant chez le joueur, il y a quelque chose qui pose problème.



Le personnage de jeu existe principalement de trois façons : à travers son design, à travers ses dialogues et à travers ses actions. Il est donc important de rendre, quitte à simplifier, le look d'un personnage mémorable. La structure géométrique de l'armure de Samus dans *Metroid* est très épurée, mais immédiatement reconnaissable. De la même façon, la voix et les expressions d'une GLaDOS — notamment sa façon de passer d'une neutralité faussement enjouée à un discours passif-agressif plus qu'inquiétant — ont fixé le personnage dans la mémoire des joueurs.

La communication

Un élément important de l'identité d'un jeu est la communication qui se joue autour du jeu, parce qu'elle va déterminer comment celui-ci restera dans les mémoires. La communication, même si elle ne fait pas à proprement parler partie du jeu, prolonge son univers d'une certaine façon, à travers la *fanbase*. Un gros éditeur de jeux AAA ne communique pas de la même façon qu'une petite équipe d'indépendants et la « mythologie » qui se construit autour de leurs jeux n'est pas la même.

Square Enix

Prenons tout d'abord l'exemple de [Square Enix](#), acteur non-négligeable de la scène vidéoludique. Quand on parle de communication chez Square-Enix, on voit généralement les choses en grand : panneaux publicitaires dans le métro ou aux arrêts de bus, spots télévisés, sites dédiés à l'interface léchée, packaging complexes avec artbooks, etc.

En faisant le choix d'investir du temps, de l'argent et du travail pour sortir les bandes son de ses jeux, les faire arranger (pour piano notamment), orchestrer et diffuser, Square-Enix a renforcé l'identité musicale de ses jeux : la série des *Final Fantasy* est reconnue pour ses musiques et un jeu comme *Nier* a été acclamé pour sa bande son audacieuse.

Graphiquement, Square-Enix a fait des efforts sur sa série phare pour homogénéiser les logos : la société a même demandé à Yoshitaka Amano, character designer attitré des anciens épisodes, de revenir sur les titres des premiers opus pour les accorder avec les autres et continue à l'artiste de signer les logos des nouveaux *Final Fantasy*, bien qu'il ne réalise plus leur design.



1987



2012

Pour la sortie de *Deus Ex : Human Revolution*, le service marketing a conçu un faux site pour l'entreprise fictive Sarif Industries, qui a ensuite été piraté par un groupuscule rebelle, lui aussi fictif. Le jeu entre réalité et fiction permet de prolonger l'univers du jeu, ce qui lui a permis d'installer dans les esprits son ambiance particulière.

Tous ces choix de marketing font que, d'une certaine manière, l'univers du jeu s'étend hors du cadre du jeu à proprement parler et que l'ambiance, l'image, l'identité de ces titres restent très fortement ancrées dans la mémoire des joueurs.

D-Pad Studio

A l'opposé de la gamme, on trouve par exemple [D-Pad Studio](#), un petit studio indépendant travaillant sur *Owlboy*, un projet de jeu de plateforme. Malgré le manque de moyens financiers, l'équipe de développement a su jouer sur la jovialité et la proximité avec les joueurs pour créer une forte communauté autour de *Owlboy*.

La communication autour d'*Owlboy* est caractéristique des jeux *indie* : l'utilisation centrale d'un blog, la présentation du développement du jeu et des problématiques qui y sont liées, un jeu de réponse entre artworks et *fan arts*, la présence très régulière à des conventions ou à des concours amateurs. Les membres de l'équipe sont par exemple représentés par de petits portraits dans le style du jeu et donne une image sympathique à la fois du jeu et de ses développeurs, comme si l'ambiance du jeu « contaminait » la communauté qui se crée autour.

Valve

Entre les deux, l'exemple de [Valve](#) est particulièrement surprenant : les développeurs de *Half-Life* ont choisi d'utiliser leur arsenal de communication pour établir une relation de connivence avec les joueurs basée sur l'humour. En jouant sur l'absurde et des répliques extrêmement spirituelles, les développeurs ont réussi à créer toute une mythologie autour du jeu, au point de créer de véritables mèmes (on ne présente plus le fameux « The cake is a lie »).

L'exemple le plus impressionnant de cette communication décalée est le Potato Sack, une offre promotionnelle disponible sur Steam regroupant 13 jeux indépendants, proposé peu avant la sortie de *Portal 2*. Des modifications subtiles basées sur le thème de la pomme de terre dans chacun des jeux menaient à un jeu de piste labyrinthique, censé déboucher sur des informations cachés sur le futur *Portal 2*. Oui, oui, vous avez bien lu.

En choisissant de ne pas considérer les joueurs exclusivement comme des consommateurs mais aussi comme des personnes avec qui l'ont peut échanger sur le jeu, Valve a réussi à étendre l'univers de *Portal* au delà du

jeu à strictement parler : ce n'est plus seulement un jeu, c'est un ensemble de références humoristiques, voire culturelles.

Conclusion

Un jeu qui a une forte identité est un jeu dans lequel rien n'est laissé au hasard et où tout ce avec quoi le joueur peut interagir à été pensé avec un réel souci de cohérence (gameplay, graphismes, écriture, musique). C'est également un jeu qui a su être promu intelligemment, non pas comme produit commercial, mais comme expérience vidéo-ludique.

Un jeu, aussi bien dans son univers qu'à l'extérieur, doit être cohérent avec lui même. Ralph Eggleston, *production designer* chez Pixar disait à propos des univers dans lesquels prennent place les films Pixar :

You don't have to build it all, you just have to know it all.

Il est temps de faire revenir les jeux d'aventure

Par [Lyonsbanner](#) | le 6 novembre 2011 19:00:00 | Catégories : [Ernest Adams](#), [Game Design](#), [Scénario](#)

Cet article est une traduction de [Lyonsbanner](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (novembre 1999).

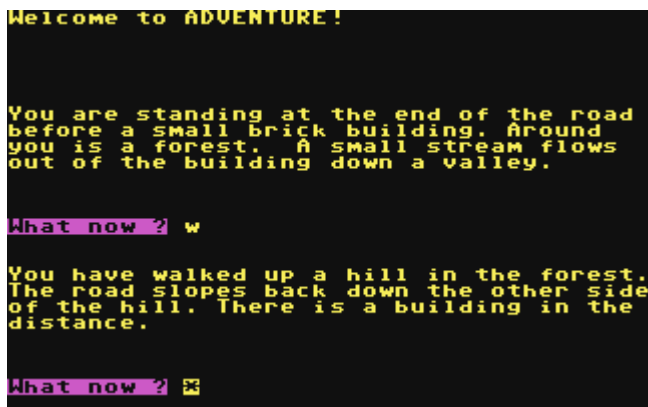
L'auteur original n'est en aucun cas responsable de la précision de cette traduction.



S'il y a bien une chose dont on n'entend plus beaucoup parler, c'est la « narration interactive ». À la *Game Developers' Conference*, il y avait beaucoup de tables rondes consacrées à ce sujet, et elles continuaient autour d'un verre au bar. C'était l'époque où les jeux d'aventure étaient rois. L'époque où *LucasArts* et *Sierra On-line* étaient au sommet de leur forme, les jeux d'aventure étaient les plus beaux et dominaient le reste du marché. Ils étaient amusants, effrayants, mystérieux et fascinants. Les jeux d'aventure présentaient des défis et exploitaient des domaines que les autres genres ne touchaient pas.

À cette époque, au début des années 90, les jeux de guerre étaient moribonds – c'était de petits jeux au tour par tour sur des grilles hexagonales qui se vendaient entre 5000 et 10000 exemplaires. Les jeux à la première personne étaient presque inexistants ; nous n'avions pas la technologie requise. Du côté de l'action, les side-scrollers faisaient la loi. *Flight Simulators* était un monde rudimentaire et très cubique. Les jeux d'aventure étaient très en avance sur les autres genres par leur richesse, leur profondeur, leurs personnages et leurs qualités artistiques, et cela se voyait dans les budgets du développement et du marketing. Beaucoup de gens travaillaient dessus et d'autres rêvaient d'en faire autant.

Les jeux d'aventure ont depuis sombré dans l'oubli, mis de côté pour être remplacé par des jeux de tirs en 3D et des jeux de stratégie en temps réel. De nos jours, le terme de "jeu d'aventure" lui-même est un peu devenu un abus de langage. C'est une contraction de « jeu à la Adventure », terme qui fait référence au tout premier jeu d'aventure, parfois appelé *Colossal Cave*, mais plus souvent connu sous le nom de *Adventure*. Pour s'offrir une vraie peur bleu ou avoir le cœur au bord des lèvres comme on l'attend dans une aventure, il est difficile de faire plus efficace que les jeux modernes en 3D comme *Half-Life* ou *Thief : The Dark Project*, surtout quand on y joue seul tard la nuit. Le terme « jeu d'aventure » désigne aujourd'hui un jeu avec des personnages, des énigmes et une intrigue qui se déroule, et généralement sans moments de stress intense.



Les cartes graphiques accélératrices 3D ont beaucoup influencé le déclin des jeux d'aventure. Les moteurs 3D permettent une grande liberté de mouvement, des perspectives illimitées, et par dessus tout, de la vitesse. L'accélération 3D est l'une des meilleures choses qui ne soit jamais arrivé à l'industrie du jeu vidéo, mais dans la hâte de faire des jeux toujours plus rapides, nous avons sacrifié la richesse visuelle dans nos décors. A quoi ça sert d'avoir un environnement de toute beauté, si c'est pour courir dans tous les sens en ne faisant attention qu'aux tirs ennemis ?

L'autre élément qui a éjecté les traditionnels jeux d'aventure hors des feux de la rampe, ce sont les jeux en ligne. Quand je suis entré pour la première fois dans l'industrie du jeu vidéo, la plupart des développeurs ignoraient l'existence d'Internet, et les jeux en ligne étaient une niche minuscule occupés par des sociétés telles que *CompuServe* et *GENie*. Les éditeurs ne prenaient même pas la peine de s'y intéresser, et encore moins de le développer. Aujourd'hui, les jeux en ligne sont à la mode, et très peu de jeux sont produits sans mode multijoueur. Certains jeux, comme *Quake* et ses successeurs, sont principalement conçus pour le mode multijoueur et le mode solo est un peu la cinquième roue du carrosse.

Il y a une vieille blague qui dit qu'il y a deux catégories de personnes dans le monde, ceux qui divisent les personnes en deux catégories, et ceux qui ne le font pas. Dans l'ensemble, je fais parti de la deuxième – les généralisations sont responsables de nombreux problèmes dans ce monde. Cependant, je crois qu'il y a deux sortes de joueur, ceux qui aiment jouer à des jeux seuls, et ceux qui aiment jouer contre d'autres personnes.

Les jeux multijoueur, malgré leur popularité, ne sont pas fait pour tout le monde. Pour une raison simple, il leur faut (tada !) d'autres personnes, et ça veut dire que vous devez trouver le temps de pouvoir jouer ensemble. Si vous n'avez pas beaucoup de temps libre et préférez jouer par petites séances, vous pouvez vous permettre de quitter un jeu sans frustrer un autre joueur quelconque. Vous pourriez évidemment jouer à des jeux rapides comme le poker et le black-jack en ligne, mais si vous préférez jouer à des jeux longs sur de courtes périodes, vous avez besoin d'un grand jeu solo.

Une autre raison pour laquelle certaines personnes préfèrent jouer à des jeux seuls est une question de tempérament. Je joue à des jeux pour m'amuser, et je voudrais que les gens avec qui je joue s'amuse aussi. Je ne suis pas là pour vous fendre le cœur ; je suis là pour passer un bon moment ensemble. Je suis sûr qu'enfant nous avons tous déjà joué au moins une fois avec un gamin qui se délecte de ses victoires, qui boude de ses défaites et se comporte comme un crétin de manière générale. Beaucoup trop d'univers en ligne sont remplis de ce genre de personne : des adolescents psychotiques dont le seul plaisir dans la vie semble être de railler les étrangers. Je respecte les bonnes manières, et j'ai reçu assez de railleries dans les cours de récréations pour tout une vie, merci bien.

Mais la raison la plus importante pour jouer seul est le sentiment d'immersion. Beaucoup de gens sont attirés par les jeux parce qu'ils aiment être dans un monde fantastique ; ils aiment le côté exploration et découverte, autant du décor que de l'intrigue. Partager ce monde avec des gens réels tend à détruire ce doux rêve. C'est une chose de prétendre que vous êtes le puissant chevalier arpentant seul la forêt ; c'est autre chose si votre ami Joe est juste à côté de vous. Joe est un produit du 20ème siècle, et contrairement aux personnages fictifs dans le jeu, il ne parle pas avec des répliques à la Geoffrey Chaucer (NDT : un poète médiéval anglais) que les fantômes médiévaux semblent exiger. ("Je vous salue, preux chevalier ! Que vous amène dans ces bois

si périlleux à la tombée de la nuit ? J'ai ouïe dire des rumeurs d'un dragon en ce lieu.") Quand Joe parle, c'est comme si c'était Joe – qui est bien dans la vie réelle, mais dont le phrasé moderne sonne faux sur les terres mystiques d'Albion. Et partager un monde avec des étrangers est encore pire. Si je suis à la recherche de gloire, de fortune et d'amour pour ma bien-aimée, la dernière personne que je souhaite avoir comme compagnon est le sieur KewL DooD (NDT : « MeC KooL »).

Ce qui m'intéresse le plus dans les jeux ce sont les personnages, les lieux, les relations et les événements qui se déroulent, et avoir une chance de pouvoir interagir avec eux. J'ai joué jusqu'à la fin de *StarCraft* (en trichant parfois) non pas parce que j'ai été captivé par le jeu de guerre en lui-même, mais parce que je voulais savoir ce qui allait arriver à Jim Raynor et à Sarah Kerrigan. Je n'ai pas l'intention de manquer de respect aux mécaniques du jeu – je les ai beaucoup appréciées – mais ce qui m'a poussé à vouloir jouer à la trentaine de missions, c'était l'histoire.

Les jeux d'aventure sont la quintessence de l'expérience en solo. Beaucoup de jeux en mode solo sont en réalité des jeux multijoueur dans lesquels la machine est un substitut d'adversaire humain, et maintenant il est possible de jouer contre de vrais adversaires, c'est comme ça que fonctionne l'industrie du jeu vidéo. Mais les jeux d'aventure ne sont pas orientés autour de la compétition ; en fait, ce ne sont pas du tout des « jeux ». Il n'y a pas d'adversaires à proprement parler, ni de conditions de victoire si ce n'est de résoudre toute les énigmes et d'atteindre la fin de l'histoire. Les jeux d'aventure font la part belle aux actions d'un individu dans un monde complexe, le plus souvent un monde où on se sert plus de sa tête que de ses armes. Si vous jouez avec quelqu'un d'autre, c'est quelqu'un assis dans la même pièce que vous et qui vous aide à réfléchir – les jeux d'aventures récompensent la pensée latérale.

Le genre a tout de même ses problèmes, le pire d'entre eux étant les coûts de développement. *Infocom* et *LucasArts* sont devenus plutôt doués pour développer des moteurs de jeux réutilisables, respectivement la *Z-Machine* et *SCUMM*, mais ce qui coûte vraiment cher, ce sont les graphismes et l'audio. Les histoires doivent avoir du contenu, et les histoires interactives ont besoin de trois à dix fois plus de contenu que les histoires linéaires. Des éditeurs ont investi des sommes colossales dans le développement de leurs jeux d'aventure (*Phantasmagoria* est sorti sur sept CD-ROM) et ils n'ont tout simplement pas calculé les recettes qu'il aurait fallu pour compenser leurs dépenses. Quand vous pouvez gagner au moins autant d'argent en développant un Quake-like pour une bouchée de pain, pourquoi s'embêter à développer un jeu d'aventure ?



En dépit de tout cela, je pense qu'on doit leur redonner une chance. Il y a encore un marché pour les jeux au rythme lent où le challenge est surtout mental. Rempli de casse-tête intelligents et de plaisirs visuels, les jeux d'aventure ont toujours été populaires auprès des femmes. Bien que la plupart des femmes utilisent bien plus souvent un ordinateur et jouent plus souvent à des jeux qu'autrefois, je crois que l'industrie a fait un pas en arrière pour ce qui est d'offrir un divertissement qui puisse plaire au plus grand nombre de joueuses. L'accent mis actuellement sur les jeux de courses, de vol ou de tir (tout ça grâce à l'accélération 3D, bien sûr) n'a pas vraiment intéressé beaucoup de femmes ; encore moins le fait de pinailler sur la gestion de production d'armes qui prend tellement de temps dans les jeux de stratégie en temps réel.

L'autre marché pour lequel les jeux d'aventure sont parfaits est celui des jeunes enfants, particulièrement si le jeu ne demande pas beaucoup de dextérité. Les enfants s'évadent très facilement du monde réel (je n'arrive pas à croire que plus jeune j'aimais *Le Sous-marin de l'apocalypse* (NDT : un film américain des années soixante)), et ils aiment comprendre des choses tout comme les adultes. L'énorme succès de la refonte de *The Legend of Zelda* sur Nintendo 64 démontre manifestement qu'il existe encore un marché, et les moteurs 3D ont aussi tout autant à apporter aux jeux d'aventures qu'aux autres genres.



Nous allons encore devoir faire face à cette question des coûts de développement, mais avec des entreprises qui dépensent couramment un million de dollars ou plus dans leurs jeux, ce n'est pas comme si les autres genres étaient raisonnables non plus. La voix-off et les cinématiques que l'on retrouve dans les jeux d'aventure sont maintenant inclus dans toute sorte de jeux. Le coût de fabrication des cinématiques est le même pour un jeu de guerre ou un jeu d'aventure.

Les jeux d'aventure plaisent à un public qui n'est pas focalisé sur la taille d'une explosion ou de la vitesse du moteur, un public que nous ignorons la plupart du temps. Mais ces gens veulent jouer à des jeux aussi. Il est temps pour les jeux d'aventure de revenir.

Introduction à GraphicsGale

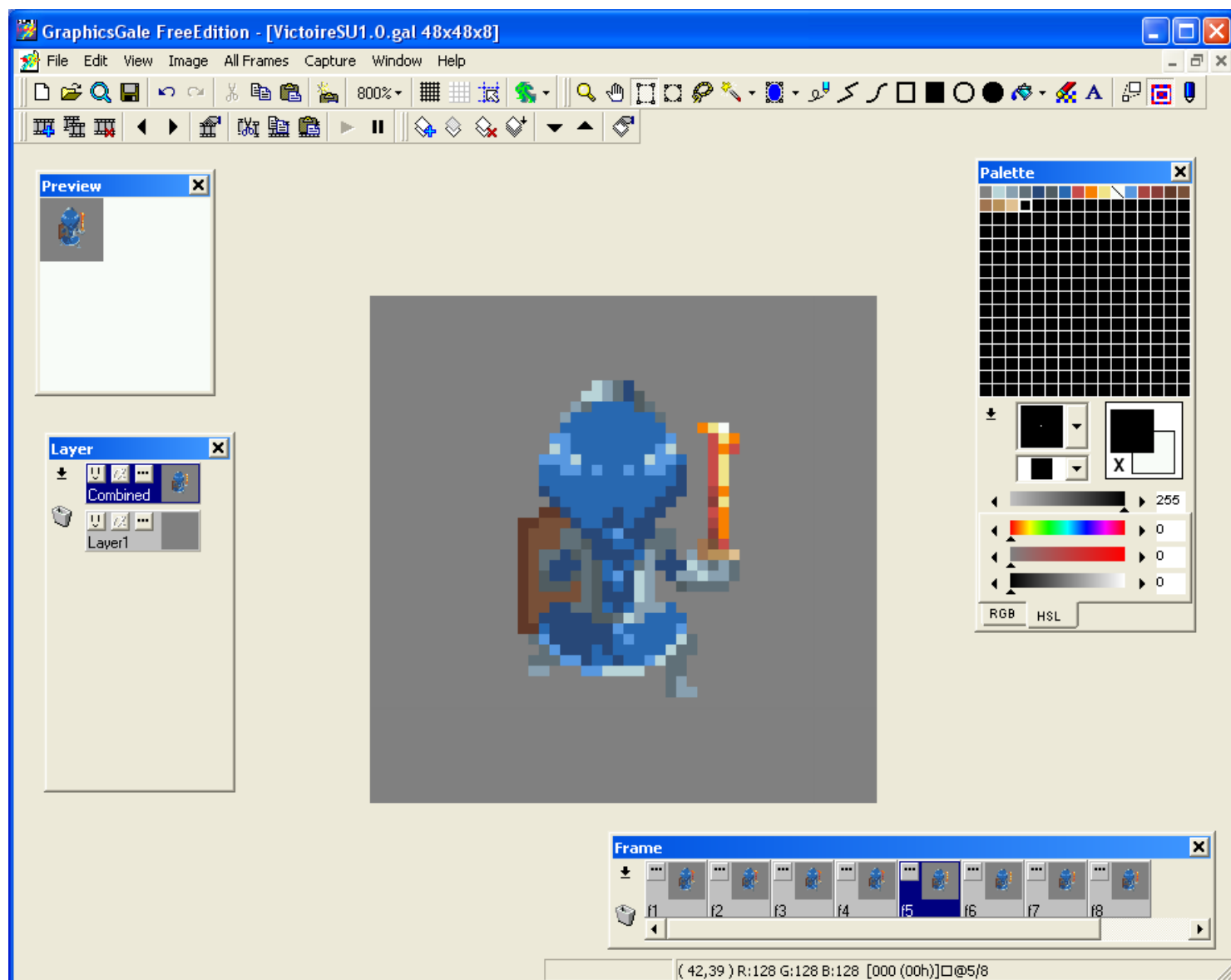
Par [Thomas Chassin](#) | le 13 novembre 2011 16:00:00 | Catégories : [Cours](#), [Pixel art](#)

GraphicsGale est un outil très puissant pour quiconque s'intéresse au pixel art, et propose une version d'essai gratuite qui reprend la majorité des fonctionnalités de la version complète (sauf les formats comme .gif, .cur, .ani et .ico qui ne sont pas supportés) téléchargeable [ici](#).

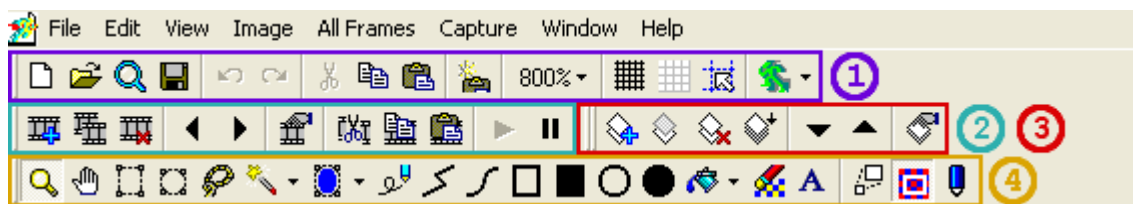
GraphicsGale permet notamment l'utilisation de plusieurs frames pouvant elles-mêmes contenir plusieurs calques et gère la transparence. C'est également un outil de gestion des palettes restreintes très puissant, mais trêve de bavardage, entrons dans le vif du sujet.

L'interface de GraphicsGale

Étudions de plus près le fonctionnement de l'interface, dont voici un aperçu:



Les menus



Les blocs (déplaçables) sont abordés dans l'ordre affiché ci-dessus et les icônes sont abordées de gauche à droite.

Bloc n° 1: Options générales

- **Nouveau** : Est-il vraiment nécessaire de détailler l'utilité de cet outil ? Il permet simplement la création d'un nouveau document. (Ctrl + N)
- **Ouvrir** : Ici aussi, rien de bien extraordinaire. Ce bouton provoque l'ouverture d'un explorateur vous permettant de sélectionner un document à ouvrir. (Ctrl + O)
- **GaleBrowser** : Ouvre le GaleBrowser, un outil de recherche plus perfectionné (compris dans le pack d'installation de la version d'essai) qui permet entre autre de prévisualiser les fichiers Gale (.gal). Je ne m'en suis jamais servi, donc je ne m'étendrai pas dessus.
- **Sauvegarder** : Sauvegarde le fichier sélectionné. (Ctrl + S)
- **Précédent** : Annule la dernière action effectuée. (Ctrl + Z)
- **Refaire** : Refait la dernière action annulée (Ctrl + Y)
- **Couper** : Copie la sélection vers le presse-papier et la supprime du fichier. (Ctrl + X)
- **Copier** : Copie la sélection vers le presse-papier. (Ctrl + C)
- **Coller** : Copie l'élément présent dans le presse-papier. (Ctrl + V)
- **Coller comme nouvelle image** : Crée un nouveau document à partir de l'élément du presse-papier.
- **Grossissement** : Permet de choisir le grossissement de l'image. L'option *setup* permet de configurer un grossissement personnalisé.
- **Grille** : Affiche la grille de délimitation des pixels.
- **Grille personnalisée** : Affiche une grille aux dimensions réglables (très, très pratique). L'option *input* permet de créer une grille paramétrable et l'option *setup* permet de définir de nouvelles grilles par défaut.
- **Curseur aimanté** : Active ou désactive le magnétisme du curseur. Le curseur aimanté se place automatiquement dans le bord supérieur gauche d'une case.
- **Peau d'oignon** : Cette option permet de visualiser en transparence la *frame* précédente ou/et suivante. Notez bien que la transparence ne dépend pas de la profondeur des couleurs (marche même en 1bit)

Bloc n° 2: Les outils

- **Loupe** : Lorsque cet outil est sélectionné, le clic gauche provoque l'augmentation du grossissement et le clic droit sa diminution.
- **Outil main** : Cet outil vous permet de déplacer librement l'image en maintenant une pression du clic gauche.
- **Sélection rectangulaire** : Définit une plage de sélection rectangulaire à partir du point sélectionné en maintenant le bouton gauche de la souris enfoncé. Notez que cet outil peut également être utilisé lorsqu'un autre outil est sélectionné en maintenant enfoncé le bouton droit de la souris. Le clic droit permet d'annuler la sélection.
- **Sélection ovale** : Définit une plage de sélection ovale à partir du point de clic.
- **Lasso** : Définit une plage de sélection personnalisée en traçant un contour en maintenant enfoncé le bouton gauche de la souris.
- **Baguette magique** : Sélectionne une plage de pixels de couleur plus ou moins uniforme selon le niveau de tolérance choisi. Fait la différence entre deux couleurs identiques en tolérance 0.
- **Sélecteur de couleur** : Sélectionne tout pixel de la couleur choisie au sein du claque sélectionné. La tolérance est paramétrable. Fait la différence entre deux couleurs identiques en tolérance 0.
- **Crayon** : Votre outil de dessin principal. Il applique le motif choisi en le centrant sur le pixel cliqué.

- **Outils segment** : Trace un segment entre deux points successivement définis. Le clic gauche définit un nouveau point, le clic gauche valide le tracé.
- **Outil courbe** : Trace une courbe passant par les points définis. S'utilise comme l'outil segment.
- **Rectangle** : Trace un rectangle d'une pression maintenue du clic gauche du point d'origine au point d'arrivée.
- **Rectangle plein** : Similaire au rectangle, mais trace une plage rectangulaire de couleur uniforme.
- **Ovale** : Trace un ovale à partir de deux points. Utilisation similaire à l'outil rectangle.
- **Ovale plein** : Trace une plage ovale de couleur uniforme.
- **Pot de peinture** : Remplit une plage de pixels de couleur plus ou moins uniforme selon le niveau de tolérance choisi. Fait la différence entre deux couleurs identiques en tolérance 0.
- **Remplacer couleur** : Fonctionne de la même manière que le crayon, mais uniquement sur la couleur secondaire. Fait la différence entre deux couleurs identiques.
- **Texte** : Ouvre une fenêtre d'édition de texte.
- **Antialias** : Active ou désactive l'antialiasing automatique lors du redimensionnement.
- **Transparence** : Active ou désactive la transparence de la sélection, définie par la couleur de transparence du calque ou de la *frame*.
- **Mode de peinture(?)** : Je n'ai honnêtement AUCUNE idée de l'utilité de cet outil, je ne m'en suis jamais servi en 3 ans, et chaque fois que j'ai essayé de l'utiliser, toutes les options étaient grisées.

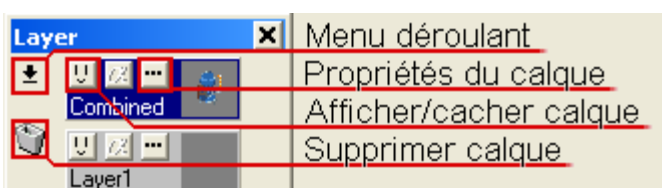
Bloc n° 3: Gestionnaire d'animation

- **Ajouter *frame*** : Ouvre une fenêtre permettant la création d'une nouvelle *frame* à la fin de l'animation.
- **Dupliquer *frame*** : Duplique la première *frame* sélectionnée et la place automatiquement la nouvelle à la fin de l'animation.
- **Supprimer *frame*** : Supprime la *frame* sélectionnée.
- ***frame* précédente** : Sélectionne la *frame* précédente.
- ***frame* suivante** : Sélectionne la *frame* suivante.
- **Propriétés** : Ouvre la fenêtre d'édition des propriétés de la *frame* sélectionnée.
- **Couper *frame*** : Supprime la(les) *frame*(s) sélectionnée(s) et la(les) ajoute au presse-papier.
- **Copier *frames*** : Ajoute la(les) *frame*(s) sélectionnée(s) au presse-papier.
- **Coller *frames*** : Ajoute la(les) *frame*(s) du presse-papier à la suite de la(des) *frame*(s) sélectionnée(s).
- **Lecture** : Joue l'animation dans la fenêtre d'aperçu.
- **Pause** : N'affiche que la *frame* actuelle dans la fenêtre d'aperçu.

Bloc n° 4: Gestionnaire de calques

- **Ajouter calque** : _ Ouvre une fenêtre permettant la création d'un nouveau calque.
- **Dupliquer calque** : Duplique le calque sélectionné et le place automatiquement au dessus de la pile.
- **Fusionner calques** : Fusionne tous les calques visibles. Le résultat est ce qui s'affiche à l'écran.
- **Supprimer calque** : Supprime le calque sélectionné.
- **Calque précédent** : Sélectionne le calque inférieur.
- **Calque suivant** : Sélectionne le calque supérieur.
- **Propriétés** : Ouvre la fenêtre d'édition des propriétés du calque sélectionné.

Fenêtre de gestion des calques

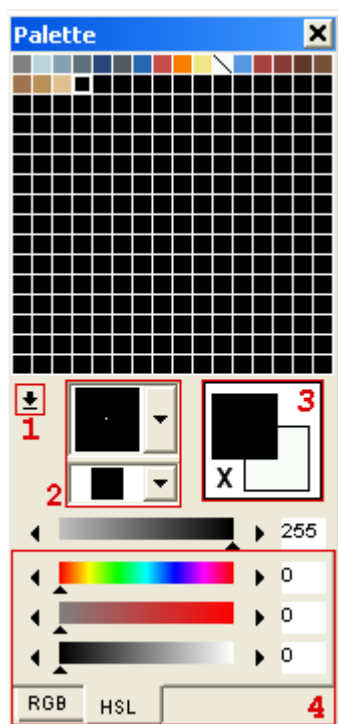


Le calque sélectionné est coloré en bleu. Pour réarranger vos calques, faites les glisser à l'emplacement désiré. Les propriétés d'un calque regroupent son nom, sa couleur de transparence et son opacité (attention, l'affichage se fait selon votre profondeur de couleur : en dessous de 15bit, vous obtiendrez un dithering). Les options du menu déroulant sont les mêmes que celle de la barre d'outils, à l'exception de l'option *Reverse*

visibility with same name, qui inverse la visibilité de tous les calques du même som (attention, l'icône de visibilité ne change pas, ce qui peut porter à confusion, raison pour laquelle je déconseille cet outil).

Vous remarquerez un bouton non légendé dont l'utilité m'échappe totalement.

Fenêtre de gestion de la palette



La palette

La palette affiche les couleurs utilisées dans l'image en dessous du 8bit (compris). Vous pouvez sélectionner une couleur en cliquant dessus avec le bouton gauche de la souris, ou sélectionner une seconde couleur en cliquant dessus avec le bouton droit. Notez bien qu'à partir du 15bit, la palette vous présente une échelle de teintes que vous pouvez modifier en double-cliquant dessus. Il faudra en revanche annuler la sélection de la couleur pour actualiser votre pinceau. La palette est lue de gauche à droite et de haut en bas.

Le menu déroulant (1)

Voici un petit descriptif des options proposées:

- **Copier valeur RGB** : Copie la couleur vers le presse-papier.
- **Copier valeurs RGB** : Copie la valeur RGB des couleurs comprise entre la couleur sélectionnée et la couleur secondaire.
- **Coller valeur RGB** : Copie la valeur RGB du presse-papier vers la couleur sélectionnée ou vers la couleur secondaire. L'option *Coller de multiples couleurs en partant de la couleur sélectionnée* permet de faire exactement ce qu'indique son nom.
- **Charger palette** : Permet de charger une palette de couleur à partir d'un fichier de palette (.pal) ou d'un fichier d'une profondeur de couleur de 8bit ou moins. L'option *Faire correspondre les pixels aux couleurs* permet d'attribuer à chaque pixel la couleur de la palette la plus proche de sa couleur actuelle. Si cette option n'est pas activée, la couleur est remplacée par la couleur qui prend sa place sur votre palette.
- **Sauvegarder palette** : Permet d'exporter votre palette actuelle en fichier .pal.
- **Palette par défaut** : Cette option réinitialise votre palette si vous travaillez en 15bit ou plus, c'est à dire que votre palette est remplacée par l'échelle de teintes initiales. Cela ne modifie pas les couleurs utilisées.

- **Faire un dégradé** : Crée un dégradé automatique entre votre couleur primaire et votre couleur secondaire sur la palette (Pratique pour créer rapidement une palette). En 8bit ou moins, les couleurs utilisées seront remplacées par les nouvelles couleurs.
- **Supprimer couleurs inutilisées** : Supprime de votre palette tout couleur qui n'apparaît pas au moins une fois. Cet outil fait la différence entre deux couleurs identiques.
- **Unir couleurs dupliquées** : Cet outil supprime les doublons de vos couleurs. Seule la première occurrence de la couleur est conservée.
- **Uniformiser couleurs** : Cette option permet de d'ouvrir un fenêtre permettant de gérer une palette globale pour tous les fichiers ouverts d'une profondeur de 8bit ou moins.
- **Recharger pinceau/tuile** : Encore une option aux fonctionnalités obscures.
- **Sélectionner première couleur pour transparence** : Affecte la couleur sélectionnée à la couleur de transparence de votre calque actuel.
- **Sélectionner seconde couleur pour transparence** : Affecte la couleur secondaire à la couleur de transparence de votre calque actuel.

Choix du pinceau (2)

Vous disposez de deux paramètres réglables ici: le pinceau et le motif de remplissage.

- **Le pinceau** : Vous pouvez choisir une des 8 formes de pinceau, de taille variable.
- **Le motif** : il existe 8 motifs de remplissage, qui permettent entre autre de créer rapidement des pages de dithering.

Couleurs sélectionnées (3)

La couleur du dessus, correspond à votre couleur principale, et la couleur du dessous à votre couleur secondaire. En passant votre souris dessus, la valeur RGB de la couleur s'affiche. Vous pouvez éditer les couleurs en les cliquant. La petite croix en bas à gauche permet d'inverser couleur principale et couleur secondaire.

Éditeur de couleurs (4)

Permet de faire varier les paramètres TSL ou RGB de la couleur sélectionnée. Notez qu'en 15bit et plus, la modification n'affecte que la couleur de votre pinceau, pas la couleur sélectionnée sur la palette.

Gestion des frames

La fenêtre de gestion des *frames* reprend les symboles qu'on peut retrouver ailleurs dans le logiciel, il est inutile de la commenter. Cependant, les options de gestion des *frames* sont intéressantes. Les propriétés d'une *frame* sont son nom, sa transparence et le délai d'attente avant la *frame* suivante. Le paramètre *Disposal* permet de gérer la méthode de suppression des *frames*. Il est possible de changer les propriétés de l'ensemble des *frames* en cochant la case *All frames*.

Quelques conseils et astuces

- Le format Gale (.gal) permet de sauvegarder vos calques et *frames*, pensez à l'utiliser pour la réalisation d'éléments complexes.
- GraphicsGale est un puissant outil de gestion des palettes limitées, créez vos fichiers en 8bit ou moins : cela vous évitera d'avoir des couleurs semblables qui traînent partout dans votre image.
- Pour les grilles d'animation ou les tilesets, la grille personnalisée est votre meilleure amie : utilisez là.
- Un antialiasing réalisé automatiquement, c'est moche : faites le vous-même.
- GraphicsGale est paramétrable (File -> Preferences), profitez-en pour l'adapter à vos besoins.

Pourquoi écrire un Game Design Document ?

Par [Pierre Guyot](#) | le 11 décembre 2011 15:44:23 | Catégories : [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juillet 2007).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

De temps à autres, je reçois des questions de la part d'étudiants ou je lis des posts de développeurs débutants qui exigent de savoir pourquoi ils devraient écrire des *Game Design Document* (GDD). Ils veulent se plonger tout de suite dans la modélisation ou le code et ils ne voient le travail sur papier que comme une perte de temps. Les joueurs ne le verront jamais, alors à quoi bon ?

Je connais cette façon de penser : j'avais la même aussi quand j'ai commencé à créer des jeux. Je me revois en train de coder en FORTRAN bien avant d'avoir une idée claire de comment mon jeu allait bien pouvoir fonctionner en temps que divertissement ou même en tant que programme. Bien sûr, à l'époque, il n'y avait pas de designers expérimentés dans le coin pour me dire comment j'aurais dû faire — tout le monde prenait les décisions au fur et à mesure.

Chez les débutants, l'une des objections les plus courantes à l'écriture de GDD est que personne ne les lit. Ça semble logique au premier abord, mais en réalité cet argument est complètement hors-sujet. Personne ne lit l'annuaire non plus et pourtant, s'il n'y avait pas de moyen de trouver les numéros, le téléphone serait beaucoup moins utile. Comme l'annuaire, la plupart des GDD ne sont pas conçus en vue d'être lus mais en vue d'être consultés. Personne ne les lit d'un bout à l'autre, mais les managers et les développeurs peuvent y trouver les informations qui concernent leurs tâches respectives.

Une autre objection classique est que, dans la mesure où les jeux sont d'abord développés sous la forme de prototypes, le prototype peut constituer la base du jeu et l'équipe peut y ajouter de nouvelles fonctionnalités, alors pourquoi écrire un GDD ? Mais ce n'est pas pour ça que le prototype existe et procéder ainsi le prive de sa valeur. Les prototypes sont censés être conçus rapidement et salement, et plus ils sont conçus salement, plus ils pourront être conçus rapidement. Ils sont faits pour le test, pas pour la conception — vous pouvez y jouer, mais vous ne pouvez pas y trouver de données ou de plans. Dans la célèbre [Méthode Cerny](#), l'auteur avertit les développeurs que tout ce qu'ils produisent pour le prototype sera jeté. Ceci les autorise à y aller très grossièrement et les assure que s'il y a des bidouilles dans le prototype, ces bidouilles ne seront pas dans le produit fini. Il est toujours dangereux de transformer le code d'un prototype en code final, et Cerny évite ce risque en faisant en sorte que cela ne se produise pas. De plus, les prototypes ne contiennent presque jamais tout les éléments du jeu fini. On utilise principalement les prototypes pour tester les mécaniques de jeu et les interfaces utilisateur. Si un jeu comporte trente niveaux, le prototype en présentera trois ou quatre, voire moins. Il reste encore à concevoir et à poser sur le papier tous les éléments restants pour que l'équipe puisse les réaliser. Un prototype ne remplace pas les documents papier — diagrammes, cartes, listes d'objets, etc. — nécessaires à la construction de ces niveaux.

Jusqu'à présent, j'ai répondu à quelques objections, mais je n'ai pas encore avancé de raisons positives pour lesquelles les GDD sont importants. Quiconque a mené un gros projet dans l'industrie du jeu vidéo sait parfaitement pourquoi ils le sont, mais c'est une question raisonnable pour ceux qui n'ont pas encore pu profiter de ce (douteux) privilège. Il y a en fait plusieurs raisons, que j'évoquerai en commençant par la moins importante et en finissant par la plus essentielle.

Raison 1 (la moins importante) : les gens qui vous financent (éditeurs ou autres) veulent des GDD comme preuves que le designer sait ce qu'il fait.

Beaucoup de gens dans l'industrie voudrait avoir l'argent d'abord et se pencher sur les détails après — évidemment, qui ne voudrait pas, s'il pouvait s'en tirer comme ça ? Parfois vous pouvez réussir à embobiner un investisseur privé pour qu'il vous laisse faire ça, en particulier s'il ne s'y connaît pas trop en affaires. Mais

un éditeur sensé ne va pas simplement placer des millions de dollars dans les mains d'un designer qui n'a pas de plan clair en tête. Les producteurs exécutifs veulent une trace écrite. Ils n'insistent plus sur une bible du jeu de trois cents pages, comme il y a quinze ans, mais ils veulent quand même quelque chose qu'ils puissent tenir entre leurs mains et montrer au département marketing. Aujourd'hui, on demanderait plutôt un document d'une trentaine de pages, mais il s'agit toujours d'un document et quelqu'un doit l'écrire. Pas de GDD, pas d'argent.

Si vous êtes auto-financé, ce n'est pas un problème, mais il y a plusieurs autres bonnes raisons pour écrire un GDD même si vous vous débrouillez pour payer.

Raison 2 : les GDD sont parfois la base d'obligations contractuelles.

La plupart des contrats de développements comporte un calendrier des différentes étapes qui définit quand le développeur devra fournir certains livrables et quand l'éditeur devra avancer plus d'argent pour le développement. Vous ne pouvez pas créer un calendrier avant de savoir quelles fonctionnalités seront dans le jeu, et vous ne pouvez pas savoir ça avant d'en avoir conçu au moins une partie, ce qui inclut une trace écrite pour pouvoir bâtir le rapport dessus.

En pratique, ces calendriers changent tout le temps, et la liste des fonctionnalités aussi. Ça n'a pas d'importance : il faut bien commencer quelque part. Pas de liste de fonctionnalités, pas de calendrier ; pas de calendrier, pas de contrat. De plus, il est dans l'intérêt du développeur d'avoir une planification aussi claire et précise que possible. Si les développeurs peuvent prouver sans doute possible qu'une certaine part du travail est faite et qu'un nouveau paiement est dû, ils sont en position de force. Si le game design consiste à brasser de l'air, les éditeurs peuvent brasser beaucoup d'air en retour pour expliquer qu'ils diffèrent le paiement et exigent plus de travail. Si vous savez ce que vous vous engagez à fournir, vous êtes certains de savoir si vous avez tenu vos engagements à temps.

En plus de la relation légale entre éditeur et développeur, il y a aussi la relation légale entre le développeur et les sous-traitants extérieurs. Aujourd'hui, la musique, le graphisme, l'animation et l'écriture sont souvent sous-traités auprès d'agences spécialisées. Pour que ces entreprises puissent faire leur travail, elles doivent disposer de documents qui leur indiquent ce qu'on attend d'elles et quand, documents qui peuvent constituer la base de leur contrat.

Raison 3 : le GDD communique vos intentions au reste de l'équipe et lui permet de planifier les différentes tâches.

En théorie, voilà pourquoi tout le monde devrait écrire un GDD : afin de transmettre l'information aux autres. Les petites équipes n'en ont pas autant besoin dans la mesure où les membres travaillent souvent dans la même pièce et se parlent tout le temps. C'est pourquoi les étudiants et les débutants ne voient pas l'intérêt d'écrire un GDD : ils pensent que s'ils peuvent parler ensemble, il n'y a pas besoin de mettre quoi que ce soit sur le papier. Néanmoins, comme nous l'avons vu, ces documents sont conçus pour d'autres raisons, contractuelles entre autres. Étrangement, communiquer avec le reste de l'équipe n'est pas la raison la plus importante pour écrire un GDD, mais c'est déjà une raison et une bonne.

Les enseignants demandent aux étudiants d'écrire des GDD même si les équipes d'étudiants n'en ont pas forcément besoin pour la communication. c'est parce que les enseignants savent que les étudiants ne travailleront pas toujours au sein d'une petite équipe et qu'ils ont besoin de s'entraîner avant de rentrer sur le marché du travail. Les grandes équipes de développement peuvent aller jusqu'à cent cinquante personnes (NDT : et même plus de trois cent sur un *Final Fantasy XIII*) et sont souvent réparties sur plusieurs bureaux, voire plusieurs pays si une partie du travail est sous-traitée. Parler tout le temps, même par téléphone ou vidéoconférence, n'est pas pratique.



Plus le jeu est grand, plus il est important de poser le design sur papier, de façon à ce que les autres puissent élaborer leur planning. Si vous voulez que votre jeu inclue 45 types de créatures, les artistes de l'équipe devront modéliser, texturer et animer chacun d'entre eux. Ils auront besoin de concept art à partir desquels travailler — une autre forme de *design document*. Les ingénieurs du son auront besoin de trouver ou créer des effets sonores pour chacune de ces créatures. Si elles sont autonomes, les programmeurs auront besoin de connaître leurs caractéristiques comportementales. Si vous ne posez pas tout ça sur le papier, comment ces gens pourront-ils savoir tout ça ? Vous ne pouvez pas vous contenter de leur expliquer ça lors d'un meeting et espérer à ce qu'ils se souviennent de tout.

Quand je faisais de la production audio et vidéo pour la série des *Madden*, j'ai écrit les scripts d'enregistrement ligne par ligne — encore une autre forme de *design document*. A la fin, ça faisait environ 75 pages. Nous devions enregistrer du contenu audio pour tous les événements qui pouvaient se produire dans le jeu et tout les commentaires possibles pour John Madden. Personne ne pourrait garder tout ça en tête et dans tous les cas, Madden avait besoin du texte pour le lire et l'ingénieur du son avait besoin d'une base pour s'organiser lorsqu'il retravaillait les prises de son brutes. Les jeux de sport nécessitent plus de GDD que vous ne pourriez le penser, parce que bien que ce soit la ligue qui fixe les règles du jeu, quelqu'un doit définir toutes les différentes stratégies [c'est-à-dire le *playbook* (NDT : le carnet contenant les diagrammes stratégiques et descriptions d'une équipe, plus particulièrement au football américain) de chacune des équipes], les animations et l'interface utilisateur pour traduire le sport en jeu. Tout ce travail produit de la documentation.

Enfin, sur certains projets, tous les membres d'une équipe ne parleront pas la même langue — littéralement. J'ai été confronté à ça quand j'ai travaillé avec THQ (Angleterre) et GSC Game World (Ukraine) sur *S.T.A.L.K.E.R.: Shadow of Chernobyl*. La plupart des développeurs ne parlaient pas anglais du tout, uniquement le russe ou l'ukrainien. Il n'était pas possible d'avoir des traducteurs présents quarante ou soixante heures par semaine : une grande partie de la communication entre éditeur et développeur se faisait donc sous la forme de documents traduits.

Raison 4 : un GDD permet de transformer des généralités en particularités.

Le processus d'écriture transforme une idée vague en plan explicite. C'est une chose de dire « Les harpies seront des créatures volantes. » lors d'un meeting, mais c'est très loin d'être suffisant pour constituer une base de travail. Ce dont les développeurs ont besoin, ce sont les détails : à quelle hauteur peuvent-elles voler ? A quelle vitesse volent-elles ? Sont-elles affectées par la météo ? Peuvent-elles atterrir ? Peuvent-elles atterrir n'importe où ? Peuvent-elles se déplacer au sol et si oui, sur quelles sortes de terrains et à quelle vitesse ? Sont-elles plus (ou moins) vulnérables dans les airs que sur le sol ? Et ainsi de suite et tout doit être posé sur papier pour que l'équipe ait toutes les informations pour réaliser le produit.

Ce serait bien si le game design consistait à rêvasser les pieds sur la table et à s'imaginer des fonctionnalités et du contenu « trop cools ». J'ai rencontré des designers qui pensent que c'est le cas. Ça ne l'est pas : ces types sont des fainéants. La plus grande partie du design consiste à penser les détails. Bien que vous puissiez toujours changer les détails plus tard lors des phases de test et d'équilibrage, vous devez bien commencer quelque part. On peut même véritablement dire que le processus de design, *c'est l'écriture des design documents*, parce que c'est le moment où vous transformez des concepts abstraits en plans concrets. Même si personne ne lit votre document, écrire une idée, c'est prendre une décision, atteindre une conclusion.

Raison 5 (la plus importante) : les *design documents* constituent une trace écrite des décisions qui ont été prises.

Le jeu vidéo est une activité hautement collaborative, bien plus que le cinéma. Contrairement à un réalisateur de film, qui règne presque sans partage, peu de designers ont un contrôle total sur leur jeu. En tant que développeurs, nous tolérons les longs horaires et les salaires comparativement bas de l'industrie du jeu vidéo parce que nous avons la chance d'apporter une contribution créative et si cela nous était retiré, il n'y aurait plus d'intérêt. Un *lead designer* ne conçoit pas lui-même l'intégralité du jeu : il doit continuellement mettre en commun les idées d'autres gens et (dans la mesure du possible avec tact) rejeter les idées qui ne collent pas.

En conséquence, une énorme partie des décisions de conception ne sont pas prises à votre bureau, mais lors de meetings, autour d'une tasse de café ou d'un repas. Certaines d'entre elles, peut-être faites par un débutant, sont juste des tentatives et doivent être mises de côté avec l'aide du lead designer et le reste de l'équipe. Dans tous les cas, quand une décision de game design est prise à la suite d'une conversation ou d'une négociation, vous devez poser vos conclusions sur le papier — même si vous savez déjà que vous allez les modifier plus tard, je le répète. La raison en est que vous avez besoin d'une trace papier, un enregistrement de ce que vous avez déjà décidé. Je me suis retrouvé dans trop de meetings où un débat éclatait parce que personne n'avait noté la précédente décision et que les gens n'étaient pas d'accord sur ce qui avait été décidé : « On n'avait pas dit qu'on ferait ça ? Non, pas du tout, on avait décidé qu'on ferait ça ! ». Tout ce qui en résulte, c'est une perte de temps et d'énergie. Si une semaine ou deux se sont écoulées depuis le précédent meeting, il se peut que toute l'équipe ait passé son temps à travailler sur la base de suppositions erronées. C'est pourquoi il devrait y avoir pour chaque meeting un secrétaire qui prend des notes et les envoie à chacune des parties concernées — et si les questions débattues sont des questions de design, ces notes font partie des *design documents* et devraient être répertoriées en tant que telles. Si les gens ne se souviennent pas de la même chose, vous pouvez revenir en arrière et vérifier ces notes.



Les *design documents* vous aident aussi à garder une trace de ce que vous avez fait et de ce que vous avez encore à faire. Si une fonctionnalité du jeu n'a jamais été décrite dans un document écrit, il y a de bonnes chances que vous l'ayez oubliée et que quelqu'un vienne vous en reparler beaucoup plus tard, ou pire, prenne une décision sur le sujet sans en parler ni à vous, ni à qui que ce soit. Le résultat peut être désastreux quand chacune des parties de l'équipe a une idée différente de ce que vous attendiez et produit un résultat incohérent. Il est bien plus facile et moins cher de corriger une erreur de design avant que le code soit écrit ou que les artworks soient réalisés.

J'ai cité cet argument comme le plus important parce qu'avant tout, les *design documents* sont des outils pour s'organiser. Il ne s'agit pas de livres ou d'histoires à lire, mais de planification : une liste des choses à mettre en œuvre. Ils peuvent prendre de nombreuses formes : diagrammes, *concept art*, références graphiques, textes explicatifs, tableaux de statistiques ou d'attributs, listes de toutes sortes, storyboards, *flowcharts* (oui, même de nos jours, mais pour la structure d'une interface utilisateur et non plus pour du code), notes de réunion, scripts d'enregistrement audio et video et synopsis pour vendre le projet à des éditeurs. Quand le jeu est presque terminé et que la majorité du travail consiste à tester et équilibrer le jeu, vous pouvez jeter les *design documents* de la même façon qu'on démonte les échafaudages à la fin d'un chantier, quoique ce soit toujours utile de les garder dans les parages comme références. Mais pendant que les choses sont en cours, les *design documents* sont essentiels pour garder la trace de ce qui se passe et de ce que vous devez faire ensuite.

Conclusion

Les *design documents* ne garantissent pas que vous ferez un grand jeu, ni un bon jeu, ni même que vous aurez fini dans les temps. En fait, si elle s'en sert mal, une équipe peut perdre un temps précieux avec ces documents et je parlerai de cet écueil dans un prochain article. Mais j'espère avoir répondu à certaines questions que l'on a pu me poser.

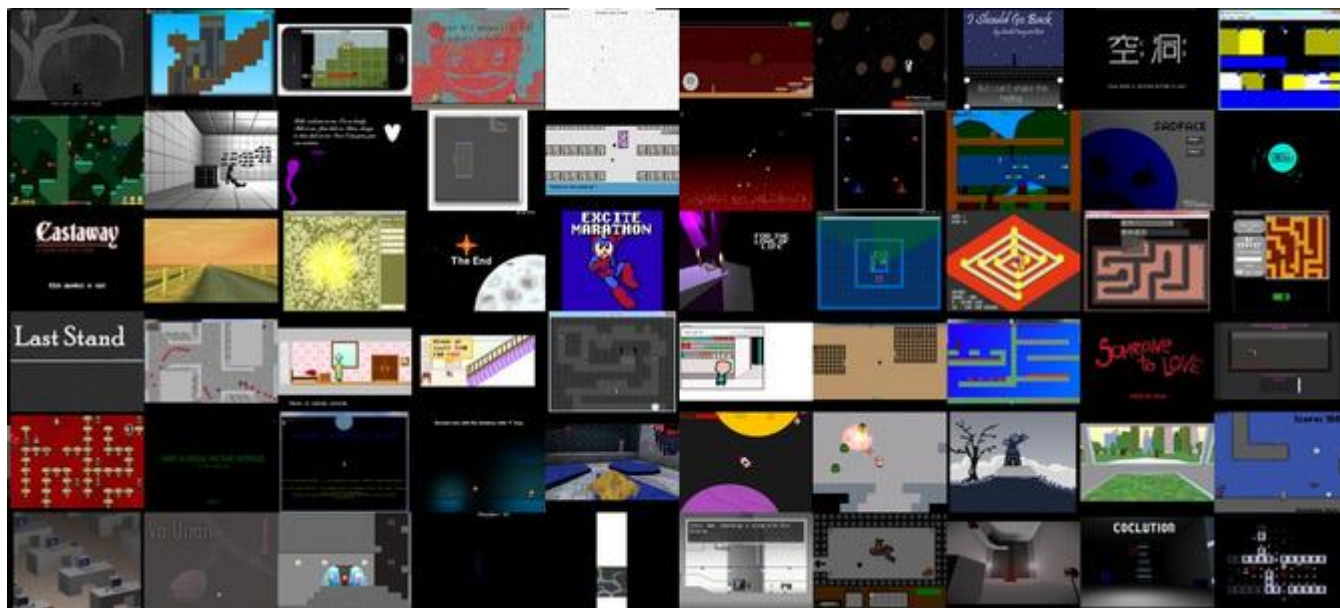
Oui, une petite équipe qui travaille sur un petit jeu, parfois sans contrainte de temps, n'a pas besoin de beaucoup en matière de documentation. Si votre carrière entière est dédiée à des jeux basiques sur portable, il se pourrait que vous n'écriviez jamais de vrai GDD (mais il faut toujours que quelqu'un pose les éléments de base sur le papier, non ?). Néanmoins, les professionnels qui travaillent sur de gros projets qui doivent absolument sortir pour Noël connaissent bien la valeur de la documentation. Elle permet de communiquer, d'organiser et de diriger tout le processus. Un manager de projet ne peut pas établir un programme, une liste de tâches et une répartition du travail — et suivre leur avancée — sans savoir ce qui doit être produit, et cette information doit exister sous une forme écrite.

Enfin, écrire (ainsi que dessiner, faire des diagrammes, des tableaux et des listes et écrire du pseudo-code), c'est ça la conception. Vous vous en passez à vos risques et périls.

Compte-rendu de Ludum Dare #22

Par [Antoine Gersant](#) | le 22 décembre 2011 16:39:54 | Catégories : [Game Design](#), [Graphisme](#), [Ludum Dare](#), [Ma vie](#), [Pixel art](#)

Quatre mois se sont écoulés depuis la précédente édition de Ludum Dare et je ne vais pas vous cacher que j'étais très impatient de participer à nouveau à cet événement. Pour ceux d'entre vous qui ne seraient pas familiers du principe et qui n'auraient pas lu mon [compte-rendu de la précédente édition](#), rappelons qu'il s'agit d'un weekend pendant lequel des développeurs du monde entier créent des jeux sur un thème imposé. Les participations peuvent se faire soit en 48 heures en solo (la compétition ou *compo*), soit en groupe en 72 heures (le *jam*).



Avant

Les préparatifs pour le Ludum Dare commencent plusieurs semaines en avance avec un formulaire où chacun (même les non-participants) peut proposer des thèmes. J'ai tenté ma chance avec *PRHÖFESSOR, I CAHN'T CONTRÖL THE MACHINE ANYMORH!* parce que sur le moment je trouvais ça rigolo. Malheureusement, des filtres automatiques ont été appliqués pour faire face à l'immense afflux de propositions de thèmes et ma suggestion n'a pas survécu au nettoyage sur les critères « moins de 3 mots » et « n'utilise pas de mots absents du dictionnaire ». Tant pis pour les scientifiques fous à l'accent allemand.

Afin d'être prêt au moment du top-départ (Samedi 19 Décembre à 3h00 du matin heure française), j'ai consacré mon après-midi à de savants préparatifs :

- Faire des courses riches en bacon, Coca Cola, cookies et bière
- Créer un squelette de projet vide prêt à accueillir les premières lignes de code
- Préparer une page web pour distribuer le (futur) jeu fini. Ça fait un peu bizarre de faire ça avant même de commencer le développement du jeu.
- Mettre à jour mon framework personnel, *Bacchus*, avec quelques fonctionnalités développées récemment sur une branche à part
- Installer et apprendre à utiliser *Chronolapse* (plus d'infos plus bas dans cet article)
- Annoncer ma candidature et mes outils sur le [blog Ludum Dare](#)
- Accorder ma guitare (on ne sait jamais)
- Nettoyer mon bureau qui avait un peu régressé au rang de poubelle à ciel ouvert ces derniers temps

Une heure avant le début de la compétition, je me trouvais fin prêt. Ouf!



Je suis ensuite aller me balader sur le blog de l'évènement et le canal IRC dédiés pour me mettre encore plus dans l'ambiance. J'y ai découvert au détour d'un lien l'existence de [Tiled](#), un éditeur de maps en 2D génériques. Je me suis dit que ça pourrait servir et j'ai consacré ma dernière heure à écrire un *parser* basique et incomplet (mais fonctionnel) pour pouvoir utiliser si nécessaire des fichiers produits par ce logiciel.

Pendant

A 3h00, le coup d'envoi est donné sur IRC : le thème est *Alone*. Je suis un peu déçu parce que j'adorais les thèmes *Forgotten Places* et surtout *Shapeshifting* pour lesquels j'ai déjà réfléchi à des concepts qui me plaisaient. Au moins le thème n'est pas *Kittens* (qui a fini bon dernier) comme l'espérait un lobby hyperactif et rigolo qui a inondé le blog Ludum Dare les jours précédents.

Les minutes passent et les bonnes idées ne viennent pas. J'envisage tout d'abord un jeu qui raconte l'histoire d'un marin seul survivant sur un navire et qui meurt à petit feu et où le joueur participe à des mini-jeux qui prennent place dans le souvenirs et les délires du personnage. J'abandonne l'idée car elle oblige à développer plusieurs gameplay de mini-jeux, ce qui ne semble pas très raisonnable. Je pense ensuite à un jeu vu de dessus où l'on incarne un petit garçon dans une école où personne ne lui parle à l'exception d'inquiétants démons que personne ne peut voir sauf lui. A la réflexion, je pense que c'était peut-être ma meilleure idée. Finalement, je me décide à parler d'un chevalier ensorcelé qui se retrouve seul dans un monde mystérieux (que je comptais au départ peupler de ses souvenirs et démons intérieurs).

J'ouvre mon *Photoshop* préféré et je commence à gribouiller des tiles de jeu de plateforme dans une ambiance crépusculaire (qui est un peu partie à la trappe petit à petit). J'aime bien commencer par des graphismes parce que ça me donne tout de suite une idée de ce à quoi mon jeu va ressembler et c'est motivant pour le programmer. Au petit matin, vers huit ou neuf heures, j'ai terminé mes tiles, un sprite de chevalier, une tombe, une épée et une fleur qui pourront toujours servir. Je m'inquiète beaucoup parce que je ne sais pas encore ce que je vais en faire exactement mais au moins les gens sur IRC et sur le blog disent qu'ils sont très jolis. Au moment d'aller me coucher, je repère que la boîte de *Dark Souls* en évidence sur mon étagère s'offre à mes yeux dès que je les détourne de l'écran — je la soupçonne très fort de m'avoir influencé sans que je le demande.

Après cinq heures de sommeil, je me lève avec la ferme intention de ne pas me coucher tant que mon jeu ne ressemblera pas à quelque chose. Je décide d'en faire un jeu de plateforme lent et où l'histoire progresse au fur et à mesure que le joueur avance dans les salles. C'est un peu ambitieux quand on sait qu'à ce moment là je n'ai toujours pas de scénario en tête mais au moins ça me fait quelque chose à coder.

Je passe toute la journée à alterner entre « je fais des maps avec Tiled » (j'ai bien fait d'écrire mon *parser* une heure avant la compétition), « je programme le jeu » et « je fais les animations du chevalier ». Avec le

recul, je pense que passer deux heures et demie à faire une animation de marche en 8 frames c'était une très mauvaise idée pour un Ludum Dare.



Arrive le début de la nuit, j'ai l'impression d'être dans un épisode de MacGyver dont le pitch est « Hum... Ok, j'ai un moteur de jeu de plateformes, des falaises, des cailloux, une épée, une fleur et un tombe et je dois les combiner pour faire un jeu ». J'écris une mini-histoire pour relier tout ça (je vous laisse la découvrir en jouant au jeu, le lien est en bas de cet article) et je la mets en place. J'utilise mes dernières forces pour faire le générique de fin et le texte d'introduction ; à ce moment là je suis plus que content d'avoir choisi d'utiliser un outil qui me permet de gérer l'avancement du projet sans tout garder en tête (ce qui est très difficile quand on est fatigué). J'ai choisi [Asana](#) que j'utilise déjà sur mes projets à long termes et dont je suis très satisfait d'une manière générale. Je vais finalement me coucher, épuisé mais ravi d'avoir un jeu qui ne demande plus qu'à être mis en musique, décoré et testé avant d'être fini.

The screenshot shows the Asana web interface. On the left is a sidebar with the Asana logo, a search bar, and navigation tabs for 'ECN', 'PERSONAL', and 'Antoine's Tasks'. Under 'PERSONAL', there are sections for 'PROJECTS', 'TAGS', and 'PEOPLE'. The 'PROJECTS' section lists 'Bacchus', 'Daily shit', and 'LD22'. The 'LD22' project is selected and highlighted. Below the sidebar, there's a 'New Project' button and a settings gear icon. The main area displays the 'LD22' project with a star icon and an 'Unfollow Project' button. It shows a list of tasks with checkboxes and completion status. The tasks are grouped into sections: 'Courses:', 'Préparatifs:', 'GAME:', and 'Post-game:'. The tasks listed are: 26 Music title screen (checked), 27 Music wind (checked), 28 Lower volume on confirm sound (checked), 29 Ladder sound (checked), 30 Additonal levels? (checked), 34 Death animation, 35 Music intro, 36 Title Screen ART, 37 Make intro and outro prettier (gothic font ?, book?), 38 More tiles, 39 Ecran fin, 41 Post Mortem on LD site, 42 Post Mortem on Les Forges, and 43 (empty). The bottom of the sidebar shows the user's name 'Antoine Gersant' with a dropdown arrow.

Task ID	Task Name	Status
26	Music title screen	Completed
27	Music wind	Completed
28	Lower volume on confirm sound	Completed
29	Ladder sound	Completed
30	Additonal levels?	Completed
Courses:		
Préparatifs:		
GAME:		
34	Death animation	Pending
35	Music intro	Pending
36	Title Screen ART	Pending
37	Make intro and outro prettier (gothic font ?, book?)	Pending
38	More tiles	Pending
39	Ecran fin	Pending
Post-game:		
41	Post Mortem on LD site	Pending
42	Post Mortem on Les Forges	Pending
43		Pending

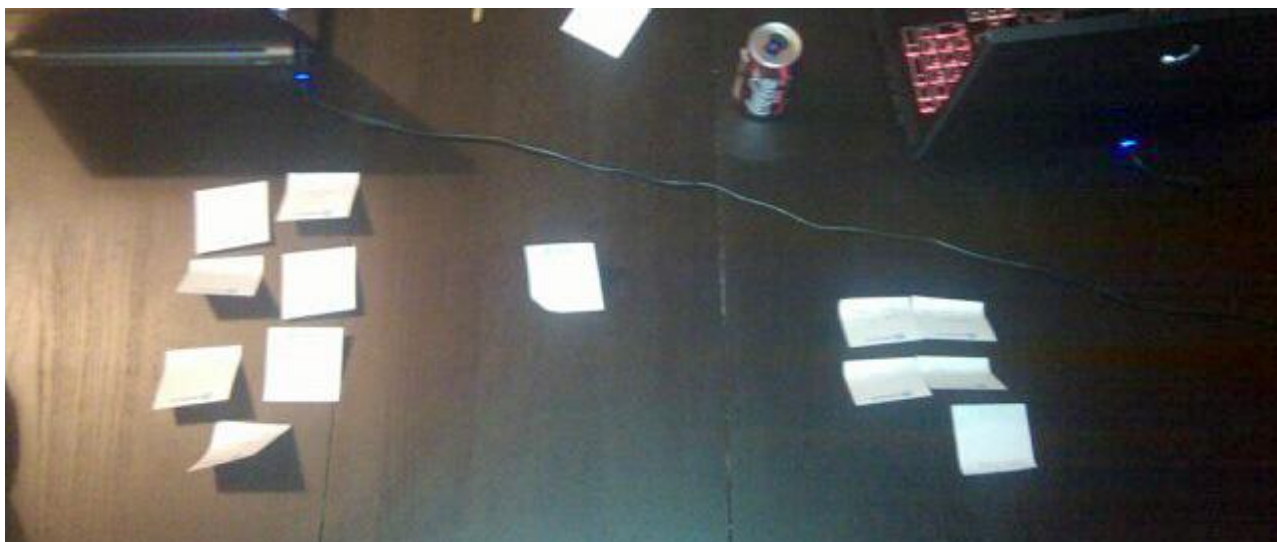
Je me lève vers 13h00 et après un copieux casse-croûte j'appelle deux amis nantais pour leur demander s'ils auront le temps de venir faire les beta-testeurs plus tard dans la soirée. Ils sont trop cools donc ils acceptent et ils me proposent même de me ravitailler, ce qui est inutile parce que j'ai encore du bacon et des cookies en quantité suffisante.

Je passe quelques minutes à faire un écran titre (je décide d'appeler le jeu *Castaway*) et un écran d'instructions très sobres pour pouvoir m'atteler au plus vite aux bruitages et à la musique. Je me dis que [BFXR](#) ne sera pas suffisant pour faire des effets sonores qui correspondent à l'ambiance du jeu et je mets à contribution mon micro pourri qui me sert d'habitude à parler sur Teamspeak. En nettoyant un peu les enregistrements et en les refaisant cent fois jusqu'à ce qu'il n'y ait plus d'ambulances dans ma rue pour tout gâcher, j'arrive à un résultat très acceptable vu mon manque d'expérience dans le domaine. Je suis particulièrement satisfait des sons de mort à la *Tekken* (vous savez, *HEUAAARRRHH HEUUuaah heuaah...* avec plein de *delay*) et des bruits de pas. Ci-dessous une photo de différents objets qui ont servi à faire les bruitages du jeu :



Je m'occupe ensuite des musiques, en commençant par les plus importantes (la musique principale et celle d'introduction). Comme il me reste du temps j'écris quelques mesures pour l'écran titre et je fais une piste « grondement mystérieux » pour les maps sans danger. Contrairement au précédent Ludum Dare, j'avais ma guitare et un petit clavier MIDI sous la main ainsi que deux grands écrans pour étaler *Reason* (mon *Photoshop* version audio) et j'ai pu travailler vite et bien selon mes standards personnels. En plus, cette fois-ci je n'ai pas perdu la stéréo sur une piste en oubliant de brancher un câble virtuel...

La soirée commence, mes vaillants bêta-testeurs arrivent armés de leurs ordinateurs et de kebabs qui sentent mauvais. Je dîne avec eux et je les installe dans le salon pour ne pas être trop déconcentré quand j'aurai à corriger des bugs compliqués. Je regarde les premiers tests avec anxiété, ils trouvent le jeu très difficile (c'était voulu) mais ils rient beaucoup à chaque fois que l'autre meurt, je trouve ça plutôt encourageant. Étant donné qu'ils repèrent des wagons de bugs, je m'aperçois qu'il va falloir s'organiser un peu et comme je n'ai pas le temps de leur faire découvrir Asana, je transforme la table du salon en un micro-scrumboard à base de post-it. Non seulement c'est marrant mais en plus c'est super pratique.



J'arrive au bout de la longue liste de bugs une heure et demie avant la fin de la compétition et j'en profite pour rajouter, sur suggestion d'un de mes testeurs, une salle cachée avec une armure bonus qui facilite la suite du jeu. Trente minutes plus tard, j'upload la version finale, je remplis le formulaire de participation sur le blog Ludum Dare (qui est à la limite du crash tellement il y a de gens dessus), je rajoute des liens de téléchargement vers la bande son (j'adore le moment où je dois donner des titres aux morceaux qui s'appellent jusqu'ici 0, 1 et 2) et les sources. Il est 2h55, je décapsule ma dernière bière et j'ai un sourire jusqu'aux oreilles.

Après

Comme la première fois que j'ai participé, l'effet Ludum Dare se fait sentir. Il est toujours aussi difficile de se rendre compte que 48 heures plus tôt je me demandais quel serait le thème et que là, paf ! j'ai fabriqué un jeu. Il est très court et pas génial mais il est complet et c'est quelque chose d'incroyable en soi. Enfin je trouve.

Je vous mentionnais tout à l'heure [Chronolapse](#), c'est enfin l'heure de compacter les 4Go de screenshots qu'a pris le logiciel pendant le weekend pour en faire une vidéo. La création de *Castaway* en moins de 5 minutes ça donne ça :

Les heures et les jours ont depuis commencé à passer et les avis affluent. Le consensus général semble être que le jeu est très joli et peaufiné mais surtout très frustrant. J'ai choisi de rendre le jeu très difficile en faisant un personnage difficile à contrôler (le chevalier a beaucoup d'inertie) mais ça n'a pas l'air de plaire. C'est sans doute l'interaction entre mes deux testeurs qui avait rendu la difficulté amusante, seul ça a surtout l'air énervant. Dommage.

En plus de ça, j'ai appris que le jeu crashait violemment sur un grand nombre de machines. Oups, il manquait une ligne de code pour éviter ça. J'ai depuis publié une version alternative qui est beaucoup plus stable.

Au final, je suis tout de même content du résultat (oui oui, je m'envoie toujours des fleurs) et j'espère qu'il vous plaira aussi. Vous pouvez essayer le produit fini (la version stable) [par ici](#). Je profite aussi de ce court paragraphe pour remercier mes deux testeurs et les différentes personnes qui m'ont encouragé tout au long du weekend.

J'ai commencé à jouer à quelques jeux d'autres personnes et voici les premières perles que j'ai trouvé parmi les 891 participations (!) de cette édition :

- [Never Alone Hotline](#) de Pierrec
- [Chromboid](#) de Arne, pekuja et BurnZeZ

- [Vatnsmyrkr](#) de Skomakarn

Si vous en voulez d'autres, je vous recommande de faire un tour sur [Game Side Story](#) qui couvre très bien l'évènement.

Vivement avril prochain pour le prochain Ludum Dare. Mes bonnes résolutions pour la prochaine fois sont de passer moins de temps sur les graphismes, de ne **pas** faire un jeu de plateformes vu de côté et d'avoir plus de contenu. Espérons que tout cela se réalise ! Enfin, je ne peux que vous encourager à participer vous aussi à cet évènement si amusant et productif. Comme l'a dit quelqu'un sur Twitter, si seulement Ludum Dare avait lieu tous les weekends... on dormirait pendant la semaine !

La faute à qui ? C'est pas moi !

Par [Lyonsbanner](#) | le 30 décembre 2011 15:30:00 | Catégories : [Analyse](#), [Michael Molinari](#)

Cet article est une traduction de [Lyonsbanner](#).

L'[article original](#) est écrit par Michael Molinari publié par [SHMUPtheory] (juin 2009).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

(NDT : Cet article a également été traduit par **Jerom**, et nous le remercions pour avoir choisi ce thème)



Vous jouez à *Raiden III*, et vous vous acharnez dessus. Vous n'avez pas perdu une seule vie, votre stock de bombes est plein à craquer, et vient ensuite le boss qui n'a aucune chance face à votre plasma autoguidé à pleine puissance. Vous êtes le maître du monde, et soudain PAN ! Vous perdez une vie à cause d'un tir furtif, dont la simple existence anéantit votre parcours sans faute, vous fait perdre vos médailles ainsi que tous les bonus cachés en une explosion aveuglante. Est-ce que vous vous attendiez vraiment à ce que tout parte en fumée ? Vous savez que vous n'avez pas la possibilité de continuer cette partie après avoir laissé passer un truc pareil. Vos certitudes ont été réduites à néant par ce coup sournois qui s'est faufilé par derrière, juste au moment où vous pensiez que vous aviez la partie en main.



J'ai rarement vu des *Shoot'em Ups* qui non seulement me laissent mourir honteusement après avoir fait un magnifique parcours mais qui trouvent en plus le moyen de me faire croire que c'est de ma faute. Je suppose qu'il y a des gens qui, dans tous les cas de figures, respireront un grand coup en se disant : « Je ferai mieux la prochaine fois ». Pour les autres ce sera plutôt « Et merde ! Quelle connerie ! » etc. encore et encore.

Jetons un coup d'œil à *Raiden III*. Il y a des vaisseaux ennemis spécialement conçus pour vous tirer dessus seulement à l'horizontal. Dans un jeu où l'écran défile à la verticale, ceux qui connaissent peuvent témoigner qu'une telle initiative est sournoise et a pour but de vous donner envie de devenir violent (envers le jeu, la manette, la télévision, etc.). Difficile pour nos pauvres petits yeux de détecter les risques de collisions à l'horizontal (c'est peut-être une des raisons pour lesquelles [le test de la ligne verticale](#) en mathématique triomphe largement de ses équivalents horizontaux). C'est pourquoi les *Shoot'em Ups* horizontaux ont généralement un rythme de jeu plus lent, afin de mieux observer ce qui se passe à l'écran pour avoir une chance de survie. On comprend sans doute pourquoi *Defender* est si difficile, quelque soit la personne qui y joue. Et puis, il y a les *Shoot'em Ups* verticaux lents qui souffrent du problème inverse : l'absence de difficulté au niveau des projectiles et des ennemis est si ennuyeuse qu'on se demande si ça valait le coup d'entrer ses initiales aussi tôt. Oui on parle de toi, *Truxton II* (mais on t'♥ quand même).

Bien, parlons maintenant des jeux où je m'en veux tout le temps de perdre. Nous avons *Ikagura* et *R-Type*. Ces deux jeux sont principalement basés sur la mémorisation et les deux ont des ennemis injustes qui tirent à l'horizontal. Ce n'est pas comme si j'étais mieux prévenu de leur arrivée, tout se joue en fait sur la prédictibilité de leurs comportements. Dans *Raiden* (la série), j'ai une bonne idée de la manière dont un ennemi va tirer. Le tout, c'est de savoir 'quand'. C'est la clé de la réussite. Commençons par *Ikagura* en nous rappelant que tout est réglé avec un bon rythme de jeu qui permet de savoir exactement quand et pourquoi un projectile se dirige vers vous. Si vous activez par accident vos lasers automatiques, changez de polarité, absorbez ensuite la totalité de la mauvaise couleur, mourez et enfin arrivez à l'écran « veuillez entrer vos initiales », c'est la faute à qui ? Eh bien à vous ! (Bon d'accord, c'est moi. Et surtout dans les chapitres 2 et 3). Je suis mort des centaines (voire des milliers) de fois dans ce jeu, et je n'ai pas reproché au jeu une seule de mes morts.



R-Type n'est que vaguement dans la même lignée. Les scénarios sont suffisamment prévisibles pour que vous puissiez gérer ce qui se passe autour de vous. Je pense que *Gradius* fonctionne de la même manière, bien que parfois je me sente dépassé par le bazar dans l'écran (Note du développeur : c'est le but !). Je sais que ça peut vaguement simuler un voyage intergalactique futuriste contre les légions du mal d'engins mal blindés, où le hasard et les mauvais tours sont bien présents. Mais sérieusement, le gameplay basé sur les réflexes a ses limites. Si vous terminez n'importe lequel de ces ignobles *Shoot'em Ups* en un seul crédit, donnez-vous une tape sur l'épaule et rejouez à *Death Smiles* en mode expert. Je continuerai d'être suffisamment crispé avec un jeu que j'ai mémorisé du début à la fin, mais que je n'ai pas encore fini.

C'est sûrement une préférence personnelle, ou peut-être que je ne suis pas aussi rapide que ces jeux l'exigent. Quoiqu'il en soit, j'ai une certaine tolérance avant l'overdose d'échec. Mais bon, Michael Jordan a raté plus de 9000 tirs de panier dans sa vie. C'est pourquoi je remonterai en selle, une fois que j'aurai oublié à cause de qui j'ai perdu, évidemment.



Dans un jeu comme *Mario Bros*, le game designer est celui qui décide :

- Quels sont les différents ennemis du jeu, leurs forces et leurs faiblesses
- Que Mario peut rentrer dans des tuyaux
- Qu'il y a un système de points
- Que Mario peut sauter 5 fois sa taille
- Que Mario peut tuer ses ennemis en leur sautant dessus
- Que les champignons transforment Mario en Super Mario
- Qu'il y a 4 niveaux par monde et 8 mondes
- Qu'il y a un temps limité
- Qu'il n'y a pas de sauvegarde
- Qu'on démarre avec trois vies et qu'à zéro, on recommence le jeu
- Qu'il y a des warp zones
- etc.

Dans un jeu comme *Mario Bros*, le level designer est celui qui décide :

- S'il met des plantes carnivores dans ses tuyaux
- Si son niveau se trouve sur des plateformes au dessus du vide
- Si tel ou tel endroit cache une vie supplémentaire
- Si le niveau est difficile ou non
- Que c'est un niveau à scrolling automatique
- Qu'il y a une étoile suivie d'un groupe d'ennemis en ligne
- Qu'il faut envoyer une carapace sur un bloc au sol pour récolter un champignon
- etc.

Le travail du level designer.

Si le level design est une mise en scène interactive, c'est que le level designer tente de raconter une histoire au joueur, de lui faire ressentir des émotions et de lui faire traverser des épreuves en lui laissant la possibilité de faire des choix et de changer une partie de ce qu'on lui présente de la façon qu'il souhaite. Le level designer doit prévoir ce qu'il veut faire vivre au joueur et doit savoir pourquoi il fait cela. Tout ce que verra et fera le joueur sera prévu par le level designer. En quelque sorte, il montre une situation et propose

plusieurs solutions pour la traverser. Il doit guider le joueur par la main sans que ce dernier ne s'en rende compte et lui lâcher à la fin afin que le joueur ait l'impression d'avoir réussi par lui-même.

Qu'est-ce qui est plaisant pour le joueur ?

- Voir des choses amusantes
- Se sentir fort
- Ressentir des pics d'émotions
- Faire face à un challenge
- Comprendre les règles du jeu et s'adapter
- Évoluer au sein du jeu, en termes de *skill* et de connaissances
- Recevoir une récompense

Tous ces éléments sont entre les mains du level designer.

Que doit savoir le level designer pour bien travailler ?

- Il doit connaître le public ciblé
- Il doit savoir ce qu'il veut montrer et faire vivre
- Il doit penser à donner le maximum de plaisir aux joueurs
- Il doit amuser le joueur, lui proposer un challenge et/ou le tenir en haleine

Quelles sont les choses qu'un level designer ne doit pas faire ?

- Décider de « faire perdre » le joueur grâce au gameplay en le « trahissant » : c'est possible, mais si le public ciblé le souhaite
- Penser à son propre plaisir à créer le niveau sans se soucier de ce que le joueur peut penser

Comment travaille un level designer ?

- Il cherche des idées, seul ou au moyen d'un brainstorming
- Il réfléchit à l'émotion qu'il veut véhiculer et/ou aux gameplays qui seront présents
- Il réfléchit à l'architecture de son niveau
- Il réalise une version « papier » de son niveau, de ses idées de gameplay et des éléments qu'il veut utiliser
- Il teste ses idées de gameplay à l'aide de prototypes (maps de test)
- Il parcourt son niveau dans le moteur de jeu et le teste
- Il y implante ses idées de gameplay et les teste
- Il fait des modifications du gameplay (jusqu'à la perfection)
- Il donne tout au long de sa production des indications spécifiques à son niveau aux artistes chargés de la composante visuelle

Le level designer s'occupe donc énormément du gameplay. Mais, qu'est-ce que le gameplay ?

Le gameplay.

Dans le fond, beaucoup de gens ne savent pas vraiment ce qu'est le gameplay. Il est souvent confondu avec mécanique de jeu ou même jouabilité.

Le *gameplay général*, c'est la façon dont se joue le jeu, ce qu'il faut y faire et comment le faire. Il est principalement lié au travail de game design.

Le *gameplay spécifique*, c'est le trio « but, challenge et récompense » :

- Le but, c'est ce que le joueur cherche à réussir, l'endroit où il doit se rendre. Il intervient à la fin d'un gameplay spécifique.
- Le challenge, c'est la difficulté mise en œuvre pour empêcher le joueur d'atteindre le but. Il doit apparaître le plus clairement possible aux yeux du joueur pour que ce dernier ne se sente pas trahi. Il doit aussi être à la hauteur de ce qu'attend le joueur et représenter un défi intéressant, non redondant et relatif au but.
- la récompense est donnée une fois le but atteint et doit être à la hauteur du challenge. Elle ne doit pas toujours être la même, sinon le joueur ne se sentira plus récompensé.

Le gameplay spécifique est principalement lié au travail de level design.

Prenons l'exemple de *F-Zero X* sur Nintendo 64 :



- But : terminer la course en vie et bien placé.
- Challenge : traverser les obstacles de la piste et dépasser ses concurrents.
- Récompense : des points pour le classement général suivant la place d'arrivée.

Les écueils du level design

Le level designer doit donc intégrer ce genre de schéma dans son niveau. Il peut le faire plusieurs fois (c'est même conseillé) et peut entrecroiser différents schémas de gameplay spécifique tant que le joueur comprend ce qu'il fait. Le level designer devra rendre ces gameplays intéressants et amusants. Le simple fait d'avoir une bonne idée de gameplay ne fait pas tout : il y a des idées excellentes sur le papier qui se révèlent mauvaises « in game » parce que le level designer n'a pas su y intégrer la notion de plaisir.

Si le joueur se sent frustré ou blasé, c'est que le gameplay est **mal équilibré**. Ceci peut avoir plusieurs causes :

- Le but n'est pas clairement défini
- Le challenge est trop difficile par rapport à la récompense obtenue
- Le challenge est trop simple par rapport à la récompense obtenue
- Le challenge n'est pas clair et induit le joueur en erreur (ou, pire, n'est même pas remarqué par le joueur)
- Le challenge n'est pas amusant

Par exemple, un joueur peut être énervé par une énigme très complexe qui ne mène qu'à une simple salle ou par un passage de plateformes difficile où le chemin n'est pas clairement indiqué. A l'inverse, un boss trop facile à battre et qui donne trop de points d'expériences risque de l'ennuyer.

La plupart des joueurs ne veulent pas sentir que le jeu est trop simple, mais veulent sentir qu'ils sont bons. Le joueur n'est pas dupe et verra la différence. Le level designer ne doit absolument pas tenter de blesser le joueur, c'est-à-dire qu'il ne doit pas s'attaquer à son estime de lui-même ou de ses aptitudes. En général, c'est une très mauvaise chose. Un pic de difficulté soudain et non justifié au milieu d'un jeu est tout simplement

mal conçu. Il ne faut jamais oublier que le gameplay est là pour amuser le joueur et non le level designer. Il ne doit pas oublier qu'il crée un *jeu* et que l'essence même du jeu est d'avoir du plaisir.

[Lire la suite →](#)

Éléments de level design II

Par [Pierre Guyot](#) | le 1 janvier 2012 22:21:16 | Catégories : [Cours](#), [Game Design](#), [Level Design](#)

Cet article est une re-publication.

L'[article original](#) est écrit par Relm publié par E-magination (avril 2008).

Les « ingrédients » sont sûrement l'une des choses les plus importantes dans le level design. Ils doivent être utilisés de façon réfléchie, de la même façon qu'une recette de cuisine : à la base, tous les ingrédients sont bons, mais si vous les utilisez mal, la préparation sera mauvaise.



Les différents ingrédients du level design.

Quels sont les ingrédients du level design ?

- Les ennemis
- Les Personnages Non Jouables (PNJ)
- Les boss
- Les obstacles
- Les éléments de décor interactifs
- Les points de sauvegarde
- Les *power up*
- Les armes
- Les objets à collectionner (*collectibles*)

C'est la totalité des éléments liés au gameplay qu'un level designer a à sa disposition.

Les ennemis.

Les ennemis servent à ralentir la progression du joueur. Il doit être facile de distinguer un ennemi d'un PNJ inoffensif et un type d'ennemi d'un autre. Chaque ennemi doit posséder ses forces, faiblesses et compétences propres.

Les PNJ.

[En anglais, on parlera de *Non Playable Character*, soit NPC.]

Il peut s'agir d'alliés du joueur, d'arbitres, de vendeurs, d'otages, etc. Ils peuvent donner des infos ou simplement contribuer à une certaine ambiance (figurants). Ils ne porteront pas atteinte à la vie du joueur. Le joueur ne peut pas les contrôler. Un PNJ ne peut pas être un ennemi, mais il peut être un adversaire. La différence entre les deux réside dans l'affrontement : un combat contre un adversaire se joue souvent en un contre un.

Les boss.

Le boss fait office d'examen sur le gameplay et les techniques de jeu que le joueur a dû apprendre jusqu'à présent. Plus difficile à vaincre qu'un ennemi normal, il illustre la finalité d'une partie du jeu.

Les obstacles.

Ils empêchent le joueur d'avancer librement en le bloquant (un mur) ou en le tuant (un trou). Ils sont non-interactifs. Il s'agit souvent d'un élément du décor ou justifié par le décor. Le plus souvent, un obstacle est immobile ou ne réalise que des mouvements simples.

Les éléments de décor interactifs.

Tout ce qui est décoratif à la base, mais avec lequel le joueur peut interagir. Un décor interactif doit être différentiable d'un décor non-interactif. Il peut devenir aussi un obstacle suite à une interaction avec le joueur. Par exemple, dans *Mario Bros*, les blocs « ? » sont à la base des éléments de décor interactifs, mais une fois que le joueur a tapé dedans, ils deviennent de simple obstacles.

Les points de sauvegarde.

Ils permettent au joueur de sauvegarder sa progression momentanément ou de façon permanente. Ils doivent être utilisés entre chaque partie d'un niveau suivant la façon dont la difficulté varie au cours de celui-ci. Ils servent à ne pas frustrer le joueur inutilement et à lui permettre d'arriver à son but plus facilement.

Les power up.

Munitions, vies, bonus d'invincibilité, trousse de soin, etc. Ils aident le joueur sur son chemin et le récompensent. Ils doivent apparaître suivant un rythme logique et pertinent. Ils peuvent être permanents ou temporaires. Ils doivent être différentiables du décor. Ils doivent être cohérents avec leurs attributs (le joueur doit comprendre l'effet d'un *power up* rien qu'en le regardant).

Les armes.

Elles augmentent la force du joueur, lui offrent de nouvelles possibilités. Elles sont permanentes (une grenade dans un jeu est un *power up* et non une arme).

Les objets à collectionner (collectibles).

Il s'agit d'objets sans utilité première, qui doivent être collectionnés pour obtenir une récompense. Par exemple, une vie pour cent pièces dans *Mario Bros*. Leur collecte peut être facultative. Ils permettent aussi de guider le joueur en formant des directrices (nous en reparlerons dans la suite) en faisant office d'appât. Ils ajoutent aussi à la *replay value*. Ils doivent être facilement différentiables d'une arme ou d'un *power up* (pour cela, on peut jouer efficacement sur les quantités disponibles dans le niveau).

C'est tout ? Oh que non ! Après ce chapitre un peu technique, parlons un peu de l'utilisation concrète de ces divers éléments dans un niveau de jeu. L'intégration de ces « ingrédients » dans un jeu est sûrement l'une des choses les plus importantes. La façon dont le level designer les placera dans le niveau aura une influence décisive sur la **compréhension du joueur**. C'est quelque chose qu'il ne faut pas sous-estimer. Il faut se rappeler que l'on invite le joueur dans un monde qu'il ne connaît pas et qu'il faut donc lui apprendre les

choses comme à un « nouveau né ». Le joueur n'est jamais bête : ça peut être dur à admettre, mais si un joueur ne comprend pas un élément du jeu, c'est que **le travail de level design a été mal fait**.

Le but n'est pas de montrer au joueur qu'on peut le perdre... C'est tout le contraire !

Directrices, repères, modules et appâts.

Je vais présenter ici quatre outils très importants pour un level designer. Judicieusement utilisés, ils permettent au joueur de ne pas se sentir perdu dans un univers inconnu ou frustré par la malchance. Le joueur, rappelez-vous, doit se sentir récompensé, fort, mais surtout pas frustré.

Dans un jeu en 3D avec une caméra mobile, il est bien plus simple de se perdre que dans un jeu 2D, mais les principes restent les mêmes. Vous vous êtes déjà sûrement dits devant un jeu « Je ne sais plus par où je suis venu » ou « Je ne sais pas où je vais, ni pourquoi ». Quand un joueur un minimum attentif ressent cela, c'est le plus souvent le résultat d'un level design bancal. Il faut savoir qu'il est difficile d'éviter cet écueil et de créer un level design logique et cohérent pour le joueur.

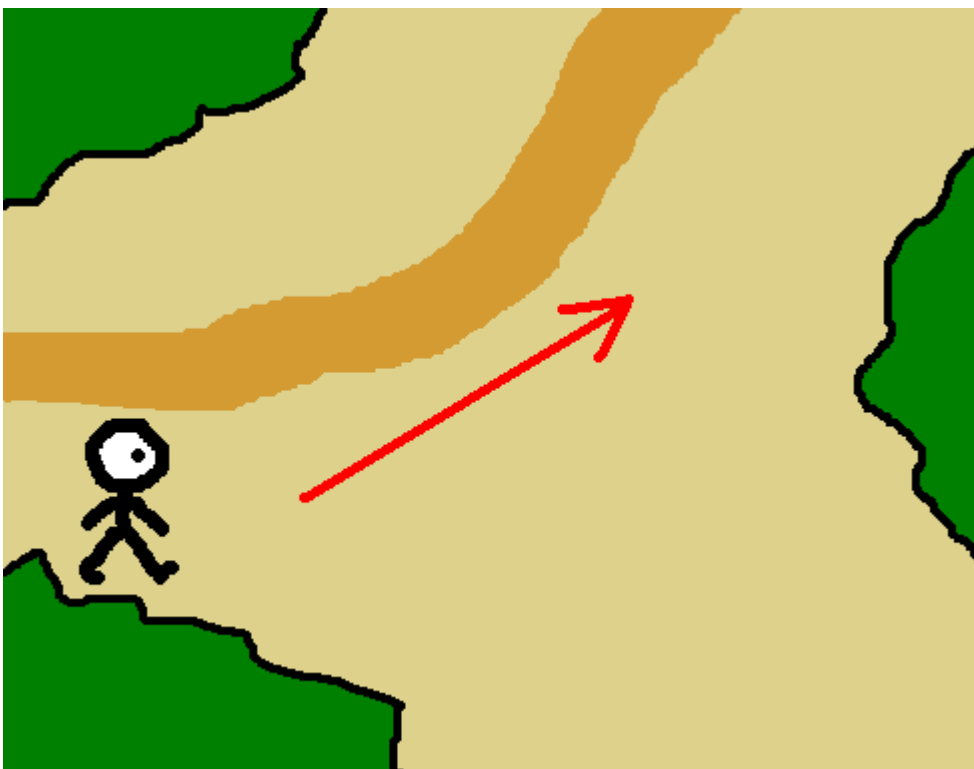
Mais commençons par les bases !

Directrices.

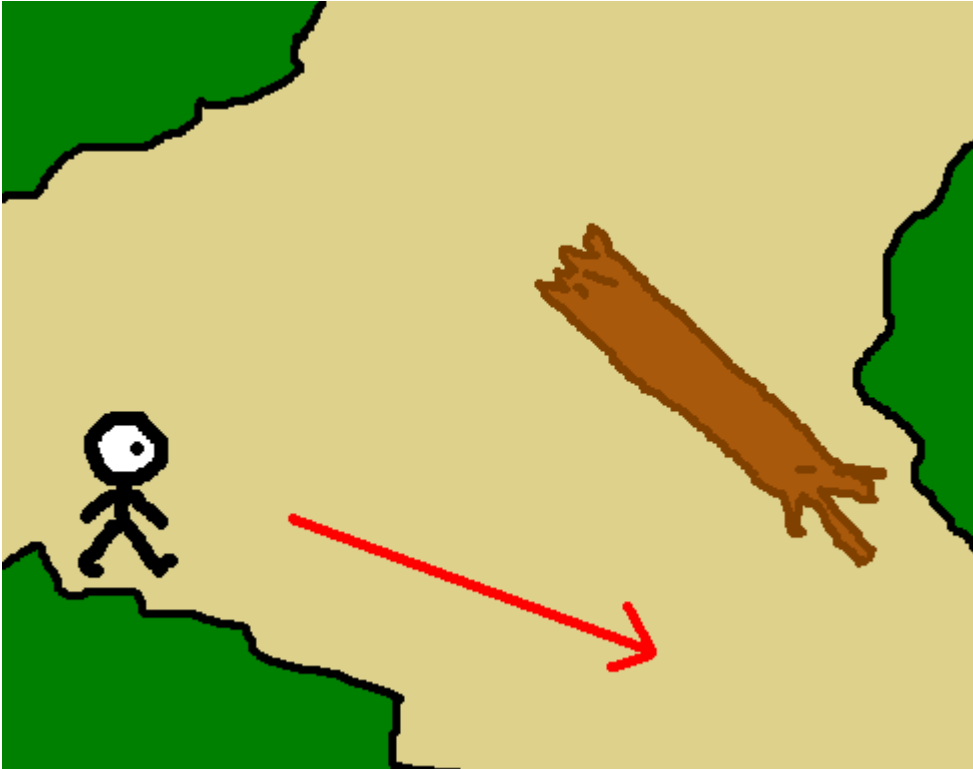
Une **directrice** est une ligne qui dirige le joueur là où le level designer souhaite l'amener. Il peut s'agir :

- D'un décor : un chemin de terre sur le sol, des fils électriques dans une rue ;
- D'un obstacle : un tronc d'arbre en travers de la route.

La directrice n'est pas quelque chose qui oblige le joueur à la suivre, c'est un élément discret qui guide le joueur sans qu'il ne s'en rende compte. Il est possible de concevoir un labyrinthe où un joueur trouvera plus facilement la sortie en utilisant des directrices. Il aura eu l'impression d'avoir eu de la chance ou simplement d'avoir eu du flair. Il peut aussi remarquer et comprendre la présence d'une directrice et s'en servir volontairement.



Dans ce plan en vue de haut, un chemin est dessiné sur le sol. Intuitivement le joueur va être guidé vers le passage au nord.



Même plan, mais cette fois-ci, un tronc d'arbre est couché sur le sol. La trajectoire du joueur va être infléchi par le tronc et aller vers le bas. Intuitivement le joueur va être guidé vers le passage au Sud.

Bien sûr, il ne s'agit pas d'ingrédients magiques : cela ne marchera pas si vous ne les utilisez pas correctement. Mais c'est une bonne base de travail. L'effet produit dépend aussi du joueur. Certains joueurs ont tendance à se dire : « le chemin semble être par là, je vais donc aller de l'autre côté avant pour voir ». Cela ne signifie pas que c'est une faille, mais justement que ça marche tout à fait ! Le joueur inattentif ou *casual* trouvera vite le chemin de l'arrivée et ne se sentira pas perdu, alors que le joueur attentif ou *hardcore* trouvera un chemin secondaire offrant par exemple un bonus et se sentira récompensé de ne pas avoir pris le chemin le plus court.

Vous pouvez aussi utiliser l'exemple à l'inverse si votre but est :

- De faire découvrir au joueur un endroit « caché » plus facilement ;
- De perdre le joueur (à utiliser intelligemment).

Repères.

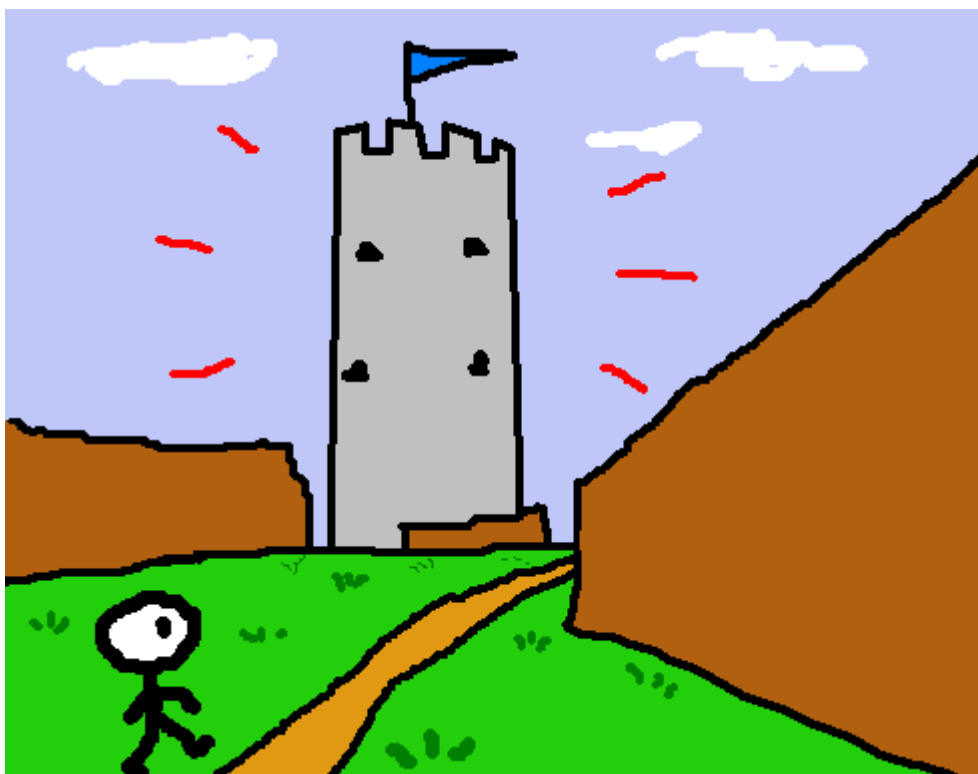
Un **repère** est un élément qui attire le regard du joueur. Il lui permet aussi de se repérer dans l'espace. Il peut s'agir :

- D'un (gros) objet ;
- D'un contraste visuel (lumière, mouvement, etc.).

Le repère est quelque chose qui va toujours sauter aux yeux du joueur, contrairement à une directrice. Cela permet au level designer de donner au joueur une idée de sa position ou de guider son regard vers un autre élément.



Dans ce plan, la torche émet de la lumière, ce qui crée un contraste avec la pénombre naturelle de la grotte. Le regard du joueur qui arrive est immédiatement attiré par ce contraste, et donc par le coffre. Le repère a servi à faire remarquer ce coffre au joueur. Cela aurait aussi pu être quelque chose de plus subtil, comme une fissure dans le mur (si l'on imagine que, comme dans un *Zelda*, le joueur peut creuser un tunnel dans un mur fissuré).



Dans ce plan, le joueur se voit proposer plusieurs chemins, mais il voit depuis n'importe quelle position la grande tour au milieu de la zone. Il peut donc se repérer dans l'espace rapidement. Ce genre de repère est très utile dans les maps multijoueurs dans les FPS par exemple. Cette tour peut aussi être le « but », l'endroit où doit aller le joueur. On lui aura dit auparavant de se rendre dans la grande tour et dès qu'elle sera dans son champ de vision, le joueur la remarquera et comprendra son but.

Un repère peut aussi fonctionner comme une directrice. Imaginez une rangée de casiers fermés à l'exception d'un seul (contraste de forme) : le joueur sera tenté d'aller regarder de plus près le casier ouvert.

Un repère peut aussi servir à attirer le regard du joueur afin d'activer un élément de gameplay : par exemple, imaginons un *survival horror* où le joueur arrive dans une salle avec plusieurs fenêtres. Sur une seule d'entre elles, les rideaux flottent légèrement (contraste de mouvement). Le regard du joueur sera directement attiré par cette fenêtre. Le level designer peut décider de faire apparaître un monstre à ce moment-là. Il laissera quelques secondes au joueur pour remarquer le repère, puis :

- il fera apparaître un monstre de derrière les rideaux (s'il veut aider le joueur) ;
- il fera apparaître le monstre dans la direction opposée (s'il veut surprendre et piéger le joueur).

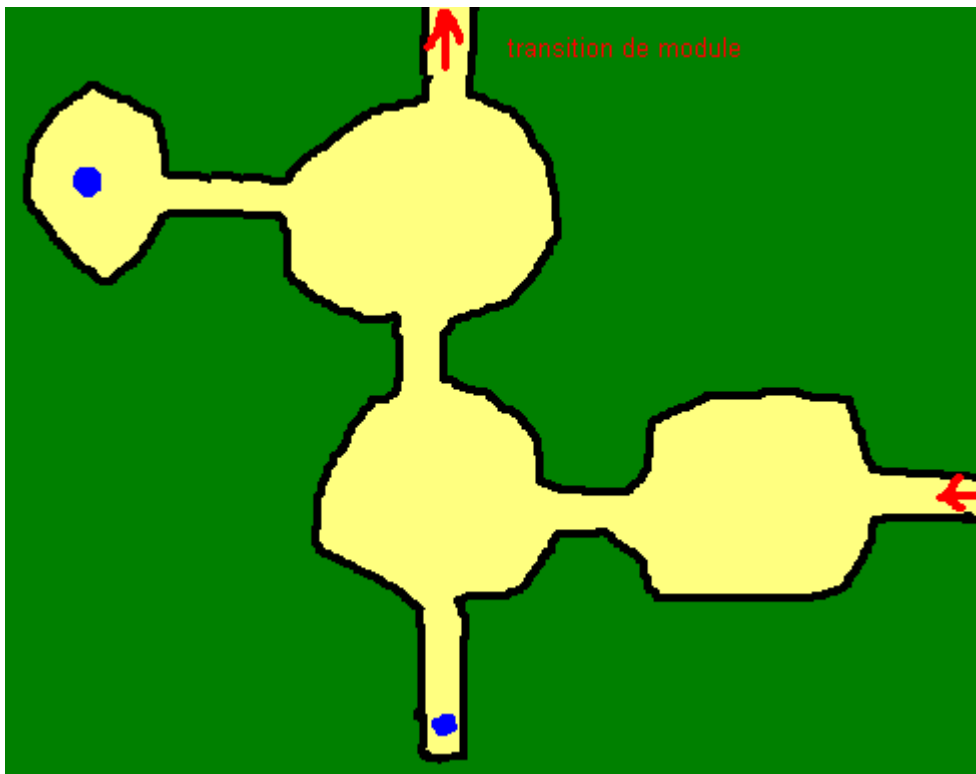
Modules.

Un **module** est une transition vers la suite du jeu. Il sert à informer le joueur qu'il va passer dans un nouvel endroit. Techniquement, il s'agit d'une map différente dans le moteur du jeu.



Dans *Resident Evil*, les portes sont pratiquement toutes des modules : le joueur sait qu'en y pénétrant, il passe à une nouvelle map et que, par exemple, des zombies ne peuvent plus le suivre.

Ici, le joueur sait qu'il peut visiter la pièce pour y trouver des objets ou tuer des zombies et il sait aussi que s'il emprunte la porte bleue à gauche, il arrivera dans une nouvelle zone. Cela permet au joueur de ne pas se sentir frustré. Car il nous est tous arrivé au moins une fois d'avancer tranquillement et de se rendre compte soudainement d'un changement de map, voire du début d'une séquence de boss.



Sur cette image, on peut voir une vue de haut d'un niveau. C'est une forêt avec plusieurs chemins et au nord se trouve un village. Chaque « salle » est grande et les points bleus sont des trésors.

Le joueur peut en explorant rater les chemins qui mènent aux trésors et arriver sans s'en rendre compte à la fin du niveau. Il touchera alors le bord de l'écran et sera téléporté dans le village où une scène se lance et dure une vingtaine de minutes : c'est possible et très énervant. En voyant cela, le joueur se dit : « J'aurais voulu voir ce que j'avais raté dans la forêt ».

Pour éviter cet écueil, on peut :

- Expliquer en montrant au joueur dès le début du jeu quels sont les moyens de sortir d'une map (comme les portes dans *Resident Evil*) ;
- Proposer une mini-map ;
- Utiliser des décors différents à l'approche de la transition (dans notre exemple du village, un sentier de terre qui se change en route pavée).

Appâts.

Un **appât** est un élément qui, contrairement aux précédents, fait partie intégrante du gameplay. Il peut donc s'agir d'ennemis, de PNJ, de bonus, etc. Il fonctionne souvent comme un repère. Un appât est là pour attirer le joueur et lui faire prendre des décisions, en allant parfois même à l'encontre de l'action des outils précédents.

Imaginons une scène dans un jeu comme *Mario Bros*. Le joueur fait apparaître une vie supplémentaire en frappant par inadvertance invisible et cette vie se met à avancer le long du chemin. Le joueur tentera de l'attraper, quitte à mettre sa vie à péril.



Les appâts peuvent aussi servir de directrices : des ennemis en ligne dans un FPS servent à guider le joueur dans un labyrinthe.

Voilà ! J'espère que cet article pourra vous être utile et si vous avez des questions, n'hésitez pas à les poser !

Making of : un arbre pour Lije

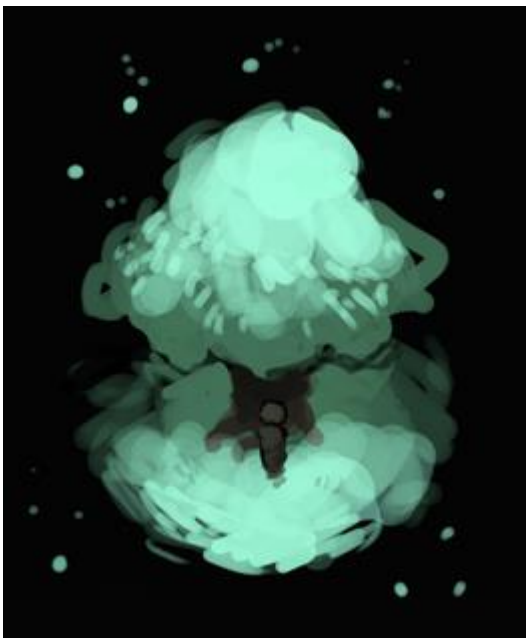
Par [Pierre Guyot](#) | le 10 janvier 2012 03:41:16 | Catégories : [Graphisme](#), [Pixel art](#)

Suite à des remarques qu'on nous a faites sur le site, j'ai pensé que cela pourrait être une bonne idée de présenter des making of pour illustrer les articles que nous avons écrits sur le graphisme. Il ne s'agit pas bien sûr d'exemples à imiter, mais de présenter une façon de travailler et d'essayer de mettre le doigt sur quelques éléments de réflexion. Si vous avez des remarques, critiques ou commentaires, je suis bien évidemment tout ouïe !

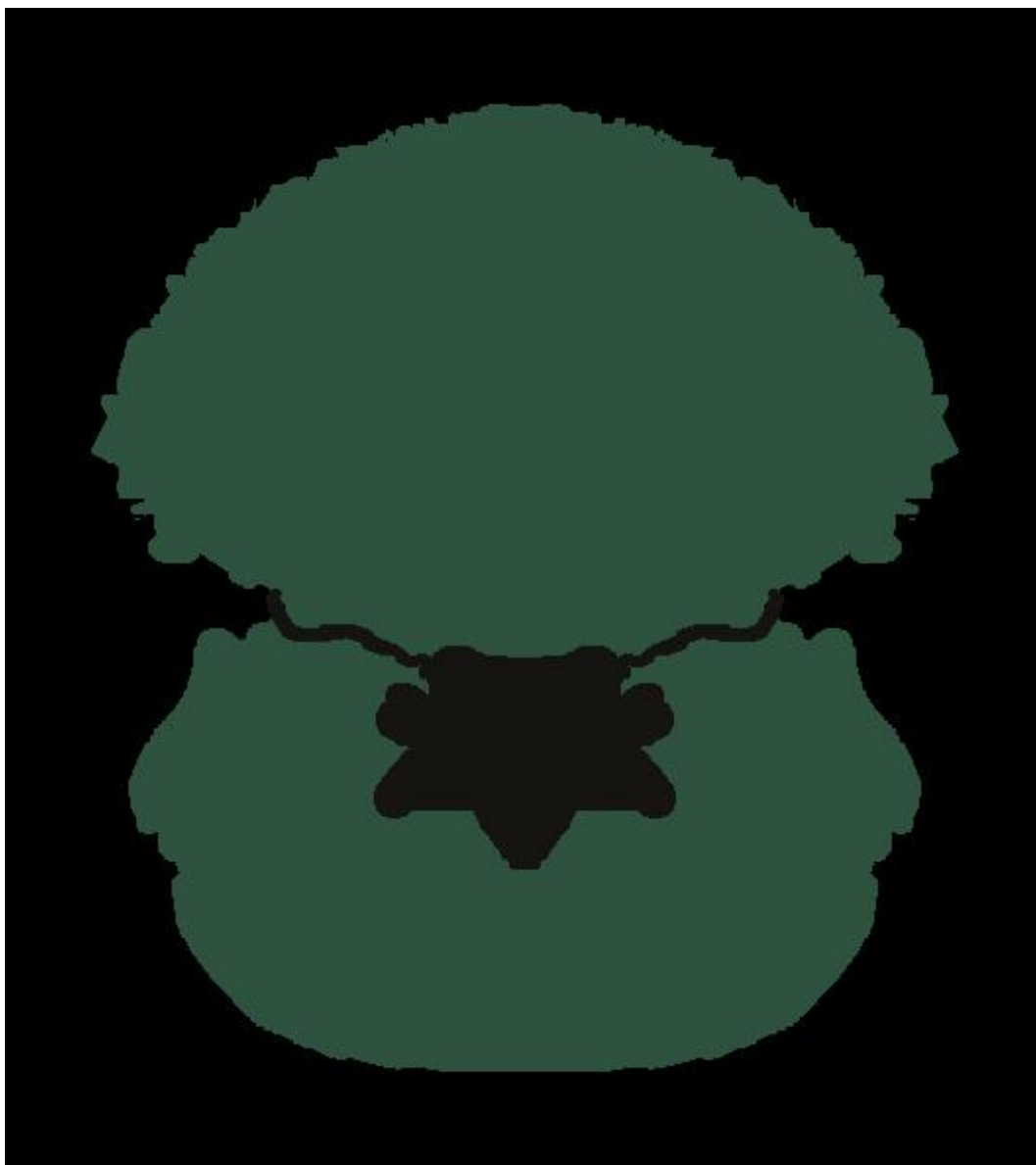
Lije

Pour ce premier essai, j'ai choisi de vous présenter le making of d'un arbre qui apparaît dans le prologue de [Lije](#), pour lequel je suis responsable des décors. Il s'agit de pixel art, bien évidemment, mais plutôt grande résolution, où j'essaye de travailler au maximum en à-plats (ce qui exclut par exemple le dithering).

Pour commencer, j'ai réalisé un rapide concept art pour avoir une idée de là où je voulais aller, en matière d'ambiances, de formes et de couleurs.

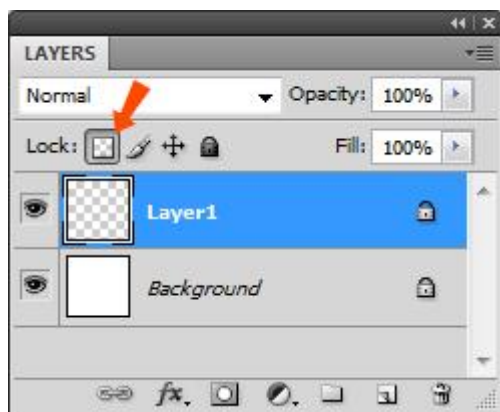


C'est très simple, voire grossier, mais ça me donne une bonne idée de ce à quoi l'écran ressemblera une fois le jeu lancé. J'ai une idée de la palette que je vais utiliser et de la forme des éléments.

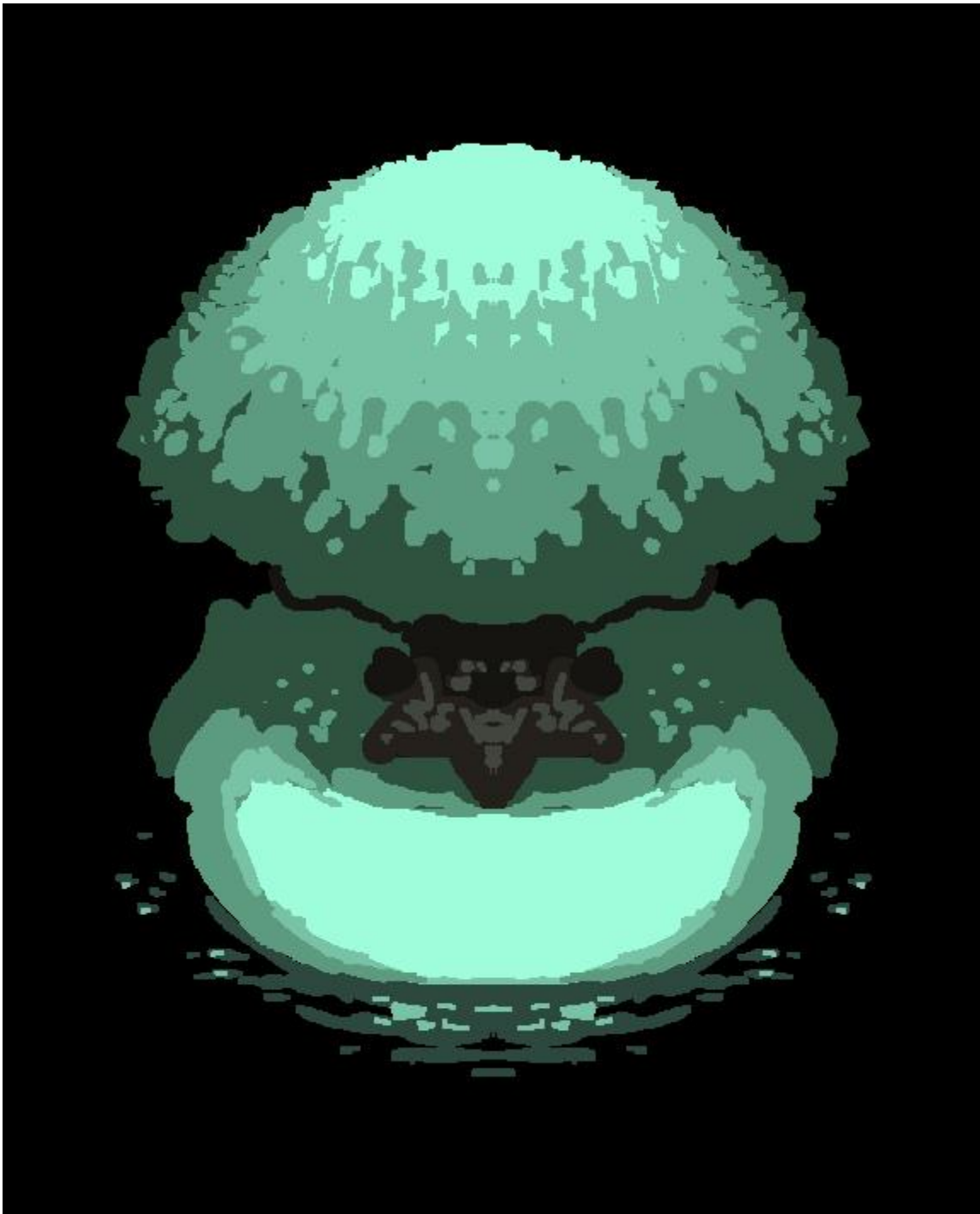


Je travaille sur un fond noir car l'arbre apparaîtra sur un fond noir dans le jeu : je peux donc avoir une meilleure idée du rendu final et je peux anti-aliaser de l'arbre vers le fond sans risque. En général, j'essaie de travailler avec la couleur du fond telle qu'elle sera dans le jeu et si jamais l'élément peut apparaître différents backgrounds, je travaille sur un fond gris neutre (#808080) et je n'anti-alias pas.

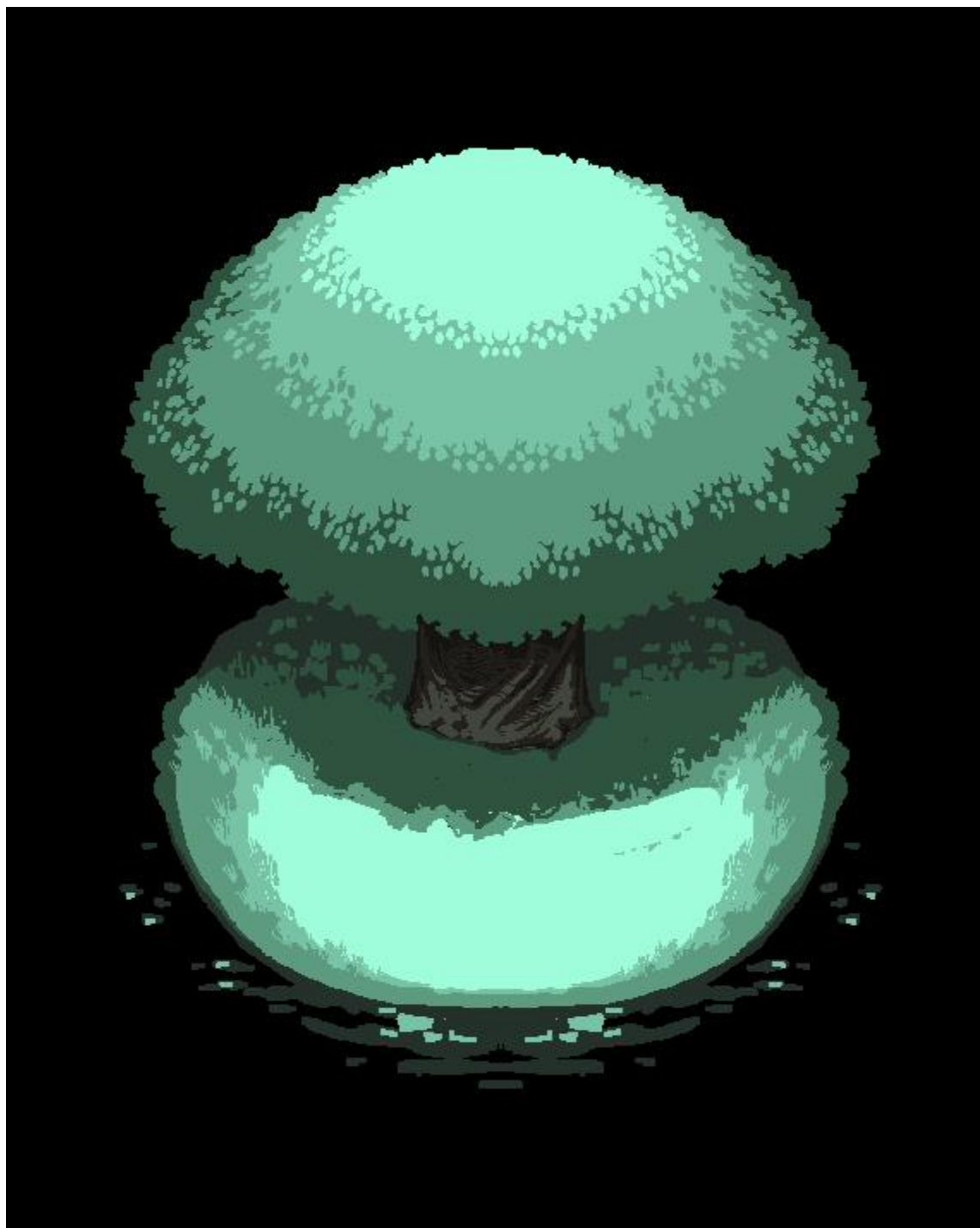
La première chose que je fais, c'est de poser les à-plats les plus importants. Ici : le fond, le feuillage, l'herbe, le tronc. J'essaie d'être assez propre, pour pouvoir sélectionner les différentes zones et travailler dessus séparément, sans « déborder ». On peut utiliser l'outil Baguette Magique pour sélectionner un à-plat ou même séparer les à-plats en différents calques verrouillés. Je rappelle au passage l'intérêt du verrouillage pour ne pas déborder, c'est vraiment très utile.



Pour poser les à-plats, je place le concept art, redimensionné pour coller avec l'échelle des personnages, à faible opacité (en général 30%) et je travaille sur un autre calque vide, placé au-dessus. De cette façon, le concept art me sert de guide : c'est particulièrement utile si votre concept de base est précis.



Je me sers ensuite des à-plats (verrouillés ou sélectionnés) pour poser grossièrement les ombres et les lumières. J'ai choisi une lumière zénithale, censée conférer une note sacrée à la scène. J'essaye de ne pas faire de *pillow shading* et de donner, dès le départ, un bon volume à l'arbre. Je n'ajoute pas trop de détails pour l'instant et je travaille avec une grosse brosse en mode pixel, avec variation de taille mais pas d'opacité (essentiel pour le pixel art !).

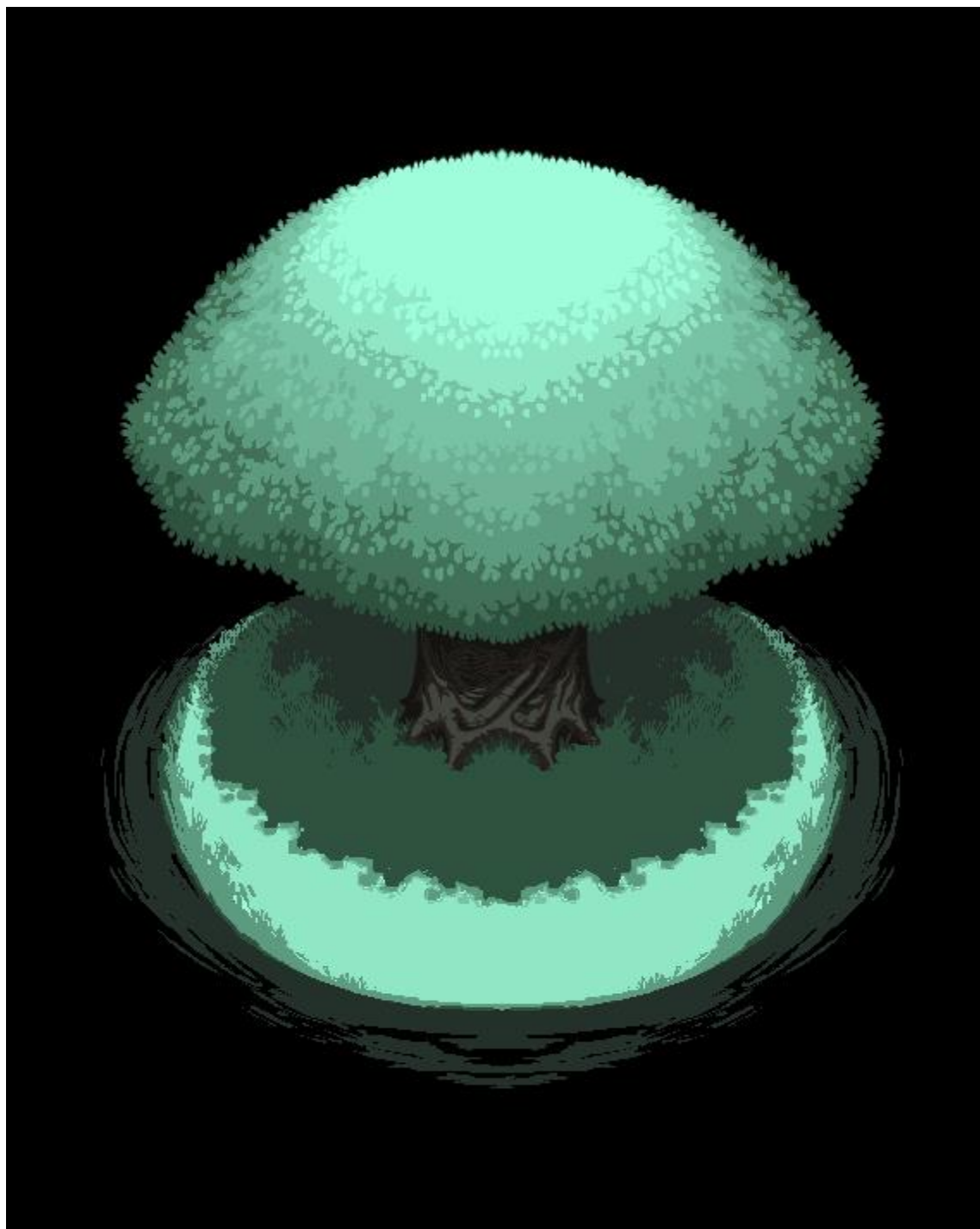


Je me permets maintenant de déborder des calques. Je commence à travailler à la brosse taille 1 pixel, c'est-à-dire que je commence à vraiment faire du pixel art. Je veille à bien conserver les grands à-plats de couleur et je travaille surtout sur les transitions entre les zones : je trace le feuillage, en essayant de rendre les feuilles par une découpe des zones d'ombre, je commence à créer une texture d'herbe en « grattant », je redessine le tronc (qui faisait trop *A Link To The Past* à mon goût) et je lui donne une texture (à la fois avec des à-plats dirigés pour rendre le mouvement de torsion du tronc et en grattant pour rendre le côté rugueux de l'écorce). Remarquez que j'utilise la symétrie sans aucun remords, ça me fait gagner un temps fou.

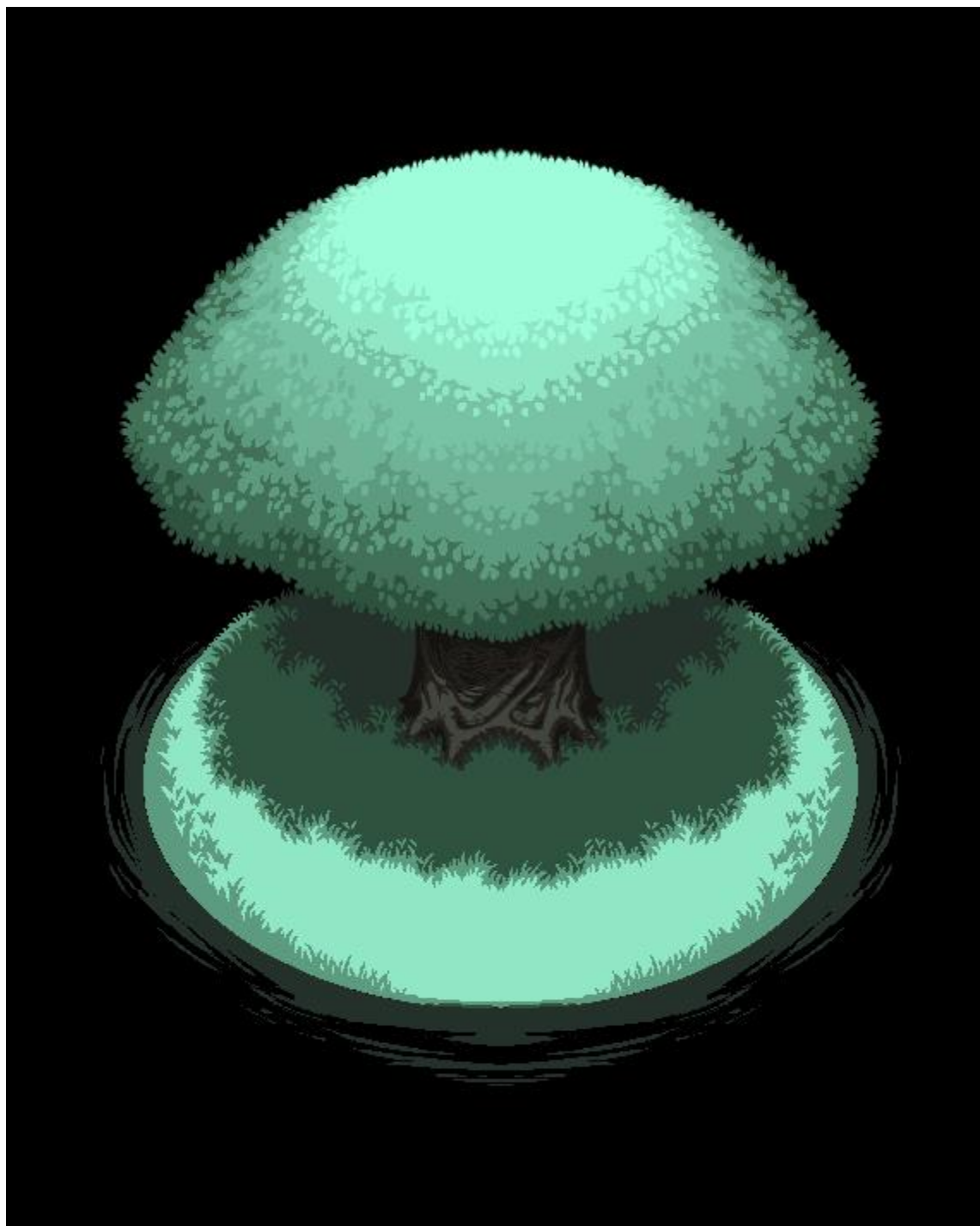
Vous noterez que j'ai rajouté une zone d'ombre supplémentaire sous l'arbre. J'ai repris pour cela la couleur du reflet sur le fond noir : dans la mesure du possible, j'essaye de réutiliser mes couleurs à plusieurs endroits différents et pas seulement d'utiliser une rampe pour chaque élément. Cela me permet de mieux « marier » les différentes couleurs.



Je rajoute des couleurs de transition à ma palette, essentiellement des verts. Je les utilise pour adoucir le rendu de l'arbre avec de nouvelles « couches » de feuilles et je compte m'en servir plus tard pour faire de l'anti-aliasing. J'ai essayé de mettre des racines sous l'arbre et je supprime quelques taches de couleur pour agrandir les à-plats.



En général, c'est le moment où je demande un avis extérieur (et même plusieurs) pour prendre un peu de recul (et c'est aussi l'occasion de faire une pause !). Avec ces conseils en poche et un peu plus de distance vis-à-vis de mon travail, je regarde mieux et je constate que plusieurs choses ne vont pas. Je reprends donc les affreuses racines que j'avais faites (je suis tombé pile poil dans les travers décrits par le [cours de Fil](#)), je redessine la forme de l'île, je retrace l'ombre portée de l'arbre et j'en profite pour dessiner proprement les vaguelettes autour de l'île. Bref, je remets ça au propre. Au passage, je raffine le contour du feuillage de l'arbre et je l'anti-alias, de façon à ce que la transition vers le fond soit plus fine.



Je dessine maintenant proprement les brins d'herbe. Je ne veux pas de texture, donc je me contente de les dessiner à la transition, brin par brin. J'y vais proprement, à la brosse taille 1 pixel, car je veux que ces contours soient définitifs. Je soigne particulièrement la façon dont les brins d'herbes passent devant le tronc d'arbre et les racines, de façon à montrer que l'arbre est bien planté dans le sol et pas juste « posé » par dessus.



Pour finir, je peaufine les transitions avec de l'anti-aliasing : j'en applique sur les différents brins d'herbe et je dessine des brins sur le bord de l'île, pour qu'ils dépassent sur le fond et donnent une impression de superposition plutôt qu'une impression de juxtaposition. Je finis d'anti-aliaser le bord de l'île et c'est bon !



Je me suis arrêté ici pour des raisons de temps (la démo de *Lije* devant sortir pour les [Alex d'Or](#)), c'est pourquoi je n'ai pas plus poussé le rendu du feuillage. Je sais qu'il ne s'agit pas exactement de pixel art (dans la mesure où la résolution est très grande), mais j'ai essayé d'en utiliser les techniques : palette limitée (12 couleurs), anti-aliasing à la main, à-plats, etc. Voici donc une idée de comment j'ai réalisé ce bout de décor. Ce n'est pas forcément très ordonné et le résultat pas forcément exceptionnel, mais j'espère que cela illustre un peu mieux les cours de pixel art du site !

Game Design & Dark Souls

Par [Antoine Gersant](#) | le 19 janvier 2012 20:07:43 | Catégories : [Analyse](#), [Game Design](#), [Level Design](#), [Ma vie](#)

Dans mon article de juin dernier sur le [game design des MOBA](#), j'ai déclaré ne plus prendre beaucoup de temps pour jouer à des jeux vidéo ; il y a effectivement une quantité assez sidérante de jeux pas déballés ou pas essayés sur mes étagères. Mais je me soigne : au fil des derniers mois, j'ai redécouvert par petites doses les mille et un plaisirs du rôle de joueur et je suis venu à bout de pas mal de jeux que j'avais l'intention de faire. Il m'en reste des wagons mais je progresse. Ouf.



Dark Souls est un jeu développé par *From Software*, sorti en 2011 sur Xbox 360 et PS3. Si on voulait enfermer le jeu dans des boîtes, on dirait que c'est un RPG d'aventure à la troisième personne en temps réel. Essayez d'imaginer un jeu à mi-chemin entre *The Witcher* et *Shadow of the Colossus* et le résultat ne sera pas trop loin de l'idée générale, sauf si vous êtes en train d'imaginer « un type avec les cheveux blancs dans un jeu où il faut draguer des monstres géants ».

Je ne sais plus exactement pourquoi j'ai acheté ce jeu, il me semble que les deux principales raisons étaient que les gens en parlaient beaucoup sur le forum de [TIGSource](#) et aussi la vue de ce trailer curieux :

A vrai dire, le jeu démarrait assez mal de mon point de vue. D'abord, il y avait des dragons dans l'introduction, ce qui est très souvent signe de mauvais goût et de clichés tolkienesques. Ensuite, on commence en prison comme dans un *Elder Scrolls* (une série que je ne porte pas franchement dans mon cœur), autant vous dire que j'étais plein d'appréhension et pas vraiment vendu d'avance.

Inutile de vous le cacher plus longtemps, si j'écris cet article, c'est bien parce que mes mauvais pressentiments ont très vite été dissipés. Voilà, j'avoue tout : je pense que *Dark Souls* est un des jeux les plus remarquables de ces dernières années. Heureusement pour moi, je ne suis pas critique de jeu vidéo : je ne vous parlerai donc pas des graphismes sublimes, de la bande son insaisissable ou encore du design exceptionnel de ce jeu (oups, trop tard). Le propos de cet article est d'analyser quelques unes des mécaniques de jeu qui font de *Dark Souls* un jeu si réussi. Je n'ai donc aucune prétention d'objectivité puisque je choisis de me focaliser sur « ce qui fonctionne bien » dans le jeu.

Dark Souls et la difficulté

Dark Souls est le successeur de *Demon's Souls*, lui-même précédé par toute une flopée de jeux *King's Field* développés par *From Software*. Le point commun de tous ces jeux, c'est qu'ils ont la réputation d'être extrêmement difficiles (et je peux attester de la véracité de ces propos en ce qui concerne *King's Field II*). Cette réputation a été le fondement de la campagne marketing autour de *Dark Souls* dont le slogan était *Prepare to Die* et qui montrait un montage de plein de façons horribles de mourir dans le jeu. Ambiance.

Beaucoup de joueurs considèrent *Dark Souls* comme un jeu à la difficulté insurmontable comme on en avait dans les années 80 (que tous ceux qui ont fini *Wizardry* lèvent la main !). Seulement, quand on y réfléchit,

on s'aperçoit que c'est très très faux et que *Dark Souls* se contente d'être un tout petit peu en décalage avec les poncifs du game design des années 2000 et plus. Explication :

Dans *Dark Souls*, le joueur peut mourir en deux ou trois coups de n'importe quel monstre. Dans un jeu moderne comme *God of War* ou *Bayonetta*, mon personnage peut encaisser des effondrements de bâtiments et des éclairs magiques par paquets de cinq avant de succomber. Je pense que les joueurs qui se plaignent que *Dark Souls* est un jeu trop difficile sont ceux qui tente d'y jouer comme ils jouent à *God of War* et *Bayonetta*. Dans ces deux jeux, lorsque l'on combat des ennemis on adopte une posture offensive qui consiste à déployer ses combos les plus puissants et à faire un joli combat chorégraphié que, parfois, on choisit d'interrompre par une esquive ou une parade afin d'éviter un coup porté par un adversaire. Ces mêmes adversaires sont extrêmement lents à lancer leurs attaques (elles sont « téléphonées », une longue animation vous laisse le temps d'anticiper le coup pour l'esquiver) et cet avantage est suffisant pour que la défense fasse figure d'« exception » dans la chorégraphie du joueur.

A l'inverse dans *Dark Souls*, si vous foncez tête baissée dans un affrontement, vous avez de bonnes chances de vous faire tuer en deux secondes. Pour réussir, il faut changer d'attitude (ce qui est très inhabituel et à contre-courant des jeux modernes) et être défensif par défaut plutôt qu'offensif. Dans *Dark Souls* les monstres ne sont pas beaucoup plus lents que vous (en général, vos attaques à vous sont téléphonées aussi) et tel un monstre surnois caché dans un angle mort de caméra de *Bayonetta*, vous devez attendre une occasion de placer un enchaînement sans risquer de prendre un coup de Jarnac qui viendrait ruiner votre barre de vie. Il en résulte que les chorégraphies de combats de *Dark Souls* tiennent plus, et ce dès les premières heures de jeu, de la gavotte que du *pole dance*. Attaques, parades, feintes, esquives — tout s'enchaîne avec un réalisme saisissant (les animations sont très bonnes) et le jeu aurait pu être sous-titré *The Real Medieval Combat Simulator*. *Metal Gear Solid 3: Snake Eater* est un autre exemple de jeu moderne où le système de jeu punit sévèrement les joueurs trop pressés et inattentifs à leur environnement. Ce choix de game design apporte une dimension tactique aux combats sans pour autant sacrifier leur dynamisme, mais a l'inconvénient de dresser une barrière d'entrée qui écarte beaucoup de joueurs incapables de se défaire de leurs habitudes et d'aborder les situations avec sang-froid.

Heureusement dans *Dark Souls*, les développeurs ont pensé à tout et même aux gens pas très doués comme moi. J'étais plutôt mauvais à *Dark Souls*, non pas parce que j'étais pressé mais parce que je manque souvent d'adresse dans les jeux (lève la main si toi aussi tu n'arrives pas à faire les esquives dans *Alan Wake* et les QTE — argh — dans *Castlevania: Lords of Shadows*). Du coup je suis mort très souvent et j'ai pu apprécier dans toute sa finesse la mécanique de la mort dans *Dark Souls*.



Quand je meurs dans un jeu classique, la continuité du jeu est brisée. L'état du jeu (position du personnage, inventaire, etc.) est ramené à un état antérieur (éventuellement avec une pénalité sur ma progression macroscopique : expérience, or ou une autre ressource non critique pour mon succès à court terme) et on me donne une seconde chance pour refaire correctement la séquence que j'ai ratée. Si je multiplie les échecs, cela devient vite frustrant et ça finit souvent sur GameFAQs en quête d'un monde plus juste.

Dans *Dark Souls*, le personnage du joueur est maudit et porte une marque sombre qui fait de lui un non-mort qui renaît éternellement à l'étincelle de lumière (lire *checkpoint*) la plus proche. La conséquence immédiate de cette décision est que lorsque je meurs, le flux du temps n'est pas brisé pour revenir dans le passé et recommencer : je ressuscite et c'est la suite logique de l'aventure. Mourir fait partie du jeu et n'est pas directement relié à la notion de défaite. On ne peut d'ailleurs pas littéralement perdre dans *Dark Souls*. Ce n'est pas pour rien que le message qui apparaît à l'écran est « You died » et non « Game Over » (dans la version bêta présentée sur un salon le message était « You went hollow », ce qui correspondait effectivement à un échec définitif dans le contexte scénaristique du jeu). Je pense que c'est une façon très habile de faire accepter sa mort au joueur et de l'encourager à poursuivre ses efforts.

La seconde conséquence de cette façon de gérer la mort est que si vous avez utilisé des objets consommables (qui sont en petit nombre dans le jeu, à l'exception des potions de soins que l'on récupère gratuitement) avant de trépasser, vous ne les aurez plus à votre disposition pour votre prochaine tentative. A l'inverse, si vous avez ramassé un objet très puissant gardé par un monstre féroce avant qu'il ne vous massacre, vous l'aurez toujours après votre renaissance ! Enfin, le jeu introduit une pénalité qui consiste à laisser sur le lieu de décès toutes les âmes (qui servent pour acheter de l'équipement et des *level up*) en possession du joueur. Il est possible d'annuler la pénalité en ramassant les âmes sans mourir à nouveau entre temps, ce qui permet de les accumuler lorsque l'on bute sur une difficulté : je tue dix monstres, je ramasse mes âmes, je meurs contre le boss, je recommence. Une fois la difficulté passée, on peut aller joyeusement dépenser son tas d'âmes durement gagnées.

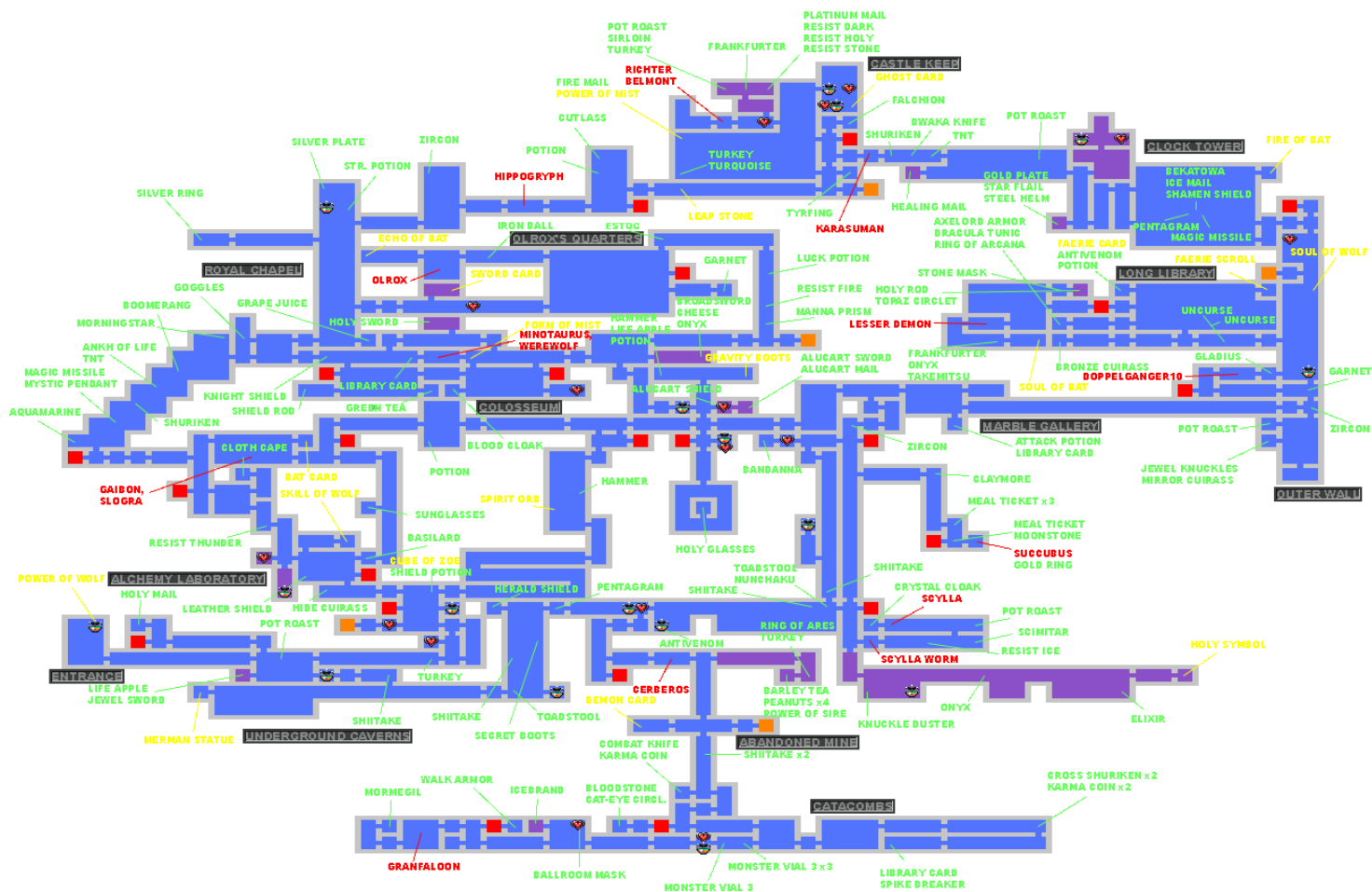
Au final, si vous êtes nul comme moi à *Dark Souls*, vous mourrez beaucoup (jamais sans une bonne raison) mais le game design du jeu fait tout pour vous faire avaler la pilule — ce que ne faisaient pas du tout les jeux des années 80 auxquels certains ont comparé la difficulté de *Dark Souls*.

Metroidvania et mécanique du rendez-vous

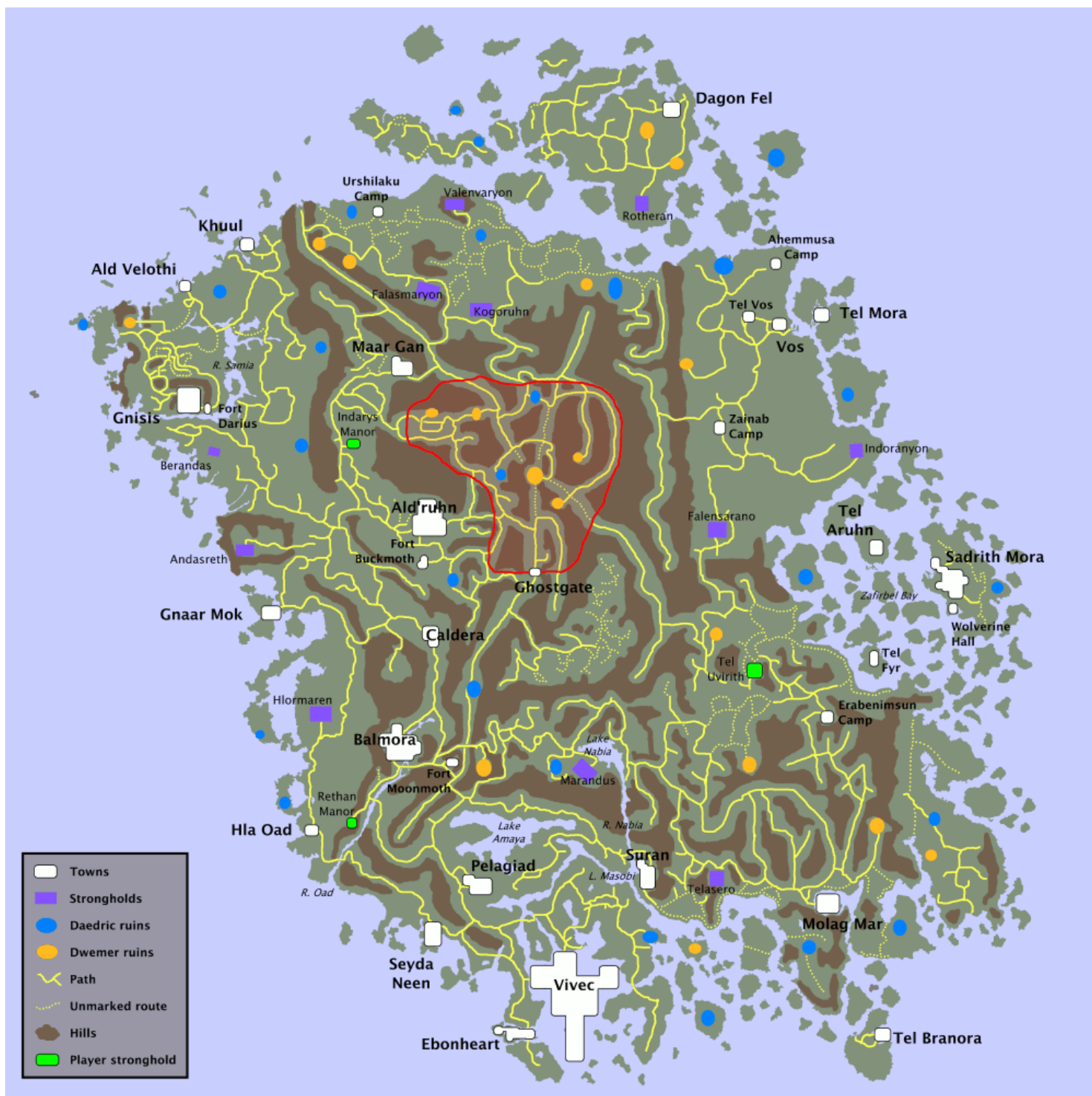
Le principal point fort de *Dark Souls* nous arrive lui aussi tout droit d'une autre décennie. La structure des jeux d'action/aventure qui sortent depuis quelques années repose le plus souvent sur l'un des deux paradigmes suivants :

- Une succession (le plus souvent linéaire) de niveaux essentiellement linéaires (*Bayonetta*, *Final Fantasy XIII*, etc.)
- Un monde ouvert à 360°, « bac à sable » dans lequel le joueur doit se déplacer entre différents points d'intérêt (*GTA III*, *Morrowind*)

Dark Souls fait renaitre une autre structure que je n'avais pas eu la chance d'observer depuis longtemps, celle des *Metroidvania* (savante contraction de *Metroid* et *Castlevania*). La structure d'un *Metroidvania* pourrait être décrite comme une multitude de niveaux essentiellement linéaires et interconnectés. Parce qu'un exemple est toujours plus parlant, voici la carte du château de *Castlevania: Symphony of the Night* sur Playstation. Désolé pour l'absence de source, je ne sais plus où je l'ai piquée :



Appréciez la structure qui alterne les zones linéaires et les embranchements qui redirigent vers l'une ou l'autre de ces zones. A comparer avec par exemple la carte de *Morrowind* (profitez en passant des attractions de *Disneyland* qui se sont cachées au milieu des noms plein de consonnes et qui finissent par « a ») :



Cette différence de structure entraîne des perceptions complètement différentes de la part du joueur. Dans un *Metroidvania*, le joueur tient à jour dans sa tête (ou sur un papier) une liste des embranchements qu'il a délaissés et des chemins bloqués (parce qu'il lui [manque une clé](#)) où il pourra revenir plus tard.

Dans un monde ouvert, cette tâche est le plus souvent effectuée au moyen d'un journal de quêtes (ou équivalent) qui rappelle au joueur tous les lieux où il peut se rendre pour faire avancer l'une ou l'autre des intrigues en cours.

Dans les deux cas, le résultat accompli est ce que j'appelle la mécanique du rendez-vous, qui consiste à donner au joueur un point de repère (dans le temps et l'espace) sur son statut futur dans le jeu. Dire au joueur « plus tard, tu iras là » (ou « plus tard, tu combattras untel ») c'est lui donner une perspective sur le jeu qui le pousse à continuer, pour peu que le rendez-vous ne soit pas d'un ennui mortel du genre « plus tard, tu iras tuer 500 rats ». En plus d'être une source d'intérêt pour le joueur, cette mécanique permet de le faire jouer plus longtemps (ce n'est pas forcément votre but) puisque le meilleur moment pour s'arrêter est celui où le

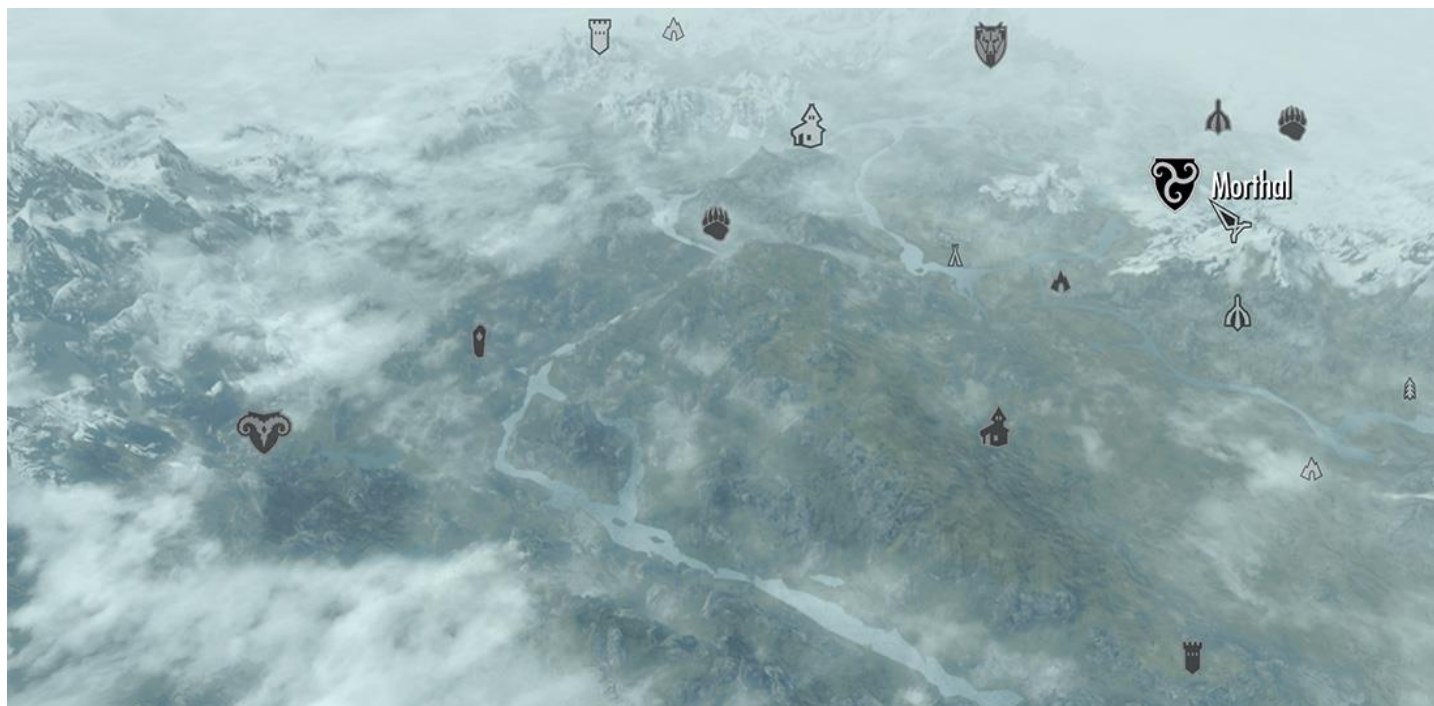
joueur n'a plus de rendez-vous à espérer. Les *Hack-n-Slash* (et les *Roguelike*) les plus addictifs comme *Diablo* ou *Torchlight* reposent sur un rendez-vous permanent avec l'étage suivant du donjon, ce qui pousse beaucoup de joueurs à jouer jusqu'à des heures déraisonnables (NDLR : Antoine en sait quelque chose).

La mécanique du rendez-vous fonctionne mieux dans les *Metroidvania* que dans les jeux ouverts pour deux raisons :

- D'abord, dans un *Metroidvania*, le joueur a une liste de rendez-vous gérée par le concepteur du jeu : elle se remplit pendant une partie du jeu, finit par diminuer au fur et à mesure qu'il brise les verrous et elle est vide quand le joueur termine le jeu. Dans un jeu ouvert, cette liste est gérée par le joueur : un joueur peu curieux qui se déplace d'objectifs en objectifs videra très vite sa liste de rendez-vous initiale et se concentrera sur la sacro-sainte « quête principale » tandis qu'un vagabond errant remplira encore et encore sa liste, souvent plus vite qu'il ne peut la vider. Les deux extrêmes sont très difficiles à équilibrer (surtout pour le joueur, ce n'est pas son métier) et la plupart des joueurs se retrouvent alors dans une situation qui ne leur convient pas. En ce qui me concerne, j'ai collectionné trop de rendez-vous dans la première moitié de *Nier* (j'ai fait le plein de quêtes annexes) et j'ai été victime d'un grand ras-le-bol qui m'a poussé à aller ensuite en ligne droite vers la fin du jeu.
- Ensuite, l'efficacité de la mécanique du rendez-vous repose en bonne partie sur la qualité du rendez-vous. Dans un *Metroidvania*, vos rendez-vous mènent les joueurs vers de nouvelles zones de la carte (qui sont la promesse de nouveaux décors, nouvelles rencontres, nouveaux objets etc.). C'est à peu près tout ce dont ils ont envie. Dans un monde ouvert, la récompense obtenue par le joueur qui atteint un rendez-vous n'est pas une nouvelle aire de jeu, c'est tout au plus un évènement scripté dans une zone qu'il pouvait déjà visiter avant.

Je ne suis pas en train d'essayer de démontrer que les mondes ouverts sont fondamentalement « moins bien » que les *Metroidvania* (même si je le crois très fort) ; je tente plutôt d'explicitier les qualités de game design que la structure de *Metroidvania* confère à *Dark Souls* et qui font défaut à bon nombre de RPG modernes comme les *Elder Scrolls*. Ne me faites pas dire ce que je n'ai pas dit hein, les mondes ouverts ont aussi de bons côtés (dont nous parlerons un autre jour).

Comme pour la difficulté, *Dark Souls* ne se contente pas de recycler de vieilles idées de game design que l'on croyait abandonnées (les *Metroidvania* n'ont pas vraiment survécu au passage à la 3D et ont un peu disparu du paysage ces quinze dernières années, à part quelques rares exceptions comme *Metroid Prime*). Les jeux d'aventure d'une manière générale n'exploitent généralement pas la 3D dans leur level design, ils sont en « 2.5D ». Si vous regardez la capture d'écran de *Skyrim* ci-dessous, vous pouvez remarquer qu'à chaque coordonnée (X, Y) de la carte ne correspond qu'une seule altitude Z explorable (sauf quand parfois il y a une grotte). C'est une structure qui a du sens puisqu'elle correspond assez bien à la réalité en plus d'être facile à concevoir. C'est comme ça que marchent à peu près tous les jeux modernes.



Je vous aurais bien fourni une carte de *Dark Souls* pour comparer mais il n'en existe aucune plus pertinente qu'une projection semblable à la carte de *Castlevania: Symphony of The Night* présentée plus haut. *Dark Souls* est à ma connaissance le seul *Metroidvania* qui utilise autant la 3D dans son level design. L'aire de jeu occupe une surface au sol assez petite mais elle s'étend en hauteur (vers le haut et le bas), non pas linéairement comme un *Diablo* mais avec de véritables embranchements. Le jeu n'inclue d'ailleurs aucune carte (une décision contestable mais qui ne m'a pas gêné outre-mesure) et il est fréquent de se dire « Hmm, ok, je suis ici, je veux aller là bas. Je peux passer par telle zone et telle zone puis prendre tel ascenseur pour y aller ; ou alors je passe par l'échelle là bas, telle zone etc. ».

Les Level Designer ont joué de cette force pour permettre au joueur de débloquent des raccourcis (ascenseurs, échelles, portes fermées) entre des zones qui semblaient jusqu'alors très éloignées. L'effet est assez saisissant d'avancer la peur au ventre dans un sombre couloir inondé avant de se rendre compte qu'on l'a déjà parcouru dans l'autre sens quatre heures plus tôt et qu'il est proche d'une zone connue et rassurante.

L'aventure, c'est pas fait pour les comptables

Bon, on commence à dresser le portrait d'un jeu sympathique avec ses combats intenses, sa difficulté intelligemment réglée et son level design semi-linéaire en vraie 3D. Continuons notre tour d'horizon avec une autre *feature* de *Dark Souls* qui contribue à le rendre si bon.

Une fois encore, il s'agit d'un « retour aux sources » intelligent. Si vous jouez aux RPG depuis plus d'une dizaine d'années, vous n'aurez pas manqué de remarquer que la quantité d'objets que vous ramassez dans ce genre de jeux n'a pas cessé d'augmenter. J'ai l'impression que la tendance a démarré dans les RPG occidentaux qui ont voulu gagner en réalisme en proposant de ramasser tout et n'importe quoi ; ce qui correspond aussi à l'essor des systèmes de *crafting*. L'idée s'est propagée aux RPG japonais : il n'y a qu'à voir la quantité de merdouilles que l'on peut (doit) ramasser dans *Final Fantasy XIII* pour s'en convaincre. Le problème, c'est qu'il est fastidieux pour le joueur de gérer un grand nombre d'objets de faible valeur et de faible utilité. Cela complique la navigation et l'oblige à retenir toute une flopée d'informations peu utiles elles aussi sur la nature des objets.

Dans *Dark Souls*, chaque objet que vous ramassez (ou presque, mais vraiment presque) est un trésor à vos yeux. Les objets que vous voyez briller au loin sont tous potentiellement des anneaux magiques qui vont vous changer la vie, des armes inédites, des armures complètes (on ne trouve jamais une chaussette gauche ou une cote de maille seule) ou encore des clés qui ouvrent l'accès à des secrets du jeu. C'est tout le

contraire d'un *Torchlight* où le *loot* est tellement banalisé qu'au bout d'un moment on ne prend même plus la peine d'identifier les objets magiques les plus faibles. D'ailleurs dans *Dark Souls* vous ne ramasserez jamais (vraiment jamais) une *Short Sword +1* ou un *Great Shield +2* ; les objets ont un nom, un *artwork* et une description (et un effet — dans le cas des objets magiques) qui sont uniques.

L'unicité des objets ramassés donne le sentiment d'une grande diversité qui pousse à la rêverie (les objets intéressants ne sont pas noyés au milieu des merdouilles). Une fois encore les développeurs l'ont bien compris et affichent pendant les chargements des descriptions d'objets énigmatiques et qui révèlent quelques informations sur le (mystérieux) *background* du jeu.

Cette décision s'inscrit dans le à mon avis cadre d'un choix de conception plus large que celui de la nature et la diversité des objets. Le jeu déploie beaucoup d'efforts pour donner l'impression au joueur qu'il n'est qu'un grain de poussière dans un univers immense qui le dépasse. Depuis certaines zones du jeu, on peut apercevoir au loin des paysages (même une ville) inaccessibles et dont on entend pas parler. Le scénario du jeu est évoqué à demi-mots, souvent par sous-entendus ou par des personnages à moitié fous : c'est au joueur de boucher les trous avec son imagination. Les noms des lieux ou certains objets font référence à des personnages et des histoires souvent inexpliquées. Certains personnages évoqués dans le prologue du jeu ne refont pas surface ensuite, etc.

Je pense que grâce à cette approche, *From Software* a assez bien répondu à la problématique qu'évoque Ernest Adams dans son article [Rendons sa magie à la magie](#) (même si ça concerne l'univers du jeu plus que le système de magie lui même). *Dark Souls* n'est pas un parc d'attractions où on se déplace de la *cave aux champignons* à la *grotte du maraudeur* avec une barbe à papa dans la main et un *kodak* sur le ventre. Le contenu n'est tout simplement pas centré sur ce que le joueur va voir, il a sa propre cohérence interne qui échappe à la capacité de compréhension du joueur. C'est tout le contraire d'un jeu comme *Fable III* où chaque bribe de contenu est usée jusqu'à la moelle et agitée sous le nez du joueur pour être sur qu'il l'ait bien vue, c'est d'autant plus dommage que le peu de contenu existant est de très bonne qualité.

Il en résulte une communauté active de joueurs ([exemple](#)) à la recherche des nombreux secrets (objets, zones de jeu, personnages, boss, etc.) de *Dark Souls*. Le jeu est sorti depuis 4 mois et il reste beaucoup de mystères à résoudre. J'ai eu la chance d'y jouer tôt, au moment où il restait vraiment beaucoup à découvrir et échanger des informations sur internet m'a rappelé les échanges entre jeunes joueurs dans les cours de récréation (mais si, souvenez vous de ces rumeurs pour déshabiller Lara Croft !). C'est encore une forme de retour aux sources intelligent. *Dark Souls* grâce à sa richesse et ses non-dits (aucun PNJ avec un point d'exclamation jaune pour vous proposer une quête — c'est à vous de trouver vos propres aventures) place les joueurs dans une situation de jeu qui rappelle un peu les années 80, du temps où il était très difficile de faire le tour d'un RPG et où GameFAQs n'était pas là pour vous aider. Bien sur, il est tentant de passer un peu trop de temps sur le wiki et de briser le plaisir de la découverte me je pense que cette dimension du jeu est suffisamment explicite pour que les joueurs la remarquent et préservent leur expérience intacte.

Voilà qui m'amène au bout de ma réflexion sur les grandes forces du game design de *Dark Souls*. J'espère que vous y aurez trouvé quelques pensées intéressantes et qu'elles vous pousseront, si ce n'est pas déjà fait, à jouer au jeu. Si vous n'en avez pas eu assez, je vous recommande la lecture (en anglais) de cet [éloge](#), de ces [pensées sur la difficulté du jeu](#) ou encore de ce très très judicieux [comparatif de la narration](#) dans *Skyrim* et *Dark Souls*.

Merci à Camille Bouquet pour sa capture d'écran de *Skyrim*.

Musique et gameplay

Par [Pierre Guyot](#) | le 29 février 2012 21:48:15 | Catégories : [Analyse](#), [Musique](#)

La musique de jeu vidéo est confrontée aux mêmes problématiques que d'autres genres, comme la musique de film. Elle conditionne la façon dont le joueur va percevoir les différents éléments du jeu et aussi la façon dont il s'en souviendra. En créant une ambiance, elle détermine la façon dont le joueur va ressentir une scène ou un moment de jeu. Elle permet par ailleurs au joueur d'identifier immédiatement les différents éléments du jeu : une musique est attachée à un lieu, à un personnage (on parle alors de thème), à une scène, à une situation de la même façon qu'un bruitage est associé à une action.



Mais le film, contrairement au jeu vidéo, est un support « invariable » : un compositeur de musiques de films connaît à l'avance la durée et l'enchaînement des scènes telles qu'elles apparaîtront au spectateur. A l'inverse, un compositeur de musique de jeux ne peut pas toujours le savoir : à l'exception de certains passages purement cinématiques, la durée et l'enchaînement des différents moments de jeu dépendent du joueur et de sa façon de parcourir le jeu. La musique de jeu est donc confrontée à des problématiques supplémentaires par rapport à la musique de film : cette musique doit-elle « boucler » et si oui comment ? Comment la musique doit-elle réagir au comportement du joueur ? Comment éviter l'impression de répétitivité due à une écoute prolongée ? Comment faire intervenir les différentes musiques aux moments adéquats sans pouvoir les prévoir systématiquement ? Comment réaliser des transitions musicales alors que c'est le joueur qui dirige l'action ?

Cet article vise à ébaucher un début de réflexion sur les fonctions et les utilisations possibles de la musique dans le jeu vidéo, pour essayer de comprendre ce qui en fait la spécificité, par rapport à la musique de films notamment, et comment l'utiliser non pas en simple accompagnement du gameplay mais comme composante à part entière de celui-ci. N'étant pas moi-même compositeur, je m'intéresserai essentiellement ici à l'utilisation de la musique et à sa construction en rapport avec le gameplay, et non à la composition. Je citerai essentiellement ici des exemples tirés du RPG (et particulièrement des *Zelda*), tout simplement parce que c'est le genre que je connais le mieux (ceux qui ont lu mes articles précédents connaissent mes névroses c:).

I. Les fonctions de la musique dans le jeu vidéo

1. Tension

La musique de jeu conditionne l'ambiance d'un jeu et plus particulièrement son rythme. De la même façon que le level design conditionne le rythme de jeu en s'étrécissant ou en s'ouvrant, la musique détermine la tension, le stress (au sens le plus positif du terme) ressentis par le joueur.

Il est très classique que le boss final d'un jeu passe par plusieurs formes de plus en plus impressionnantes (avec parfois une forme finale extrêmement faible). Cette montée en puissance se traduit par le gameplay (les statistiques et les points de vie du boss changeant, ses attaques et son comportement aussi) et par le graphisme (les différentes formes sont de plus en plus grandes, de plus en plus impressionnantes). Mais

fondamentalement, c'est la musique qui va véritablement créer cette impression de *crescendo*, à mesure que les thèmes successifs deviennent de plus en plus rapides, complexes, intenses. Essayez donc de battre le boss de fin d'un *Final Fantasy*, d'un *Skies of Arcadia* ou d'un *Grandia* en coupant le son : la tension dramatique est immédiatement anéantie.

La musique conditionne par ailleurs la perception que le joueur a de son environnement. Elle définit comment le joueur va ressentir le lieu et la phase de la gameplay dans lesquels il doit évoluer.



La plupart du temps, les lieux où le joueur est en sûreté bénéficient d'un accompagnement musical serein, rassurant : dans un *Resident Evil*, quand un joueur rentre dans une *safe room*, un (très joli) thème au rythme plus régulier, plus calme l'informe immédiatement qu'il est en sécurité. A l'inverse, dans les endroits où le joueur peut être attaqué, la musique repasse sur des clusters dissonants et des nappes sourdes, autrement plus angoissants. Le game designer peut donc fournir une information au joueur sans avoir à la lui donner explicitement : plutôt que d'écrire à chaque entrée de porte « *Safe room* » ou « *Attention danger !* », ce qui tuerait l'immersion, il fait directement ressentir au joueur cette information.

Qui plus est, si le game designer peut se servir de la musique pour informer le joueur, le guider, il peut donc s'en servir pour brouiller les pistes et le piéger.

Resident Evil nous fournit à cet effet un autre exemple, caractéristique des jeux qui cherchent à surprendre ou à faire peur au joueur. Les différentes salles où le joueur doit résoudre des énigmes constituent un ralentissement du rythme de jeu, car elles sont synonymes d'une absence de combats. Il se pourrait que le joueur en profiter pour respirer un bon coup. Ce n'est pas forcément une mauvaise idée pour varier le rythme, mais si ça dure trop longtemps, le joueur trop confiant ne sera plus dans l'état d'esprit adéquat pour se laisser effrayer. Pour compenser ceci, les développeurs de Capcom ont choisi d'accompagner ces séquences par l'une des musiques les plus stressantes de toute la bande-son. La baisse de rythme induite par le gameplay est contrebalancée par une utilisation judicieuse de la bande-son, afin que le jeu maintienne chez le joueur un niveau de tension minimal.

2. Les transitions musicales

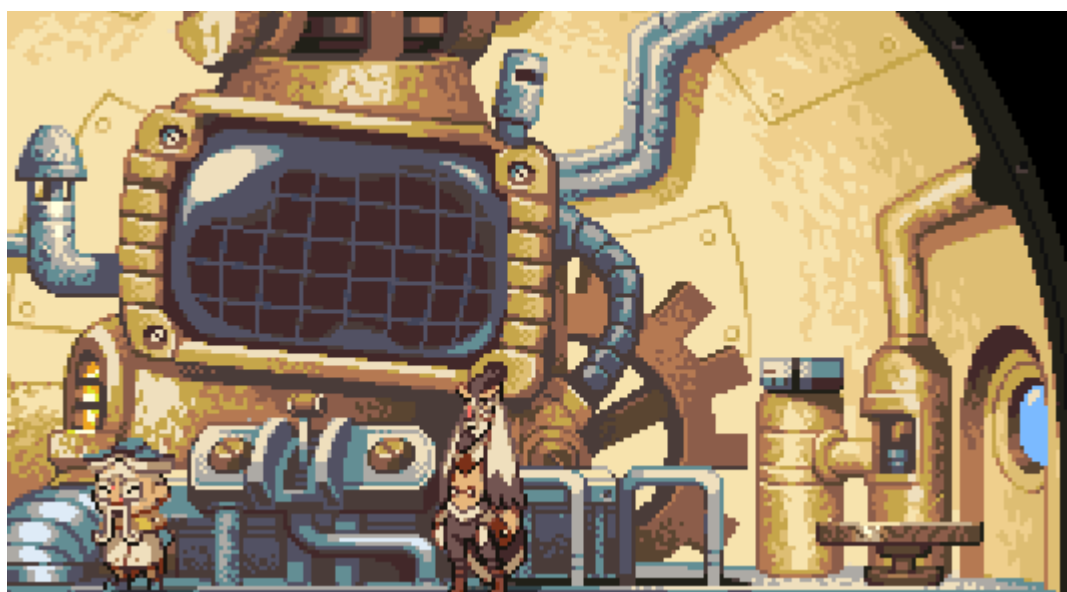
Les transitions musicales dans un jeu constituent des [modules](#) et à ce titre font partie des éléments de level design : elles doivent donc s'adapter au type de gameplay.

Prenons l'exemple de la série des *Zelda*, notamment à partir d'*Ocarina of Time*. Les combats ne sont pas distincts de l'exploration : Link traverse une plaine, rencontre un ennemi et commence à combattre directement, sans changement d'interface. Pour accompagner l'entrée en combat, la transition est

progressive, la musique de la plaine est progressivement « recouverte » par la musique des combats (qui évoque d'ailleurs en général beaucoup plus une tension nerveuse qu'un duel épique).

Prenons maintenant l'exemple d'une autre série, les *Final Fantasy*. Dans la plupart des opus, les combats et l'exploration sont très nettement séparés. Le joueur se déplace sur la carte, sans interface, jusqu'à tomber sur un ennemi. Là, rupture nette : l'écran subit un effet visuel très vif (il vole en éclats, par exemple) accompagné d'un gros *swooooosh*, les ennemis apparaissent, l'interface change du tout au tout (des jauges et des chiffres apparaissent un peu partout), le style graphique aussi (passage d'une caméra fixe à une caméra mobile, de 3D pré-calculée à 3D temps réel, de Super Deformed à réaliste, changement d'angle de vue, etc.). En réponse à cette transition, la musique change du tout au tout : on passe du thème de donjon mystérieux et éthéré au thème de combat *upbeat* et dramatique.

Dans les deux cas, la transition musicale s'effectue de manière cohérente avec le reste du jeu, elle respecte le gameplay. Elle peut être brutale (changement de piste et effet sonore) ou très douce (modulation, changement de rythme), mais dans tous les cas, elle doit s'intégrer au reste du gameplay.



Un autre exemple d'utilisation de la musique comme module entre deux phases de jeux : dans *Owlboy*, lorsque le personnage principal, Otus, marche sur une île ou sur la terre ferme, l'exploration est accompagnée d'une version 8-bit du thème principal du jeu. Dès qu'Otus s'envole, la version orchestrale du thème est superposée à la piste 8-bit. Le changement de genre musical permet de passer sans à-coup d'une phase de gameplay à une autre et d'indiquer au joueur qu'il doit changer de façon de jouer.

A l'inverse, une transition ratée peut détruire l'immersion dans le jeu. C'est notamment le risque avec une transition entre musique et silence.

Je citerai ici l'exemple malheureux de *Dragon Age : Origins*. Dans un RPG occidental, l'exploration des villes prend en général beaucoup de temps, la musique qui accompagne ces lieux est donc essentielle pour immerger le joueur dans l'univers et ne pas le lasser. Les développeurs ont choisi, plutôt que de laisser tourner le thème musical d'une ville en boucle, d'alterner phases de musique et phase de silence (ou de quasi-silence). Si cette technique permet de « rallonger » la durée des pistes sonores pour éviter la lassitude, elle est à l'origine d'une rupture de l'immersion. Quand la musique reprend après un long silence, la transition trop brutale trahit le changement et le joueur pense immédiatement « Tiens, ils ont remis la musique ? ». Cette utilisation de l'accompagnement musical ne fait pas sens, car il y a une transition musicale qui ne correspond à aucune transition du point de vue du graphisme ou du gameplay.

3. Musique et temps

Si la musique permet de créer une tension, c'est essentiellement parce qu'elle permet au joueur de mesurer plus ou moins clairement la proximité du danger. Ceci peut servir à créer une ambiance, mais aussi à élaborer des solutions de game design.

Prenons ici l'exemple des donjons d'*Ocarina of Time* : à plusieurs reprises, le joueur doit activer un interrupteur, puis effectuer une action en temps limitée avant que l'interrupteur ne s'éteigne. Comment faire comprendre ce principe de jeu au joueur sans passer par de fastidieuses explications ? On pourrait proposer d'utiliser un chronomètre, qui indiquerait le temps restant pour indiquer l'épreuve. Mais Nintendo a su se servir de l'accompagnement musical pour se montrer plus subtil.

Quand le joueur active un tel interrupteur, un tictac (semblable à celui d'une bombe par exemple) se fait entendre. Instinctivement, le joueur sait qu'il doit faire vite. De plus, quand le compte à rebours s'approche du zéro, ce bruitage s'accélère, ce qui permet au joueur d'évaluer le temps qui lui reste. Ceci permet d'éviter l'affichage d'un chronomètre qui nuit à l'immersion et de ne donner au joueur qu'une idée du temps qui lui reste, pour faire grimper un peu plus l'adrénaline chez lui.



C'est le même principe utilisé dans un autre opus de la série, *Twilight Princess*. Dans l'un des donjons, le joueur doit ramener du fond du donjon un orbe lumineux. Mais quand il retire l'orbe en question de son socle, une main géante se lance à sa poursuite pour le récupérer et le joueur doit réussir à fuir, sans se faire attraper, jusqu'à l'entrée du donjon. Dans la mesure où le joueur est occupé à fuir et à désactiver les mécanismes qui bloquent son passage, il ne peut pas forcément regarder où est la main qui le suit et savoir combien il lui reste de temps. L'accompagnement musical permet de répondre à ce problème : une musique du type de celle des *Dents de la mer* se superpose au thème ambiant à mesure que la main se rapproche. Quand la musique change, le joueur sent la tension augmenter et sait qu'il doit s'apprêter à se battre et non plus s'occuper de l'orbe. C'est un mécanisme qui permet de développer le gameplay et la narration, sans briser l'immersion comme pourrait le faire, disons, l'utilisation d'un radar dans un coin de l'écran.

La musique peut donc servir le gameplay, parce qu'elle est l'une des seules façons, autre que spatiale, de mesurer un temps, une durée. Les variations d'une mélodie permettent au game designer de faire percevoir au joueur l'écoulement du temps. C'est pour cette raison, même si c'est beaucoup plus discret, que la répétition d'une même musique, par exemple dans un donjon, donne au joueur une sensation d'immersion :

intuitivement, il mesure le temps qu'il y a passé, par exemple en se rendant compte qu'il commence à connaître le thème musical de l'endroit.

II. La musique, composante à part entière du gameplay

1. *Feedback immédiat*

Le joueur peut donc être influencé par la musique, y réagir, de la façon la plus simple (avoir peur) à la plus complexe (évaluer la proximité d'un danger). Mais le joueur n'est pas uniquement passif vis-à-vis de la musique, il peut aussi agir dessus. De la même manière qu'un joueur aime à agir sur son environnement ludique au travers du gameplay (il détruit un objet, pousse un bloc, etc.) ou du graphisme (il change de lieu, passe du jour à la nuit, etc.), il aime à interagir avec son environnement musical.

L'*immediate feedback*, ou feedback immédiat en est sans doute l'exemple le plus flagrant. L'idée est de récompenser immédiatement toute action du joueur par une réaction à l'écran : dès que le joueur appuie sur un bouton, il se passe quelque chose.

Dans un *Mario*, dès que le joueur saute ou frappe, le tout est accompagné d'un bruitage adéquat. Dans un *Sonic*, dès que le joueur collecte un anneau, il entend un bruitage caractéristique. Ceci augmente considérablement le plaisir de jeu : imaginez donc jouer à un *Zelda* où vous ramasseriez des rubis sans le moindre bruitage !

2. *L'environnement musical*

Mais l'interaction joueur-musique ne se limite pas qu'à des exemples aussi simples. La musique de jeu vidéo, si elle partage des points communs avec la musique de film, possède une caractéristique essentielle qui l'en distingue : elle est intégrée au gameplay. L'une des choses qu'un joueur recherche dans un jeu, c'est la possibilité de transformer son environnement et donc aussi son environnement musical.

Permettez-moi de citer une fois de plus l'exemple de la série des *Zelda*. Dans les Bois Perdus d'*Ocarina of Time*, Link doit s'orienter en écoutant la musique qu'il entend. La musique sert d'élément de gameplay, elle guide le joueur. Inversement, le joueur transforme la musique du lieu, comme n'importe quel autre élément du jeu, en se déplaçant : selon la direction qu'il choisit d'emprunter, il module le volume de la musique. Un autre exemple, à la fin du jeu, est l'ascension dans la tour centrale du donjon final (reprise par ailleurs dans *Twilight Princess*). Lorsque le joueur avance, il entend la musique devenir plus forte (voire se transformer dans *Twilight Princess*) ce qui lui annonce qu'il se rapproche de la confrontation finale. Mais s'il recule, la musique redevient plus douce. Ainsi, c'est le joueur qui contrôle la musique, qui choisit comment il construit la tension dramatique de cette ascension qui n'est pas scriptée : la musique obéit à ses pas.

Dans *The Wind Waker*, les affrontements sont ponctués par de courtes notes de violon, qui se font entendre à chaque coup. Ceci donne la possibilité au joueur de rythmer le combat, non seulement du point de vue ludique, mais aussi du point de vue musical. Chaque note se fait plus aiguë que la précédente, ce qui fait que la tension augmente avec chaque coup et c'est le joueur lui-même qui contrôle cette progression de la tension. *Twilight Princess* reprend — encore une fois — le même principe et l'affine. Dans le monde du crépuscule, les notes de violon qui dynamisaient le combat sont remplacées par des sons — ou plutôt des bruits — très désagréables et passablement angoissants. Le contact des monstres en devient plus stressant et incite à plus d'hésitation chez le joueur : ce changement sonore induit d'une certaine façon un changement de rythme et donc de gameplay. Poussé plus loin, c'est ce principe qui est à l'origine du concept de l'étrange *Rez* : le joueur, en détruisant les ennemis et les différents éléments du décor, produit des sons qui viennent un à un s'intégrer à la bande-son du jeu, initialement composée d'une simple pulsation. Le joueur crée son environnement sonore par le rythme de son action.



Dans *Skies of Arcadia*, c'est dans les combats contre les boss que les développeurs ont réussi à intégrer ce principe. Lors de ces affrontements, la musique peut changer selon que le joueur est en position de force ou de faiblesse. Le combat commence avec une musique de combat relativement neutre et passe à un thème sombre et discordant si les points de vie de l'équipe du joueur descendent trop bas. A l'inverse, si les points de vie du boss diminuent suffisamment, la musique se change en un thème de victoire épique. Les transitions étant finement réalisées, ceci n'est pas forcément évident lors d'une première partie. Cette idée, réutilisée dans *The Last Remnant*, permet à la musique de s'intégrer en tant que punition ou en tant que récompense selon les cas. Faire changer l'environnement sonore devient d'une certaine façon le but du joueur, la musique devient un élément que le joueur peut modifier.

3. Le diégétique et l'extradiégétique

Le jeu vidéo, comme le cinéma, a l'occasion de jouer sur deux plans : le réel dit *diégétique*, c'est-à-dire l'univers du jeu (ou du film) tel qu'il est perçu par ses personnages, et le réel dit *extradiégétique*, c'est-à-dire tel qu'il est perçu par le joueur (ou le spectateur).

Dans la plupart des cas, la musique de film ou de jeu est un élément extradiégétique : les personnages n'entendent pas la bande-son qui accompagne la scène. Une musique entendue par les personnages, par exemple s'ils assistent à un concert ou s'ils sont à côté d'un poste de radio, est à l'inverse un élément diégétique.

Le compositeur peut jouer avec cette frontière entre musiques diégétique et extradiégétiques pour créer une atmosphère particulière ou articuler le gameplay et la musique. La musique des Bois Perdus d'*Ocarina of Time* est-elle véritablement diégétique ? Link est censé entendre la mélodie jouée par son amie Saria, mais dans ce cas, d'où viennent les instruments qui l'accompagnent ? Comment pourrait-il l'entendre de si loin ?

Reprenons l'exemple de l'ascension de la tour à la fin du jeu. Link entend Ganondorf, qui joue de l'orgue au sommet de la tour. Il est ici amusant de constater que la musique qui guide Link vers le combat final n'est autre que le thème musical de Ganondorf. Comment un personnage pourrait-il avoir conscience de son propre thème ? Il en va de même pour Saria ou, pour changer de jeu, pour Cloud. Dans *Final Fantasy VII*, lorsqu'il joue de la flûte pour réveiller la Forêt Endormie, c'est l'un des thèmes principaux du jeu qu'il entonne, alors qu'il ne devrait même pas en avoir conscience.

Conclusion

La musique de jeu, même si elle partage nombre de points communs avec la musique de film, trouve sa spécificité dans l'interaction qui existe entre elle et le joueur. Au même titre que les autres composantes du jeu, elle fait partie intégrante du gameplay et doit être réfléchi en tant que telle : ce n'est pas pour rien que beaucoup de grands jeux sont aussi restés mémorables pour leurs musiques.

Si nous avons commencé à discuter de la question de l'utilisation des musiques, on peut aussi légitimement s'interroger sur les contraintes qui portent sur la nature même des morceaux utilisés. Certains jeux choisissent d'explorer une large palette de genres et de registres, d'autres au contraire restreignent jusqu'au choix des instruments utilisés. Certains choisissent de privilégier des thèmes musicaux forts et reconnaissables, d'autres préfèrent des compositions d'ambiance plus discrètes et parfois à la limite du bruitage, mais qui laissent plus respirer les voix des personnages. J'espère pouvoir aborder (ou demander à quelqu'un d'autre de le faire) ces questions dans un prochain article. En attendant, vos critiques, remarques et suggestions (musicales notamment) sont plus que bienvenues !

Bonus

Pour conclure cet article, la rédaction des Forges a souhaité se faire un petit plaisir et vous faire partager certains des morceaux de la musique de jeux vidéo qui l'ont marquée.

La sélection de Thomas :

- Dearly Beloved, *Kingdom Hearts*, Yoko Shimomura
Le thème de l'écran-titre de *Kingdom Heart* premier du nom est pour moi l'une des musiques les plus marquantes du jeu vidéo. Simplement jouée au piano, accompagnant un menu sobre et élégant, c'est la musique apaisante qui précède l'enchaînement de combats, qu'on prend le temps d'écouter au moins une fois avant de partir à l'aventure. L'entendre évoque instantanément la licence et les heures de jeux accumulées sur les divers titres.
- Main Menu Theme, *Age of Empires II: The Age of Kings*, Stephen Rippy
Encore une musique de menu, c'est décidément une habitude. Cette fois ci, c'est un thème épique, entraînant, qui évoque l'époque médiévale, thème central de *The Age of Kings*. C'est également l'annonce de féroces combats, mais cette fois-ci l'attente est plus fébrile que sereine: on a envie d'en découdre, là, maintenant, et d'assiéger du château à la tête d'une armée de cavaliers en armure.
- Route 24/25 Theme, *Pokémon Rouge* et *Pokémon Bleu*, Junichi Masuda
Ce dernier choix est guidé par la nostalgie principalement. *Pokémon*, c'est ma découverte des jeux japonais, c'est ma Game Boy reçue pour mon anniversaire (pensez-vous, une console qu'on peut emmener partout alors qu'on ne connaissait que la grosse tour dans la chambre des parents !), c'est la découverte des RPG. Et le thème de la route 24, c'est *Pokémon* résumé en 10 minutes: une musique entraînante, un parcours semé d'embûches et l'envie d'aller voir ce qu'il y a au bout.



La sélection d'Antoine :

- Gallantry, *Raiden*, Akira Sato
Ce thème est la musique du premier niveau de *Raiden* (une série de shoot them up des années 90). C'est une musique 16-bit entraînante qui illustre ce que le jeu vidéo a pu apporter à la musique. A cause de limitations techniques, les compositeurs ont du ruser pour obtenir des sons intéressants et ont donné naissance sans le savoir à la scène *chiptune* qui est si riche aujourd'hui.
- Landmass, *WipEout 2097*, The Future Sound of London
La série des *WipEout* (des jeux de course à la F-Zero pour hardcore gamers) a pour habitude de constituer ses bandes sons en faisant travailler des artistes à la pointe de la scène *electro* anglaise (en tout cas c'était le cas à l'époque). Cette piste et le reste de la bande son m'ont fait réaliser qu'il pouvait y avoir des échanges fructueux entre le jeu vidéo et des artistes qui ne sont pas des compositeurs dédiés. Accessoirement, c'était aussi un de mes premiers contacts avec la musique électronique de bonne qualité et The Future Sound of London fait toujours partie de mes groupes préférés.
- Mantis' Hymn, *Metal Gear Solid*, KCE Japan Sound
Je suis désolé d'avoir choisi trois jeux des années 90, je devais être très émotif dans mon enfance. Je crois qu'avant cette piste je ne faisais jamais attention à la musique dans les jeux sauf quand elle était très entraînante ou exaspérante. Il s'agit d'un morceau inquiétant qui exploite habilement le chipset sonore de la PSX pour créer des glissandi dans les chœurs, chouette ambiance. J'ai remarqué l'existence de cette musique quand Snake a explicitement fait remarquer « Tiens !? La musique s'est arrêtée. » quelques minutes avant l'affrontement contre le boss de la zone.



La sélection de Pierre :

- *Blinded By Light, Final Fantasy XIII*, Masashi Hamauzu
Essentiellement, mes choix sont dictés par mon amour des musiques « à thème ». Le thème des combats de cet opus m'a marqué par son dynamisme, sa résistance hors-norme à la répétition (alors que la boucle est extrêmement courte) et surtout sa capacité à associer les dimensions artificielle (batterie, guitare électrique, etc.) et organique (violons, harpes, etc.) qui font toute la direction artistique du jeu.
- Mullen, *Grandia*, Noriyuki Iwadare
Grandia est une œuvre immense, émouvante, adulte, véritable transposition d'un Miyazaki en jeu vidéo. Et sa bande-son n'y est pas étrangère. Le thème du personnage de Mullen en est l'exemple le plus frappant : en dix secondes, on sent tout le conflit intérieur de ce personnage profondément déchiré entre sa noblesse, son courage et son arrogance, sa soif de pouvoir. Le thème *impérial* par excellence.
- *Heavenly Flight, Dragon Quest VIII : l'Odyssée du Roi Maudit*, Koichi Sugiyama
Antoine râlera tant qu'il voudra, je n'en démordrai pas : la bande-son des *Dragon Quest* est remarquable. Cette musique, qu'on entend sur la carte du monde lorsqu'on la survole à dos d'oiseau, traduit tout le sens du voyage qui anime *Dragon Quest VIII* de bout en bout. Personnellement, c'est grâce à elle que je me suis littéralement senti *saisi* chaque fois que je voyageais dans cet univers.



Mon blog AAA – épisode 1

Par [Antoine Gersant](#) | le 3 avril 2012 18:02:42 | Catégories : [Ma vie](#)

Il y a quelques mois, à l'occasion de la 22ème édition du Ludum Dare, j'ai eu le plaisir d'inaugurer une nouvelle catégorie d'articles sur les Forges intitulée « Ma Vie ». Il est maintenant temps de remplir plus avant cette catégorie au moyen d'articles bloguesques sur mon quotidien. Rassurez vous, il n'est pas question de parler de mon chat ou du contenu de mes sandwiches — tout du moins je vais essayer de me retenir.

Les lecteurs qui me connaissent un peu (ou les curieux qui cliquent sur les liens en petits caractères en bas de page) savent que je suis étudiant à l'École Centrale de Nantes (une école d'ingénieurs). Mon cursus là-bas touche à sa fin et je viens de quitter Nantes pour un stage de fin d'études de 6 mois. Etant donné que construire des ponts et optimiser des centrales nucléaires ne sont pas vraiment des activités qui correspondent à mon projet professionnel, j'ai plutôt été tenter ma chance du côté des studios de développement de jeux dits AAA (grosse équipe, gros budget, etc.).

Quelques candidatures et entretiens plus loin, me voici aujourd'hui fraîchement débarqué à Guildford (en Angleterre), en train d'écrire cet article depuis la cantine de Lionhead Studios (où les sodas sont gratuuuuuuuuits). Lionhead est le studio qui développe la série Fable (et qui a aussi fait Black and White) et je suis très très enthousiaste à l'idée d'y travailler pendant 6 mois. A vrai dire c'est l'un des studios où j'espérais le plus être pris et ça n'a pas loupé =)



L'objet de cette série d'articles sera de raconter le déroulement de mon stage et la vie dans un studio AAA (où les sodas sont, rappelons-le, gratuuuuuuuuits) telle que perçue par un jeune développeur candide et expatrié. Mon propos sera limité par un sévère *Non Disclosure Agreement* qui m'interdit de parler de beaucoup de choses, mais je ferai de mon mieux pour trouver des sujets à aborder. Au pire je vous détaillerai le menu de la cantine et les céréales disponibles pour le petit-déjeuner (je pense que leur prix de 50 pence n'est pas concerné par mon *Non Disclosure Agreement*).

Enfin, n'hésitez pas à laisser des commentaires sur cet article si vous avez des questions particulières ou des points que vous souhaitez me voir aborder. Je ferai de mon mieux pour y répondre !

[Lire la suite →](#)

Les joies du game over

Par [Pierre Guyot](#) | le 11 juin 2012 01:44:47 | Catégories : [Game Design](#)

Il y a un déjà un petit moment, Bonhomme et moi discutons de son projet en cours, *Blade of Victory*, et plus particulièrement de la posture à adopter quant au game over. Je n'y avais pas trop réfléchi plus que ça auparavant et au bout d'un moment, je me suis rendu compte que c'était un sujet beaucoup plus difficile que ce à quoi je m'attendais. J'ai eu le temps d'oublier et puis récemment, j'ai dû réaliser un écran de game over pour un autre projet et je me suis dit que c'était un peu bête de ne pas coucher toute cette réflexion sur le papier. Aujourd'hui, je m'intéresse donc au game over.



Cet article est l'occasion d'un premier essai sur [Level Up](#) : à la suite d'une discussion sur le forum, j'ai essayé de rédiger une synthèse de ce qui a pu se dire et d'aboutir à un article plus riche que si je m'étais lancé seul dedans. En espérant qu'il y en ait beaucoup d'autres, je vous invite à nous faire part de vos critiques, commentaires et réflexions et surtout, à aller sur Level Up ! :D

I. Game over et difficulté.

1. Crime et châtement.

Classiquement, le game over sanctionne (dans tous les sens du terme) la façon dont un joueur joue. La façon de choisir le type de game over adapté à un gameplay va beaucoup dépendre de la façon dont on souhaite « punir » le joueur et de la difficulté qu'on cherche à lui proposer. On trouve ainsi toute une gamme de game over, du plus cruel au plus clément.

Le game over classique — sans doute le plus pénalisant — consiste à effacer intégralement la progression du joueur, ce qui l'oblige à recommencer depuis le dernier point de sauvegarde, voire depuis le début du jeu. Son caractère punitif dépend beaucoup de la répartition et de la fréquence d'apparition des points de sauvegarde. Le joueur peut aussi être pénalisé par une perte plus partielle : de l'argent, des objets, de l'expérience, etc. Dans *Dragon Quest* ou *Pokémon*, le game over se solde par la perte d'une bonne partie de son argent (la moitié me semble-t-il) mais permet au joueur de conserver tout le reste. Ce type de game over ne rompt pas la continuité de l'histoire : le joueur continue de jouer *après* la rousste qu'il s'est prise et non pas comme si elle n'avait *jamais* existé. Le game over peut aussi consister à renvoyer le joueur à son point de départ. Quand on meurt dans un *Zelda*, on ne perd rien, mais on réapparaît (avec peu de vie) soit dans un lieu prédéterminé (l'entrée de la grotte dans le premier opus, la maison de l'oncle dans *A Link to the Past*),

soit à l'entrée du donjon. Ce n'est pas forcément très pénalisant, mais suffisamment pour que le joueur soit stressé à l'idée de perdre. A l'autre bout du spectre, des jeux comme *Final Fantasy XIII* et *Kingdom Hearts* proposent au joueur de recommencer juste avant le combat, ce qui permet de ne pas interrompre le rythme du jeu.



La pénalité peut n'apparaître qu'avec la répétition, comme si le joueur avait un « droit à l'erreur ». C'est le principe des *continues* dans d'anciens jeux comme *Starwing*, mais c'est aussi ce qu'on retrouve dans la mécanique des combats de *Prince of Persia : les Sables du Temps* : tant que le joueur a du sable, c'est-à-dire qu'il n'a pas encore trop mal joué, il peut remonter le temps pour se sauver de la mort.

2. Un équilibre délicat.

Le game over, comme tout mécanisme permettant de régler la difficulté du jeu, repose sur un équilibre délicat entre frustration et tension.

Si le game over est trop fréquent et trop pénalisant, le joueur risque d'en avoir assez et d'arrêter de jouer. La différence entre *The legend of Zelda* premier du nom et beaucoup de jeux mal conçus de la même époque, c'est le choix d'un game over non pénalisant : le jeu est dur, c'est un véritable challenge, mais il encourage le joueur à persévérer grâce à un système intelligent de game over. Le joueur perd souvent, mais ne repart pas systématiquement de zéro. Si le game over avait été pénalisant comme dans un *Final Fantasy* classique, la plupart des joueurs auraient sans doute très rapidement arrêté de jouer.

A l'inverse, un game over trop « laxiste » ne permet pas au joueur de ressentir une véritable tension, de s'impliquer dans le jeu. Pour citer un grand dramaturge français :

A vaincre sans péril, on triomphe sans gloire.

Persona 3 joue avec la répartition des checkpoints pour sadiser le joueur. L'unique donjon que vous parcourez pendant tout le jeu, le Tartarus, fonctionne par étages et vous propose régulièrement des téléporteurs vers l'entrée, à *sens unique*, mais très rarement de *vrais* téléporteurs. Pour ainsi dire, le jeu ne vous propose quasiment jamais de « vrais » checkpoints et donne au joueur le sentiment d'être en « apnée » dans le donjon. Le risque de game over est bien présent — d'autant plus que celui-ci est impitoyable, à savoir retour à la dernière sauvegarde sans sommation — ce qui permet de créer une vraie tension dans les

donjons : quand on retourne *enfin* au point de sauvegarde, il y a un sentiment *réel* de soulagement, de récompense, de gratification.



Il y a ici un compromis délicat à trouver entre donner au joueur envie de continuer à jouer malgré le game over et lui donner la sensation qu'il prend des risques.

Un exemple subtil est celui que propose *Drakengard 2* : si vous perdez une bataille, deux alternatives s'offrent à vous. Si vous abandonnez, vous retournez à l'écran-titre et votre progression, votre argent, votre expérience sont définitivement perdus. Si vous recommencez immédiatement la bataille, vous ne perdez que l'argent gagné mais par l'expérience. Ce principe est très fin. Il pousse le joueur à recommencer immédiatement plutôt que d'abandonner, mais sans lui faire grâce du game over. Le jouer est puni, mais on lui donne une chance de se « racheter », ou pour le voir sur un mode plus ludique, de se « refaire ». Le joueur n'est au final pas vraiment pénalisé s'il fait preuve de persévérance. Qui plus est, ce système permet d'éviter que le joueur ne se retrouve définitivement bloqué, car il a toujours la possibilité de faire de l'XP jusqu'à réussir à passer la prochaine bataille. Impossible donc de se retrouver dans une impasse ou de voir sa sauvegarde foutue.

II. Rupture et continuité.

Cette approche classique du game over sous l'angle de la pénalité n'est pourtant pas suffisante. Après avoir réfléchi un bon moment sur la notion de game over, je me suis rendu compte que je ne pourrai parler de celui-ci sans évoquer sa contrepartie, la sauvegarde. Pourquoi ? Parce que game over et sauvegarde sont deux mécanismes de jeux complémentaires qui permettent au game designer de contrôler le rythme, le flot du jeu, que ce soit en termes de gameplay ou de narration. Ce sont un peu les fonctions *stop* et *play* du jeu.

1. Les fonctions de la sauvegarde.

La sauvegarde a essentiellement deux fonctions dans un jeu.

D'une part, elle permet au joueur de conserver ses données, tout simplement. Si jamais un incident fâcheux se produit — coupure de courant, bug — la sauvegarde lui permet de récupérer ce qu'il a accompli. C'est cet aspect qui est le plus flagrant dans les systèmes de mots de passe. Ce dernier n'est plus trop à l'ordre du jour parce qu'on dispose maintenant de la mémoire nécessaire pour créer des fichiers de sauvegarde, mais aussi parce que ce système crée moins de liens entre un joueur et sa partie : une sauvegarde n'y est pas un objet unique, personnel.

D'autre part, la sauvegarde permet de faire une pause. Le joueur peut éteindre la console sereinement : c'est le caractéristique « À table ! » suivi de « Attends deux secondes, je sauvegarde ». Dans cette optique, on

trouve les classiques fichiers de sauvegarde, mais aussi plus rarement des sauvegardes dites rapides ou temporaires : le joueur est renvoyé à l'écran-titre et sa sauvegarde disparaît dès qu'elle est chargée. Cette fonctionnalité est liée au rythme du jeu et on l'utilise surtout dans le cas où les séquences de jeux sont longues, comme dans un Tactical RPG à la *Final Fantasy Tactics Advance*. Ceci permet au joueur de mieux contrôler le rythme du jeu et de ne pas être obligé de jouer par tranche d'une heure minimum, ce qui par exemple incompatible avec des jeux qui visent les plus jeunes ou prévus pour des parties courtes (en particulier sur les consoles portables).



Au delà de ces deux fonctions, la sauvegarde doit respecter une condition essentielle : en sauvegardant, le joueur ne doit pas pouvoir foutre en l'air sa partie. Dit comme ça, ça a l'air tout bête, mais même des jeux pro récents s'y trompent (je vous renvoie au passage sur les checkpoints hors de propos dans cet [article](#) d'Ernest Adams). *Final Fantasy Tactics* premier du nom *aurait pu* être un chef-d'oeuvre. Mais la structure de la sauvegarde lui confère une difficulté incongrue : si vous sauvegardez au cours d'une série de missions, comme le jeu vous incite lourdement à le faire, vous risquez de vous retrouver en train de recharger juste avant un combat trop dur pour vous (grâce à une courbe de difficulté complètement discontinue), sans aucune possibilité de faire du level grinding. Vous imaginez, une partie de 30 heures bonne pour la poubelle ?

2. Respecter le rythme du jeu.

Le game over et la sauvegarde, en tant que modes d'interruptions du jeu, permettent de contrôler son rythme.

Considérons ici des jeux au game over peu pénalisant, *Final Fantasy XIII* et *Kingdom Hearts*. Dans les deux cas, il s'agit de jeux qui se veulent rapides, nerveux, rythmés : les combats sont dynamiques, essentiellement basés sur la vitesse, la structure de la narration est linéaire (une pure ligne droite pour le premier, des mondes très linéaires pour le second). Dans cette optique, les développeurs ont choisi de proposer au joueur de recommencer immédiatement le combat perdu. Cette décision permet d'inciter le joueur à continuer de jouer et surtout, ne brise pas le rythme tendu du jeu. *Final Fantasy XIII* a été pensé comme une course-poursuite par ses développeurs — que l'on trouve ça réussi ou non — et le choix d'un game over qui n'interrompt pas le rythme du récit est cohérent avec cette vision du jeu. Exemple plus extrême encore, *Super Meat Boy*. Le gameplay de ce jeu repose essentiellement sur la possibilité de réessayer *très* rapidement après un échec. Imaginez ce qui se passerait s'il y avait une latence de, disons dix secondes entre chaque essai : *Super Meat Boy* passerait du statut de jeu culte à celui de médiocre plateformer.

A l'inverse, considérons maintenant l'exemple de *Persona 3* et *Persona 4*. Ces jeux sont basés sur un rythme quotidien, avec un système de calendrier où le joueur construit pas à pas, étape par étape, degré par degré ses relations avec les différents personnages. Ce rythme progressif se retrouve dans des donjons dans lesquels on avance peu à peu, étage par étage : le choix d'un game over pénalisant — à savoir le retour à l'écran-titre — est fait de telle sorte qu'on hésite à avancer dans le donjon, on craint de perdre le butin accumulé à tout instant et donc, on avance à pas comptés parce que tenter de traverser dix étages d'un coup pour atteindre le prochain vrai checkpoint peut s'avérer très risqué. Le système de game over permet ici d'assurer un rythme de jeu cohérent avec la volonté des développeurs. Dans *League of Legends* et autres MOBA (et même plus généralement dans beaucoup de jeux multijoueurs), l'attente après la mort de son avatar n'est pas une simple pénalité, mais un élément stratégique à part entière puisqu'elle va influencer sur le rythme de la bataille et son issue, souvent de manière décisive.

Que le rythme du jeu soit lent, rapide, saccadé ou continu, le game over (et donc en creux, la sauvegarde) permet de contrôler ce rythme, de l'interrompre ou de le fluidifier. Quand vous réfléchissez au type de game over que vous allez choisir ou la façon dont vous allez placer vos checkpoints, il faut toujours vous demander ce que le joueur devra subir après le game over et si ceci correspond au *rythme* que vous cherchez à définir pour le jeu. Surtout, n'oubliez pas que l'attente, si elle est injustifiée, peut être extrêmement pénible pour le joueur : quelle punition plus grande pour un joueur que le non-jeu ? Non seulement vous avez perdu, votre honneur est sali, vous êtes pénalisé dans le jeu, mais *en plus* vous devez poireauter (et allez que je me farcis de nouveau les noms des développeurs, éditeurs, technologies utilisées, l'écran-titre et tous les menus, sans compter les temps de chargement en bonus).



Le placement des opportunités de sauvegarde est à ce titre essentiel. Un joueur qui a traversé toute une partie d'un niveau difficile et se voit obligé de tout recommencer pour une minuscule erreur peut tout à fait décider d'abandonner purement et simplement. C'est ce qui arrive par exemple quand le joueur, pour aller retenter la séquence difficile qui le bloque, doit retraverser à chaque fois tout le niveau (qu'il commence à connaître par cœur) parce que le level designer n'a pas prévu assez de points de sauvegarde. On peut essayer de résoudre ce problème en multipliant les checkpoints, comme dans un *Alan Wake* : la démultiplication des points de sauvegarde permet de compenser la difficulté de chaque « segment » de jeu et on peut recommencer directement la séquence qui nous bloque (et qui donc, nous intéresse :P). Mais sur ce sujet, le grand [Miyamoto](#) s'exprime sans doute bien mieux que moi.

III. La question de la narration.

Le game over, en tant que fin de la partie, peut aussi constituer la fin de l'histoire. C'est un élément narratif à part entière et la question de l'immersion du joueur doit être prise en compte.

Le premier danger pour la narration, c'est la répétition : devoir se manger deux fois — voire plus — les mêmes scènes, les mêmes passages après un game over est particulièrement pénible (et une option permettant de passer dialogues ou cut-scenes est souvent la bienvenue). Pire, ceci peut tuer l'émotion ou l'intensité de scènes pourtant bien construites et qui avaient marqué la première fois.

Un autre problème est celui du choix, question chère aux amateurs de RPG. Il est confortable de pouvoir sauvegarder à volonté, mais ceci peut ruiner la profondeur d'un jeu. Dans les RPG de Bioware comme *Dragon Age : Origins* ou *Mass Effect*, la possibilité de pouvoir sauvegarder à tout moment entre en contradiction avec le gameplay dans la mesure où elle retire le poids, l'importance véritable des choix effectués par le joueur : même s'il regrette une décision, le joueur a toujours l'opportunité de revenir en arrière. Les choix — déjà peu intéressants et tristement manichéens — sont alors vidés de toute leur substance : leur irréversibilité.

Mais le game over et la sauvegarde ne sont pas seulement des risques pour la narration : ce sont aussi des opportunités de l'enrichir et d'entrelacer plus subtilement le récit et le gameplay.



Dans *Dark Souls* et son prédécesseur *Demon's Souls*, le game over est intégré à la narration et au scénario mêmes. Lorsque le joueur meurt, il revient à son lieu de départ, mais *après* sa mort : tout ce qu'il a fait reste valable. S'il a pu récupérer l'arme surpuissante caché dans un coffre juste derrière le boss, tant mieux pour lui car il la conserve et pourra se servir de ladite arme pour assommer ledit boss. A l'inverse, s'il a gâché son mégalixir, il l'a dans le baba. En intégrant cet aspect au scénario (avec notamment l'apparition d'une sorte de « hub » pour morts, le Nexus), les développeurs ont su créer un système très intéressant et qui contribue tout particulièrement à l'ambiance extrêmement oppressante du titre.

Certains jeux jouent sur l'opposition diégétique/extradiégétique (ce que voit le personnage/ce que voit le joueur) avec de faux game over. Il s'agit souvent de séquences humoristiques comme le délirant « Fission Mailed » de *Metal Gear Solid 2* et la fausse mort (avec déroulé complet du générique de fin !) du malheureux Guybrush Threepwood dans *The Curse of Monkey Island*. Les points de sauvegarde sont parfois intégrés à l'histoire du jeu (Masato Kato s'en est fait une spécialité avec des jeux comme *Chrono Cross* et *Xenogears*, mais nous n'en dirons pas plus :3). Pour pousser le bouchon encore plus loin, on pourrait parler

de l'intrigant *Baroque* (ou du moins de sa version Saturn, si je ne fais pas d'erreur) : la sauvegarde, possible uniquement à la fin de chaque niveau de la tour qui constitue l'essentiel du jeu, peut *échouer* aléatoirement. Imaginez l'angoisse du joueur qui vient de voir la possibilité de sauvegarder lui être refusée, après une traversée déjà éprouvante.

La sauvegarde, au même titre que le game over, n'est donc juste un « système » pour conserver les données, elle est aussi un élément de narration à part entière et on peut l'utiliser comme telle : les sauvegardes de *Resident Evil* sont pour ainsi dire des objets (les fameux rouleaux encreurs) et leur utilisation fait tout l'intérêt *narratif* et l'ambiance des *safe rooms*. Le sentiment de soulagement, de *respiration* (pour reprendre la métaphore précédente :P) dans ces lieux est un grand moment de jeu vidéo : le gameplay y fait la narration et réussit à créer une émotion profonde chez le joueur.

Conclusion

Le game over et la sauvegarde sont deux systèmes étroitement liés, qui fonctionnent à la fois comme éléments de gameplay et de narration. Ce sont deux éléments symétriques qui permettent de régler la difficulté d'un jeu, mais surtout son rythme. Mais indépendamment de ces considérations théoriques, n'oublions pas que leur implémentation n'est pas anodine.

Tout d'abord, elle induit des contraintes techniques et un travail de programmation. Quand on travaille sur un jeu amateur ou indépendant, avec des contraintes de temps ou d'argent très strictes, ceci n'est pas à écarter. En termes de programmation, on peut dire que les solutions de sauvegarde les plus simples sont justement celles proposées par *Final Fantasy* ou *Zelda*, c'est-à-dire sauvegarder l'état du jeu à un moment précis et y revenir bêtement (*Final Fantasy*), ou ne pas toucher à cet état et réinsérer le joueur dedans en le déplaçant légèrement s'il déclenche une condition de défaite (*Zelda*).

Par ailleurs, elle participe de l'intégration de la fin de partie dans le game design. Un système comme celui de *Dark Souls* ou celui de *World of Warcraft*, où la mort constitue une partie du jeu, demande un gros travail supplémentaire sur le game design, et donc un gros travail technique. Décider ce que l'on sauvegarde et ce que l'on retire au joueur devient encore plus complexe. Le choix d'un système de game over, en tant qu'élément de gameplay, doit être réfléchi en fonction des autres éléments de gameplay et non comme une *feature* gratuite.



Enfin, tout bêtement, son habillage, graphique et sonore, doit s'intégrer dans le reste du jeu : ayons une pensée pour la délicate *Soul Poetry* de *Baten Kaitos* ou la mélancolique *A Light Already Lost* de *Chrono Cross*.

Vilain Game Designer ! Pas de biscuit pour toi ! VII

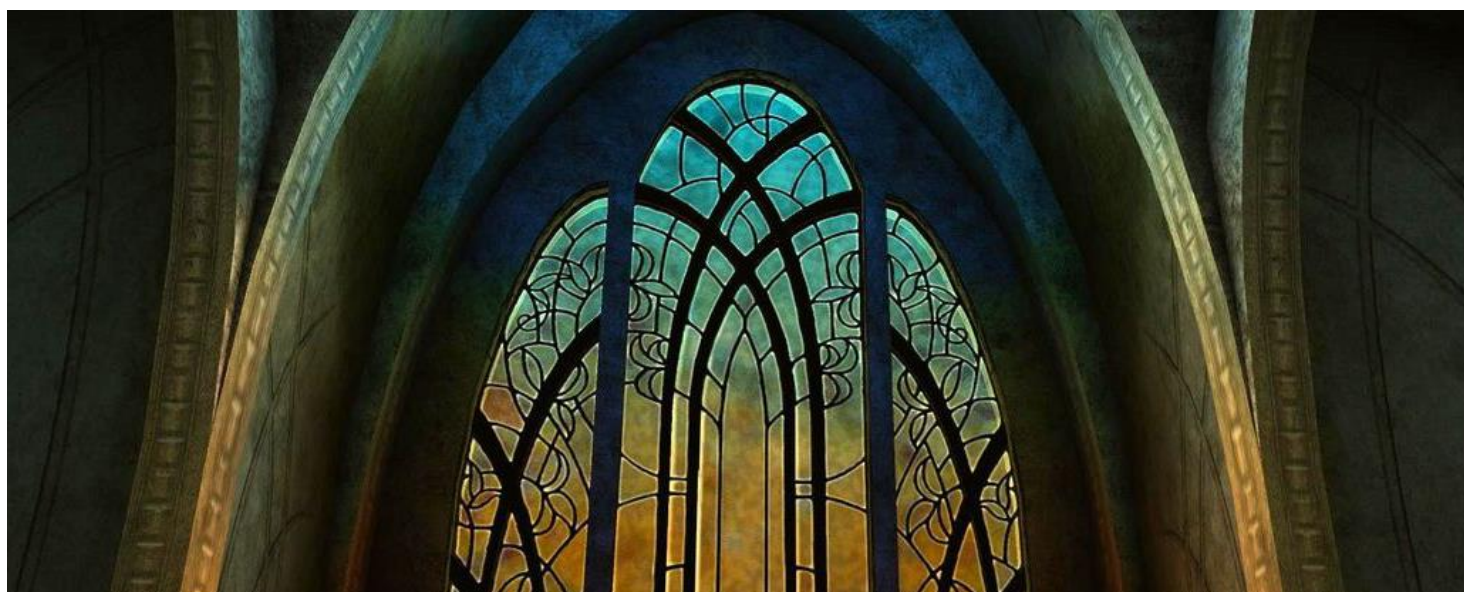
Par [Lyonsbanner](#) | le 11 juillet 2012 00:26:08 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Lyonsbanner](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juin 2006).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Nous savons tous que l'industrie du jeu vidéo souffre d'un important *turn over*: des débutants enthousiastes rentrent dans le monde du business ; les conditions de travail et les longues heures les poussent à bout ; ils finissent par démissionner pour trouver une occupation plus saine et de nouvelles recrues arrivent, toutes prêtes à se faire exploiter à leur place. Mis à part la vie et le talent gâchés que cela représente, ceci signifie que les studios de jeux vidéo n'ont pas de mémoire institutionnelle, et c'est en partie pour ça que nous continuons à faire des erreurs de conceptions.



D'un autre côté, ça me donne de quoi écrire quelque chose chaque année ! Je vous présente Vilain Game Designer ! Pas de biscuit pour toi ! VII. Cette liste de TDC annuelle [NDT : *Twinkie Denial Condition*, littéralement, condition pour refuser un biscuit, à savoir à un mauvais game designer)] est, malheureusement, longue, et comme d'habitude, m'a été envoyée par vous, mes fidèles lecteurs. Si vous voulez m'en envoyer un peu plus, écrivez à notwinkie@designersnotebook.com [NDT : en anglais]. En attendant, c'est parti.

Pas de nouvelles fonctionnalités après les premiers niveaux

Le renouvellement est l'un des nombreux aspects qui rend un jeu vidéo amusant et l'une des qualités qui sépare ce genre des jeux de société, par exemple. Mahdi Jeddi écrit pour se plaindre des jeux qui présentent toutes leurs fonctionnalités dans les premiers niveaux, mais qui ensuite n'offrent rien de nouveau dans les dernières étapes du jeu. Il nous dit :

S'ils ont un budget limité, ils peuvent répartir la présentation des nouvelles fonctionnalités dans chaque niveau, et peut-être faire quelques niveaux spéciaux pour les présenter. De cette manière, le jeu conservera sa fraîcheur jusqu'à la fin et le joueur sera sauvé de l'ennui.

Oyez ! Oyez ! Quel est le pire : un jeu qui présente ses fonctionnalités petit à petit mais régulièrement ou un jeu qui les présente toutes d'un coup et qui n'en donne plus du tout ensuite ? Je préférerais beaucoup jouer au premier. Évidemment, cela dépend du genre, mais offrir un peu de neuf de temps à autre aide sans aucun doute à maintenir l'intérêt du joueur. Trop de jeux finissent par devenir lassants à mourir dans leur dernier tiers voire plus et le joueur doit faire un effort considérable s'il veut voir la fin. On ne fait pas ce métier pour créer des choses ennuyeuses à mourir.

Changement radical des règles contre le boss de fin

Les boss nécessitent toujours une approche différente des ennemis ordinaires — c'est une convention bien connue des jeux vidéo, et nous le savons tous. Mais quand les changements sont si grands qu'ils rendent l'expérience antérieure inutile, le jeu est injuste envers le joueur. David Peterson écrit :

Dead or Alive 3 change complètement les règles au moment du combat contre le boss final et rend caduc tout ce que vous avez appris auparavant. Quand vous jouez au mode Histoire, vous combattez un tas de personnages différents au cours du jeu. À la fin, vous affrontez le « big boss » (je ne sais toujours pas qui il est ni pourquoi il est entouré de flammes) et tout change. Il vous tire dessus ; la caméra est verrouillée dans une position complètement différente du reste du jeu, qui décale tous les combinaisons que vous avez l'habitude d'utiliser d'un quart de tour sur la manette et le jeu finit par se réduire à « Bouge un peu, bloque, bouge un peu, etc. » Tous les mouvements et les compétences que vous aviez appris ne servent plus à rien maintenant. Argh !!!

Yep, clairement une TDC. Les combats contre des boss devrait se baser sur ce que le joueur a appris et non repartir de zéro.

Mauvaise traduction

Santiago Hodalgo nous écrit d'Espagne pour souligner à quel point une mauvaise traduction peut tuer tout le fun d'un jeu. Je lui cède la parole :

Avant, seul un petit nombre de jeux étaient traduits en espagnol, principalement les jeux d'aventures et c'étaient sans doute les seuls correctement traduits. Je me souviens particulièrement des jeux *Lucasarts* parce qu'ils avaient de bonnes traductions, et les blagues américaines avaient même été adaptées pour le jeu en espagnol... Aujourd'hui, avec des jeux plus complexes et un contenu multimédia riche, je pense que le travail de traduction et de localisation est devenue une tâche plus importante, et alors que certains jeux sont correctement traduits, beaucoup d'autres ne le sont pas.

J'ai joué à *Call of Duty* en anglais pour la première fois, et les voix ajoutaient au réalisme, mais dans la version espagnole la traduction était extrêmement pauvre. Dans la version originale, les officiers criaient des ordres, insistaient sur certains points ; la voix des soldats paniqués reflétait leur peur ; les civils cachés vous interpellaient à voix basse et on distinguait même des accents différents selon la nationalité des personnages. Dans la version espagnole, il n'y a pas la moindre intonation de voix, c'est juste de la lecture. C'est le même timbre de voix pour chaque ligne de dialogue, peu importe la situation ou l'endroit."

Quoi qu'il en soit, *Call of Duty* est un jeu génial, alors qu'est-ce qui ne va pas ici ? Qui, chez Activision, a une estime assez basse de son travail pour bâcler d'une telle façon la traduction ? Le marché hispanophone n'aurait-il pas droit à une aussi bonne qualité de production que le marché anglophone (vue la taille du marché hispanophone *rien qu'aux États-Unis*, la personne en question ferait bien de vite se ressaisir) ? Vilain Game Designer ! Pas de biscuit pour toi ! [NDT : les joueurs français comprennent sans aucun doute très bien le problème, quand on pense à des traductions comme celle — « mémorable » — de *Final Fantasy VII*]

Pas de sous-titres

Et puisque nous évoquons le problème de la langue, Mihail Mercurjev écrit d'Ukraine pour nous dire :

Les sous-titres rendent le jeu accessible aux personnes étrangères et aux personnes qui sont forcées de jouer sans le son. Mais de nombreux jeux connus (*Myst*, *Uru* et *Outcast* par exemple) n'en ont pas. En outre, la plupart des dialogues dans *Myst* sont tellement bruités que c'est à peine compréhensible, même pour quelqu'un dont c'est la langue maternelle. Dans *Commandos*, quand vous n'avez pas de son, le briefing de la mission ne sert plus à rien, il y a juste la carte qui défile d'un côté ou de l'autre. Pour une raison inconnue, le jeu *Driver* refusait de diffuser certains sons sur mon ordinateur, je n'ai donc pas pu écouter certains messages... et pas de sous-titres pour compenser, évidemment.



Je pense que tous les messages vocaux doivent être renforcés par des sous-titres ou un autre système visuel. Si vous pensez que l'utilisation de sous-titres ruine le gameplay, désactivez-les par défaut. Dans certains cas, quand vous traduisez le jeu dans une autre langue, on peut se contenter simplement d'afficher les sous-titres sans toucher aux voix. Par exemple, ce genre de traduction a bien fonctionné dans *Desperados* ; si les voix étaient doublées en russe, les personnages perdraient leurs charmants accents.

Je ne suis pas d'accord avec la dernière phrase — ils ne perdraient pas nécessairement leurs accents si les acteurs avaient fait un doublage *correct*. Mais comme nous l'a fait remarquer Santiago, les sociétés de traduction n'en prennent pas toujours la peine. Il y a une autre bonne raison d'inclure des sous-titres, c'est de permettre aux joueurs malentendants de pouvoir profiter du jeu. Nous sommes *très* loin derrière les autres médias de ce point de vue. Les programmes télévisés et les DVD sont aujourd'hui systématiquement sous-titrés, voire accompagnés d'une traduction en langue des signes. *Half-Life 2* a fait un bon travail sur les sous-titres en repérant les différents protagonistes par des couleurs. Pour en savoir plus sur comment rendre votre jeu accessible aux personnes malentendantes, je vous renvoie vers le site <http://www.deafgamers.com/>.

À moins que vous ne fassiez un jeu sur un support minuscule ou quelque chose dans le genre, vous devriez considérer que les sous-titres sont nécessaires. C'est un bon exercice de game design — comme régler la taille des caractères et ajuster le niveau sonore.

Des éléments d'interface illisibles

Santiago nous propose un autre TDC : un viseur que vous ne pouvez même pas voir. Il écrit :

La plupart des jeux vous laisse le choix d'utiliser un viseur ou non, et certains d'entre eux vous propose plusieurs réticules différents. Mais dans certains jeux, ils sont transparents ou sont une couleur qui se confond avec l'arrière-plan. Dans le pire des cas, vous finissez par ne plus savoir ou vous tirez. J'ai choisi d'utiliser un viseur, alors s'il vous plaît, faites qu'ils soient visibles à l'écran !

Half-Life Opposing Force propose des viseurs verts et transparents, une image différente pour chaque arme. Certains d'entre eux sont tellement transparents qu'on ne les voit presque plus, et si vous mettez les lunettes de vision nocturne (qui sont indispensables dans certaines parties du jeu pour pouvoir le finir), votre écran devient entièrement vert... y compris le réticule de visée ainsi que le reste de l'interface qui deviennent complètement illisibles.

Ça me semble être un problème de test. Malheureusement, la plupart des testeurs ne sont pas les personnes adéquates pour corriger des problèmes d'ergonomie d'un jeu. Ils connaissent les moindres recoins du jeu. Ils savent exactement ce qui va se passer et où tirer. Ils n'ont pas besoin de viseur... ni de mini-carte, ni de personnages qui leur donnent des conseils, ni de mode facile, ni aucune des fonctionnalités qui rend le jeu accessible à quelqu'un qui y joue pour la première fois. Les testeurs n'ont jamais remarqué que les viseurs étaient inutiles... mais quelqu'un aurait dû le faire.

J'ai assisté l'année dernière à une excellente conférence de Melissa Federoff, qui a écrit sur l'ergonomie du jeu vidéo et qui jusqu'à récemment travaillait sur ce sujet chez Microsoft Game Studios. Elle a raconté une histoire drôle sur des concepteurs qui regardaient les nouveaux joueurs essayer de traverser un niveau de jeu. La moitié du temps les joueurs finissaient par déboucher sur un cul-de-sac et tomber dans un précipice. Les level designers étaient déroutés. « Pourquoi est-ce qu'ils continuent à faire ça ? » ont-ils demandé. Melissa n'a pas pu s'empêcher de crier : « Parce que ce ne sont pas eux qui ont créé ce niveau, crétin ! Ils ne savent pas ce qu'il y a dedans et, contrairement à vous, ils n'y ont pas joué dix mille fois ! »

OK, j'exagère ; Melissa est trop professionnelle pour traiter un level designer de crétin. Elle réserve sans doute ça aux designers d'interface qui ne donnent pas au joueur les moyens de pouvoir configurer le jeu comme il l'entend. Mais quand vous faites tester un jeu, il vous *faut* quelqu'un qui n'y a encore jamais joué. Et vous devez prendre en compte ses remarques et agir en conséquence.

En parlant de reconfigurer, Ben Ashley écrit pour se plaindre des...

Réglages non sauvegardés

J'ai déjà mentionné les mauvais mécanismes de configuration dans le [cinquième article](#) de cette série, mais je n'avais pas réalisé le nombre délirant de façons de se planter sur une fonctionnalité aussi triviale.

Battlefield 2 ne sauvegarde pas votre configuration avec votre profil de jeu, donc si vous changez d'ordinateur, vous devez reconfigurer toutes les touches... et dans un jeu comme *Battlefield 2*, il y en a *beaucoup*. Et quand vous créez un nouveau profil, nous écrit Ben, ce nouveau profil réinitialise toutes les commandes. En outre, d'après le test de GameSpot, vous devez parcourir plusieurs pages pour désactiver une touche afin de pouvoir l'attribuer à une autre fonctionnalité.

Plusieurs pages ? Mais quelle horreur ! On pourrait croire que la liste déroulante n'a jamais été inventée. Mettez une liste pour *toutes* les touches déjà affectées, *toutes* les touches non affectées, et *toutes* les fonctions non associées sur *un seul* écran. De cette façon, la personne qui joue a toutes les informations dont elle a besoin devant elle pour configurer le jeu selon ses préférences. Après, laissez-la sauvegarder autant de configurations différentes qu'elle le souhaite sur sa propre machine, et s'il s'agit d'un jeu en ligne, faites-le sur le serveur, et laissez le joueur les charger à sa convenance.

Impossible de configurer les menus

Et je n'ai pas encore fini. Il semblerait que les mauvais menus de configuration rebutent encore plus les joueurs que nous le pensions. C'est pourquoi Mihail Mercuryev ajoute :

En règle générale, la plupart des menus de configuration (contrôles, sons, petits ajustements graphiques) devraient être accessibles n'importe où dans le jeu. De toute évidence, il y a des exceptions. Vous ne pouvez modifier les règles au milieu d'une course, et les changements de résolution feraient laguer votre ordinateur, ainsi que les autres joueurs en ligne, pendant quelques secondes. Certains changements nécessitent de longs calculs ou même de recharger entièrement le niveau. Mais à quoi ça sert de refaire des calculs quand vous changez juste les commandes ? Ou lorsque vous désactivez les effets de fumée ?"

Grand Prix 3 ne vous permet pas de modifier quoi que ce soit (à part le son) durant une partie. Dans *Fallout*, vous devez créer un personnage et regarder une vidéo d'introduction avant de pouvoir changer la luminosité, le volume ou activer les sous-titres. Pour *Driver*, c'est encore pire : vous êtes obligé de finir la première mission (censée déterminer votre niveau) avec les contrôles de base, et après seulement vous pourrez les modifier. *Counter-Strike* vous oblige à vous reconnecter lorsque vous changez la luminosité.



Heu... La luminosité ? Je dois me déconnecter du jeu parce que je veux régler la *luminosité* ? Qui a eu cette idée ?

Vous devez vous dire "Et alors ? Ces histoires de configuration sont complètement triviales comparées aux problèmes d'équilibrage du gameplay." Oui, et c'est pour ça qu'il n'y a aucune excuse à ne pas le faire correctement ! Concevoir une bonne configuration n'est pas la partie la plus passionnante du travail d'un game designer, mais c'est important pour les joueurs, car s'ils ne peuvent pas configurer le jeu comme bon leur semble, vous allez leur donner une mauvaise expérience du jeu même si le gameplay est bien conçu. Je ne devrais pas avoir à prouver que je joue bien avec vos stupides configurations avant de pouvoir en créer une moi-même, parce que au cas où vous ne l'auriez pas remarqué, c'est moi qui joue, pas vous !

Faites votre travail correctement — y compris les parties ennuyeuses — ou démissionnez et donnez le à quelqu'un d'autre qui en sera capable. Il y a des milliers de concepteurs volontaires qui seraient heureux de prendre la relève après vous.

Oubli du principe des vases communicants

Je terminerai avec un problème étrange qui nous est également signalé par Mihail Mercurjev. Nous sommes habitués à des compromis étranges avec les lois de la physique — des voitures qui s'arrêtent net en heurtant un banc de parc indestructible, etc. Un jour, nous disposerons d'ordinateurs suffisamment puissants pour modéliser ces lois correctement. Mais l'erreur de level design dont Mihail nous parle n'a rien à voir avec la puissance des ordinateurs. Il écrit :

Dans certains jeux (*Quake* premier du nom, *Duke Nukem 3D*), vous plongez dans l'eau, et vous en ressortez quelques mètres en dessous. Si c'est devenu une convention, oubliez-la et commencez à étudier la physique! Quand des réservoirs communiquent, l'eau devrait s'élever au même niveau de chaque côté (à moins qu'il ne s'agisse d'une cavité qui emprisonne une bulle d'air, comme une cloche de plongée).

Je dois admettre que je n'y avais jamais pensé, mais il a bien raison. Si vous plongez sous l'eau au point A et que vous ressortez au point B, A et B doivent être à la même altitude parce que la physique l'exige [NDT : pour les curieux, ceci est dû à la loi fondamentale de l'hydrostatique ou [loi de Pascal](#)]. Si vous plongez vers un niveau inférieur et ressortez à l'air libre plus bas... pourquoi cet endroit ne se remplit pas simplement d'eau ?

Un jour, peut-être que nous pourrions simuler la physique de l'eau de façon réaliste plutôt que se contenter d'un filtre bleu sur la caméra, et alors nous aurons réussi [NDT : l'article original date de 2006]. En attendant, chers level designers, si vous voulez éviter la fureur de Mihail, souvenez-vous de la loi des vases communicants. Ou situez votre jeu dans une cloche de plongée.

Conclusion

Mes remerciements à Mahdi Jeddi, David Peterson, Santiago Hodalgo, Mihail Mercurjev, et Ben Ashley pour cette avalanche annuelle de TDC, et souvenez-vous, *vous seul* pouvez empêcher ces erreurs de design stupides. Et si vous en trouvez d'autres — même si, ou plutôt en particulier si elles sont tirées d'un jeu génial — faites nous en part.

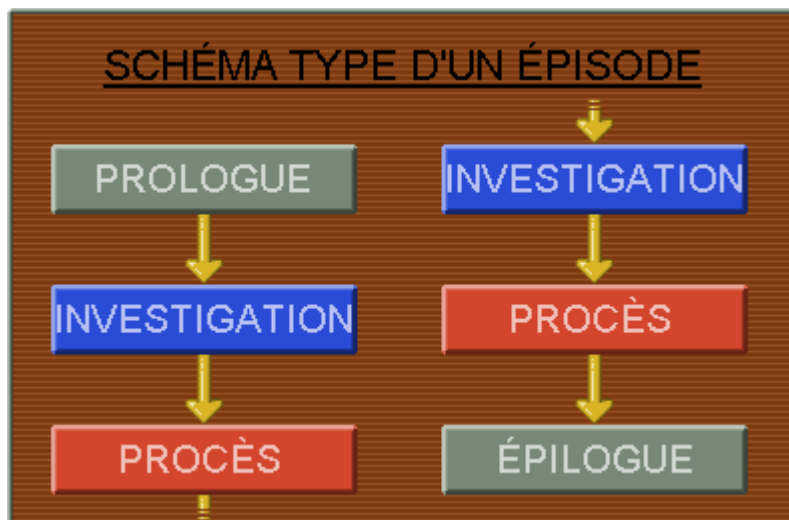
Game Design & Ace Attorney

Par [Lyonsbanner](#) | le 21 juillet 2012 16:34:32 | Catégories : [Analyse](#), [Game Design](#)

Si je vous demande de me citer un avocat de la défense habillé en bleu, avec une tête de hérisson et qui crie « OBJECTION ! » à tout bout de champ, vous me répondez immédiatement : Phoenix Wright.



Les *Ace Attorney* sont une série de jeux d'enquête découpés en deux phases bien distinctes : l'investigation et le procès. Dans un premier temps, le protagoniste, avocat de son état, va sur le terrain pour enquêter, récupérer des indices et des preuves et faire la lumière sur les circonstances du crime. Dans un deuxième temps, il revient au tribunal pour mener interrogatoires et contre-interrogatoires, affronter le procureur général (et le juge !) dans le but de prouver l'innocence de son client.



Des jeux avec des énigmes, il en existe à la pelle, comme les *Myst* ou la série *Sherlock Holmes*. Mais des jeux où l'on incarne un avocat de la défense, personnellement, je n'en avais jamais vu. C'est là l'innovation de Capcom : en plus de devoir résoudre le mystère, il faut être capable de prouver l'innocence de son client.

Wikipédia décrit la série des *Ace Attorney* comme des jeux d'aventure. Ce n'est pas entièrement faux mais il n'y a pas que ça. Si l'on s'en tient à la première phase de jeu, l'enquête, on peut considérer qu'il s'agit d'un jeu d'aventure au même titre qu'un *The Longest Journey* ou un *Grim Fandango*. Mais la deuxième phase, celle qui fait tout le sel et l'originalité du jeu, ne rentre pas du tout dans cette catégorie : elle s'apparenterait plus du fameux duel d'insultes des *Monkey Island* et s'oriente beaucoup plus vers le dialogue que la recherche.

Première phase : l'enquête

Des deux phases qui constituent le jeu, l'enquête est la moins importante. Elle est conçue pour amener et préparer la phase de procès. Elle expose rapidement les personnages, les lieux, les objets et les enjeux et permet au joueur de réunir les indices et objets nécessaires pour le procès à venir.

La plupart des jeux d'aventures sont basés sur l'exploration, la recherche d'objets, la possibilité d'agir sur des éléments ou de résoudre des casse-têtes que le joueur peut rencontrer (découvrir un nouveau lieu, ouvrir une porte avec une clef, tirer sur un levier, trouver la solution d'une énigme, etc.), comme dans *Atlantis* ou *Myst*. Les dialogues peuvent se faire rares voire inexistantes et on touche parfois au désert (le poétique *L'Amerzone* est terriblement dépeuplé, d'où son ambiance profondément mélancolique).

Dans *Ace Attorney*, la phase d'enquête est au contraire réduite à l'essentiel et est conçue pour mettre en avant le dialogue au détriment de l'exploration. Tout est fait pour que le joueur avance aussi vite que possible afin de se concentrer sur les dialogues et sur la phase de procès qui va venir. Le joueur évolue dans des décors fixes, sans déplacements (ce qui n'est pas le cas pour un *Simon the sorcerer* par exemple, où le personnage met dix secondes pour aller d'un bord à l'autre de l'écran). Un simple clic suffit pour examiner les objets, examen qui donne le plus souvent lieu à un dialogue – hilarant – entre les personnages. L'interface est réduite au minimum (cinq boutons à tout casser) pour pouvoir laisser le joueur se concentrer sur les dialogues qui sont longs, très animés et nombreux. Une fois écouté, le dialogue est coché et peut-être réécouté en accéléré, ce qui permet au joueur de revenir rapidement au passage qui l'intéresse. Pour être sûr que le joueur sente qu'il est dans la bonne direction et qu'il aille plus vite, les développeurs utilisent un petit effet visuel et sonore : lorsque quelque chose a changé dans un décor (disparition ou apparition d'un personnage, nouvel objet, etc.), une boîte de dialogue affiche le nom du lieu, la date et l'heure avec un effet machine à écrire (et le bruitage associé, bien sûr). Tout est fait pour remettre le dialogue au centre du gameplay et accélérer le rythme de jeu : l'important, c'est d'aller jusqu'au procès !



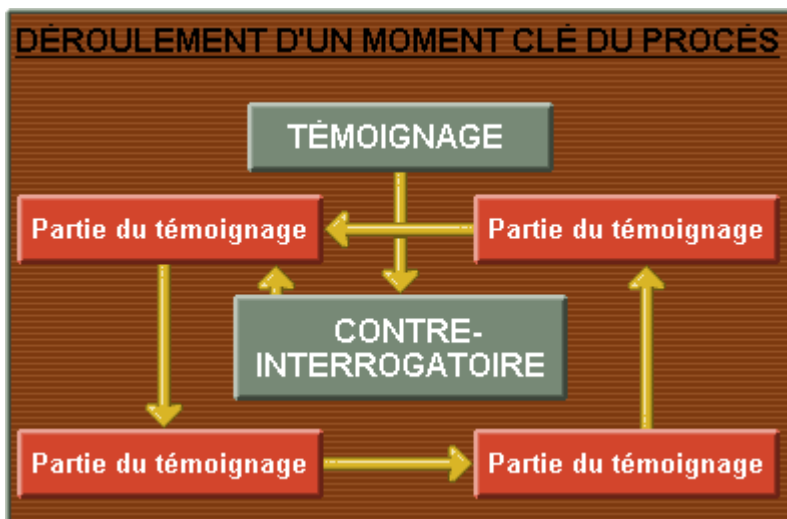
Toutes les fonctionnalités du classique jeu d'aventure ont été liftées pour pouvoir coller au cœur du jeu : vous incarnez un avocat, d'où un gameplay centré sur le dialogue et sur le procès. L'inventaire, par exemple, est réduit au plus simple : pas de combinaison hasardeuses d'objets (Dieu merci !), de jolies icônes avec une description brève et efficace, pas besoin de faire de grands efforts pour se rappeler comment il l'a obtenu. Plutôt que de s'embarquer sur des fonctionnalités trop complexes avec les objets, les game designers ont choisi de les insérer au maximum dans les dialogues. Les objets ne doivent pas, la plupart du temps, être utilisés sur le décor mais montrés à des personnages comme preuves. La moindre utilisation donne lieu à un dialogue plutôt qu'à une animation. Cette direction se retrouve jusque dans la direction artistique du jeu : les décors sont peu nombreux et extrêmement basiques, alors que les personnages ont droit à de grands portraits (les pixel artistes de Capcom ont fait des merveilles) et de très nombreuses animations pour émailler les

dialogues ; la bande-son est ponctuée de nombreux bruitages pour rythmer les conversations avec les personnages et l'ensemble en retire un côté anime pas déplaisant du tout.

Deuxième phase : le procès

Passons maintenant au procès. Le (relatif) sentiment de liberté que le joueur a pu connaître durant la phase d'enquête se volatilise, car il ne peut tout simplement plus se déplacer, il est obligé de se plier aux règles, confiné dans une salle où règne l'ordre et la justice, j'ai nommé : le tribunal. Le joueur n'est bien évidemment lâché comme ça dans l'arène : le procureur présente les faits (« Le suspect est accusé d'avoir tué la victime à tel endroit et telle heure. ») et l'inspecteur nous détaille le rapport d'autopsie. Enfin, le témoin oculaire raconte sa version des faits. Si le joueur a décidé de faire une longue pause avant d'entamer la phase de procès, c'est aussi une excellente occasion pour lui de se remettre dans le bain.

Et c'est là que vous entrez en scène ! Eh oui, la présomption d'innocence n'a pas cours dans le monde cruel des *Ace Attorney* et c'est à vous de réussir à disculper votre client. Il est possible que le témoin mente, qu'il ait dit quelque chose d'incohérent ou qu'il ait oublié un élément capital lors de sa déposition. Le témoignage prend la forme d'une boucle dont vous ne pourrez sortir qu'en trouvant la bonne réplique ou la bonne preuve et en la présentant au moment adéquat.



Pour cela, le joueur peut choisir de faire pression sur le témoin, de le « cuisiner », ce qui permettra de lui faire révéler de nouvelles informations qu'il ajoutera à son témoignage, ou de lui présenter une preuve qui démontre que sa déposition est incohérente. Si vous trouvez la bonne option, vous progressez dans le procès et sinon, vous verrez la jauge de patience du juge diminuer. Arrivé à zéro, vous serez confrontés à une nouvelle version du classique game over : **Coupable**.

Pourquoi avoir mis en place un tel système ? Il y a plusieurs raisons à cela. Pour commencer, imaginez ce qui aurait pu se passer s'il n'y avait pas cette barre : le joueur doit montrer un élément du dossier, il lui suffit donc de présenter bêtement un par un tous les objets qu'il possède. En ajoutant cette jauge, on force le joueur à s'arrêter un moment et à réfléchir à la situation. Autrement dit, on fait monter la pression !

Cette pression, on va la retrouver partout dans le jeu et c'est elle qui va accaparer toute l'attention du joueur, non seulement quand le juge met une amende mais également quand le joueur doit faire face à un choix, quand il se demande quelle preuve pourra faire l'affaire dans une situation donnée, quand le procureur revient à la charge, etc.

La musique joue ici un rôle très important. Si la bande-son des phases d'enquête est tout ce qu'il y a de basique, les morceaux conçus pour accompagner les phases de procès ont bénéficié d'un soin remarquable. A chaque avancée dans le procès, à chaque preuve que vous présentez, à chaque coup que le procureur vous assène (très souvent littéralement), la musique accélère. La piste qui accompagne le procès est décomposée

en différents mouvements correspondant à des rythmes différents et qui vont servir à faire monter ou à faire redescendre la pression en accélérant ou en ralentissant le tempo. On peut rien qu'à l'oreille savoir à quelle phase du procès on se situe. Et ce n'est pas tout ! À partir du moment où le joueur présente la bonne preuve, la tension atteint son paroxysme : la musique s'arrête net. On respire un bon coup, satisfaction du joueur qui se dit : « Ouais, c'est ça, j'ai trouvé ! ». On se rapproche d'un cran de la vérité et bien entendu, le procureur ne va pas se laisser faire et va tenter de faire voler vos raisonnements en éclats.

L'aspect graphique de la direction artistique vise aussi à installer cette narration forte qui fait l'intérêt des procès dans le jeu. Dans la plupart des jeux d'aventure, on peut deviner si un personnage est content, triste ou en colère en fonction de l'intonation du doubleur. Mais dans *Ace Attorney*, point de doublage (sur une DS, c'est pas forcément évident). Capcom a pallié ce manque en concentrant les efforts des graphistes sur les sprites des personnages : ceux-ci sont très grands, très soignés et disposent d'animations fluides et variées (on sent l'expertise *Street Fighter*). Certaines expressions et mimiques (souvent accompagnées de bruitages) sont volontairement exagérées afin de souligner les passages dramatiques. Les seules voix utilisées ont été réservées pour les points culminants de la narration, les moments où Phoenix lâche son célèbre « OBJECTION ! ». Par ailleurs, l'interface du jeu n'est plus du tout la même que dans les phases d'exploration. On voit le protagoniste à la troisième personne et non plus à travers ses yeux. L'écran affiche tour à tour différents angles de vue du tribunal et le passage de l'un à l'autre permet de mieux immerger le joueur : les mouvements de caméra, en s'accéléralant (voire en devenant instantanés), soulignent le rythme de la narration au même titre que la musique et donnent un côté cinématographique (presque similaire à une série) à l'ensemble.

Alors, il est parfait ce jeu ?

Ace Attorney a beau être une version rajeunie du jeu d'aventure textuel, il n'en souffre pas moins de certains défauts.

Le scénario est parfaitement linéaire, qu'il s'agisse de la phase d'enquête ou de la phase de procès. Même si le joueur a l'impression qu'il est libre de visiter n'importe quel lieu à n'importe quel moment lors de la phase d'enquête, il doit cependant épuiser tous les sujets de conversations d'un protagoniste avant d'aller pouvoir parler au suivant. Cette linéarité se manifeste aussi par l'obligation de présenter la bonne preuve au bon moment. Il arrive parfois que la logique du scénario échappe au joueur ou pire, que celui-ci prévoit le dénouement (pas forcément du procès, mais d'une séquence au moins) et essaye de brûler les étapes : le jeu n'est pas conçu pour ça et l'intrigue, pourtant brillamment ficelée, révèle son caractère « construit », artificiel, linéaire.



Cette linéarité est par ailleurs la cause logique d'une mauvaise rejouabilité : il y a relativement peu d'intérêt à refaire les enquêtes une fois celles-ci découvertes (ou du moins pas avant un long moment) car leur intérêt réside principalement dans leur intrigue et dans les rebondissements dont elles sont émaillées. Il n'y a pas ici un style d'écriture ou une finesse d'observation sociologique qui justifieraient une relecture comme pour les romans d'Agatha Christie (du moins, pas dans les traductions française et anglaise du jeu).

Affaire classée

Les *Ace Attorney* ont su apporter du renouveau au genre du jeu d'aventure en faisant le choix de se recentrer sur le dialogue et la discussion : en utilisant à bon escient des fonctionnalités de gameplay (dialogues accélérés, absence de déplacements) et en proposant une direction artistique d'une économie remarquable (efforts concentrés sur les personnages et non les décors, utilisation de la musique comme un véritable métronome.), Capcom a su créer un rythme et une rapidité de jeu hors norme et à éviter les travers du genre (bavardage, personnages transparents, etc.).

Mon analyse se termine ici. Comme c'est mon premier article, j'espère qu'il vous aura plu et qu'il vous aura ouvert de nouvelles perspectives sur le game design. La saga des *Ace Attorney* est un retour aux sources du jeu d'aventure et elle ne semble pas prête de s'arrêter.

Huit façons de faire un mauvais tutoriel

Par [Lyonsbanner](#) | le 16 août 2012 22:00:00 | Catégories : [Analyse](#), [Ernest Adams](#), [Game Design](#)

Cet article est une traduction de [Lyonsbanner](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juin 2011).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

À l'aube de l'industrie du jeu, il y avait des jeux vidéo (sur console ou arcade) et des jeux sur ordinateur. Avec les jeux vidéo, vous vous retrouviez directement dans le grand bain : seul face à une horde d'ennemis avec très peu d'instructions, soit vous couliez soit vous nageiez. La plupart du temps vous couliez, ce qui constituait par ailleurs la source de revenus des salles d'arcade.



Les jeux sur ordinateur étaient plus complexes que les jeux d'arcade, c'est pourquoi des manuels étaient fournis aux joueurs avant de commencer à jouer. De nos jours, on s'attend à ce que les joueurs ne lisent pas le manuel, du coup, il leur faut un tutoriel à la place. Les tutoriels présentent au joueur l'interface et le gameplay. Ils doivent expliquer au joueur comment interagir dans l'univers du jeu, comment atteindre l'objectif principal, et (brièvement) pourquoi.

Récemment, j'ai eu le privilège de siéger en tant que juré à l'*Extra Credits Innovation Awards*, ce qui signifiait que je devais jouer – et donc, apprendre à jouer à plusieurs jeux rapidement.

Un ou deux tutoriels étaient si mauvais que j'ai décidé que nous ferions mieux d'en parler. Les mauvais manuels et/ou les mauvais tutoriels sont déjà des TDC [NDT : *Twinkie Denial Condition*, voir articles précédents de la série], mais les « Vilain Game Designer, Pas de Biscuits pour toi ! » que j'ai publiés ne sont pas aussi détaillés que ça.

Les tutoriels existent pour enseigner des choses au joueur, et les game designers ne sont pas des enseignants par nature. Nous sommes habitués à créer des challenges, pas à en expliquer les principes. Pendant la majeure partie du jeu, on attend des joueurs qu'ils apprennent par eux-mêmes grâce à l'observation et à l'expérimentation.

Comme l'a souligné Raph Koster, la partie la plus fun du gameplay est d'apprendre à maîtriser le jeu, mais ce procédé est inefficace. Les tutoriels ne doivent pas fonctionner comme ça. Ils doivent expliquer aux joueurs ce qu'il faut faire et leur montrer ce qui se passe à l'écran quand ils font quelque chose. Il faut laisser les joueurs maîtriser les bases du jeu – les guider dans le petit bain plutôt que de les faire plonger dans

l'inconnu. Mais j'ai récemment découvert qu'il y a plusieurs manières de mal le faire. En voici quelques unes.

Forcer le joueur à faire le tutoriel.

À chaque fois qu'un joueur tombe sur un game over, faites-lui refaire le tutoriel. Même s'il y a déjà joué une douzaine de fois auparavant. Lapidez-le à coup d'explications sur des choses qu'il sait déjà. Irritez-le avec des défis inutilement ennuyeux. Faites-lui perdre 10 ou 20 minutes de son temps avant qu'il puisse s'amuser à nouveau.

Les jeux contiennent souvent des tutoriels inévitables parce qu'ils sont intégrés dans le premier voire le deuxième niveau. Ce n'est pas mauvais en soi si le joueur peut sauter les conseils, ou les interrompre. Mais inonder le joueur d'une montagne d'informations qu'il connaît déjà est fastidieux et ennuyeux.

La solution la plus simple pour résoudre ce problème est de rendre le tutoriel optionnel, de le séparer du jeu en lui-même. Ça fonctionne pour beaucoup de concepts de jeux. Que ce soit des soldats, des athlètes, des pilotes, ou dans notre cas, des rois et des urbanistes ; tous ont besoin d'une formation avant qu'ils puissent faire leur vrai travail. Si vous voulez vraiment que votre tutoriel fasse partie du jeu principal, assurez-vous que le joueur ait la possibilité de passer les instructions afin qu'il puisse jouer au vrai jeu immédiatement.

Donnez au joueur beaucoup de lecture.

Donnez-lui des pages et des pages et des pages de notice d'utilisation à lire, en ne lui laissant rien faire d'autre que d'appuyer sur un bouton pour passer d'un texte à un autre. Écrivez-le avec un langage pseudo-médiéval rempli d'anachronismes, ou pire encore, dans un monologue dicté par un personnage mentor ennuyeux avec beaucoup de propos irritants. (« Le bouton A permet de brandir votre épée! Essayez maintenant. Oui, cela est fort bien. ») Affichez le tout avec une police de caractères moche qui a été initialement prévue pour les titres ou les sous-titres, mais pas pour les paragraphes.

J'ai joué une fois à un jeu japonais où le tutoriel consistait à appuyer sans arrêt sur un bouton pour faire défiler le texte pendant 10 minutes. Bien sûr, une fois arrivé à la fin, j'en avais oublié la moitié. Les joueurs s'en souviendront mieux s'ils apprennent par la pratique.

Tant que j'y suis, ne faites pas lire au joueur une énorme quantité de texte sur ce qui s'est passé dans l'histoire non plus. Le défilement du générique d'introduction du tout premier Star Wars ne dure qu'une minute et 14 secondes, de « C'est une époque de guerre civile » à « restaurer la liberté dans la galaxie ». Si c'était suffisant pour George Lucas, c'est suffisant pour vous. Laissez les joueurs découvrir le reste du background par eux-mêmes.



Ecran de Ville

Votre ville est un Château et vous voyez donc l'écran de ville caractéristique des Châteaux. Il y a huit types de villes différentes de Château dans Heroes III, et chacune dispose de bâtiments et de créatures différents. La

Mauvaise description des boutons et des éléments de menu.

Vous pouvez rendre votre tutoriel encore plus agaçant en incluant des références à un bouton sans clairement l'identifier. Par exemple, dire « Appuyez sur le bouton Vendre situé en haut à gauche de l'écran » quand il y a cinq boutons différents à cet endroit et qu'aucun d'entre eux ne s'appelle Vendre. De même, dites-leur de choisir l'élément X du menu quand cet élément X du menu n'est pas visible. Mettez cet élément X du menu un ou deux niveaux plus bas dans l'arborescence sans aucune instruction pour pouvoir le trouver.

Les boutons avec des icônes dessus prennent bien moins de place à l'écran que ceux avec du texte, et ils n'ont pas autant besoin d'être signalés, c'est donc généralement une bonne idée. Cependant, le joueur a besoin d'apprendre ce que le bouton signifie et ce qu'il fait, et le tutoriel doit le montrer.

Quand votre texte se réfère à un bouton, profitez-en pour mettre en évidence ce bouton à l'écran – mettez une lueur dorée tout autour ou quelque chose comme ça, et faites-le avec un effet pulse ou flash.

Si vous voulez faire référence à quelque chose qui se trouve dans l'arborescence du menu, assurez-vous d'indiquer clairement le chemin. Ne dites pas : « Vous pouvez le trouver dans l'onglet Graphiques du menu de personnalisation », mais dites : « Sélectionnez Aide > Préférences > Personnaliser et choisissez l'onglet Graphiques ». Ça peut sembler évident, mais vous seriez surpris de voir que peu de jeux le font.



Sauter les étapes.

Faites faire au joueur un tutoriel avec des étapes, et brutalement, en plein milieu, arrêtez de lui donner des instructions. Laissez-le en plan face à un écran rempli de boutons et de menus en faisant en sorte qu'il n'ait aucune idée de ce qu'il est censé faire. Quand il essaye de faire quelque chose après un moment d'hésitation, affichez à nouveau des instructions mais qui n'ont aucun rapport avec ce qu'il vient de faire.

Certains tutoriels ont besoin d'être plus approfondis que d'autres. Plus le jeu est inhabituel, ou le genre obscur, plus vous devez expliquer ce qui se passe. Je ne suis pas contre les tutoriels limités, mais contre les tutoriels qui commencent par fournir des explications et qui soudain disparaissent ensuite. Quel que soit le nombre d'explications, vous devez en mettre jusqu'au bout.

Punissez le joueur inexpérimenté.

Quand le joueur fait la moindre erreur, ramenez-le en arrière et réexpliquez-lui tout dans les moindres détails – ou faites en sorte qu'il revienne en arrière lui-même. Faites monter son niveau de frustration en le punissant de petites erreurs qu'il aura accumulées.

Quand j'essaie d'apprendre comment sauter dans un jeu, s'il y a bien une chose que je déteste c'est de tomber dans un trou profond où il me faut deux bonnes minutes pour remonter avant de pouvoir tenter le saut à nouveau. C'est assez mauvais dans un gameplay normal, mais c'est inexcusable dans un tutoriel.

Quand un joueur rate son saut, ou toute autre action de ce genre, il doit être en mesure de pouvoir réessayer immédiatement.

Un des jeux auquel j'ai joué lors de l'*Innovation Awards* était un tutoriel dans lequel vous perdez ! Le but que j'essayais d'atteindre n'était pas très clair, mais j'ai essayé de faire ce qu'on me disait, et à la fin, ça me disait que j'avais perdu. Je peux difficilement imaginer un début de jeu plus décourageant. Finir le tutoriel devrait donner au joueur un sentiment de satisfaction. Si le joueur échoue dans le tutoriel, alors le tutoriel lui-même est un échec.

Baby-sittez ou humiliez votre joueur.

Dites au joueur « Très bien ! » avec une voix exagérément guillerette quand il a parfaitement compris comment faire pour appuyer sur la touche Entrée. Faites-le à chaque fois qu'il fait ça. Glorifiez-le comme vous le feriez à un enfant de quatre ans qui vient d'apprendre comment boutonner sa propre chemise. D'un autre côté, humiliez votre joueur quand il fait quelque chose de mal. Dites-lui à quel point il est nul. Les nouveaux joueurs adorent être sévèrement critiqués pour ne pas savoir ce que le tutoriel est censé leur apprendre.



Ce point ne nécessite pas plus d'explications. Un tutoriel, émotionnellement parlant, devrait être positif et encourageant sans être condescendant. Dans la vie, la critique constructive est utile, mais il ne faut pas en abuser.

Forcez le joueur à faire tout le tutoriel.

Il est possible que durant votre tutoriel, le joueur sente qu'il a compris ce qu'il faut faire et qu'il soit prêt à aller de l'avant, et donc à commencer le jeu. Oh, non. Pas de chance. Vous avez fait tellement d'efforts dans ce tutoriel et il va devoir le finir qu'il le veuille ou non. Après tout, c'est votre jeu, pas le sien.

Si vous avez un niveau qui est entièrement un tutoriel, laissez le joueur le quitter pour qu'il puisse passer à autre chose. Si votre tutoriel est intégré dans un niveau ordinaire, laissez le joueur sauter les passages du tutoriel.

Et pour finir...

Ne leur donnez pas de tutoriel du tout.

Concevez ce que vous pensez être une interface totalement intuitive et ingénieuse, de préférence sans HUD et autres indicateurs à l'écran qui empêchent le joueur de s'immerger complètement dans votre magnifique univers. Faites-le apparaître clairement sans qu'il y ait besoin d'instructions. Jetez-les dedans et souhaitez-leur bonne chance.

Un des jeux que j'ai eu à juger à l'*Innovation Awards* faisait exactement ça. L'interface utilisateur m'a écœuré – un jeu en 3D à la première personne où la vue bougeait comme si j'étais ivre, apparemment des efforts ont été fait pour transmettre l'idée que mon avatar se sentait malade et désorienté. Ça marchait si bien que je ne m'y suis pas attardé. Il m'a aussi fallu cinq minutes d'expérimentations avec cette interface utilisateur sot-disant intuitive rien que pour savoir comment ouvrir une porte.

L'interface utilisateur entièrement intuitive a été le Saint-Graal de l'interaction homme-machine pendant des décennies. Je vais être franc : ça n'existe pas. Sauf si vous faites un jeu sur ordinateur sur comment utiliser un ordinateur, les périphériques d'entrée de la machine ne sont pas susceptibles de correspondre à l'activité du monde réel que le jeu simule.

J'ai travaillé sur *Madden NFL* aussi loin que remonte l'ère de la Sega Genesis, et je peux vous dire que choisir un type de jeu, puis taper dans le ballon, le passer à un coéquipier, l'attraper et courir avec, ce n'est pas rien avec trois boutons et une croix directionnelle. Nous avons réussi – mais nous avons encore à écrire le manuel du jeu. Il n'y avait pas assez de place dans une cartouche de Genesis pour un tutoriel.

L'arrivée de la Wiimote et, plus récemment, de Kinect remet en avant ce sujet, et certaines personnes sont très enthousiasmées par des interfaces du style *Minority Report*. Ces gens ont oublié – ou n'ont probablement jamais su – pourquoi la souris d'ordinateur a été inventée avant le reste. Les écrans tactiles sont épuisants à utiliser à la longue, car il faut garder le bras levé tout le temps. Bouger ses bras dans tout les sens, comme dans *Minority Report*, est encore plus fatigant. Ça a l'air très excitant et cool, mais on ne peut pas faire ça pendant des heures en fin de compte.

Les joueurs n'ont pas besoin d'un tutoriel s'ils disposent d'un équipement spécial, bien sûr – une manette de jeu haut de gamme pour les simulateurs de vol (mais ils vont encore appuyer sur une touche du clavier pour faire sortir le train d'atterrissage), ou les Bongos DK pour jouer à *Donkey Konga*. Mais vous ne pouvez pas compter sur le fait que le joueur possède son propre équipement, à moins que vous ne les fournissiez avec le jeu, et ceci signifie qu'il a toujours besoin d'un tutoriel ou d'un manuel pour une manette ordinaire.

En d'autres termes, votre interface utilisateur doit associer les contrôles de la machine à l'univers du jeu et en à un moment ou un autre, d'une façon ou d'un autre, vous allez devoir expliquer ces associations. Et votre interface ne sera certainement pas aussi intuitive que vous le pensez. Vous jouez à votre propre jeu pendant des mois ou des années, ça devient une seconde nature pour vous, mais pour les joueurs, c'est nouveau. Ils ont besoin d'un tutoriel. J'insiste.

Conclusion

Certains d'entre vous ont sans doute remarqué que ce n'est pas une critique très constructive. J'ai identifié un bon nombre de choses à ne pas faire, mais je n'ai pas expliqué comment faire un bon tutoriel. C'est parce qu'il est très difficile de créer des règles générales dans un domaine aussi diversifié que le nôtre. Un tutoriel pour un jeu de tir sera forcément différent de celui d'une simulation de construction et de gestion. Le seul conseil positif que je peux vous offrir et qui couvre tout les cas de figure est : assurez-vous de faire tester votre tutoriel par des personnes qui sont complètement novices – et profitez en pour leur faire tester le jeu complet, tant qu'on y est.

Éléments d'anatomie

Par [Pierre Guyot](#) | le 10 septembre 2012 01:48:02 | Catégories : [Cours](#), [Dessin](#), [Graphisme](#)

Si vous vous intéressez à la création de jeux sur le plan graphique, vous risquez très fortement d'avoir un jour ou l'autre à dessiner des êtres humains ou du moins des créatures dont l'anatomie s'en rapproche. Certains me diront : « Je dessine dans un style particulier, pourquoi apprendre l'anatomie ? ». D'une part, on ne sait jamais d'avance si on ne sera pas amené à un moment ou un autre à devoir réaliser des illustrations « réalistes ». D'autre part, connaître et maîtriser les bases théoriques du dessin « classique » permet de mieux styliser, même pour des styles très éloignés du réel.



Cet article vise à dégager quelques conseils pour se lancer dans l'apprentissage de l'anatomie. Il est volontairement simple et se limite à ce qui me semble être l'essentiel pour débiter : j'ai essayé d'y mettre ce que j'aurais voulu qu'on me dise quand j'ai commencé à dessiner. Pour un cours plus poussé, je vous invite à jeter un œil aux références proposées dans la conclusion de cet article. Je tiens à préciser que cet article n'est pas un « cours » à proprement parler et qu'il s'y trouve sans doute un certain nombre d'oublis (et d'erreurs), j'essayerai donc de le mettre à jour aussi régulièrement que possible :)

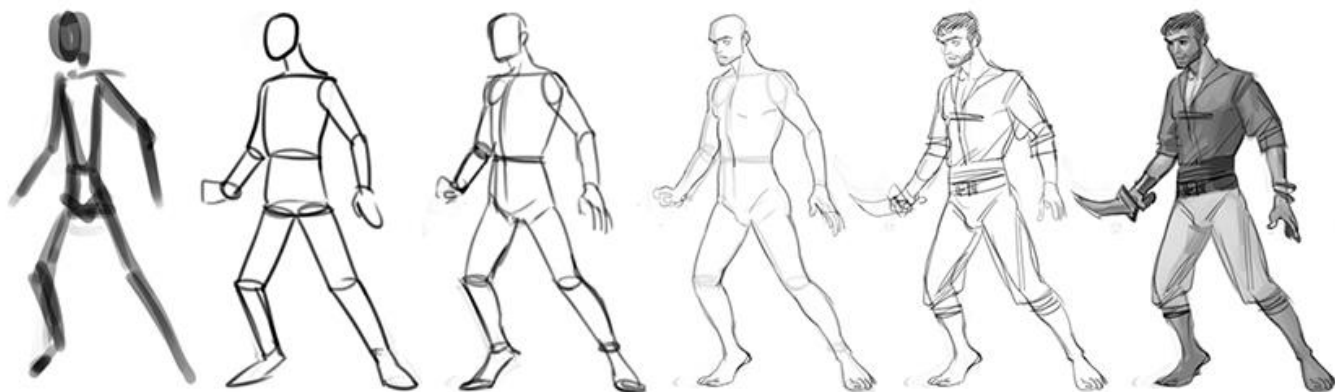
Du plus simple au plus complexe.

Je vois souvent des gens qui, pour s'entraîner à l'anatomie, essaient de copier des références photographiques jusqu'à avoir une ressemblance parfaite. Je pense que ce n'est pas une très bonne façon de s'y prendre. En raisonnant de cette façon, on ne cherche qu'à imiter ce qu'on voit sans comprendre comment ça fonctionne. Si vous essayez de travailler l'animation de personnages, cette méthode ne vous expliquera pas comment le corps humain bouge, se déplace, se déforme. Par ailleurs, ceci ne vous donnera pas une

bonne notion de volume mais seulement un bon rendu 2D. En général, ça se repère très vite : les dessins réalisés avec référence photographique sont beaucoup moins bons que ceux réalisés sans et on a souvent l'impression d'un collage, d'un « effet 2D » mais pas d'un bon volume (ça aboutit souvent à ce que les anglophones appellent « flat face »). Non pas que les références photographiques soient inutiles (on n'a pas tous les jours la chance de disposer d'un modèle vivant), mais elles doivent être utilisées à bon escient.

Une autre façon de procéder que j'ai souvent vue sur des forums de graphisme, c'est d'essayer d'apprendre le nom et les emplacements de tous les muscles et de tous les os du corps humain. Ce n'est pas absurde comme façon de procéder, c'est très analytique mais ce n'est peut-être pas une bonne façon de commencer. Souvent, les gens qui commencent par là mettent beaucoup de temps avant de pouvoir construire une silhouette correcte et on a l'impression qu'ils accumulent tous les muscles du corps sans réussir à les faire « tenir » sous la peau. Et généralement, c'est un peu aride de dessiner des pages et des pages de rotules avant de s'autoriser de dessiner le moindre bonhomme (ça peut paraître caricatural, mais un bon nombre de sketchbooks de débutants commence de cette façon sur de gros forums de graphisme comme conceptart.org).

Je crois (et je le répète, c'est un point de vue qui n'engage que moi) qu'avant de se lancer dans l'anatomie au « sens médical » du terme, il faut d'abord s'intéresser à la question des proportions et de la silhouette. D'une manière générale en dessin (et dans beaucoup d'autres domaines), c'est une bonne chose de partir du plus simple pour aller au plus complexe. Ainsi, avant de vouloir s'attaquer à un rendu photo-réaliste ou à une connaissance pointue de la structure du corps humain, il est plus important de savoir poser une silhouette simple, mais bien proportionnée. Le dessin suivant sert à illustrer ce processus : je pars d'un *stickman* tout bête qui va devenir un mannequin, puis se doter de muscles, d'une peau, de cheveux et de poils, de vêtements et finalement, d'ombre et de lumière.



[Grande version](#)

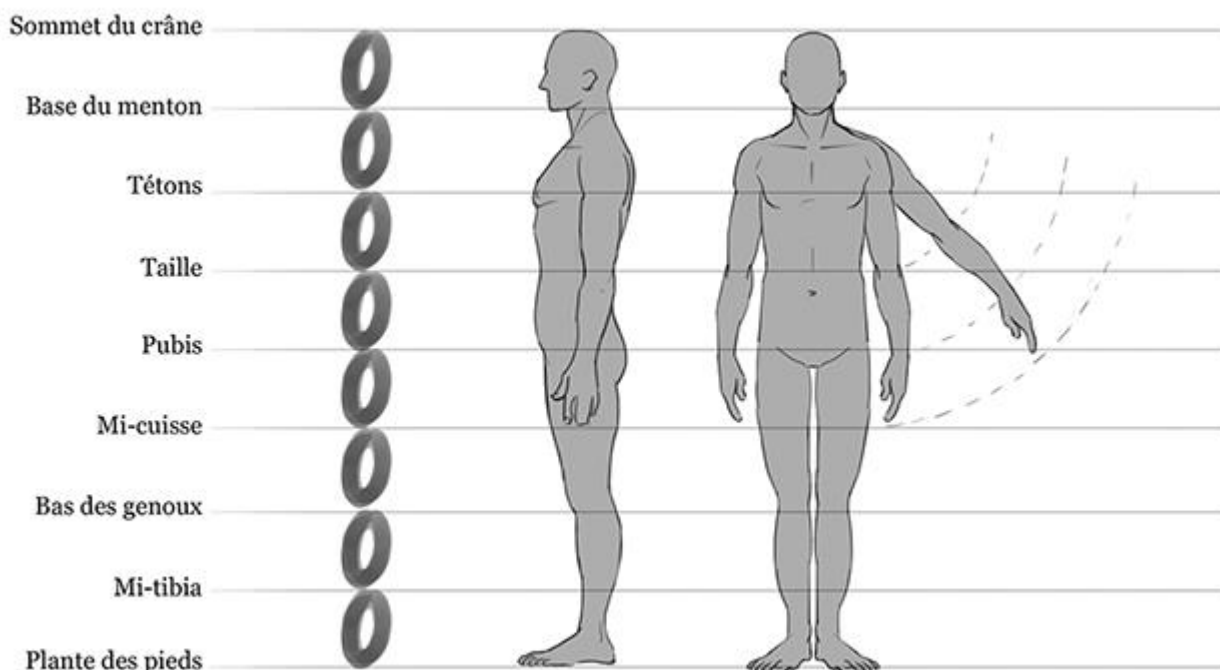
Proportions et repères.

Je vais présenter ici un certain nombre de rapports de proportion qui se conservent plus ou moins dans le corps humain. Il s'agit d'un modèle classique, qui n'est pas forcément universellement vérifié (il est un peu idéalisé) mais qui donne de bonnes bases pour commencer à apprendre. Les règles qui suivent ne sont pas absolues et c'est à vous de jouer avec pour réussir à produire de la variété dans les personnages que vous dessinez.

Le corps.

Pour découper le corps verticalement, le plus simple consiste à le décomposer en huit têtes. Vous me demanderez pourquoi huit et vous avez même peut-être entendu parler d'un canon anatomique à sept têtes. Il ne s'agit pas d'un absolu : on trouve des gens qui en hauteur font six ou neuf têtes et vous serez sans doute

amené à tricher avec ces divisions si vous dessinez des personnages fantastiques. Mais l'avantage du modèle à huit, c'est qu'il est facile de diviser un segment en huit : vous placez un segment vertical dans votre décor, qui donne la hauteur de votre personnage, vous le découpez en deux, puis vous recoupez en deux chaque segment et vous recoupez encore. Diviser en sept, c'est nettement plus complexe, vous en conviendrez :p



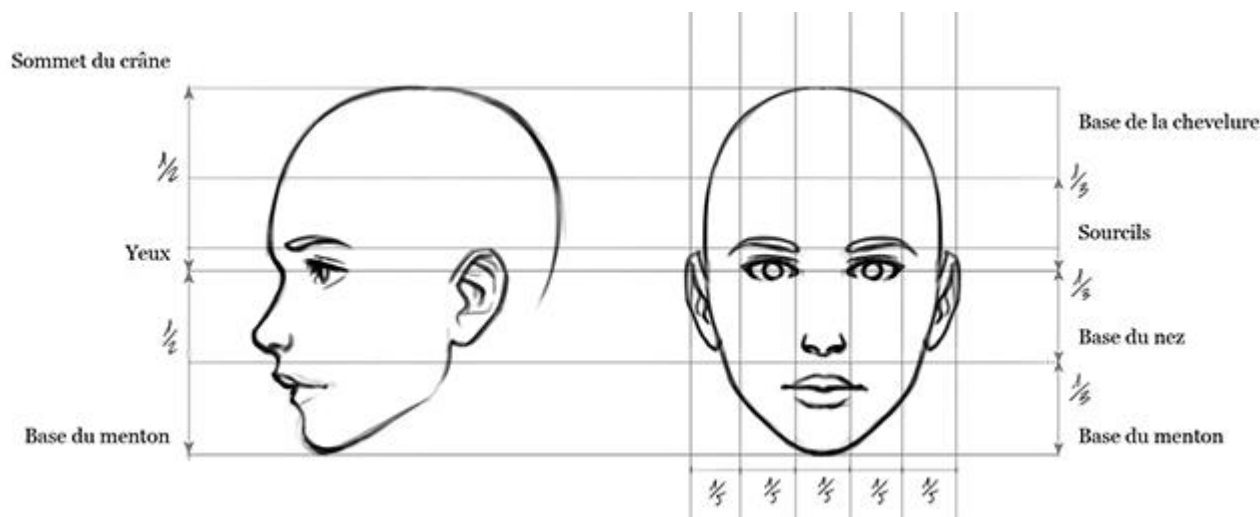
[Grande version](#)

Ce découpage en huit têtes vous permet de placer différents repères. Ces proportions, même quand le corps se déforme, se conservent : par exemple, si vous tendez le bras du personnage, vous devez pouvoir les retrouver en traçant ces arcs-de-cercle que vous voyez sur le dessin (en règle générale, utilisez les cercles pour reporter une longueur).

Il y a un certain nombre de repères que je trouve personnellement très utiles :

- Le pubis arrive pile-poil à mi-hauteur du corps : ce repère est particulièrement important si vous faites du dessin d'observation et il est toujours vérifié.
- Lorsque le bras est relâché le long du corps, la saignée du coude arrive au niveau de la taille.
- Lorsque le bras est relâché le long du corps, le poignet arrive au niveau de l'aîne.
- Lorsque le bras est relâché le long du corps, le bout des doigts arrive à mi-cuisse.
- Le tibia est beaucoup plus droit qu'on ne le pense, ce qui donne cet effet de « décroché » au niveau de la jambe de profil (il vaut même mieux exagérer sa concavité que l'inverse).

La tête.



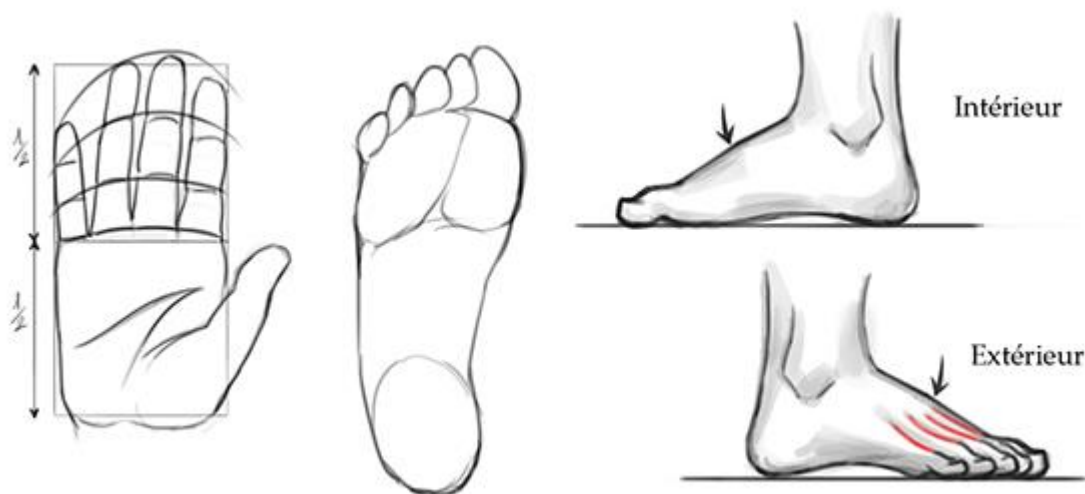
[Grande version](#)

Pour structurer le visage, vous disposez principalement de deux découpages verticaux et d'un découpage horizontal : verticalement, les yeux partagent en deux le visage, de la base du menton jusqu'au sommet du crâne. Toujours verticalement, la zone entre l'implantation des cheveux et la base du menton est découpée en tiers égaux par la base du nez et les sourcils. Horizontalement enfin, on peut découper le visage en cinq yeux (il y a donc « un œil » d'espace entre les deux yeux). Vous retrouvez ces repères sur le schéma ci-dessus.

Par ailleurs :

- Les yeux sont à mi-hauteur de la tête (et non aux trois-quarts comme on le voit souvent) : ce repère est aussi très important et il est toujours vérifié.
- Les yeux sont espacés de la largeur d'un œil et la tête fait cinq yeux de large en tout.
- Les pupilles (au repos) et les commissures des lèvres sont alignées verticalement.
- Les oreilles font à peu près la même hauteur que le nez et sont plus ou moins alignées avec.
- De profil, l'oreille est collée à droite (sur l'image) d'une verticale qui sépare la tête en deux parts égales.

Les pieds et les mains.



[Grande version](#)

Il y a peu de repères sur la main mais malheureusement, ça ne vous rendra pas la tâche plus simple, bien au contraire. C'est un sujet qui pourrait nécessiter un article entier à lui seul, je me contenterai donc ici d'aller à l'essentiel. Pour le pied, pensez à le décomposer entre la pointe (séparée en deux parties) et le talon. Entre les deux, la voûte plantaire s'arque et ne touche pas le sol la plupart du temps (d'où l'allure de nos empreintes).

- Dans une main tendue, la taille de la paume est égale à la taille des doigts.
- Le pouce arrive à la moitié de la main tendue.
- Les doigts se découpent en trois phalanges (certains me diront que c'est très bête, mais je vois cette erreur suffisamment souvent pour la signaler).
- Grossièrement, la paume est carrée.
- Essayez de suivre des arcs lorsque vous dessinez des phalanges, ceci donne plus de mouvement.
- En raison de la structure des os, le dessus des pieds n'est pas concave, mais convexe : on y voit une bosse, plus ou moins prononcée, que j'ai indiquée par une flèche.
- Si on regarde l'extérieur du pied, on voit *tous* les orteils et pas seulement le plus petit.
- Quand le pied est tendu (c'est la même chose pour la main), les os apparaissent sous la peau et donnent lieu aux lignes indiquées en rouge sur le dessin.

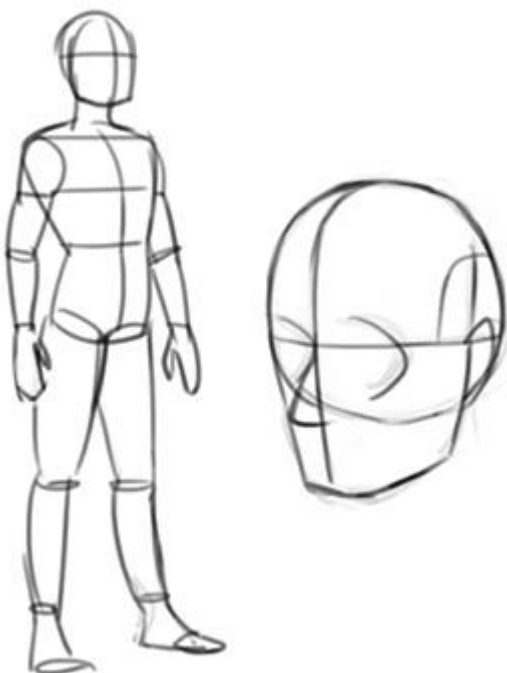
À noter : pour vous entraîner à dessiner des pieds, les statues antiques constituent d'excellents modèles.

Les découpages présentés ici ne sont pas, je le rappelle, des absolus. L'idée est essentiellement d'avoir un jeu de proportions simple, mais efficace pour commencer, quitte à le modifier et à le transformer plus tard. Essayez de varier et de comparer à ce que vous pouvez observer dans la réalité.

Décomposition en volumes simples.

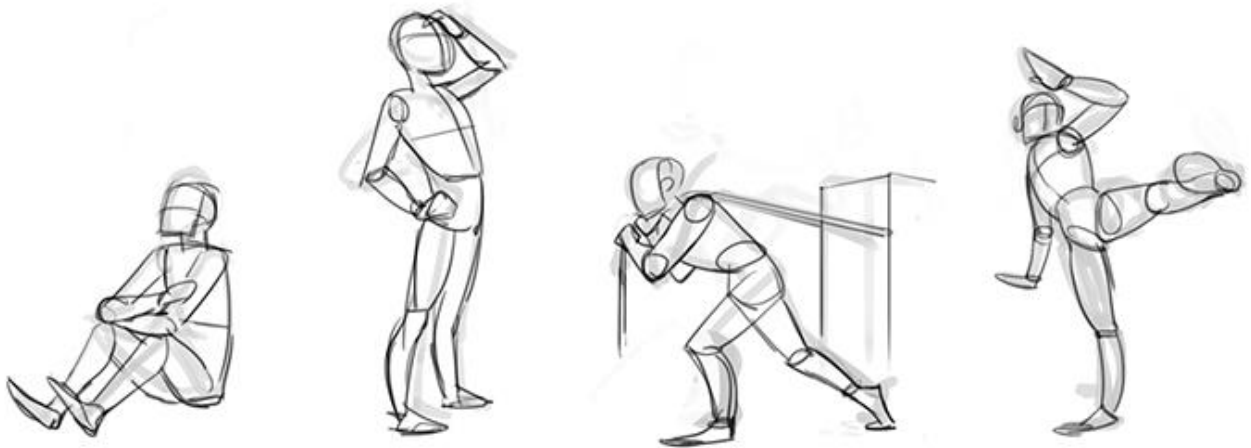
Maintenant que nous avons présenté des éléments de théorie sur le corps, intéressons nous maintenant à la pratique. Comment apprendre à dessiner un corps humain ?

L'idée consiste à simplifier le corps humain en volumes élémentaires, comme des cylindres ou des sphères, très simples donc, mais plus faciles à poser dans l'espace et à proportionner correctement. Imaginons que je veuille dessiner un personnage : je remplace son avant-bras par un cylindre (éventuellement un peu étréci), la tête par une sorte d'ovoïde, etc. J'obtiens un mannequin de ce style :



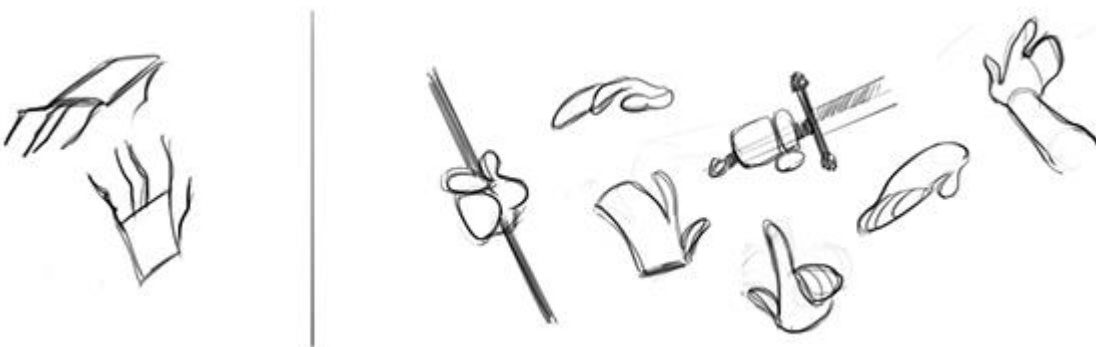
[Grande version](#)

Celui-ci est certes simple, mais il est beaucoup plus facile à animer, à « mettre en pose », à placer dans l'espace. Par ailleurs, il est plus facile de vérifier les rapports de proportions sur celui-ci. Des volumes simples permettent de ne rien oublier, de penser en volume et non en 2D. Souvent, quand les gens pensent en termes de contour, les corps ont tendance à manquer d'épaisseur. Si jamais vous avez besoin de réaliser une pose complexe, c'est la meilleure façon d'avoir une base propre pour commencer à travailler. L'essentiel est donc, avant de se lancer dans le détail, d'avoir un mannequin bien posé qui servira de fondation solide pour le travail : ce qu'on recherche, c'est une bonne *construction*.



[Grande version](#)

Pour les mains, on voit souvent les gens s'ingénier à dessiner chaque phalange comme un cylindre, mais à mon avis, c'est pousser le détail trop loin et ça a tendance à tuer le naturel dans le dessin des mains. Je vous recommanderais plutôt de vous limiter à un simple carré pour la paume et des traits sans épaisseur pour les doigts. Ça prend moins de place et c'est plus lisible, surtout sur des dessins de petite taille. Une variante consiste à dessiner la main sous la forme d'une « moufle », que vous pouvez améliorer en séparant l'index des autres doigts, comme ci-dessous. Ceci fonctionne aussi très bien pour les pieds.



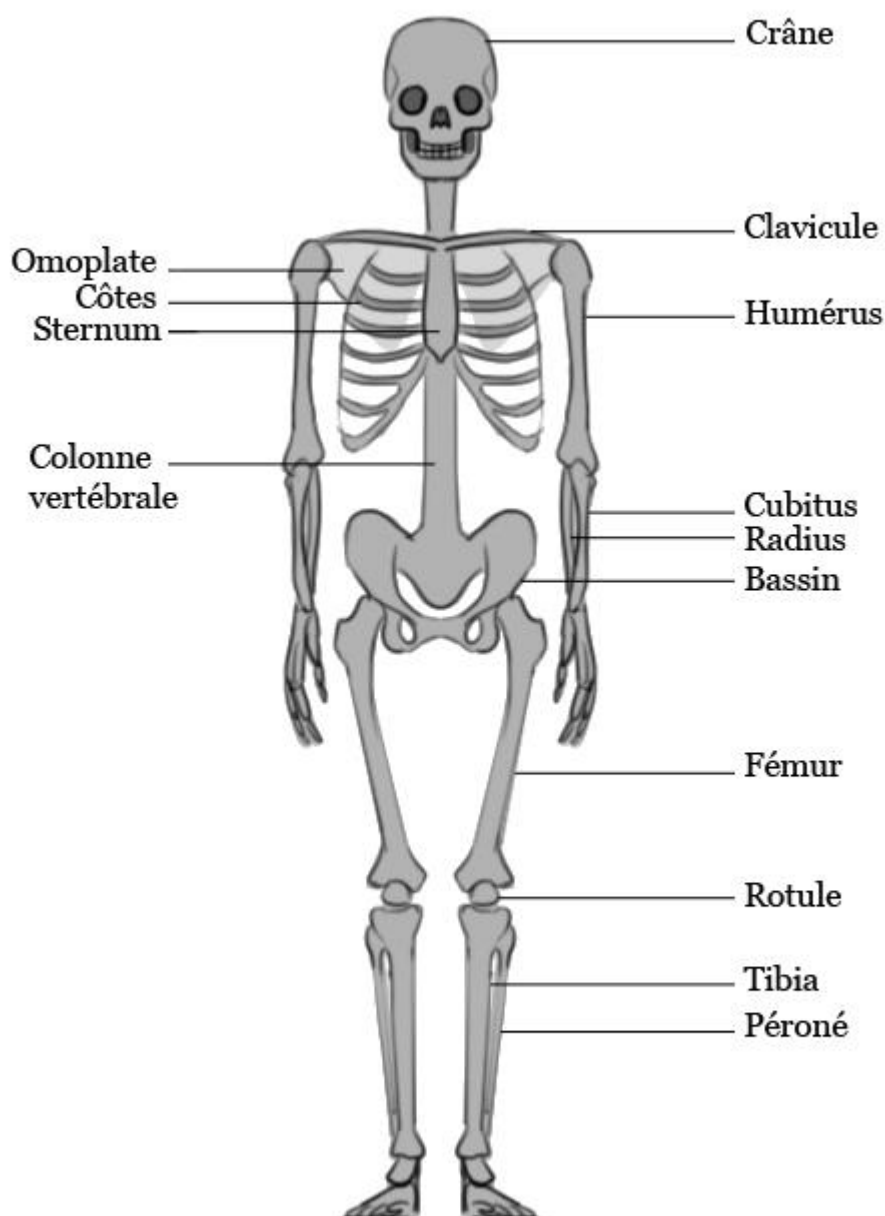
[Grande version](#)

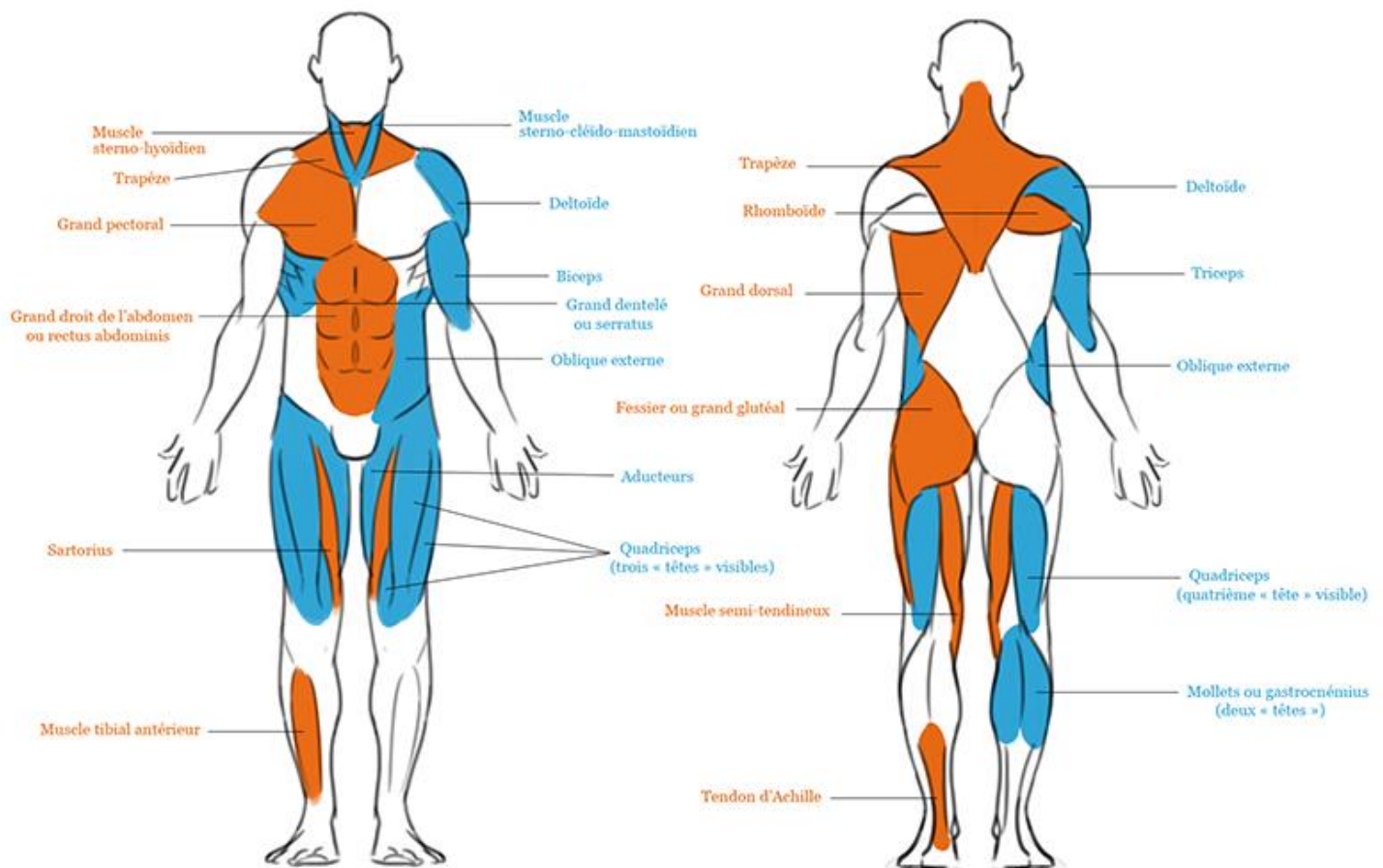
À vous de voir à quel point vous vous sentez à l'aise et quel type de simplification vous aide le plus. L'idée est de construire votre *propre* mannequin, pour pouvoir mieux poser vos dessins dans l'espace et leur donner un véritable volume. Si vous voulez travailler sur la solidité de votre dessin, essayez d'utiliser des formes très géométriques et de complexifier au maximum. Si vous voulez travailler sur la souplesse et le mouvement du dessin, essayez au contraire de le simplifier, par exemple en réduisant le dos du personnage au « S » de la colonne vertébrale. Même s'il est très simple au départ, vous aurez tendance à sophistiquer le mannequin avec l'expérience. Au fur et à mesure, vous y penserez de moins en moins et sans doute vous finirez par dessiner sans, mais c'est parce que dans une certaine mesure, vous l'aurez *intériorisé*.

Personnellement, j'aime bien dessiner les jointures comme si elles étaient transparentes et ajouter les lignes médianes des objets (par exemple, la croix sur le visage), ça permet de mieux sentir les volumes.

De chair et d'os.

Maintenant, que nous avons vu les principes du mannequin et de la construction, parlons un peu d'anatomie à proprement parler. Je ne sais pas vraiment s'il est nécessaire de connaître le moindre os, le moindre tendon, le moindre muscle du corps humain pour bien dessiner, mais si on s'intéresse à ça, je pense qu'il vaut mieux commencer par un modèle simplifié. Ci-dessous, un modèle très simple avec le nom des principaux os et muscles (disons ceux auxquels vous aurez affaire en priorité). N'oubliez pas, ces muscles existent chez les hommes *et* les femmes, j'insiste.





Pour les grandes versions, c'est par [ici](#) et [ici](#).

Le bleu et l'orange ne sont là que pour distinguer les différents muscles, ils n'ont pas de signification particulière ici. Remarquez que les fémurs ont tendance à rentrer vers l'intérieur, mais pas les tibias (ça permet de mieux comprendre certaines torsions de la jambe). Par ailleurs, le quadriceps est bien constitué d'un seul muscle, mais divisé en quatre « ventres » ou « chefs » (d'où son nom), ce qui explique pourquoi on a l'impression de voir quatre muscles au lieu d'un.

Je vous avouerai au passage que je serai bien infoutu de vous détailler les quinze extenseurs du poignet et la structure des métatarses : retenez déjà l'essentiel et travaillez la construction, ce sera déjà beaucoup (et même beaucoup plus qu'un certain nombre de professionnels du jeu vidéo :p).

Le sexe (oh, comme c'est vendeur !)

Séréotypes genrés.

Je vais maintenant me pencher sur un point assez délicat dans l'anatomie, la question de la représentation des sexes.

Dans le jeu vidéo et dans beaucoup d'autres médias, les sexes sont représentés de façon stéréotypés. En particulier, l'anatomie féminine est représentée de façon fantasmée (de gros seins, des tailles de guêpe, des jambes interminables), voire anatomiquement incorrecte (le tumblr [Eschergirls](#) se fait par exemple une spécialité d'épingler de telles dérives). D'une part, ce sont des représentations sexistes (et on ne parle même pas des costumes conçus pour exhiber le maximum de chair). D'autre part, d'un point de vue purement artistique, on perd toute la variété et la diversité anatomiques que la nature propose. Il y a des femmes avec

de gros nez, avec des épaules carrées, il y a des hommes avec des traits fins, avec de longs cils : en tant que personnages, ils ne sont pas moins intéressants qu'une bimbo ou qu'un bodybuilder.

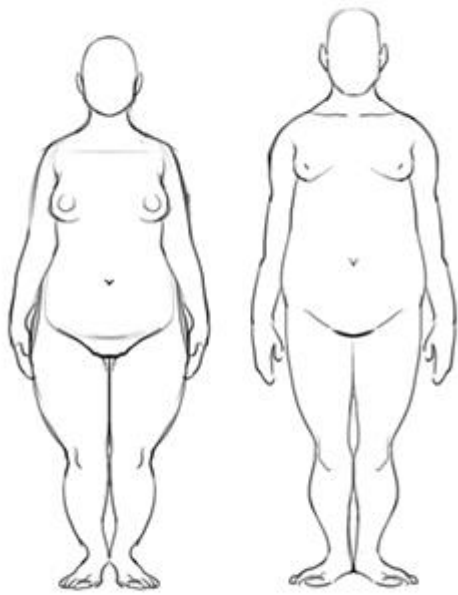


Il faut travailler à enrichir son vocabulaire graphique pour éviter les stéréotypes. On ne peut pas dessiner un personnage féminin correctement si on se contente de plaquer sur un corps les attributs supposés de la féminité : cils, lèvres, longs cheveux, etc. L'exemple-type, c'est cette façon de féminiser un personnage masculin en lui ajoutant des cils, un ruban et/ou du rose : c'est comme ça qu'on crée Minnie à partir de Mickey. Sans entrer dans l'étude de genre, les différences anatomiques homme-femme sont plus profondes, plus subtiles et moins marquées.

Précisons une fois de plus les choses : les observations que je présente ici ne sont que des tendances et pas des vérités absolues.

Répartition des graisses et silhouette.

La répartition des graisses se fait différemment chez les hommes et les femmes. Vous vous rappelez peut-être de cours de biologie où l'on vous a parlé de profils androïde et gynoïde de répartition des graisses.

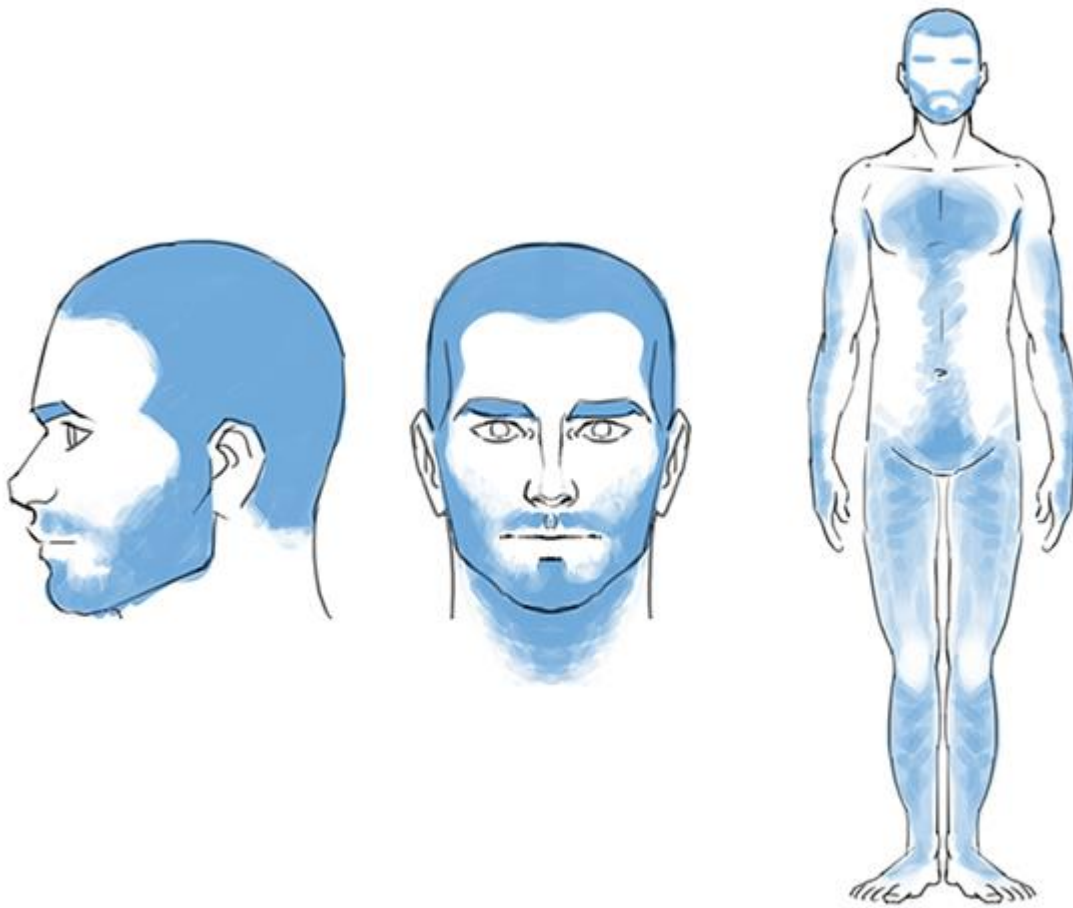


[Grande version](#)

Chez les femmes, les graisses ont tendance à s'accumuler préférentiellement sur les fesses, les cuisses et les hanches : c'est ce qu'on appelle le profil gynoïde. On le retrouve par exemple dans la stylisation « en sablier » pour représenter une silhouette féminine et même dans des représentations très anciennes (statues de déesses de la fertilité par exemple). Chez les hommes, les graisses ont tendance à s'accumuler sur la partie supérieure du corps et sur l'abdomen : c'est ce qu'on appelle le profil androïde. Ces différences de profils ne sont pas toujours très marquées et sont plus visibles chez des personnes en surpoids. Elles tendent à caractériser des silhouettes différentes.

Pilosité.

La pilosité est plus abondante chez l'homme que chez la femme (j'espère ne rien vous apprendre ici). Les poils ne poussent pas n'importe où sur le corps mais sur des régions privilégiées, c'est un point important pour comprendre la façon dont la barbe et les cheveux s'implantent. On voit souvent des implantations fantaisistes qui sapent tout le travail de rendu sur les mèches (c'est par exemple souvent le cas chez Hyung Tae Kim). On peut explorer des coupes de cheveux délirantes comme dans beaucoup d'animes et de jeux (les *Ace Attorney* s'en sortent très bien) mais il existe déjà une très grande variété parmi les coupes de cheveux réalistes.



[Grande version](#)

Quelques remarques au passage sur les coupes de cheveux et l'implantation de la barbe :

- J'ai remarqué que les hommes (je ne sais pas vraiment pourquoi ça m'a semblé revenir plus souvent chez les hommes que chez les femmes) ont tendance à systématiquement dessiner les oreilles par dessus les cheveux comme si elles étaient forcément dégagées, mais il arrive souvent qu'elles soient couvertes par les cheveux, ne serait-ce que partiellement.
- Dans une coupe de cheveux d'homme, la patte a une grande importance. Dans une coupe « classique », elle arrive à mi-oreille. Au-dessus, ça fait bizarre, au-dessous ça donne l'impression d'être long (à vous de jouer avec !). S'il y a de la barbe, les pattes peuvent se fondre dedans.
- Les « coins » sous la bouche où je n'ai pas mis de barbe peuvent finir par être recouverts, mais la barbe y pousse beaucoup moins vite qu'ailleurs, voire pas du tout.
- La barbe, avec le temps, peut monter très haut sur les joues.
- La barbe et la moustache ne sont pas nécessairement connectées.
- Pour suggérer une très courte barbe, on peut utiliser un changement de teinte vers le bleu, le vert ou le gris plutôt que de dessiner chaque poil.
- La calvitie apparaît sur l'arrière du crâne, mais aussi par l'avant, où elle a tendance à faire reculer la ligne d'implantation des cheveux. Ceci se fait plus rapidement sur les tempes que sur le haut du front en général.

Références.

Pour conclure cet article, j'aimerais citer un certain nombre de références extrêmement utiles.

Tout d'abord, les livres absolument incroyables d'[Andrew Loomis](#). Si on met de côté un style très années 50 (et la vision publicitaire assez sexiste qui va avec), ces ouvrages sont des merveilles de pédagogie. A

l'inverse, je vous déconseille très formellement les livres de Burne Hogarth et de George Bridgman. Pour s'entraîner au dessin d'après modèle, le site [Posemaniacs](#) est extrêmement bien conçu pour réaliser des dessins d'observation, en particulier pour les temps de pose courts. Je vous recommande aussi chaudement les guides très bien faits d'[AlexDS1](#) et [Coelasquid](#) (dont [celui-ci](#) qui se concentre principalement sur les muscles). Enfin, pour des références photographiques de bonne qualité, le site [Artsy Poses](#) devrait vous être utile [le site étant down pour le moment, voici un autre lien pour un [site du même genre](#) qui nous est signalé par ilos (merci à lui !)].

Au passage, sachant à quel point il est difficile d'avoir accès à des modèles vivants, pensez à l'autoportrait, c'est toujours un très bon exercice.

En vous souhaitant un bon courage pour la suite, je vous rappelle que ces quelques conseils ne vous seront pas d'une grande utilité si vous ne les mettez pas en application. Une seule chose à faire : s'entraîner :D

Compte-rendu de Ludum Dare #24

Par [Antoine Gersant](#) | le 27 septembre 2012 23:44:15 | Catégories : [Game Design](#), [Graphisme](#), [Ludum Dare](#), [Ma vie](#)

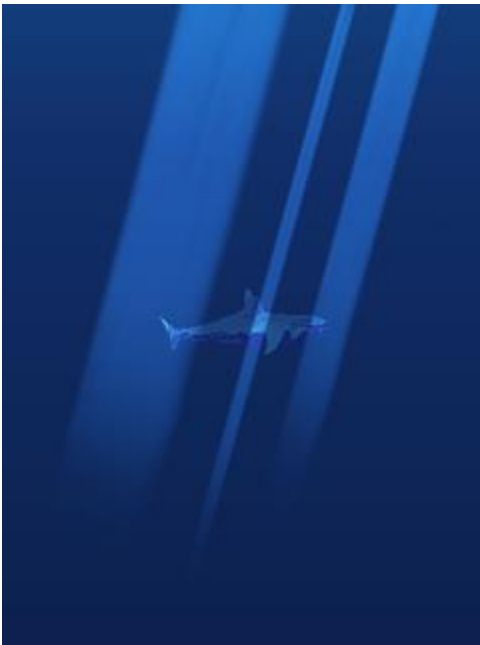
Après vous avoir fait faux-bond au Ludum Dare #23 qui a eu lieu en avril dernier, j'ai remis le couvert le mois dernier pour la 24ème édition de cet évènement si particulier. Pour ceux d'entre vous qui ne seraient pas familiers du principe et qui n'auraient pas lu mon [compte-rendu de la 22ème édition](#) (ou [celui de la précédente](#)), rappelons une fois de plus qu'il s'agit d'un weekend pendant lequel des développeurs du monde entier créent des jeux sur un thème imposé. Les participations peuvent se faire soit en 48 heures en solo (la compétition ou *compo*), soit en groupe en 72 heures (le *jam*).

L'avantage de participer pour la 3ème fois au Ludum Dare c'est que je commence à avoir mes petites habitudes, à commencer par les préparatifs du vendredi qui précède le début de la compétition :

- Création d'un projet vide prêt à recevoir graphismes et code
- Création d'une page web où héberger le jeu fini. J'y ai passé des plombes, c'est toujours difficile de choisir les couleurs d'une page web destinée à contenir un jeu pas encore créé.
- Des courses ! En plus du nécessaire à sandwiches, j'ai investi dans une montagne de cookies et dans une sélection de [délicieuses bières locales](#) (pour ceux qui ne suivent pas les articles — ou plutôt l'absence d'articles — taggés « ma vie », rappelons que je suis toujours en Angleterre). Je n'ai pas encore testé toute la gamme mais c'est en bonne voie. Par contre, il me reste plein de cookies.
- N'ayant pas écrit de musique depuis environ 6 mois, je suis allé faire un tour sur Youtube pour regarder quelques tutoriaux sur Reason (le logiciel que j'utilise pour faire de la musique) histoire de me remettre dans le bain. Il s'avère qu'une nouvelle version du logiciel était sortie entre temps, j'ai bien fait de me mettre un peu à jour =)

Le thème de cette édition, annoncé tard dans la soirée (2 heures du matin), était « évolution ». Comme toujours, un grand mouvement de panique a suivi (et pas seulement sur le canal IRC #LD48 dédié). Après avoir envisagé dans un premier temps un jeu dans lequel la technique et les features auraient évolué au fur et à mesure de la progression (ce qui a finalement été fait dans [Evoland](#) de Nicolas Canasse), je me suis dit que j'allais plutôt faire un jeu dans la veine de *Space Station Silicon Valley* sur Nintendo 64. L'idée de base était de faire un jeu dans lequel le joueur aurait incarné une bactérie dans un univers futuriste pollué et utilisé des déchets radioactifs pour évoluer de bestiole en bestiole et résoudre des puzzles. Étant donné que tout cela n'avait aucune chance d'être fait en 48 heures, j'ai réduit le concept à des espèces sous-marines et vues de côté pour diminuer un peu l'échelle. Ça reste un plan d'action complètement irréaliste mais au moins c'était un point de départ.

J'ai ensuite passé ma soirée/nuît à faire des petits poissons sous Photoshop (ainsi qu'un baril radioactif et un joli dégradé de 2000 pixels de haut dont nous reparlerons plus tard). Avant d'aller me coucher, j'avais ceci :

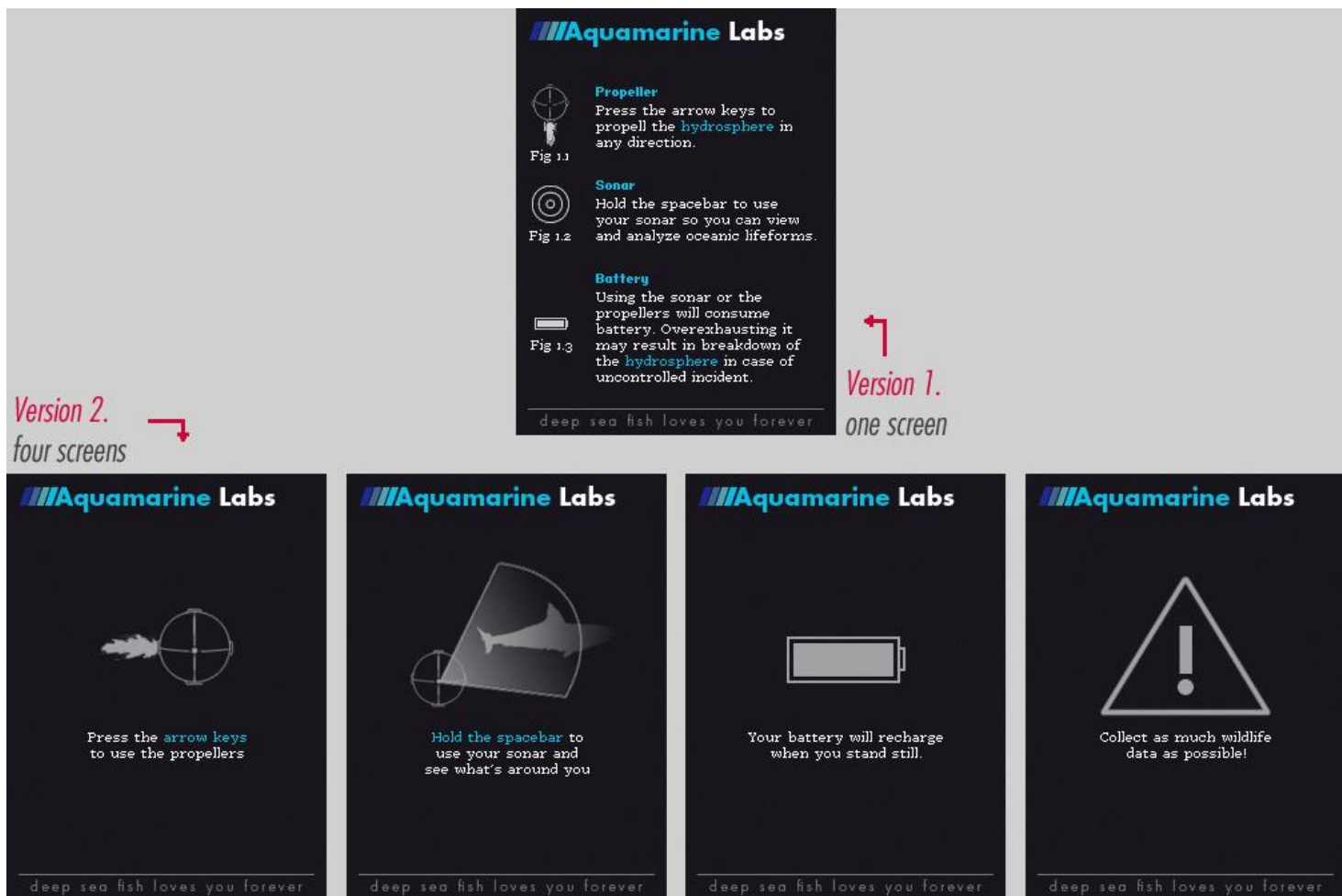


Le jeu final ressemble vaguement à ça mais avec le recul je trouve que ce mockup était plus joli (avec son échelle vue de loin).

Le lendemain matin (c'est-à-dire le lendemain midi), j'ai ouvert les yeux et je me suis rendu compte (comme à chaque fois) que je n'aurais jamais le temps de coder mon idée de départ. S'en est suivie une heure de réflexion pour tenter de répondre à la question « Bon alors, qu'est ce que je peux faire comme jeu avec des poissons et un baril radioactif tout en respectant le thème de la compétition ? ». Si cela vous rappelle mon compte-rendu précédent, c'est normal. Ma réponse à cette question est le concept que j'ai finalement implémenté : le joueur joue un robot d'exploration sous-marine chargé de collecter des données sur les espèces océaniques. Ça tenait à peu près la route, j'ai donc passé la journée du samedi à coder tout ça. Je suis finalement allé me coucher quand mon ordinateur s'est éteint sans prévenir et a refusé de se rallumer.

Je me suis demandé si je n'allais pas devoir finir mon jeu au bureau (il était versionné via SVN donc je ne risquais pas de perdre les données) mais j'ai finalement réussi à redémarrer ma machine après l'avoir ouverte et dépoussiérée un peu. Je crois que le dimanche a été ma journée préférée, comme à chaque édition du Ludum Dare. Une fois arrivé à ce stade, le jeu marche à peu près et il s'améliore à vitesse grand V en rajoutant de la musique, des sons et pleins de petits détails visuels. En plus, j'ai toujours l'impression de [piloter un vaisseau spatial quand je compose de la musique](#).

Une fois mes finitions effectuées, ce qui nous amène au milieu de l'après-midi, j'ai fait tester le jeu à quelques heureux cobayes. Pour une fois, ils n'ont pas trouvé trop de bugs mais on fait des suggestions qui m'ont beaucoup aidé. J'ai ainsi ajouté un indicateur de progression dans l'interface et complètement repensé le tutorial qui est passé de « un écran plein de texte » à « 4 écrans avec une grosse icône et un peu de texte ». Je pense qu'il est un peu moins rebutant qu'il ne l'était au départ.



Un (gros) problème que ni moi ni les testeurs n'avions repéré est le fait que le jeu rame sur beaucoup de PC, ce qui n'aurait pas été bien difficile à corriger (par exemple en n'affichant pas un gros dégradé de 2000 pixels de haut semi-transparent par dessus le jeu).

Au final, j'ai terminé mon jeu plusieurs heures avant la deadline. Je pense que j'aurais dû en profiter pour ajouter d'autres espèces de poissons mais je ne voulais pas modifier la répartition des espèces en fonction de la profondeur (qui n'est de toute façon pas très bien équilibrée). Les avis sur le site du Ludum Dare sont plus positifs que la dernière fois et je pense avoir globalement fait un meilleur jeu. Si vous voulez y jouer, c'est [par ici](#).

Pour finir, quelques menues anecdotes sur le jeu :

- Le titre du jeu, *Deep Sea Fish Loves You Forever*, est une référence pas très discrète au remix [Goodbye Thinker](#) de Jillian Aversa et Mattias Häggström Gerdt.
- La société "Aquamarine Labs" a été nommée en référence à *Ecco The Dolphin* et en pensant très fort à Aperture Science.
- La fumée des réacteurs du robot est faite en particules plutôt qu'animée à la main (pour gagner du temps). Les explosions des mines aussi mais c'est moins surprenant.
- Les [claves](#) dans la musique du jeu sont une référence (un peu plus subtile cette fois-ci) à [The Future Sound of London](#) dont c'est un peu la signature.

Cours de pixel art 10 : Ça manque d'animation par ici !

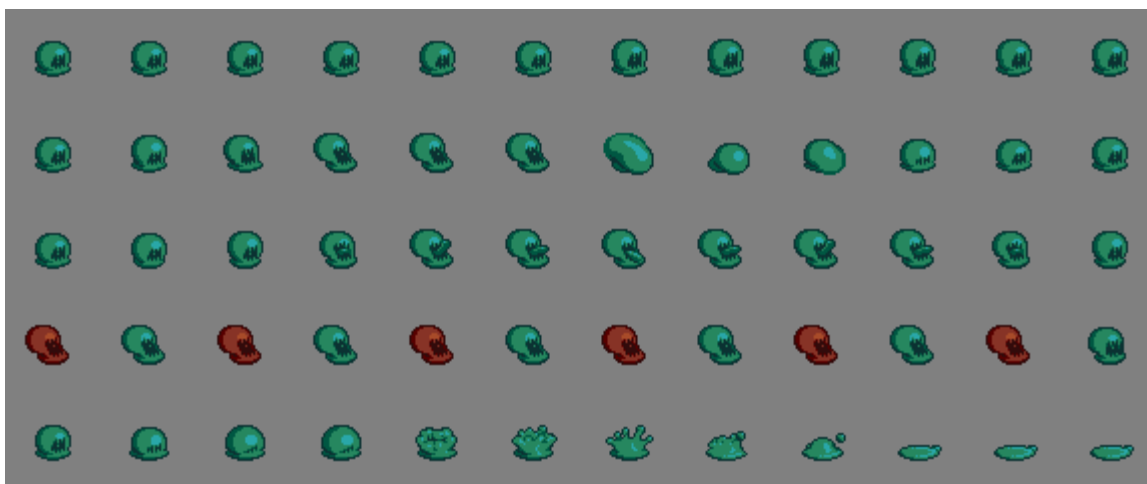
Par [Thomas Chassin](#) | le 1 octobre 2012 22:00:00 | Catégories : [Cours](#), [Graphisme](#), [Pixel art](#)

Aucun tutoriel sur le pixel art, qui plus est dans le cadre de la création d'un jeu vidéo, ne saurait faire l'impasse sur l'animation, c'est pourquoi nous vous proposons aujourd'hui un article sur le sujet.

Commençons par une rapide explication du principe d'animation tel que nous l'envisagerons ici.

Si vous avez déjà vu une bobine de film, vous n'êtes pas sans savoir qu'elle est composée d'une série d'images fixes affichées très rapidement les unes à la suite des autres (24 images/secondes en général). L'animation en pixel art repose sur ce même principe : l'affichage d'une succession de cadres ou *frames*, bien que le nombre d'images uniques par seconde soit généralement très inférieur à celui de l'animation classique, résolution oblige.

Je ne parlerai pas de la gestion des *frames* au sein d'un logiciel ni de leur organisation au sein d'un fichier, puisque cela dépend de la façon dont votre image est lue mais sachez que dans la très grande majorité des cas, les *frames* sont toutes regroupées en une seule image appelée *spritesheet*.



Un peu de préparation.

À vos crayons !

Avant d'animer, il est important de savoir à peu près ce que vous voulez obtenir comme résultat. Le plus simple pour cela est de le mettre sur papier : oui, vous allez me le dessiner (n'ayez pas peur, pas besoin d'être bon) ! Bien sûr, il serait absurde de dessiner toutes les *frames*, mais il est important de visualiser les principaux « points de passages » ou *key frames* de votre animation, i.e. les étapes-clefs qui définissent votre mouvement, mais également les étapes de mouvements complexes, comme par exemple des transformations ne se situant pas dans le plan de l'image (un personnage de jeu de plate-forme tournant sur lui-même par exemple).

C'est ce qu'on appelle l'animation par parties ou *pose to pose animation* : on décide d'un certain nombre d'étapes intermédiaires puis on rajoute des *frames* de transitions entre elles. Dessiner, même de façon sommaire, les *key frames* vous permettra de mieux visualiser le travail à effectuer et vous aidera grandement au moment de la réalisation de votre *sprite*.

Prenons un exemple pour clarifier ce propos un peu obscur : un coup d'épée. Celui-ci se découpe comme suit : le personnage est au repos, il arme son coup, frappe puis réarme, ce qui fait un total de 4 « points de passages » à définir pour pouvoir animer votre personnage, le reste des images servant de transition entre ces derniers.



Les références sont vos amies pour la vie.

Si vous n'êtes pas sûr de la façon dont se découpe votre animation, le grand secret de la vie et de la mort consiste à vous documenter, aussi bien du côté animation (regardez ce qui se fait en général) que du côté réalité (rien de mieux que l'observation pour vous débarrasser de vos *a priori*).

Notez que pour l'animation de personnages, il est possible de mimer soi-même les mouvements, mais cette option est peu recommandée lorsque vous êtes sur votre lieu de travail, ou même en public de manière générale. Normalement, cela devrait suffire à obtenir un découpage satisfaisant de l'animation, mais ne vous inquiétez pas si vous éprouvez quelques difficultés, le coup de main viens avec la pratique, et si possible des critiques constructives.

Un peu de théorie.

Deux, trois généralités.

Rentrons maintenant dans la partie plus technique de cet article, si vous le voulez bien (en fait, vous n'avez pas le choix). Pour que votre animation soit réussie, il est important d'avoir une bonne maîtrise de la répartition du mouvement dans les *frames*. En effet, le mouvement se définit par la différence de position entre deux représentation d'un même élément d'une *frames* à l'autre.

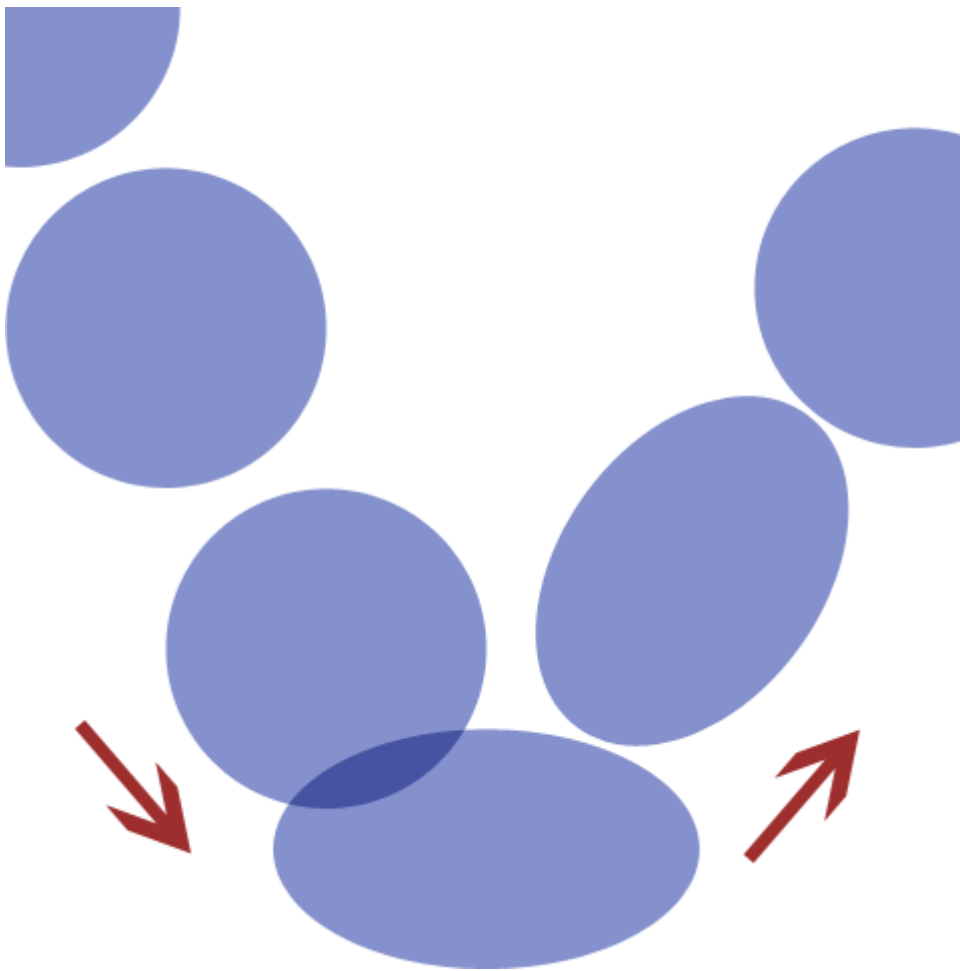
De fait, plus vous consacrez de *frames* à un mouvement, plus il apparaît lent, et inversement, plus vous supprimez de *frames* intermédiaires, plus il apparaît rapide. La première conséquence logique est que dupliquer une *frame* permet de marquer une pause dans le mouvement, la seconde est que plus vous chercherez à représenter des mouvements rapides, plus il vous faudra un nombre élevé de *frames* pour un même intervalle de temps (on peut aussi parler de fréquence d'échantillonnage ou *sample rate* pour les intimes).

Un peu de dynamisme et de stylisation.

L'élasticité.

Si vous ne vivez pas au sein d'une colonie de golems au fin fond d'une caverne, vous aurez peut-être remarqué qu'il existe des matériaux rigides et d'autres plus élastiques. Ces derniers ont la capacité de se déformer sous l'action des forces qui leurs sont appliquées.

La technique du *squash and stretch* (compression et étirement) consiste à étirer ou compresser un corps au contact d'un solide, tout en conservant son volume (c'est important). L'exemple le plus flagrant est celui d'un ballon qui rebondit : d'abord sphérique, il s'écrase au moment du contact avec le sol, puis s'étire lorsqu'il rebondit. Ce principe peut se généraliser à de très nombreux mouvements en animation, qu'on anime un visage, un personnage, un objet, etc.



[Un rapide aperçu en .gif](#)

Plus la compression et l'écrasement sont forts, plus on obtient un effet « toon ». Ce principe permet de fluidifier les actions et de leur donner plus de force.

Inertie et cinétique.

N'ayez pas peur, je ne vais pas vous faire un cours de physique ! oubliez vos cours, nous n'allons ici nous intéresser qu'à certains effets physiques utiles à l'animation, à savoir qu'un corps en mouvement tend à le rester et qu'il en va de même pour un corps immobile. Trois effets en découlent :

La continuité du mouvement.

La continuité du mouvement (ou *follow through*) a pour conséquence que lorsqu'un corps non rigide s'arrête, ses parties restent en mouvement quelques instants avant de s'arrêter à leur tour. On peut constater ce phénomène de façon très simple : faites quelques pas puis arrêtez-vous. À moins d'avoir tout le haut du corps dans le plâtre, vos bras ne deviendront pas absolument immobiles à la seconde où vous arrêterez d'avancer : leur mouvement continuera quelques instants à cause de l'énergie cinétique accumulée puis ils s'arrêteront de bouger à leur tour.

L'effet de traction.

L'effet de traction (ou *drag*) est très proche de la continuité du mouvement : c'est plus précisément son inverse. L'effet de traction peut s'observer lorsqu'on exerce une force sur un corps souple, par exemple du tissu. L'ensemble du corps ne se mettra pas en mouvement simultanément si vous tirez dessus : viendra d'abord la partie que vous tirez, puis le reste, du point de traction jusqu'aux extrémités.

Le chevauchement de mouvements.

Le chevauchement de mouvements (*overlapping action*) est le résultat de l'action conjuguée des deux phénomènes précédents. Lorsqu'un corps passe d'un mouvement A à un mouvement B, il y a un instant où les deux mouvements se « chevauchent » : la partie à l'origine du mouvement B se déplace en accord avec ce dernier mais le reste du corps, en raison du *drag*, continue encore à suivre le mouvement A. Il y a donc coexistence de deux mouvements possiblement antagonistes jusqu'à ce que l'ensemble du corps suive le mouvement B. Comme expliqué précédemment, la transition de mouvement se fait progressivement de la partie à l'origine du mouvement vers le reste du corps.

Les phases de transition.

Vous pensiez que c'en était fini de l'inertie ? Que nenni mes braves : en revoilà ! Tout corps qui se déplace, à moins qu'il ne reçoive ou ne transfère brutalement une quantité d'énergie cinétique importante, passe par un régime appelé transitoire en début et en fin de mouvement. Concrètement, cela signifie que le corps a besoin d'accélérer lorsqu'un mouvement est initié, et qu'il lui faut ensuite décélérer avant de s'arrêter. Ces phases de transition, en plus de rendre votre animation plus crédible, permet de véhiculer des informations supplémentaires au joueur. Par exemple, une lente accélération et/ou une tout aussi lente décélération indiquent que l'objet qui se déplace est lourd : pensez à une masse par exemple, c'est une arme lente, mais une fois le coup parti, il est très difficile à arrêter (les marteaux et les grandes épées de *Monster Hunter* en sont un bon exemple).

Pour se la péter dans les dîners mondains (suite) : le motion blur.

Le *motion blur*, ou flou cinétique pour ceux qui sont fâchés avec Shakespeare et ses copains, est un phénomène optique avec lequel nous vivons tous les jours : lors de l'observation d'un mouvement rapide, on peut voir une traînée floue derrière le corps en mouvement. Cette traînée est en fait due à la persistance rétinienne et n'est qu'une illusion d'optique. Ça vous fait une belle jambe ? Et pourtant c'est un principe très très utile pour l'animation.

Lors d'un mouvement très rapide, il y a un déplacement important qui s'effectue d'une *frame* à une autre, ce qui risque de rendre votre animation peu compréhensible par le spectateur, et ce risque augmente proportionnellement à la complexité de la trajectoire suivie.

Alors comment faire pour conjuguer mouvement rapide et animation compréhensible ? Une idée ? Si vous avez répondu « Le *motion blur* ! », on va peut être faire quelque chose de vous, sinon retournez à la case départ.



Pour réussir un *motion blur*, le mieux est souvent... de ne pas s'en préoccuper ! En effet, le motion blur vient **après** le découpage de votre animation et la réalisation de vos *frames*. L'idéal est de réaliser vos animations en notant les *frames* nécessitant du motion blur, puis de revenir les travailler après coup. Si vous utilisez un logiciel gérant les calques (ce que toute l'équipe vous recommande très fort), tracez la trajectoire suivie par l'élément en mouvement d'une *frame* à l'autre sur un calque à part, elle vous servira de base pour la réalisation du motion blur. Et ensuite ? Eh bien, il ne reste plus qu'à suivre quelques règles :

- La « traînée » est comprise entre la position de départ et celle d'arrivée, et suit la trajectoire de votre mouvement.
- Plus un mouvement est rapide, plus la « traînée » est longue.
- Considérez votre traînée comme un solide, les règles de perspectives et autres s'y appliquent normalement.
- Vous pouvez utiliser un dégradé pour accentuer l'effet de flou.

Protips.

Commencer simple.

Si vous commencez par faire des *frames* hyper complexes avec un éclairage à sources multiples, *ambient lighting* et tout le fatras, vous allez pleurer à chaudes larmes le jour où vous vous rendrez compte que vous avez raté votre animation. Commencez par faire des formes à base d'à-plats de couleur pour animer rapidement. Une fois que l'animation vous satisfait, augmentez le niveau de détail progressivement en vérifiant à chaque fois que le résultat est conforme à vos attentes.

Animer séparément les différentes parties.

Travaillez avec des calques. Sérieusement. Il est infiniment plus facile de modifier une animation dont les éléments sont sur calques séparés. L'idéal serait de créer un calque par élément si vous n'êtes pas sûr de vous. Notez également que certaines animations ne reposent que sur le déplacement de certains éléments les uns par rapport aux autres, sans déformation quelconque.

Prendre l'animation comme un « tout ».

Lorsque vous travaillez sur une animation, quelle qu'elle soit, vous travaillez sur un ensemble de *frames* successives. L'important est que votre animation soit réussie, pas que chaque *frame* individuelle soit un chef-d'œuvre irréprochable sur lequel vous avez passé une demi-éternité.

Non seulement garder ceci à l'esprit vous permettra d'éviter une importante perte de temps et d'énergie, mais cela vous permettra également de faire gagner en qualité votre travail. En effet, que chaque *frame* prise à part soit une réussite ne vous garantira pas que votre animation sera bonne, puisque, bien au contraire, les *frames* de transitions auront très souvent l'air « moche ».

C'est un fait inévitable. Il suffit de regarder un quelconque court ou long-métrage d'animation et de mettre en pause aléatoirement pendant que les personnages changent de pose. La probabilité qu'ils aient l'air ridicule est très élevée, et pourtant l'animation **dans son ensemble** est de bonne qualité.

La force des détails.

La différence entre une bonne animation et une très bonne animation réside (entre autres) dans le soin apporté aux détails ou actions secondaires. Ces actions secondaires ont pour rôle principal d'asseoir l'animation principale, de la rendre plus vivante et de lui donner plus de force. Par exemple, lorsqu'un personnage se déplace et qu'il porte une cape et/ou a des cheveux longs, ceux-ci ne sont pas statiques : il réagissent à l'action principale, ici la marche. Il est cependant important de hiérarchiser les différentes actions, de manière à ce que l'action principale soit clairement identifiable : si ce n'est pas le cas, tout le travail réalisé sur les animations secondaires ne fera que desservir l'action dans son ensemble.

Et maintenant bavez un peu devant ce que font les grands !

Et pleurez un coup au passage, ça vous fera du bien.

- [How to Make Her Turn](#), par [Syosa](#)
- [Maska](#), par [KO-KI](#)
- [Arrgh, I guess](#), par le très grand [fool](#)

Un peu de pub

Par [Pierre Guyot](#) | le 27 octobre 2012 16:01:22 | Catégories :

Un petit peu de pub pour Je Game Moi Non Plus, un site consacré au jeu vidéo et à la culture. Je laisse la parole à Gaëtan Blaise, l'un des membres de l'équipe et le chef du projet [Lije](#), dont vous avez pu un peu entendre parler dans [cet article](#).



Je Game Moi Non Plus est un podcast qui parle du jeu vidéo d'une manière assez particulière. Plutôt que de traiter d'un jeu ou d'actualité, ses animateurs choisissent un thème, font des recherches, réfléchissent un peu, et tentent d'explorer le sujet en profondeur. Cela peut les amener à discuter de l'industrie, de sociologie, de game design, de cuisine, de littérature... et toujours de jeu vidéo. L'objectif est d'apporter à l'auditeur une vision du jeu originale, et de lui apprendre des choses intéressantes. L'émission, mensuelle, dure environ 1h30.

Pour réaliser le podcast, nous avons réuni une équipe multidisciplinaire : on compte un psychologue, un fin connaisseur du Japon et maître des arcanes du marketing, un graphiste, un journaliste et un ingénieur. De temps en temps, un invité nous rejoint et des hors-série, les *Time Extended* sont l'occasion d'interviewer une personnalité du monde du jeu vidéo.

Le dixième numéro vient de sortir et est consacré aux jeux vidéos qui ont pour cadre l'école.

Le podcast est assez récent et le format n'est pas encore parfait. Les futurs auditeurs sont donc invités à réagir et à critiquer sur les différents canaux de diffusion.

- [iTunes](#)
- [Youtube](#)
- Le site de l'émission : <http://www.jegamemoinonplus.com>

N'hésitez pas à contacter les animateurs [ici](#) ou sur :

- Twitter : @jgmnp
- [Facebook](#)

Transgresser les règles (Ernest va au cinéma)

Par [Lyonsbanner](#) | le 3 novembre 2012 18:02:58 | Catégories : [Ernest Adams](#), [Game Design](#), [Univers](#)

Cet article est une traduction de [Lyonsbanner](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (juillet 2000).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Attention, spoiler ! Cet article dévoile certains éléments du film *Matrix*, bien que jusqu'à ce samedi soir, j'étais probablement la seule personne dans le monde occidental qui ne l'avait pas encore vu. Il est donc peu probable que ceci gâchera le plaisir de mes lecteurs.



Nous, les développeurs de jeux vidéo, nous adorons les films sur les ordinateurs, surtout s'ils parlent de réalité virtuelle et autres concepts cools dans le même genre, alors j'ai pensé que je ferais mieux de regarder *Matrix*. Après tout, je me suis précipité pour aller voir *Tron* au cinéma quand il est sorti, et ceci s'est avéré être une expérience fascinante. Non ? Vous n'avez pas adoré cette scène où Bruce Boxleitner embrasse passionnément... un programme informatique ? Je vous l'accorde, elle ressemblait à une femme, mais le film explique sans ambiguïté qu'il s'agit d'un morceau de code. Bon, j'admets qu'à l'occasion, j'ai moi-même été tenté, après m'être débarrassé d'un bug particulièrement vicieux dans mon code, de donner à mon moniteur un petit bisou, une petite bise de rien du tout pour célébrer ma victoire. Mais je n'ai jamais vraiment pensé à rouler une pelle à un sous-programme et franchement, j'ai du mal à imaginer ce que Bruce Boxleitner a pu retirer de cette expérience.

Donc, je suis finalement allé voir *Matrix*. Wow ! Un film où absolument tout se passe à travers un filtre vert ! Cool. Je n'y aurais jamais pensé. Alors, comment est-ce qu'il s'en sort comme film sur les ordinateurs ? Eh bien, pour commencer, je lui donne un 10/10 pour le style. Il y a tout ce qu'il faut : des costumes moulants noirs et brillants, un environnement urbain réaliste, des lunettes de soleil inutiles, le tout chargé de violence gratuite au ralenti. Joli travail de caméra. Dommage que tout soit vert, mais peu importe – les mondes projetés par les ordinateurs sont toujours verts, on dirait (sauf pour *Quake* qui est marron). Quoi qu'il en soit, mon cool-o-mètre était au maximum du début à la fin.

Quelque éléments de l'intrigue sont un peu faibles – une résurrection invraisemblable, celle de Tank, et une résurrection un peu plus excusable, car Neo est littéralement un *deus ex machina* — mais dans l'ensemble, c'est intéressant, bien que les allégories bibliques me donnent d'habitude envie de hurler à l'auteur : « Trouve toi ta propre intrigue, espèce de feignasse ! ». Il y avait quelques incohérences aussi, mais avec suffisamment d'effets spéciaux un film peut bien se passer de logique, non ?

Cependant, il y a une chose qui a rompu mon immersion : la prémisse du film. En gros, c'était quelque chose comme « les êtres humains humains sont cultivés par des machines diaboliques afin de récupérer ce que leurs corps produisent – de l'électricité et de la chaleur. »

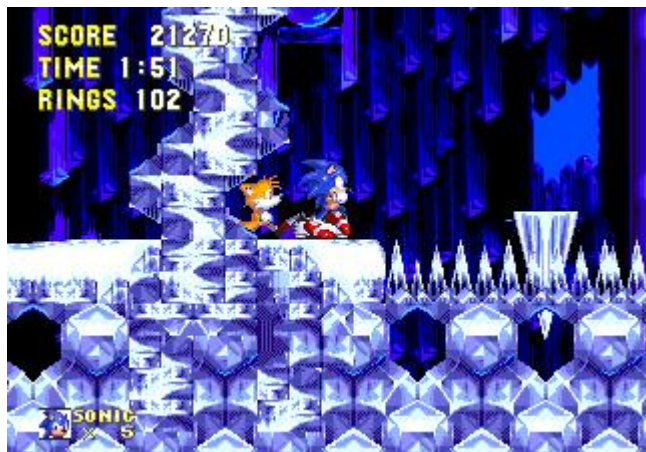
Je suis désolé, mais c'est tout simplement stupide. Ce n'est pas simplement que ce n'est absolument pas pratique – je suis prêt à faire semblant qu'il est possible d'y arriver en travaillant suffisamment dur. Ce qui est stupide, c'est que c'est complètement inutile. Il y a cette petite chose qu'on appelle la [première loi de la thermodynamique](#), dont je suis certain que toute machine diabolique digne de ce nom la connaît par cœur. Cette loi énonce tout simplement que vous ne pouvez pas extraire plus d'énergie d'un système que vous n'en faites entrer dans ce système. Les machines dépensent nécessairement plus d'énergie afin de maintenir ces humains vivants – sans compter la simulation du monde artificiel – qu'ils n'en récupèrent. Imaginez si, à la place des humains dans les cuves, les machines diaboliques gardaient des rangées de hamsters, chacun dans sa propre roue, le tout relié à un générateur. Ils doivent encore nourrir les hamsters ! Et l'énergie dans la nourriture pour hamster *doit* être supérieure à l'énergie produite par les hamsters. Ils feraient mieux de brûler cette nourriture dans une centrale électrique à la place.

Petite parenthèse : d'où la nourriture pour hamster tient-elle son énergie ? Du soleil, mais bien sûr dans *Matrix*, les humains ont bloqué la lumière du soleil, donc les machines ne peuvent pas produire de graines pour nourrir les hamsters. Mais il y a encore le charbon, le pétrole, le gaz, l'uranium, le vent, les marées et l'énergie géothermique à exploiter – *n'importe lequel* de ces moyens serait plus efficace que d'utiliser des humains ou des hamsters.

Donc, 10/10 pour le style, mais seulement 3/10 pour savoir de quoi ils parlent. Maintenant, je suis sûr que ce n'est pas un point de vue très partagé. Je peux déjà entendre les râleurs dire : « Eh mon gars, t'es un sacré nerd ! Ce film était totalement incroyable et faut vraiment être un gros geek pour pinailler sur la première loi de la thermodynamique dans un film aussi génial. »

Et je plaide coupable. Ma tolérance à l'ignorance scientifique, du moins dans des mondes où la science compte, semble s'amenuiser peu à peu. Par exemple, je suis absolument livide face à la décision du *Kansas State School Board* d'exclure l'évolution du programme des écoles publiques [NDT : et il n'est pas le seul]. *Matrix* est un techno-thriller, et à mon avis, il devrait être impeccable sur ce point — du moins sur les éléments qui concernent le cœur de l'intrigue.

Ceci dit, je sais bien que je joue les hypocrites. Sonic est un hérisson irradié avec des chaussures de tennis rouge, bordel. Moi, qui travaille dans l'industrie du jeu vidéo, j'ai vraiment du culot pour exiger des autres médias qu'ils respectent les lois de la nature. Et puis qui est-ce que ça intéresse après tout ?



Eh bien, parlons-en. En fait, en tant que game designer, vous devez être très attentifs à savoir qui ça intéresse et ce que votre public acceptera ou n'acceptera pas. Il y a cette qualité que possèdent les bons jeux et sur laquelle je n'arrive pas à mettre un nom. Ce n'est pas la crédibilité : la plupart des jeux vidéo ne sont absolument pas crédibles. C'est plus que simplement créer une forme d'immersion. Je peux seulement la décrire en creux : quand un jeu ne la possède pas, les joueurs se mettent souvent à engueuler leur écran et à balancer leur manette. C'est cette qualité qui consiste pour un jeu à être fidèle à ses propres règles, à lui-même et au joueur [NDT : on pourrait peut-être parler de cohérence interne ?]. Si vous escroquez le joueur, si vous le trompez, si vous l'arnaquez ou si vous faites quelque chose qui n'a aucun sens (ou pire encore, qui nécessite que *le joueur* fasse quelque chose qui n'a aucun sens), alors cette qualité vous échappe.

Cette qualité, quelle qu'elle soit, n'est pas la même d'un genre à l'autre. Il y a plusieurs manières d'enfreindre les règles du monde réel dans un jeu vidéo, et les conséquences ne sont pas toujours les mêmes.

Il y a d'abord la physique sous-jacente dans une simulation. Le type qui a dirigé l'équipe de développement 3DO d'EA était titulaire d'un doctorat en physique, et il insistait sur le fait que les jeux 3DO qui simulaient le monde réel – comme *John Madden Football*, pour prendre un exemple complètement au hasard – devaient avoir une physique réaliste. Donc, l'équipe a travaillé et travaillé pour rendre la physique aussi réaliste que possible. Ceci a eu exactement les conséquences auxquelles on peut s'attendre, et au final, ils ont enlevé toute cette physique réaliste. Après ça, le jeu est devenu bien meilleur.

Il est très courant de tricher avec la physique dans les simulations. Le fait que ça fonctionne ou pas dépend de ce qui vous pousse à le faire. Si vous le faites pour améliorer le gameplay, comme c'était notre cas, alors allez-y et que Dieu vous bénisse. Si vous le faites pour rendre le jeu plus difficile, en créant une situation où un joueur a du mal à faire face parce que ça ne ressemble pas au monde réel, alors j'estime que vous trichez. Il y a des moyens plus honnêtes pour équilibrer votre jeu.

Mais supposons que nous ayons brisé une autre règle du monde réel : les règles du football elles-mêmes. C'est absolument interdit. M. Madden ne tolérerait jamais ça, pas plus que nos clients. Dans les jeux de sport, c'est un domaine où on ne peut pas tricher. Si vous simulez un sport du monde réel, vous devez respecter les détails du sport, sinon les joueurs, sans oublier les critiques, vont vous passer un savon.

Les simulations d'engins militaires sont un autre domaine où vous devez respecter les règles — la plupart du temps. Les fans de matériel militaire connaissent les moindres détails de l'équipement et ils veulent que la simulation soit exacte du point de vue de l'apparence et du point de vue de la performance. Cependant, tous les gens qui achètent des simulations de vol ne sont pas forcément des grands fans de matériel militaire. Certains d'entre eux veulent juste voler et faire péter des trucs. Pour eux, vous devez faire en sorte que l'avion soit facile à piloter et que les commandes soient assez simples pour faire péter des trucs. Plus important, vous devez faire savoir à vos clients quel genre de jeu de simulation vous leur vendez : soit un jeu hyper-précis, soit un jeu clair et facile. Il faut que ce soit clairement indiqué sur la boîte et lors de la campagne publicitaire. Si vous êtes ambigu sur ce point, vous serez probablement perdant sur les deux tableaux.

Alors qu'en est-il de la science-fiction, comme dans *Matrix* ? La science-fiction enfonce tout le temps les règles. Comment décider quelles sont celles avec lesquelles nous pouvons tricher ? Pour commencer, distinguons le fait de tricher pour des raisons de gameplay et celui de concevoir une prémisse qui va à l'encontre des connaissances scientifiques actuelles. Tricher pour le gameplay est acceptable. Une prémisse impossible peut être acceptable ou non : cela dépend de ce que le public acceptera d'avalier. Une chose qui n'est *pas acceptable* c'est que tout soit permis. C'est l'un des problèmes avec le personnage de Superman : l'étendue de ses pouvoirs semble immense et dans le film, il réussit même à inverser le cours du temps. Non, désolé, mais là, ils m'ont largué : s'il peut faire ça, il peut arranger tout et n'importe quoi.

D'un autre côté, une technique de science-fiction séculaire consiste à briser une seule règle et de voir ce qui se passe ensuite. H.G. Wells a parfaitement réussi cela dans *La Machine à remonter le temps* et aussi dans *L'homme invisible*. Ses protagonistes sont des humains normaux et sont soumis à toutes les faiblesses humaines, mais l'un peut voyager dans le temps et l'autre est invisible. Ses romans explorent les

conséquences possibles de tels pouvoirs – c'est dégrisant dans le premier cas, tragique dans le second. Je vous suggère d'essayer de *briser le minimum de règles possibles*. Plus vous brisez de règles, plus votre jeu perd en crédibilité, voire gagne en ridicule.



L'autre chose à prendre en compte, c'est à quel point la violation est flagrante, et si oui ou non elle est bien connue du public. Prenez l'hyperespace, par exemple. La plupart des histoires de science-fiction dans l'espace dépendent d'une certaine forme de voyage à vitesse supraluminique. La théorie de la relativité d'Einstein semble l'interdire dans un espace normal, mais ça ne nous empêche pas de spéculer sur l'hyperespace, les trous de vers ou autre chose du genre pour arriver au même résultat. Le public est habitué à ça, et vous n'êtes pas obligé de dire avec précision comment ça fonctionne – en fait, ça vaut mieux si vous ne le faites pas. Han Solo parle de « passer en vitesse-lumière » et c'est tout ce qu'il y a à savoir.

Cependant, vous devez éviter des choses que le public trouvera absurde de façon évidente. L'hyperespace n'est pas vraiment absurde, parce que rien dans notre expérience au quotidien ne l'interdit *a priori*. J'ai trouvé l'idée qu'exploiter des humains pour récupérer leur chaleur corporelle était complètement absurde, mais j'imagine que la plupart des gens qui ont regardé *Matrix* n'y ont pas pensé plus que ça. Dans *Le mystère Andromède*, Michael Crichton [NDT : c'est, entre autres, l'auteur de *Jurassic Park*] a émis l'hypothèse d'une forme de vie qui fonctionne sans protéine. Là, c'est hors de mon domaine de compétence. Les biologistes doivent probablement trouver ça drôle, mais je ne suis pas assez instruit pour le savoir, et ça ne me dérange pas. D'un autre côté, l'intrigue n'est pas centrée là-dessus non plus. La révélation-clef du film **[attention spoiler !]**, c'est que l'organisme en question ne peut survivre que dans une fourchette de pH très étroite – l'équilibre entre acidité et alcalinité. C'est tout à fait crédible, parce que les organismes de la Terre sont sujets aux mêmes limitations.

Quand vous faites des jeux dans un style fantasy ou cartoon (et je place *Sonic* et la plupart des autres jeux d'action dans cette catégorie), alors débarrassez-vous des règles. La magie existe, les dragons crachent du feu, des ours transportent des oiseaux dans des sacs à dos – ce que vous voulez. Ça ne veut pas dire qu'il ne faut pas qu'il n'y ait pas de règles : je veux dire par là qu'il faut que vous écriviez les vôtres. Elles doivent avoir un sens dans le contexte de votre monde, et vous éviterez ainsi que le joueur jette sa manette en criant « Mais c'est n'importe quoi ! ».

Vous pouvez proposer un véritable challenge au joueur, vous pouvez le frustrer (dans une certaine limite). Mais vous devez éviter tout ce qui pourrait le mettre en colère, ou pire, s'attirer son mépris ou son dégoût. Les jeux vidéo sont des mondes à part entière : les règles sont faites pour y être brisées. Mais soyez conscients que vous êtes en train de les briser et choisissez soigneusement celles que vous brisez.

Mon blog AAA – épisode 2

Par [Antoine Gersant](#) | le 21 novembre 2012 00:51:51 | Catégories : [Ma vie](#)

Cela fait maintenant sept mois que j'ai publié [le premier épisode de cette chronique](#) qui était censée être mensuelle. La ponctualité rédactionnelle n'a jamais été le grand point fort des Forges (et surtout pas les articles avec mon nom dessus) et j'espère que vous m'excuserez du délai. Au moins, je ne devrai pas avoir trop de mal à faire plus vite pour le prochain épisode.



Aujourd'hui je vais essayer de présenter un peu l'organisation du quotidien, les gens et les méthodes de travail chez Lionhead.

Comme le prétendent les légendes et les making-of de jeux, l'ambiance dans un studio de développement est très détendue et agréable à vivre. La plupart des gens ont toujours le mot pour rire même en période de crise (dans les limites de l'humour anglais), le dress code est inexistant et il y a parfois des bières dans les frigo pour les grandes occasions. Les horaires de travail sont pensés pour des gens normaux (10h00 – 18h00) et sont flexibles dans la limite du raisonnable. En plus de ça, vu que Lionhead appartient à Microsoft les employés bénéficient de plein d'avantages très appréciables, comme par exemple les jeux Xbox Live Arcade gratuits.

Évidemment, tout cela ne marche que parce que l'entreprise fait confiance à ses employés (et elle a bien raison). En contre partie de la *Dolce Vita* et du boulot en short, les gens sont vraiment motivés par leur travail et il est courant de faire des heures sup', par intérêt ou parce que c'est un peu la crise (mais rien de scandaleux comme on peut parfois le lire dans la presse). En bref, la qualité de vie est bien là et c'est chouette de faire partie d'une entreprise qui prend un peu soin de ses employés, ne serait-ce qu'à coup de goodies et de barbecues.

Tout cela se retrouve aussi dans le travail. Je n'avais jamais été aussi bien installé en tant que stagiaire. On m'a fourni un vrai bureau (pas caché dans un coin de l'open space), un PC avec des specs de la NASA ou presque, trois écrans HD et deux kits de développements Xbox 360. Les peluches ne sont par contre pas fournies par Lionhead mais je peux vous dire que j'étais loin d'avoir le bureau le plus décoré (surtout quand on se promène du côté des artistes et des level designers).



Côtés outils, il y a évidemment toute une batterie de logiciels Microsoft (Visual Studio pour programmer, Outlook pour les mails et le calendrier, Lync pour les discussions instantanées, etc.) et ils fonctionnent étonnamment bien les uns avec les autres. En plus de ça, plusieurs outils sont utilisés pour le travail en équipe et la gestion de projets :

- [Perforce](#) est un système de gestion de versions dont je suis tombé amoureux, il combine un peu le meilleur de SVN (que j'utilise pour mes projets personnels) et Git.
- Product Studio est utilisé pour recenser les bugs, récapituler le travail restant et permettre aux producteurs de faire plein de diagrammes (accompagnés de lolcats) pour savoir si le projet est en avance ou en retard. Je vais d'ailleurs en profiter pour glisser deux mots sur le métier de producteur car je n'avais pas une idée très précise de ce métier avant d'arriver chez Lionhead (je voyais ça un peu comme un mec en maillot de bain au bord d'une piscine qui boit des cocktails avec des petits parasols dedans et qui, une fois de temps en temps, décroche son téléphone pour décaler la date de sortie du jeu ou signer un gros chèque). En fait ils remplissent des rôles indispensables. Ils travaillent avec les prestataires externes, ils définissent les priorités de l'équipe et répartissent le travail. C'est particulièrement important en fin de projet quand on sait qu'on aura pas le temps de faire tous les changements et itérations que l'on aimerait : ce sont eux qui jaugent ce qui est le plus important pour la qualité du produit fini (si on demandait à un codeur ce serait de corriger des baisses de framerates, pour un graphiste ce serait de changer telle ou telle texture, et pour un designer ce serait de rajouter des sorts dans le système de combat, etc.). Quand les testeurs relèvent des bugs (plusieurs dizaines de fois par jour), ils atterrissent en premier lieu chez les producteurs qui leur assignent une priorité et déterminent qui est à même de les corriger (bien sûr parfois ils se plantent un peu). Enfin, ils prennent soin de tout ce qui est gestion de projet : ils fixent le calendrier, le planning, les objectifs et la date de sortie (en accord avec les gens du marketing) et s'efforcent de les faire respecter ; c'est comme ça que je me suis retrouvé à travailler sur *Fable: The Journey* qui avait besoin de main d'œuvre supplémentaire pour tenir ses délais.

Je ne me lancerai pas dans des descriptions des autres métiers clés du jeu-vidéo, je suppose que si vous lisez les Forges vous avez déjà une vague idée de ce que font un programmeur, un animateur ou un testeur.

Même avec tous ces beaux outils d'organisation, le plus important c'est quand même d'aller parler avec les gens – c'est encore ce qu'il y a de plus pratique dans bien des cas. C'est moins galère que de taper un mail ou de faire rebondir un bug aux quatre coins du studio et le plus souvent les gens sont contents d'avoir de la

visite à leur bureau (en tout cas, moi, je suis toujours content d'en avoir). En plus ça permet de faire un peu connaissance et de se rincer l'œil sur le chemin car il y a toujours des trucs incroyables sur les écrans des graphistes.

C'est d'ailleurs un bon entraînement contre la timidité que d'aller à la rencontre d'un programmeur qui a bossé sur plein de jeux auxquels vous jouiez quand vous aviez 10 ans pour lui poser une question technique. A cette occasion, j'ai découvert le syndrome de l'imposteur (et j'ai aussi découvert qu'il avait un nom) : il s'agit de ce détestable sentiment (infondé) que vous n'avez rien à faire là et que vous n'avez pas du tout les qualifications nécessaires pour faire votre travail (dans mon cas, ça donne : « Oh la la ! Tous les programmeurs sont super balaises, qu'est ce que je fais là moi ? »). Un rapide tour sur Google ne donne pas beaucoup de crédit scientifique à ce syndrome, mais je me suis assez bien reconnu dedans, attendez-vous à y faire face si vous mettez les pieds dans un studio AAA. J'ai fini par m'en débarrasser à force de voir mon code passer les *code reviews* (votre code est relu par un collègue pour être sûr qu'il n'y a pas un gros bug bien flagrant dedans) sans trop d'accrocs.

Entre les outils logiciels et les gens sympathiques, les informations circulent plutôt bien dans le studio et j'ai été agréablement surpris par la brièveté et la (relative) rareté des réunions. En six mois, je ne me suis jamais retrouvé prisonnier d'un .ppt casse-pied ou d'une réunion sans intérêt pour moi. Les discussions sont tournées autour de la résolution de problèmes et de la prise de décisions, on sent bien que personne n'a envie de perdre son temps.

Je pense avoir donné les grandes lignes de l'organisation du studio et de son ambiance telle que je l'ai perçue. Il manque quelques informations qui auraient été intéressantes à partager mais je ne tiens pas à trop faire le malin avec mon NDA (Non Disclosure Agreement), n'hésitez pas pour autant à poser des questions ou laisser vos commentaires, je ferai de mon mieux pour y répondre. Le prochain article parlera plus en détail du travail que j'ai effectué sur *Fable: The Journey* et un peu de technique (mais pas trop, on est pas sur developpez.com non plus).

Réponses aux questions

Merci pour vos commentaires sur le précédent article. Je me permets de répondre ici aux questions que j'y ai trouvées pour que tout le monde en profite.

Quelles études as-tu faites? (Tipyx)

J'ai passé un Baccalauréat S puis je suis allé en [CPGE](#) au Lycée Fénélon. Au bout de deux ans, j'ai passé les concours d'écoles d'ingénieurs et je suis rentré à l'École Centrale de Nantes (trois ans de formation) dont je suis tout juste diplômé. L'ensemble n'est pas très pratique pour se vendre à l'étranger où les formations de qualité se font toujours en université (du coup, ça leur semble pas terrible alors qu'ici c'est plutôt prestigieux).

Pour les stages au niveau du métier, sont-ils indépendants du cursus que tu suis ? (plink)

Pas vraiment. D'une part les entreprises n'auraient aucune envie de prendre en stage un individu qui n'a pas de qualifications pour les aider dans leur travail. D'autre part, s'il s'agit d'un stage obligatoire, je doute que beaucoup d'écoles/universités laissent leurs élèves partir faire n'importe quoi. A Centrale Nantes, j'ai toujours eu à relier mes stages au métier d'ingénieur.

Quelles sont tes qualifications (en gros, quel(s) langage(s) maîtrises-tu ou faut-il maîtriser ?) (Tipyx)

Mes qualifications détaillées sont sur [mon portfolio](#) mais je pense que le plus important pour trouver un stage/emploi de programmeur dans le jeu-vidéo c'est de maîtriser le C++, un langage plus haut-niveau (genre Java ou C#) et un langage de script (Lua et Python sont les plus recherchés). L'industrie n'a pas l'air

de trop aimer les bricoleurs et je pense qu'il est aussi important d'avoir de solides notions théoriques (algorithmique, compilation, etc.) ainsi qu'une bonne culture informatique.

Comment as-tu été pris (mon ambition à moi c'est de pouvoir effectuer mes stages à Montréal chez Bioware), qu'est-ce qui a été déterminant dans tes compétences ? (BrutalBabylon)

Chez Lionhead (et dans plusieurs autres studios) il y a des [programmes dédiés](#) (pas cette année on dirait) mais le plus souvent il faut envoyer un mail de candidature spontanée. Je pense que ce qui a été le plus déterminant dans mes compétences, c'est d'avoir déjà bossé dans un autre studio (2 mois dans un petit studio Nantais), d'avoir des projets finis et présentables (merci [Ludum Dare](#) !) et d'être intéressé par des domaines pas très populaires auprès des programmeurs (les interfaces graphiques et les outils).

Le texte

Par [Pierre Guyot](#) | le 1 janvier 2013 17:43:49 | Catégories : [Scénario](#), [Univers](#)

Dans un certain nombre de critiques, d'articles, de posts et d'interviews que j'ai pu lire récemment, une même idée semble revenir de façon assez récurrente : le jeu vidéo doit redonner au gameplay une place dominante face à la narration et donc devenir moins « bavard ». Il y a une mouvance assez récente chez les joueurs et les game designers qui semble réticente à utiliser le texte. Le texte, d'une certaine façon, représenterait l'échec d'un gameplay noyé sous la narration (en particulier sous la forme des cinématiques-fleuves qu'on trouve dans les productions AAA).



L'article [Less Talk More Rock](#), écrit par les concepteurs de *Sword & Sworcery* en est un exemple particulièrement intéressant (et dont je vous recommande chaudement la lecture, ne serait-ce que pour les illustrations qui l'accompagnent).

L'idée développée par l'auteur est que le texte n'est pas le mode d'expression « naturel » du jeu vidéo et qu'il n'est pas le moyen le plus fort pour communiquer une idée : une image vaut mieux qu'un long discours. Et en effet, des jeux comme *Journey*, *Limbo* ou *Sword & Sworcery* lui-même ont réussi à se passer du texte comme moyen de narration. Néanmoins, si l'on peut remettre en cause une génération de jeux noyée sous des dialogues bavards, il me semble que rejeter le texte comme élément de narration dans le jeu vidéo, c'est aller un peu loin.

Le jeu vidéo est un médium qui a été beaucoup plus influencé par le cinéma que par la littérature, et plus particulièrement ces dernières années, où l'industrie n'a eu de cesse d'essayer d'imiter les blockbusters du cinéma. C'est, je pense, cette vision cinématographique du jeu vidéo qui explique partiellement la position des auteurs de l'article : le cinéma est un art où il est essentiel de montrer plutôt que dire quand on en a la possibilité. Mais l'image ne permet pas tout et il reste un certain nombre de possibilités offertes par le texte qui restent sous-exploitées par les développeurs de jeux : ce sont ces « espaces textuels » que cet article cherche à mieux cerner.

Fonctions classiques du texte.

Pour commencer, intéressons nous un peu aux diverses fonctions que le texte permet de remplir dans un jeu.

Communiquer.

L'un des premiers intérêts du texte dans un jeu vidéo, c'est de communiquer rapidement des informations au joueur.

Il peut s'agir d'une phase de didacticiel au début du jeu où vous présentez au joueur les commandes, les possibilités qui lui sont offertes ou les règles du jeu. Dans ce cas-là, on peut avoir un pavé de texte à disposition du joueur qui souhaite aller le lire ou bien un texte plus entrelacé avec les phases de jeu, avec par exemple de boîtes de dialogues qui viennent ponctuer un niveau introductif. Dans ce cas-là, le texte s'adresse directement au joueur.

D'une façon moins directe, le texte peut être une occasion de glisser des informations sur l'univers du jeu à travers les dialogues des PNJ. Il peut s'agir de simples éléments de background, mais aussi de pistes pour avancer dans la quête principale ou trouver des éléments de jeu annexes. En général, ce genre de dialogue prend place sous la forme d'allusions : le PNJ ne parle pas directement au joueur, mais parle au personnage pour faire comprendre certaines choses au joueur.



Pour une discussion plus avancée de la question des dialogues chez les PNJ, je vous renvoie à [cet article](#) : les auteurs se sont intéressés à de nombreux points de la conception de *Final Fantasy VI* et sont attachés en particulier à comprendre comment l'ensemble des dialogues du jeu est construit de façon à diriger le joueur sans le prendre pour un imbécile.

Caractériser les personnages.

Le texte, sous la forme du dialogue, est aussi l'occasion de présenter et de caractériser les personnages. C'est notamment un élément essentiel pour donner de la variété et de la profondeur aux différents protagonistes.

D'une part, le contenu du texte est l'occasion de présenter les personnages : le protagoniste parle de son histoire, de ses aspirations, de ses doutes, de ses peurs. D'autre part, la forme même du texte permet aussi de préciser ce personnage : sa façon de parler (hésitante, assurée, neutre), son registre de langage (soutenu, familier, ordurier), ses tics verbaux (répétitions, expressions, fautes) et beaucoup d'autres d'éléments permettent de révéler au joueur le caractère du personnage qui s'exprime.



Le jeu amateur *Off* sur RPGMaker (si vous n'y avez pas joué, foncez le [télécharger](#)) est un exemple particulièrement réussi de caractérisation des personnages à travers les dialogues. Rien qu'en lisant une réplique, on sait immédiatement qui est le personnage qui parle : le batteur est laconique, le juge déploie une prose particulièrement raffinée, Dedan est ordurier, Vadereloha est tragique... Ces dialogues sont une des forces du jeu, au point qu'une équipe de traducteurs amateurs américains a intégralement traduit le jeu en anglais.

Un exemple encore plus extrême est celui de *Chrono Cross*. Le jeu proposant au joueur de recruter 50 personnages, il était impossible de réécrire tous les dialogues en tenant compte de toutes les combinaisons possibles. Une bonne partie des dialogues est donc écrite de façon neutre, puis passée au travers d'un algorithme qui lui donne la caractérisation propre à un personnage : certains parlent entièrement en majuscules, d'autres ponctuent leurs dialogues de « Moi » et de « Toi » (en français dans le texte), d'autres encore rajoutent des accents divers sur leurs voyelles, etc.

Proposer des choix.

Le texte est aussi un moyen de présenter de façon simple des choix. Bien qu'il puisse y avoir d'autres façons de présenter un choix au joueur, le texte constitue le moyen le plus simple, le plus direct de le faire, en général sous la forme d'une liste d'alternatives.

C'est aussi ce qui fait parfois aussi sa difficulté d'utilisation : comment rester subtil dans la formulation des alternatives ? Un jeu comme *Mass Effect* (tout du moins le premier opus) pâtit lourdement de la forme de ses dialogues : les alternatives sont toujours divisées selon un axe conciliant/pragmatique, l'alternative conciliante en haut, l'alternative pragmatique en bas. Si ce système favorise la clarté, il nuit aussi à la subtilité, car la présentation des alternatives trahit toujours le manichéisme des choix.

Le texte, un outil pas forcément exploité.

À côté de ces fonctions classiques et plutôt bien cernées par les développeurs, d'autres possibilités pour jouer avec le texte dans le jeu existent et sont souvent mises de côté par les créateurs.

Des espaces textuels sous-utilisés.

L'ensemble des noms d'attaques, d'armes, d'équipements, d'objets forment un ensemble qui, s'il est utilisé de façon cohérente, participe à la profondeur de l'univers du jeu. Les descriptions de ces objets sont des espaces où le scénariste peut placer des informations d'une façon intéressante : le joueur qui n'en a pas envie n'est pas obligé de les lire, le joueur qui les lit se sent plus immergé, voire récompensé par des informations plus ou moins dissimulées. Dans *Demon's Souls* et son successeur *Dark Souls*, les descriptions des différents

items disponibles dans l'inventaire recèlent un nombre surprenant de micro-histoires, voire d'indications sur la façon de récupérer certains pouvoirs.

Dans la série des *Mother* ou dans *Radical Dreamers* (la suite de *Chrono Trigger* et le « brouillon » de *Chrono Cross*), les combats sont racontés au moins partiellement sous la forme de phrases et non de visuels ou de chiffres. Ces phrases qui décrivent les combats sont l'occasion d'ajouter de la variété aux actions du joueur (ne serait-ce qu'en changeant la manière de raconter une simple attaque de base) voire d'élaborer des micro-histoires à l'intérieur même des combats.



Si *Radical Dreamers* proposent des joutes classiques où la scénarisation permet de faire la tension chez le joueur, les *Mother* jouent la carte de l'humour. Ainsi, au cours de certains combats contre de petits robots appelés Frightbots, ceux-ci se contentent de laisser leurs alliés attaquer à leur place et de raconter des histoires d'horreur (oui, vous avez bien lu) : les combats contre eux sont régulièrement émaillés de phrases comme « Le Frightbot raconte une histoire à vous glacer les sangs. » ou « Le Frightbot raconte une histoire d'horreur, mais néanmoins émouvante. ». Si sur le plan du gameplay, ces ennemis sont inutiles, ils apportent néanmoins beaucoup sur le plan de l'ambiance et de l'univers du jeu. Nombre de joueurs se souviendront par ailleurs de la fameuse phrase « You cannot grasp [SPOILER]'s attack true form ! ».

Ainsi, tous ces petits bouts de texte sont en général considérés comme inutiles ou simplement basiques, mais il y a matière à travailler dessus pour produire des effets intéressants, voire une profondeur de jeu supplémentaire.

Typographie et éclatement.

La forme même du texte peut faire l'objet d'un travail. Les scénaristes et les traducteurs vont ainsi parfois jusqu'à utiliser les subtilités de la typographie pour produire des effets uniques.

Dans *Final Fantasy X*, lors de la scène où Tidus est confronté à des réminiscences de son père à qui il en veut tout particulièrement, les auteurs ont joué à la fois sur l'éclatement spatial du texte (la voix de son père « résonne » partout sur l'écran de façon aléatoire) et sur le timing (les phrases apparaissent très brièvement, comme si elles étaient criées). C'est un moyen visuel très efficace pour rendre l'impression que Tidus est particulièrement oppressé par ces voix intérieures. Dans le huitième opus de la série, c'est la casse qui est mise à rude épreuve : lors du dialogue précédent le combat final contre Ultimecia, cette dernière alterne de façon chaotique majuscules et minuscules, créant ainsi une forte dissonance avec le décorum lisse et raffiné et donc une impression de malaise chez le joueur.

Dans *Nier*, une phase entière du jeu se déroule sous la forme de texte. Au début, la forme en est classique, mais, au fur et à mesure, les phrases et les mots s'éclatent de plus en plus jusqu'à ce que le texte devienne incompréhensible, ce qui traduit d'une autre façon l'effondrement que décrit le texte lui-même. Les personnages vont même jusqu'à commenter la bizarrerie de cette façon de jouer. Ce jeu sur la frontière entre joueur et personnage peut devenir très intéressante. On pourrait tout à fait imaginer un personnage qui commenterait lui-même le texte qui le prononce et la façon dont celui-ci s'écrit : « Oh regarde ! Quand le vieux sage dit « Cristal », ça s'affiche en rouge, ça doit être sacrément important ! » !

Dans l'exemple de Nier, l'effet obtenu est tout particulièrement caractéristique du texte et ne peut pas être reproduit par d'autres vecteurs.

Dissonances.

Le texte permet aussi de produire des effets de dissonance que le visuel ou le gameplay ne permettent pas de reproduire. Lorsque le texte est censé décrire ce qui se passe à l'écran, que se passe-t-il si celui-ci dévie et entre en contradiction avec ce qui est visible à l'écran ? Un effet de malaise, de dissonance.

Imaginons qu'à l'écran, un personnage doublé parle avec des sous-titres. Au début, le texte n'est que la transcription de ce que dit le personnage. Mais au bout d'un moment, le joueur réalise que certains mots cruciaux diffèrent entre le texte écrit et le texte dit. Il en résulterait une forte impression de malaise, le joueur penserait sans doute « Est-ce que j'ai vraiment bien lu/entendu ? ». On peut aussi imaginer un joueur entrant dans une pièce, au milieu de laquelle se trouve un objet quelconque, mettons un cube rouge. Imaginons maintenant que le texte qui décrit les actions du joueur affiche : « Vous entrez dans une pièce au milieu de laquelle trône un cube *bleu*. ». De la même façon, vous générez une impression de malaise.



Si ce genre d'effet, qui joue sur la limite entre diégétique et extradiégétique (le fameux « Fourth Wall »), n'a pas forcément sa place dans un certain nombre de jeux, il peut permettre des effets très intéressants pour des jeux d'horreur ou qui traitent de la thématique de la folie.

Nier en propose un exemple frappant dès son ouverture : après que le joueur a lancé une nouvelle partie, les mots « Été 2049 » s'affichent à l'écran. Juste après qu'ils ont disparu, le jeu s'ouvre sur une ville couverte de neige. La première réaction qui vient à l'esprit, c'est quelque chose du genre « Quelque chose ne va pas. ». En cinq secondes, une impression très forte est faite sur le joueur et conditionne tout le ressenti qu'il aura face au prologue du jeu.

Dans une moindre mesure, *Off* joue aussi de cette dissonance pour réaffirmer la bizarrerie de son univers. Au fur et à mesure que le joueur avance dans sa quête, il découvre chacun des quatre éléments (que je vous laisse le soin de découvrir :p). À chaque fois, la même petite phrase vient conclure la description de l'élément : « C'est le premier des quatre éléments. ».

Références et méta-texte.

Sur plusieurs jeux d'une même série ou d'un même auteur, les noms et les descriptifs peuvent se répandre, de façon à créer une attente chez le joueur. Dans des séries longues comme les *Final Fantasy*, les *Dragon Quest* ou un peu différemment, les jeux *Tri-Ace*, le fait qu'un certain nombre de noms reviennent de façon récurrente (comme Shiva, Ifrit, Dharma, Zenithia, Gabriel Celesti, etc.) donne l'impression que ces noms font référence à quelque chose, à une *mythologie*. Cette utilisation du texte crée une forte impression de profondeur et de cohérence de l'univers, car elle est transversale. C'est un peu le même mécanisme

qu'utilisaient les *Final Fantasy* au départ en faisant référence à la mythologie ancienne : à force d'être employés, ces mots sont devenus partie d'une mythologie interne à la série.



Cette utilisation de la répétition peut aussi être utilisée pour donner au joueur des informations cachées. Les jeux de Yasumi Matsuno, par exemple, n'ont pas de lien apparent mais se déroulent dans le même univers, Ivalice. Si les descriptions de chaque jeu sont efficaces et bien écrites, un certain nombre d'entre elles ne révèlent leur véritable sens qu'une fois que le joueur a fait plusieurs des jeux de la série. Par exemple, la description de certains objets de *Vagrant Story* éclaire le dénouement ambigu de *Final Fantasy Tactics*. C'est aussi un moyen de récompenser les joueurs qui s'y intéressent sans pénaliser les autres. C'est une façon particulièrement efficace de créer la sensation de « secret » qui a été encensé dans *Dark Souls* et dont beaucoup ont [déploré l'absence](#) dans les derniers *Zelda*.

La répétition de noms et de répliques à travers différents jeux permet de créer un « méta-texte » qui donne de la profondeur, de la densité à l'univers et aide à construire ce qu'on pourrait appeler une véritable mythologie.

Mise en forme.

Pour finir, j'aimerais revenir sur quelques aspects plus triviaux de l'écriture et sur la façon de la mettre en forme. Si certains points semblent évidents, un rappel ne me semble cependant pas inutile quand on voit les ratés qu'il y a pu avoir sur les jeux de certaines très grosses boîtes (oui oui, je pense à toi, Square Enix). Je mettrai cependant l'aspect graphique de côté pour cette discussion, celui-ci sera sans doute traité dans un prochain article :p

Orthographe et syntaxe

Il faut impérativement que l'orthographe, la syntaxe et la concordance des temps soient irréprochables. Ceci semblera évident à beaucoup de monde, mais nombre de projets amateurs ont une orthographe déplorable et certains projets de grands studios aussi ! Un des exemples qui m'a le plus marqué à titre personnel est la traduction française abominable d'*Okami* (qui est pourtant une merveille par ailleurs) : outre la vulgarité crasse de certains noms, les traducteurs ont semblé très attachés à ne pas conjuguer correctement les verbes. Même si le français n'est pas la langue la plus simple à maîtriser, dites vous que c'est un critère qui permet d'écarter rapidement des jeux bâclés pour les joueurs et les testeurs. Si nécessaire, faites vous relire, si possible par plusieurs personnes.



Rythme d'affichage.

C'est un problème qu'on retrouve souvent dans les RPG, tout particulièrement ceux qui ont été réalisés par des amateurs. S'il peut être intéressant de jouer sur le timing avec lequel le texte s'affiche (pour marquer des pauses par exemple), il est insupportable pour un joueur de ne pas pouvoir passer rapidement les dialogues, tout particulièrement s'il les a déjà lus. On se souviendra bien évidemment du fameux hibou d'*Ocarina of Time* qui, non content de parler très lentement et d'empêcher le joueur de passer à la suite, prend un malin plaisir à tout répéter (pour la petite histoire, c'est parce que les développeurs ont oublié de changer le rythme d'affichage, adéquat pour des caractères japonais, lors de la localisation en Occident :p).

Des jeux comme les *Persona* récents et les *Ace Attorney* ont très bien négocié ce problème : dans *Persona*, en maintenant une touche appuyée on peut faire défiler rapidement les dialogues, mais cette même touche ne permet pas de valider quand le jeu propose un choix, ce qui évite de cliquer par mégarde. Dans les *Ace Attorney*, il est possible de faire défiler les dialogues en accéléré la deuxième fois qu'on les lit, ce qui permet de retourner directement au passage qui vous intéresse.

Traduction.

Bien que ce problème ne se posera sans doute qu'à assez peu de créateurs amateurs, la traduction est un point qui est souvent négligé, même par les plus gros studios. Je mentionnais plus haut la localisation lamentable d'*Okami*, mais on pourrait tout aussi bien parler de la traduction de *Final Fantasy VII*, dont les dialogues sont truffés de fautes d'orthographe, de contresens, voire dépassent des boîtes dialogues ou celle de *Dissidia* où l'anglais et le français se mélangent allégrement (oui oui, Square Enix, c'est encore à toi que je parle).

Une mauvaise traduction peut très facilement rompre l'immersion du joueur. Par ailleurs, pour des scénarios un peu élaborés, par exemple dans un RPG qui ferait intervenir un peu de politique, il est particulièrement important de faire attention à la façon dont les marques de politesse passent d'une langue à l'autre. Le passage du « you » anglais au « tu/vous » français est particulièrement maltraité en général. *Saga Frontier II*, malgré une excellente histoire, en est une triste victime : quand un soldat de seconde classe s'adresse à un roi, un général ou à un ministre, il ne le tutoie pas et même un traducteur amateur devrait le savoir.

A l'inverse, une traduction réussie peut apporter beaucoup. Pour s'en convaincre, il suffit de jouer aux opus récents de la saga *Persona* où les traducteurs ont fait attention de préserver les marques de politesse japonaises, ce qui leur permet non seulement d'avoir des dialogues fidèles à l'original, mais aussi de préserver l'ambiance *anime* très particulière de ces jeux.

Minimalité.

Dans l'ensemble, soyez paresseux et faites en toujours le minimum. Essayez de vous en tenir à l'essentiel pour ne pas lasser le joueur, mais aussi parce qu'écrire des dialogues, c'est beaucoup de boulot (et vous en aurez sans doute assez par ailleurs). En général, si une scène ou une ligne ne vous paraît pas absolument essentielle, si vous n'arrivez pas à exprimer la fonction qu'elle sert dans votre jeu, autant l'enlever.



J'ai été assez frappé en rejouant à certains opus de la série des *Zelda* récemment et notamment, j'ai été assez attristé de voir une dégradation dans l'utilisation des dialogues.

Prenons l'exemple de l'opus Super Nintendo, *A Link to the Past* : lorsque Link récupère un objet (mettons la baguette de glace), la description qui s'affiche à l'écran est d'une concision remarquable.

« Vous avez trouvé la Baguette de Glace ! » : on vous donne le nom de l'objet. « Utilisez là pour lancer des bourrasques glacées et geler vos ennemis. » : on vous explique à quoi elle sert. « Attention à votre compteur de Magie ! » : on vous donne la règle d'utilisation.

En trois lignes de texte (qu'il est de plus possible de passer rapidement si on en a envie), le jeu introduit l'objet de façon complète et efficace. C'est une utilisation minimale (et admirable) du texte dans un jeu.

Regardons maintenant du côté de son successeur, *Ocarina of Time* : lorsque Link récupère un objet quelconque, non seulement le joueur doit se farcir une description longue, qu'il ne peut pas passer, mais surtout, le jeu lui rappelle comment équiper ledit objet avec les boutons C. Pourtant, c'est la même règle pour tous les objets et le joueur l'a déjà vu quinze fois en une heure de jeu, sans compter les Kokiris qui le lui rappellent au début du jeu. Rapidement, le joueur se retrouve à relire régulièrement un laïus qu'il connaît déjà par cœur, qui ne lui apporte aucune information nouvelle et qui parasite son rythme de jeu (pour rappel, chaque fois que vous ramassez un objet dans un flacon, soit assez souvent, vous devez vous retaper ce même laïus !).

Entre les deux jeux, il y a un véritable fossé en matière d'utilisation du texte (et qui malheureusement, va dans le mauvais sens). Il me semble qu'en tous les cas, il vaut toujours mieux essayer de « dégraisser » le texte autant que possible. Et surtout, ne prenez pas les joueurs pour des abrutis !

Conclusion

Malgré une mouvance dans le jeu vidéo qui cherche à s'affranchir du texte, celui-ci reste un outil très puissant et souvent mal exploité. Même s'il n'est pas évident d'écrire des dialogues subtils, il est toujours possible d'assurer une qualité minimum en réfléchissant bien aux fonctions qu'on prête au texte et en demandant l'aide d'autres personnes pour des relectures (il peut s'agir de personnes n'ayant pas un intérêt particulier pour le jeu vidéo, un avis extérieur est toujours le bienvenu !). Assurer une orthographe, une syntaxe et un respect minimal des registres de langue est à la portée de tout le monde et en particulier des

indépendants et des amateurs. Si c'est un sujet sur lequel même l'industrie pêche, c'est aussi l'occasion de montrer que sur certains points, le jeu indépendant peut faire aussi bien voire mieux que le jeu pro ! :P

Compte-rendu de Ludum Dare #25

Par [Lyonsbanner](#) | le 13 janvier 2013 22:43:05 | Catégories : [Game Design](#), [Graphisme](#), [Ludum Dare](#), [Pixel art](#)

Eh oui ! Cette fois, c'est moi, Lyonsbanner (à votre service), qui vous présenterai le compte-rendu de la 25^{ème} édition du *Ludum Dare* qui s'est déroulé du 14 au 17 décembre. Vous pouvez retrouver tous les comptes-rendus précédents dans une nouvelle catégorie « [Ludum Dare](#) ». Pour ceux qui ne connaissent pas le principe, il s'agit d'un week-end pendant lequel des développeurs du monde entier créent des jeux sur un thème imposé. Les participations peuvent se faire soit en 48 heures en solo (la compétition ou *compo*), soit en groupe en 72 heures (le *jam*).

L'année 2012 a été plus que fructueuse, car il y a eu un total de 4651 jeux qui ont été réalisés dont 1327 pour cette 25^{ème} édition. C'est énorme ! D'autant plus qu'à l'heure actuelle l'ensemble des jeux Ludum Dare depuis la première édition a dépassé [le nombre total des jeux sur consoles toutes catégories confondues](#). Parmi tous les participants, j'étais dans le lot et cet article retracera mon parcours.

Malgré mon expérience en tant que développeur d'application, je n'étais pas très à l'aise face à la réalisation d'un jeu en temps limité. À vrai dire, je n'avais pas de « vrais » outils à ma disposition pour créer un jeu vidéo, juste quelques connaissances avec [Canvas](#) que j'ai découvert lors du Microsoft Days' 11. Je n'étais pas sûr de présenter quelque chose et finalement, je me suis dit que dans le pire des cas, j'aurais au moins participé. D'ailleurs, une des particularités du *Ludum Dare* est d'aider les participants à être *in the move*, et ça quelque part, c'est très motivant !

N'étant pas habitué aux nuits blanches, plutôt que d'attendre l'annonce du thème à quatre heures du matin, j'ai décidé de recharger mes batteries afin d'être d'attaque le lendemain, ce qui est d'ailleurs fortement recommandé par les organisateurs même si la plupart préfère déroger à la règle. C'est alors qu'est apparu sous mes yeux ébahis le thème de cette édition : « **You are the Villain** ». En français, on peut le traduire par « Vous êtes le Scélérat, le Traître, le Vaurien, le Rustaud », bref, il fallait donc que je réalise un jeu où le joueur joue le rôle du méchant, un peu comme l'excellent *Dungeon Keeper*. Après avoir pris connaissance du thème en long, en large et en travers... je sors faire une promenade afin de méditer sur mon projet et de trouver des idées (ça aussi ça fait parti des recommandations).

Et c'est parti !

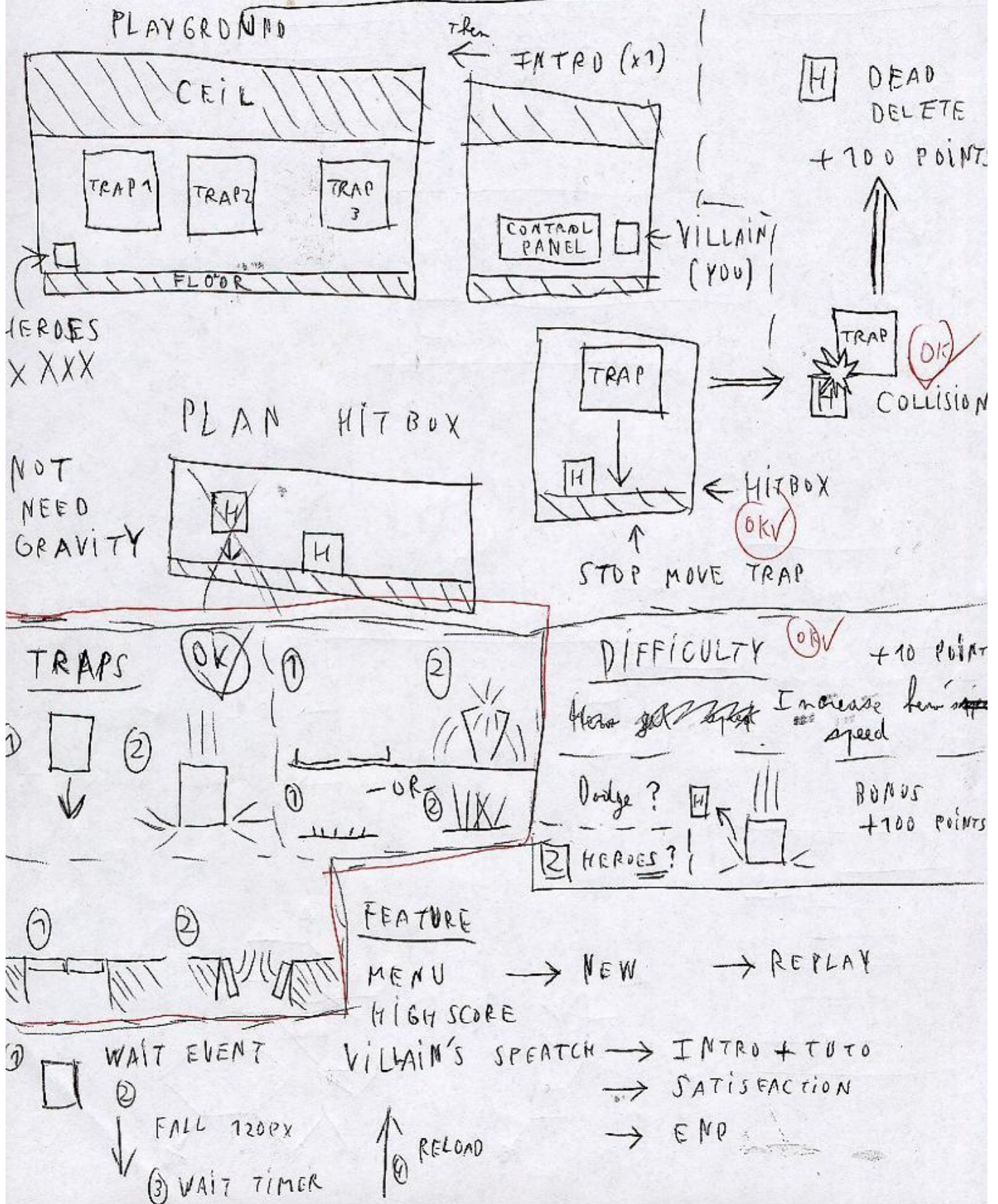
De retour chez moi, j'attrape ma planche à écrire et en cinq minutes, je jette sur le papier des notes sur les éléments de jeu, le gameplay ainsi que les théories sur les événements comme les collisions et le système de scoring. Voyez plutôt :

24x5
= 120

LUMDUM DARE 25

(Scales)
(Taste)
(Vain)
(Prestige)

YOU ARE THE VILLAIN

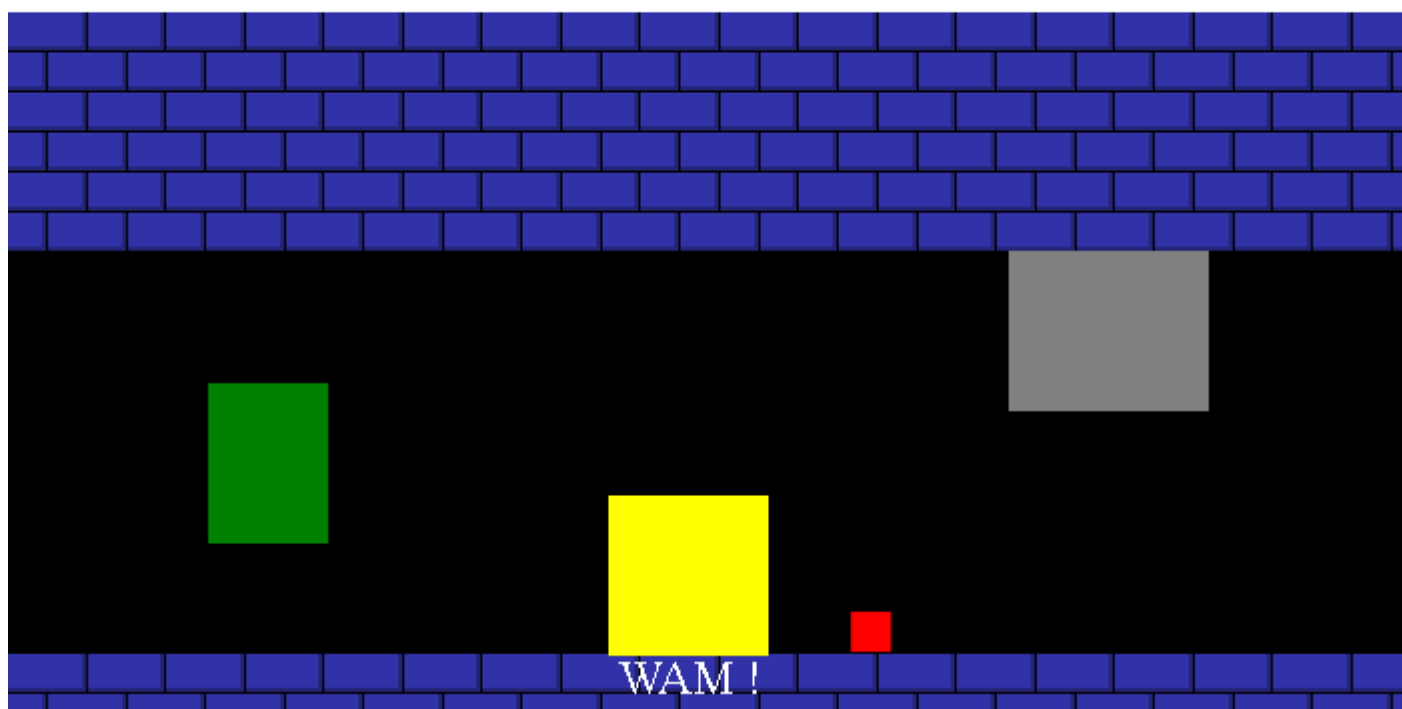


Le concept est simple : le joueur a à sa disposition des pièges qu'il doit activer afin d'empêcher le héros de sortir par la droite de l'écran. Chaque fois que le joueur réduit le héros à l'état de pizza, il gagne des points, si ce dernier atteint la droite de l'écran, le joueur perd un cœur. Quant au titre, j'ai choisi « Tripe Traps » : je n'étais pas très inspiré, mais au moins les deux mots commencent par un T et ont chacun cinq lettres (la classe, quoi !).

Fier de mon idée, je décide d'aller faire un tour sur [Twinoid](#) où quelques membres participaient également au concours, ainsi que sur le [TwitchTV de Deepnight](#), un des administrateurs de Twinoid qui croule sous les *awards* du Ludum Dare. Discuter de nos projets et même les tester entre nous est non seulement très convivial mais aussi très motivant !

J'ai passé toute la journée à dessiner et à faire bouger des carrés de couleurs un peu partout. Les gros carrés représentaient les pièges qui devaient se fracasser avec violence sur le sol afin d'écrabouiller le petit carré représentant le héros. Les premiers essais étaient rigolos, notamment les pièges qui passaient à travers le sol quand on les réenclenchaient, ou encore le héros qui se faisait tuer alors que le piège était en train de remonter tranquillement. Minuit passé, je me suis dit qu'il était temps d'intégrer des images afin « d'habiller » l'ensemble. Charger une image dans un Canvas, ce n'est pas un souci, mais dès qu'il y en a plusieurs, curieusement la tâche devient assez complexe. J'ai donc passé le reste de la soirée à stresser et à chercher une solution. En discutant sur IRC, j'apprends l'existence de *Crafty* et de *EaselJS*, des bibliothèques qui auraient pu me sauver la mise mais toutes les tentatives pour les intégrer dans mon code se sont soldées par des échecs. J'ai finalement trouvé un autre moyen d'y arriver, un peu barbare sûrement, mais au moins j'ai obtenu ce que je voulais. J'avais l'esprit tranquille, le sommeil me guettait et je m'affalai comme une masse sur mon lit.

Tripe Traps



Au petit matin, je saluai mes compagnons d'IRC qui soit dormaient encore, soit venaient de se réveiller, soit étaient déjà réveillés depuis la veille au matin (voire la veille du matin d'avant). Les premiers avis que j'ai reçus étaient positifs, le jeu était jouable même s'il n'y avait pas encore de score et même s'ils adoraient jouer avec des carrés. Ni une ni deux, j'avale un chocolat chaud et je replonge dans mon passionnant programme pour arranger tout ça.

Au Ludum Dare, le jeu final est noté sur [plusieurs critères](#) (voir la partie *Judging*). Comme je n'avais plus qu'une journée pour finir mon jeu, je me suis dit que je devrais d'abord m'appuyer en priorité sur le

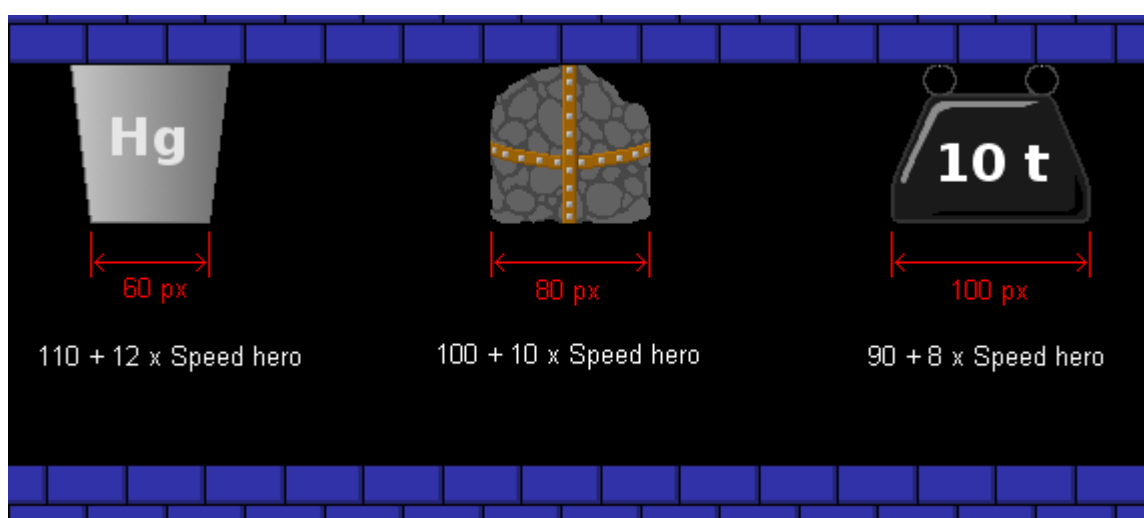
graphisme et l'humour. En revanche pour le son, je pensais que j'allais manquer de temps (et ce fût malheureusement le cas).

Grâce aux cours de pixel art que vous pouvez retrouver sur le site, j'ai dessiné mon premier piège qui se trouve être une sorte de gros bloc de pierre. Le deuxième était un peu plus large pour des raisons de gameplay (que j'évoquerai tout à l'heure) et plutôt que de faire des pièges qui se ressemblent, j'ai donc dessiné un poids de dix tonnes un peu comme on en voit dans les mangas lors de situations comiques. Le troisième piège était plus petit que les deux autres, je comptais faire un bloc de marbre, ou encore créer une herse qui se déclencherait à même le sol. Seulement, le code était déjà en place et pour des raisons de gameplay, la herse n'allait pas. Le bloc de marbre n'était pas assez original à mon goût. Finalement, j'ai opté pour quelque chose d'encore plus loufoque : un seau de mercure. Il ne me restait plus qu'à dessiner le petit héros animé sur deux frames, les « vies » du joueur et le « méchant ».

Le gameplay et le score.

Comme je l'ai dit plus haut, le score augmente dès que le joueur réussit à aplatir le héros avec les pièges. Mais c'est là que je me suis rendu compte qu'il fallait équilibrer le jeu à tout point de vue.

1. Le héros se déplace à une vitesse constante, ce qui rend le jeu trop simple et fait grimper le score au firmament.
Solution : le héros se déplace de plus en plus vite.
2. Au bout d'un moment, le héros va très vite et à tendance à « surgir » tellement vite par la gauche qu'il devient très difficile de le toucher avec le premier piège.
Solution : les points obtenus sont plus importants pour le premier piège, récompensant les joueurs qui sont capables d'anticiper. De plus, un petit chrono s'affiche avant l'apparition du prochain héros, ce qui laisse le temps aux joueurs de souffler un peu.
3. Les pièges ont tous la même taille et il est bien plus facile d'anticiper le déplacement du héros en utilisant le dernier piège.
Solution : rendre le premier piège plus petit pour défier les vétérans (et les récompenser en conséquence) et rendre le troisième piège plus grand afin de ne pas pénaliser les joueurs débutants (en contrepartie, le score obtenu est moins important).
4. Nous obtenons toujours la même tranche de score alors que le héros se déplace de plus en plus vite.
Solution : le score obtenu est multiplié par le facteur de vitesse du héros.

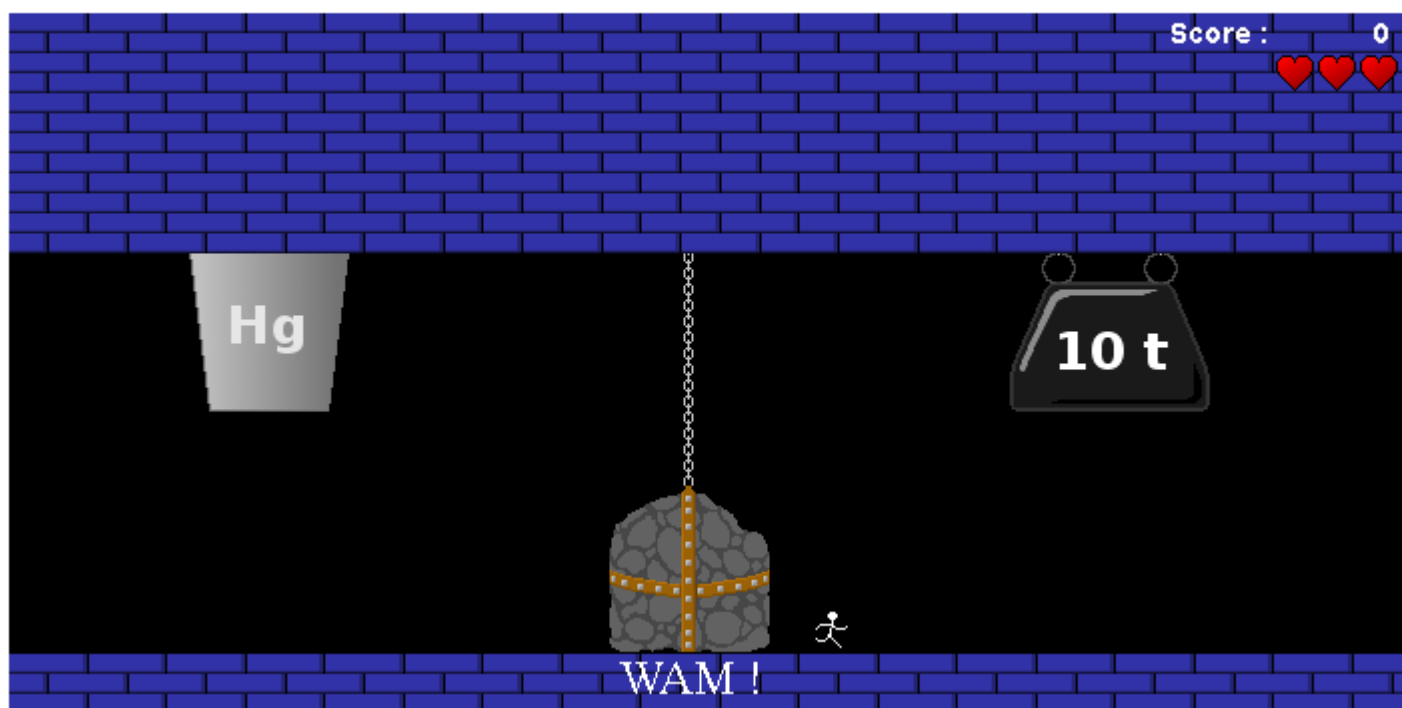


Ouf ! Ça en fait des ajustements. Une autre tâche m'attendait encore : le texte. Que serait un jeu si les développeurs n'en expliquaient pas un minimum les règles ? À quel point le jeu serait ennuyeux s'il n'y avait aucune information qui montre au joueur sa progression, comme l'objectif ou le score ? Quoi de plus normal que de rendre vivant le « méchant » qui vous encourage dès que vous vous débarrassez du héros ou qui se

retrouve lui-même confronté face au héros ? Et bien sûr, ne serait-il pas naturel d'indiquer les touches à utiliser pour pouvoir y jouer ?

Le plus gros était fait et mon jeu commençait vraiment à ressembler à quelque chose. Il n'était que 2h30 quand... QUOI ?! Déjà ? Aïe, aïe, aïe ! Il me restait à peine une heure et demie avant la fin du temps imparti ! Il ne me manquait plus que le son, le temps de chercher comment intégrer ce son dans mon code et avec quel logiciel, mais je devais d'abord comprendre comment envoyer mon œuvre sur le site du Ludum Dare (honte à moi !). C'est malheureusement la fatigue qui a eu raison de moi, et je me suis dit que pour un premier essai, c'était déjà pas mal, j'étais fier de ce que j'avais fait alors que deux jours auparavant, l'idée même de créer ET de finir un jeu vidéo me semblait irréalisable. Et pourtant, je l'ai fait et voilà le résultat :

Tripe Traps



Verdict et résultats.

Trois semaines après la compétition, c'est-à-dire le 8 janvier 2013, les résultats ont été annoncés. Vous pouvez consulter [ma page](#) sur le site du Ludum Dare et jouer à [Tripe Traps](#) (pour les anglophobes, il vous suffit d'appuyer sur S pour jouer et rejouer. Pour activer les pièges, respectivement de gauche à droite, les touches 1, 2 et 3).

Dans le classement général, les points forts qui ont été le plus appréciés sont l'humour, le fun et le respect du thème imposé. En revanche, l'aspect graphique est en dessous de la moyenne, ce qui est assez logique quand j'y repense : ça reste assez sommaire et a plutôt tendance à accentuer le côté gag de la chose plutôt que la beauté artistique. Le plus drôle, c'est que j'ai quand même eu le droit à une petite note sur l'audio, qui est bien mieux évalué que les graphismes (beaucoup de gens, moi le premier, font l'erreur de mettre une étoile sur cinq alors qu'il est possible de ne pas en mettre du tout).

Vous trouverez [ici](#) la liste des jeux de tous les participants de Twinoid, ça vaut le détour.

Un de mes amis m'a fait remarquer que le gameplay de mon projet était similaire au gameplay de [celui-ci](#). J'y avais joué il y a très longtemps et si ça se trouve, j'ai été influencé par d'anciens souvenirs de ce jeu. Mais il est également possible que j'aie été inspiré par le donjon de Bowser dans *Super Mario Bros* ;)

Vilain Game Designer ! Pas de biscuit pour toi !

VIII

Par [Lyonsbanner](#) | le 4 février 2013 22:29:48 | Catégories : [Ernest Adams](#), [Game Design](#), [Pas de biscuit](#)

Cet article est une traduction de [Lyonsbanner](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (septembre 2007).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

Une fois de plus, le moment est venu pour une nouvelle édition annuelle de Vilain Game Designer ! Pas de biscuit pour toi ! Depuis l'année dernière, j'ai reçu une bonne quantité de *Twinkie Denial Conditions* (NDT : littéralement « condition pour refuser un biscuit », à savoir à un game designer qui a mal fait son travail) de mes lecteurs, que je vous présente maintenant, pour votre éducation et surtout pour votre plaisir. De plus, j'ai finalement tenu une vieille promesse : mettre en place une [base de données](#) rassemblant tous les TDC, triés par catégorie. Il vous suffit de cliquer sur le lien pour vous retrouver sur mon site Internet.



Et c'est reparti ! Certains d'entre eux sont tellement énormes que j'aurai vraiment dû en parler des années plus tôt.

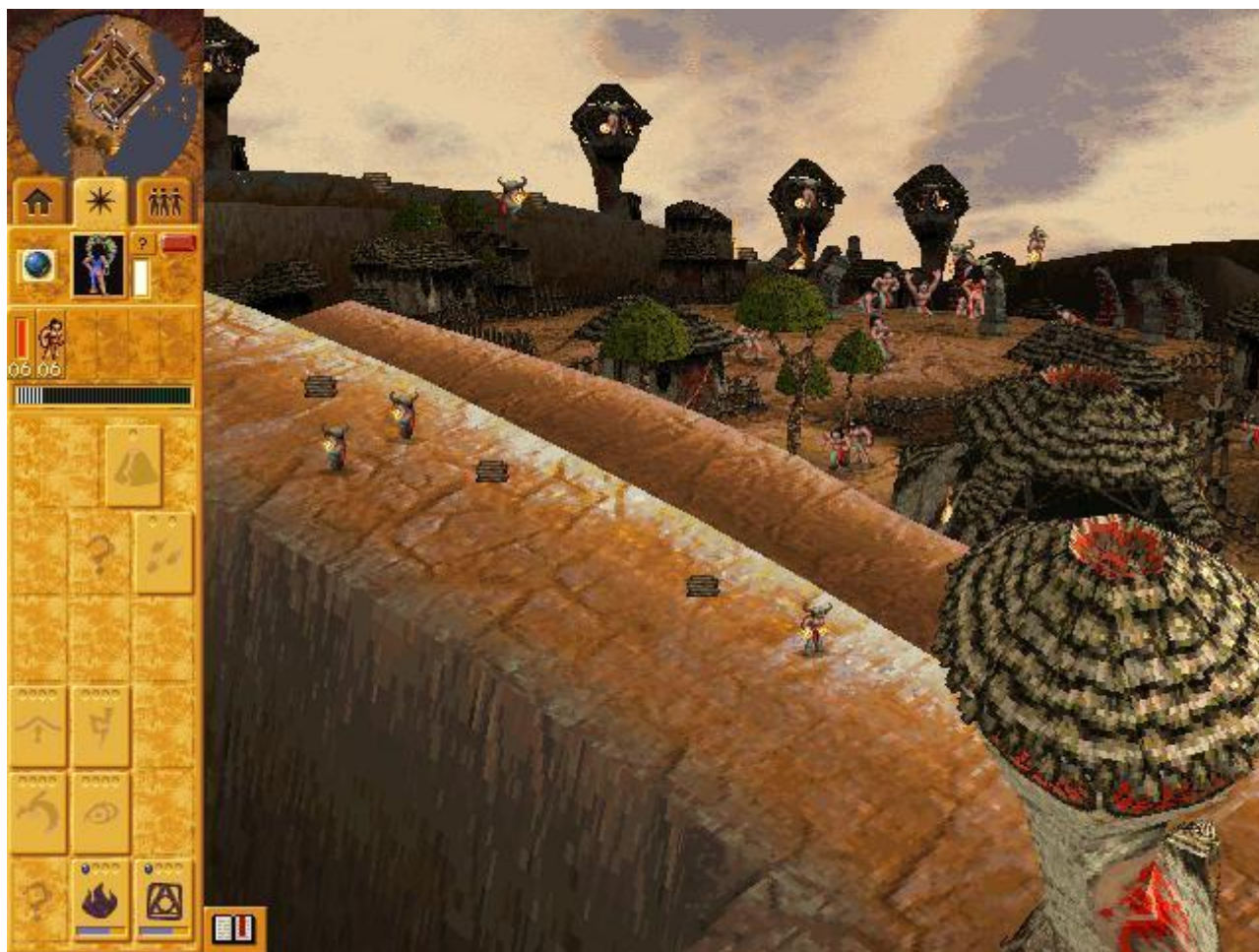
Niveaux obligatoires et terriblement atypiques

Celui-ci me met hors de moi, et apparemment je ne suis pas le seul. Joel Johnson écrit :

Je voudrais vous parler de ces passages affreusement irritants dans les jeux où les designers se sont dit « il faut que ça change ». Je n'ai rien contre les mini-jeux, mais quand le jeu est un FPS à l'exception de deux niveaux où il faut conduire une voiture, style jeu de course, ce n'est pas amusant du tout. C'est juste un remplissage qui sert à masquer le fait qu'il n'y a pas beaucoup de contenu dans le jeu principal. Parmi ces passages, on compte aussi les missions type infiltration obligatoires, assez répandues dans les FPS (si vous voulez faire un jeu d'infiltration, faites un fichu jeu d'infiltration), les sections de *rail-shooting* dans les FPS, les sections « jeux de rythme » dans les jeux du genre *Grand Theft Auto*, etc. Les mini-jeux optionnels sont amusants et peuvent apporter un changement de rythme bienvenu, mais le mot-clef ici, c'est **optionnel**. Un niveau où, pour avancer, le joueur doit utiliser des compétences complètement différentes de celles réclamées par le reste du jeu peut être affreusement énervant.

Bullfrog s'est souvent rendu coupable de ça – je me souviens de certains niveaux terriblement atypiques dans *Dungeon Keeper*, *Magic Carpet* et *Populous : The Beginning*. Ils étoffaient le jeu, mais ils sont faits de telle

sorte que tout ce que vous aviez appris auparavant devient inutile, ce qui est très pénible. Faites en sorte de les rendre optionnels.



Ne pas fournir d'information à court terme sur les objectifs

La première fois que ma femme s'est assise pour jouer au jeu d'aventure textuel *Colossal Cave*, elle a pu lire ces mots d'introduction :

Vous êtes au bout d'une route devant un petit bâtiment de brique. Autour de vous, c'est la forêt. Un petit ruisseau coule hors du bâtiment jusqu'à un ravin.

Et puis plus rien. « Je suis censée faire quoi ? » demanda-t-elle à l'homme qui lui présentait le jeu. « Ce que vous voulez ! » lui répondit-il fièrement (c'était en 1979, et les jeux avec des parseurs étaient tout nouveaux). Mais elle ne savait pas ce qu'elle voulait faire. Le jeu ne lui donnait aucun but particulier à accomplir, et nous avons vécu ainsi avec le même TDC depuis près de 30 ans – et ça arrive encore, croyez-le ou non. Andrew Harrison écrit pour nous dire :

Quand je jouais à *Metal Arms : Glitch in the system* (sur PS2), il m'arrivait parfois de vouloir charger une partie à partir d'un checkpoint qui ne donnait aucune indication claire sur ce que j'étais censé faire : pas d'information dans le menu pause, personne à qui parler, pas moyen de revenir à une cinématique explicative antérieure, pas même un écho sur mon radar. Souvent, je ne faisais que me promener jusqu'à ce que je trouve des ennemis pour ensuite les tuer, en espérant que mon objectif était de les éliminer. Si le *vrai* but était de détruire des pièces d'une machine ou d'appuyer sur un bouton, je pouvais potentiellement errer pendant longtemps avant de mettre le doigt dessus. Je pense que les designers devraient essayer d'éviter ce genre de situation.

Et il a parfaitement raison, ils devraient ; en fait, c'est une des [règles de game design](#) énoncées par Noah Falstein : *fournir des informations à court terme sur les objectifs*. Et s'il charge une partie sauvegardée, il faut donner au joueur un récapitulatif, un résumé ou quelque chose d'autre qui lui permette de comprendre ce qu'il est censé faire.

Stratégies dominantes

L'expression « stratégie dominante » est un terme de la théorie mathématique des jeux. Elle se réfère à un état de fait dans laquelle un certain mode d'action (une stratégie) donne toujours le meilleur résultat indépendamment des circonstances. Une stratégie dominante ne garantit pas forcément la victoire, mais c'est toujours le meilleur choix possible. En conséquence, il n'y a aucune raison d'utiliser une stratégie différente. Un jeu avec une stratégie dominante est fautif, car il n'offre pas de véritables décisions au joueur.

Les stratégies dominantes ne se cantonnent pas au domaine des mathématiques et on en trouve aussi dans les jeux vidéo. Joel Johnson nous écrit :

La plupart des jeux de nos jours, qu'il s'agisse de jeux d'action, d'aventure, de stratégie ou autre, donne au joueur une grande variété d'options ou de méthodes pour attaquer des unités ennemis. Un des gros problèmes que j'ai remarqués est qu'il n'est pas rare que la majorité de ces options (mouvements spéciaux, sorts, unités, etc.) soit complètement inutile, parce qu'une seule méthode surpasse toutes les autres. Par exemple, prenons *Halo*. Sniper avec le pistolet était la seule façon valable de jouer, du moins pour moi et pour beaucoup de gens avec qui j'ai joué. Il y a peu d'intérêt selon moi à utiliser d'autres méthodes d'attaque quand je peux tuer quelqu'un à l'autre bout de l'aire de jeu rapidement et facilement. Je m'amusais beaucoup plus à sniper au pistolet que les autres, qui eux utilisaient un fusil de sniper. Il y avait une certaine ironie là-dedans. En tout cas, Bungie [NDT : les développeurs de *Halo*] a bien fait son travail et a sévèrement revu à la baisse le pistolet pour *Halo 2*. J'étais déçu au départ, mais le jeu était devenu beaucoup plus intéressant à jouer.

Ceci est un parfait exemple du problème. Choisir le pistolet est une stratégie dominante, ou presque. Parfois, les stratégies dominantes se retrouvent dans les jeux parce qu'ils n'ont pas assez été testés ; parfois parce que le concepteur était tellement fan d'une stratégie particulière qu'il ne pouvait pas se résoudre à l'affaiblir, au détriment de l'équilibre du jeu. Résultat des courses : il doit y avoir des avantages *et* des inconvénients dans toutes les stratégies possibles, les rendant plus ou moins intéressantes selon la situation.



Amnésie de début de partie

Passons de l'équilibrage de jeux à la question de la narration. Andrew Stuart nous parle des jeux qui commencent par quelque chose du genre :

« Vous vous réveillez dans un endroit étrange. Vous ne savez pas qui vous êtes ni comment vous êtes arrivé ici. Vous êtes amnésique et votre objectif est de trouver qui vous êtes et ce que vous faites ici. » C'est dur à avaler, mais on dirait que tous les jeux du genre me mettent dans la peau d'un amnésique qui se réveille. C'est normal après une nuit arrosée, mais dans chaque jeu ? Ça suffit !

Il y a quelques années, j'avais mis le doigt sur ce problème de l'amnésie au cours d'une [conférence](#) à la Game Developers Conference. Le problème se pose parce que le joueur ne sait rien à propos de l'univers du jeu quand il commence la partie. Dans beaucoup de jeux d'aventure, la première chose qu'il doit faire est de passer en revue tous les tiroirs dans ce qui est supposé être son propre appartement pour voir ce qu'il y a dedans – ce qui est proprement ridicule. Un personnage dans une vraie histoire n'a pas à faire ça, parce que le personnage appartient déjà à l'univers du jeu. Ainsi, dans l'industrie du jeu vidéo, nous faisons beaucoup de jeux dans lesquels le personnage du joueur est amnésique afin de justifier l'ignorance même du joueur.

C'est une solution bas-de-gamme au problème. Les spectateurs d'un film ne connaissent pas l'univers du film non plus, c'est pourquoi les introductions sont soigneusement élaborées afin d'accompagner rapidement le public vers le cœur du film. De temps en temps, quand la situation est vraiment complexe, les films recourent à la narration en voix off, mais ce n'est pas nécessaire la plupart du temps. Examinons l'échange de dialogue suivant au début du premier épisode de [The Sandbaggers](#), le meilleur feuilleton d'espionnage jamais créé :

Secrétaire : Wellingham a appelé. Il veut vous voir.

Burnside [avec sévérité] : Vous voulez dire le sous-secrétaire permanent du Foreign Office ?

Secrétaire [avec sévérité également] : Je veux dire votre beau-père.
Burnside : **Ex**-beau-père.

En quatre lignes, sans même l'avoir rencontré, nous avons fait la connaissance de Wellingham, son travail et sa relation avec le personnage principal de la série, Burnside. Nous avons également appris que Burnside a divorcé, mais il a encore des liens professionnels avec son ancien beau-père. Finalement, nous remarquons que Burnside est un peu formel sur les noms (ce n'était pas rare en Grande-Bretagne en 1978) et que sa secrétaire lui tient tête. Ça fait beaucoup d'informations en dix secondes de dialogue, et c'est infiniment plus brillant que les explications interminables d'un quelconque mentor de jeu vidéo. Nous devons étudier ces introductions de films ou de séries afin d'apprendre comment les faire nous aussi. Et tant qu'on y est, plus de personnage-joueur amnésique.

Conditions de victoire mal vérifiées

Interstate '76 était un jeu de voitures proposant d'utiliser un large panel d'armes complètement loufoques. L'un des niveaux avait un problème assez cocasse, mais ennuyeux. Le jeu vous indiquait de trouver le moyen de sortir d'une zone fermée entourée par un mur en béton. La « bonne » solution était de trouver une rampe cachée, de grimper dessus et de voler par-dessus le mur – ce qui vous faisait atterrir dans une fosse, mais c'était essentiel pour la suite de l'histoire. Cependant, certains joueurs habiles ont réalisé qu'il pouvaient poser une mine par terre près du mur, puis foncer dessus à toute vitesse. L'explosion propulsait la voiture en l'air et, avec l'impulsion, la faisait voler par-dessus le mur. Si la voiture était assez solide, elle se fracassait au sol mais n'explosait pas. Ces joueurs avaient rempli la condition de victoire, mais le jeu ne le reconnaissait pas, et le niveau ne se finissait jamais. Le jeu testait seulement l'utilisation de la rampe et pas la présence de la voiture à l'extérieur du mur.

Quand vous dites à un joueur de faire quelque chose et que vous vérifiez qu'il l'a fait, vous devez vérifier qu'il a fait ce que vous lui avez *demandé* de faire, et non pas ce que vous avez *voulu* qu'il fasse. Dans les jeux récents avec des environnements complexes (par exemple les *Grand Theft Auto*), il y a de fortes chances que le joueur réussisse à remplir la mission d'une façon que le game designer n'aurait pas imaginée — et le joueur doit être récompensé pour ça.



Toujours sur le même thème, venons-en aux...

Vérifications illogiques des conditions de victoire

Éviter les conditions de victoires incorrects ne signifie pas que vous devez chercher la petite bête sur des détails précis. Si le joueur a effectué une action qui comporte naturellement une condition de victoire, il doit également être récompensé en conséquence. Andy Lundell dit :

C'est déjà pénible quand les objectifs de missions sont illogiques, mais quand vous commencez à pénaliser le joueur pour avoir pris une décision logique, là, ça va trop loin. On voit souvent ça dans les FPS et parfois dans les RTS en solo. Le meilleur exemple en est *Red Faction*. Il y avait une mission qui vous disait que vous deviez détruire un ordinateur spécifique dans une station spatiale. Une fois que vous y êtes arrivé, on vous disait que vous deviez faire sauter la station spatiale **tout entière** et vous échapper dans une capsule de sauvetage. J'ai donc, en toute logique, supposé que je pouvais juste faire sauter la station spatiale sans me soucier d'aller détruire l'ordinateur spécifique. J'ai fait sauter la station, plongé dans une capsule de sauvetage et... et... le jeu a planté. Nous étions supposé faire sauter l'ordinateur et **ensuite** faire sauter la station (il n'y avait pas d'explication sur la nécessité de faire cet effort). Apparemment, le jeu ne pouvait gérer le fait que le niveau soit terminé sans que l'ordinateur ait spécifiquement été détruit, je ne pouvais donc plus que revenir au menu principal. Tout ça parce que j'ai tenté de faire quelque chose d'intelligent au lieu de faire bêtement ce que les level designers voulait me faire faire !

Un petit conseil, chers level designers : si une condition de victoire (faire sauter la station) inclut naturellement une autre condition (faire sauter l'ordinateur), il n'y a aucunement besoin de vérifier la seconde – et ce genre de fautes risquerait de vous faire priver de biscuit.

Perdre le contrôle de la caméra au mauvais moment

Depuis l'arrivée de la 3D, nous avons dû travailler beaucoup plus pour modéliser des mondes, en particulier dans les jeux d'action. Avec les scrollings 2D (horizontal ou vertical) et les vues isométrique, la vie était bien plus simple. Les perspectives 3D fixes à la troisième ou à la première personne ne sont pas complexes non plus, mais toutes deux ont leurs limites (que se passe-t-il dans les jeux à la troisième personne si l'avatar du joueur est dos au mur ?). De nos jours, nous avons fait beaucoup d'efforts pour créer des caméras intelligentes à la *Ico* et nous n'avons pas toujours bien fait les choses. Loren Schmidt nous écrit :

Vous jouez à un jeu de plateforme à la troisième personne. Vous courez le long d'un couloir et devant vous, une grande fosse remplie de pics que vous pouvez tout juste passer en un seul saut... et soudain, la caméra se retourne à 180 degrés : évidemment, vous ratez votre timing et votre personnage impuissant plonge vers une mort inéluctable.

C'est encore pire lorsque dans le même temps, on passe d'une caméra contrôlable à une caméra fixe, comme on peut le voir dans les deux derniers *Prince of Persia*. La plupart de ces jeux se jouent avec une caméra que le joueur peut contrôler, mais de temps en temps, la caméra saute tout à coup vers une position prédéfinie (parfois mal placée). C'est tout particulièrement fatal lors des séquences de combat ou de saut au-dessus de précipices mortels.

Je comprends le but visé ici – juste avant une séquence d'action, le game designer a souvent besoin de verrouiller la caméra de manière à garantir au joueur une vision claire de ce qui se passe et à bien définir la façon dont les commandes vont faire bouger le personnage à l'écran. Mais changer brusquement de point de vue *pendant* que le joueur saute ou se bat pour sa vie va le gêner à coup sûr. Ne faites pas ça ! Il vaut mieux laisser le contrôle de la caméra au joueur, même si ce n'est pas l'idéal, plutôt que de le désorienter en changeant subitement de perspective sans crier gare.



C'est tout pour cette année. Curieusement, je n'ai pas reçu de grosses plaintes au sujet des menus de configuration (une source d'irritation constante). Une personne m'a tout de même écrit pour se plaindre des listes de sauvegardes non-triées ou ordonnées de manière si peu commode qu'il faut fouiller pour trouver sa sauvegarde la plus récente ; même si je reconnais que c'est une nuisance, je me dis que ce n'est pas assez grave pour justifier un refus de biscuits.

Comme toujours, je veux entendre vos réclamations ! Jetez un œil à la [No Twinkie Database](#) pour voir si j'en ai déjà parlé, et si ce n'est pas le cas, envoyez-moi un mail à notwinkie@designersnotebook.com (NDT : en anglais) et faites-le moi savoir !

Craft, alchimie et autres créations d'objets

Par [Lothindil](#) | le 12 février 2013 02:14:53 | Catégories : [Game Design](#), [Level Design](#)

Cet article est une re-publication.

L'[article original](#) est écrit par Lothindil publié par Jeux de Lothi (décembre 2012).

En tant que grande joueuse et conceptrice de jeux, j'essaye régulièrement d'analyser en profondeur les mécanismes de certains jeux. En ce moment, je joue surtout à deux jeux : *Dragon Age : Origins* et *Guild Wars*. Ces deux RPG, bien que très différents dans leur concept et leurs conceptions, possèdent cependant un mécanisme commun, que j'ai retrouvé dans de nombreuses autres productions, y compris amateurs : le *crafting* ou alchimie.



Le terme anglais (*crafting*) a l'avantage d'être plus complet, car il parle de toutes les formes *roleplay* de transformation ou de création d'objets. Il n'y a pas d'équivalent direct en français : on pourrait parler de « forge », « herboristerie » ou encore « alchimie ». Mais ces termes ajoutent une nuance plus précise qui n'existe pas dans le terme « crafting ». N'étant pas fan de l'anglais dans mes jeux, j'ai décidé d'adopter le terme « alchimie », à défaut de mieux. Il apporte un côté fantastique.

Quelques définitions.

L'alchimie, ou mélange d'ingrédients, se retrouve dans pas mal de jeux vidéos et de MMORPG. Citons en vrac *Guild Wars* (1 et 2), *Aion*, *World of Warcraft*, *Oblivion*, *Morrowind* et autres *Minecraft* (pour ne citer que quelques-uns des plus connus).

Le principe est simple : prendre un ou plusieurs objets (ressources) pour en faire un ou plusieurs autres objets. Les noms varient beaucoup, dépendant partiellement des contraintes : forge, métiers, professions (ce qui entraîne des confusions avec les classes; *profession* en anglais), etc.

Maintenant, quand on est game designer et qu'on travaille sur l'alchimie, un certain nombre de questions se pose. On retrouve d'ailleurs ces choix à travers les différents jeux que j'ai cités plus haut.

Dans la suite, pour simplifier, nous nommerons « ingrédients » les produits de base nécessaires à la réalisation d'une formule et « produits » les produits issues de l'alchimie. Notons qu'un produit peut être un ingrédient s'il est à son tour utilisé dans une recette. Nous parlerons alors de « ressources secondaires ».

« Quoi ? » : La valeur ajoutée.

C'est sans doute la question essentielle à se poser AVANT toutes les autres questions, car les choix que vous ferez plus tard dépendent partiellement de la réponse que vous apportez à cette question.

Le « quoi ? » pose la question de l'intérêt qu'aura le joueur à pratiquer l'alchimie. Quelle sera la valeur ajoutée du produit final par rapport aux produits de base ? L'équation est cependant plus complexe que « cout en pièces d'or du produit > cout en pièces d'or des ingrédients ». Il faut en effet mettre dans la balance toutes les contraintes nécessaires pour faire l'alchimie (recettes, rareté des ingrédients, possibilités de pratiquer, possibilité d'échec, ingrédients qui sont des ressources secondaires, etc.) ainsi que la rareté du produit fini (un produit fini impossible à trouver autrement dans le jeu aura une valeur ajoutée bien plus importante qu'un objet qu'on peut acheter dans une boutique en ayant dix niveaux de plus).

Il est nécessaire donc de répondre aux questions suivantes avant de passer à la suite :

- Quels types d'objets seront réalisables avec l'alchimie (consommables, équipements, ressources à haute valeur ajoutée, etc.) ?
- Les produits de l'alchimie sont-ils des produits uniques/rares/achetables avec de l'argent réel ?

Bref, **quelle est la valeur du produit ?**

Plus la valeur d'un produit est grande, plus les contraintes pour le fabriquer peuvent être fortes. Notons que dans un jeu multijoueur, les contraintes rajoutent encore de la valeur ajoutée à un produit accessible uniquement via l'alchimie. En effet, plus ces contraintes sont fortes, plus il sera rare et donc plus sa valeur sera forte.

Prenons deux exemples fort différents : dans *Aion*, les métiers permettent à haut niveau de créer des objets à très forte valeur ajoutée (équipements rares supérieurs à ce qu'on trouve normalement dans le jeu), mais les contraintes sont très nombreuses et très lourdes.

A l'inverse, dans *Oblivion*, l'alchimie permet de créer uniquement des potions à valeur ajoutée réduite (consommables dont on trouve des exemplaires en magasin ou sur des monstres), mais les contraintes sont assez rares et assez faibles.

Les différentes contraintes.

Où ?

Où le joueur va-t-il pouvoir faire l'alchimie ? Il existe globalement deux réponses : « quelque part » ou « n'importe où ».

Dans le premier cas, le PJ va devoir trouver l'endroit où il peut faire son alchimie. Généralement, cela se fait dans des ateliers dans les villes principales. C'est le cas de jeux comme *Aion* ou *Guild Wars 2*.



Dans le second cas, le joueur va pouvoir faire ça au cours de ses séquences d'exploration. C'est le cas dans *Oblivion*.

Notons que la contrainte de lieu devient d'autant plus importante que le sac est limité en taille et que les explorations du PJ le conduisent loin de la ville. Ainsi, forcer un joueur à retourner en ville pour faire de l'alchimie tout en limitant son sac à 40 objets (identiques ou différents) a toutes les chances de le dégouter et de le pousser à l'abandon très rapidement. A l'inverse, si l'exploration se passe toujours dans un rayon court autour de la ville (ou mieux s'il y a moyen de revenir en ville quand on veut très facilement) revient quasiment à permettre au joueur de pratiquer n'importe où.

L'origine des ingrédients.

Globalement, la contrainte de l'origine des ingrédients se base sur plusieurs paramètres :

- **Leur accessibilité** : le problème ici est de savoir où l'on peut obtenir l'objet. Si c'est « partout », ça sera forcément moins contraignant que « dans la dernière caverne du donjon de Raslabeurk qui est à une semaine de marche du dernier village ».
- **Leur valeur monétaire** : ceci a été évoqué plus haut. Forcément, plus un objet vaut cher, plus il sera considéré comme « contraignant ».
- **La quantité disponible** : un objet disponible en illimité chez un PNJ (fût-il au bout du monde) sera forcément moins contraignant que celui qui n'existe qu'en dix exemplaires dans tout le jeu (fût-il dans le sac à dos dès le départ).
- **La facilité d'obtention** : l'herbe que l'on peut arracher à mains nues avec un taux de réussite de 100 % sera forcément moins contraignante à obtenir que l'herbe magique qu'on doit récolter à la serpe d'or avec 2 % de taux de réussite. Dans le même genre, la fourrure miteuse qu'on a 75 % de chances d'attraper sur un lapin de niveau 1 sera nettement moins contraignante que l'écaille de dragon qu'on ramasse une fois sur cinq sur le cadavre d'un dragon de niveau 150.

Le matériel.

Il est ici question d'un matériel fixe, par exemple une pièce d'équipement ou un objet qui doit rester dans l'inventaire pour pratiquer l'alchimie. La question de ce matériel va souvent de pair avec la question « où » posée plus haut. Il est en effet rare qu'on impose à la fois un endroit où aller et un équipement à porter. Dans *Oblivion* par exemple, on impose pour faire de l'alchimie d'avoir au minimum un pilon.

Les contraintes porteront pour l'essentiel sur :

- **La difficulté d'obtenir** l'objet en question : plus l'objet sera difficile à obtenir, que ça soit pour des raisons financières ou parce qu'il est caché dans un coin, plus la contrainte sera forte.
- **L'usure du matériel** : un matériel qui s'use sera plus contraignant qu'un inusable et d'autant plus contraignant que son usure est rapide. Cette contrainte peut au besoin être baissée en autorisant le joueur à avoir plusieurs exemplaires de l'objet ou à le réparer.
- **L'encombrement du matériel** : dans un jeu où le moindre emplacement dans l'inventaire compte (je pense à un sac fait comme dans *Diablo*), transporter le matériel sera en soi une contrainte importante. Plus la place disponible est grande et moins l'objet est encombrant, plus la contrainte est faible. Dans *Oblivion*, l'objet de base prend très peu de place.
- **L'adaptation du matériel au produit** : nous y reviendrons dans la partie suivante.

Différents « types » d'alchimie : les métiers.

Voilà une contrainte plus subtile qui, mal utilisée, peut s'avérer totalement inutile ou, au contraire, trop efficace au point de rendre l'alchimie inutilisable.

L'idée est de diviser l'alchimie en différentes activités, en fonction des produits généralement. La plupart des jeux qui l'appliquent utilisent un principe de métiers en distinguant la forge, la couture, la cuisine, etc.

L'erreur de base, qui rend la contrainte complètement inutile, est de permettre aux joueurs d'exercer tous les métiers. En effet, s'ils peuvent tout faire, pourquoi ne pas leur permettre directement d'avoir accès à tout, sous un seul nom ? Ou utiliser ces distinctions pour améliorer l'ergonomie sans influencer le gameplay ?

L'erreur inverse, qui peut rendre l'alchimie inutilisable, est de restreindre drastiquement l'utilisation d'un métier à certains joueurs. Pour peu que le jeu compte peu d'actifs (chose qui sur la toile arrive bien plus souvent qu'on ne le croit), il peut très bien arriver que certains métiers soient inexistantes. Ce qui rend leurs produits très rares... Imaginez maintenant que le métier en question produise entre autres des ressources secondaires pour d'autres métiers. Même avec de nombreux actifs, si le métier possède des limitations trop fortes (typiquement les métiers payables en argent réel ou demandant d'avoir un très haut niveau), le problème peut se produire. J'ai croisé cette erreur dans de nombreux jeux amateurs et semi-amateurs qui comptaient sur des milliers d'actifs...

Cette contrainte est souvent renforcée par deux autres : la présence d'un matériel propre au métier (dans *Allods* par exemple, il est nécessaire d'acheter une ceinture correspondant à son métier) et le niveau (sur lequel nous allons enchaîner).

Les niveaux de métiers.

Le principe est simple : comme pour les niveaux du personnage, à chaque fois que vous réussissez une action, vous gagnez de l'XP. Au bout d'un certain seuil d'expérience, vous gagnez un niveau. Il existe d'autres systèmes sans niveau, où les points servent à être dépensés pour acheter des recettes. Les contraintes sont similaires aux deux systèmes.

C'est évident : plus il faut d'expérience, plus la contrainte est forte.

C'est au game designer pour l'essentiel de bien calibrer la difficulté. Pour gagner de l'expérience, il faut des ingrédients ; pour réunir des ingrédients, généralement, il faut farmer. Et tout le monde n'aime pas forcément farmer durant des heures et des heures, sans autre but que de monter son niveau de métier. Il est donc essentiel de bien doser la difficulté de gagner des niveaux (de manière à ce que les objets haut niveau restent proportionnellement rares), la quantité de production nécessaire pour atteindre ce niveau (si un PJ doit produire 50 fois la même armure pour atteindre le niveau nécessaire à l'armure suivante, il sera tenté de vendre — et donc de mettre sur le marché — les 49 armures inutiles, rendant l'objet nettement moins rare et attrayant) et la puissance de l'objet acquis au niveau supérieur.

Il existe plusieurs façons de gérer cette montée de niveau. Certains poussent au coopératif (le joueur qui va monter son métier ne pourra pour ainsi dire pas le faire seul et donnera l'équipement excédentaire à ceux l'ayant aidé). Cette solution est attrayante à première vue, mais méfiez-vous, la réalité est souvent autre et les « récolteurs » sont souvent lésés et finissent dégoutés par les métiers. De surcroît, le problème du nombre d'actifs se pose à nouveau. S'il faut 50 armures pour grimper de niveau, et qu'on veut éviter l'arrivée des armures sur le marché, c'est donc qu'il faut 49 récolteurs derrière 1 créateur. A moins d'avoir un jeu à plusieurs milliers de joueurs, c'est pratiquement intenable, que ce soit pour le fabricant qui va galérer à trouver ses ressources ou pour les récolteurs qui parfois devront attendre les 48 précédentes armures pour avoir la leur...

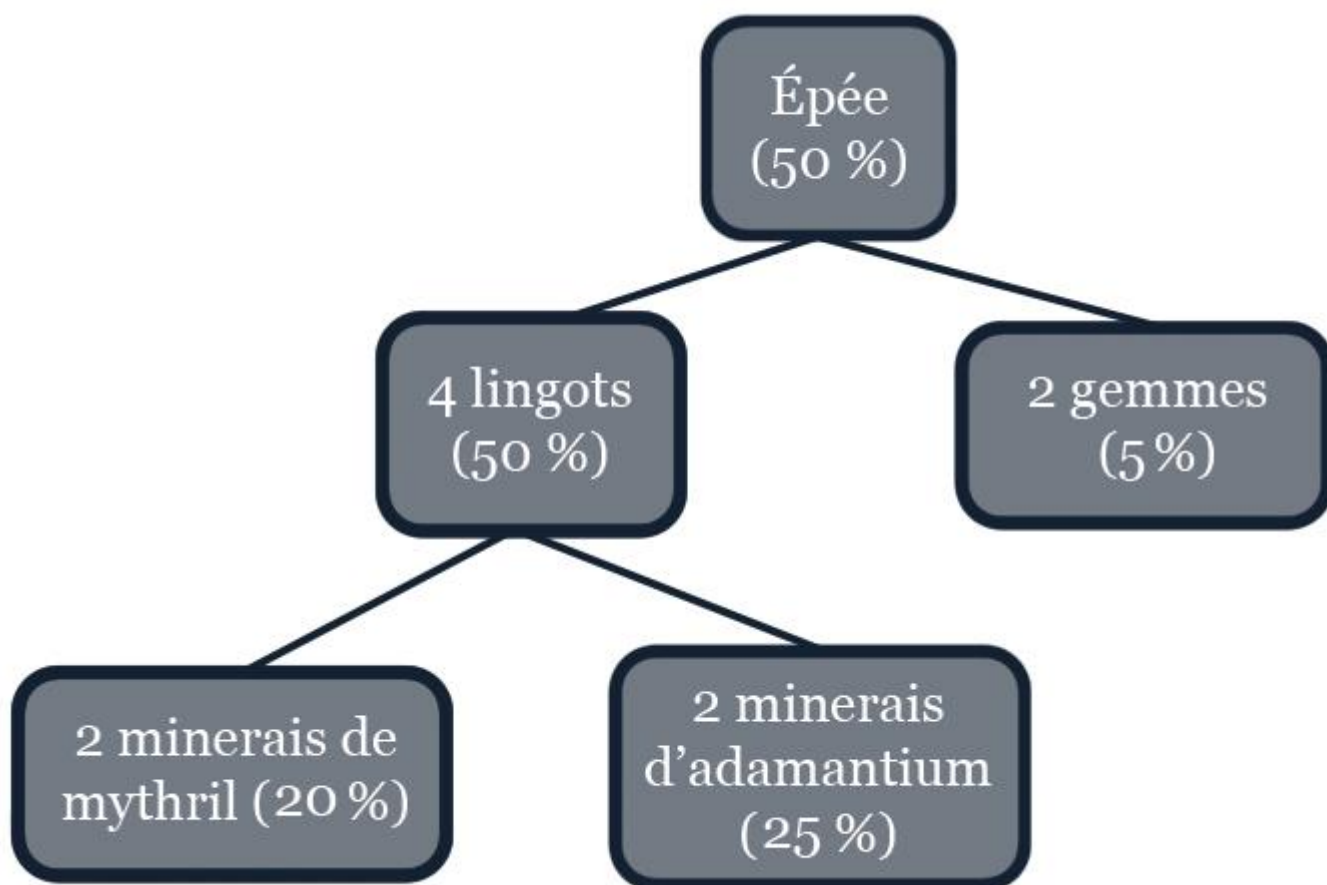
D'autres jeux permettent de monter de niveau sans produire des objets qui iront sur le marché. C'est le cas d'Aion par exemple, qui propose un système de commande. Le PNJ demande un certain nombre d'objets, fournit une partie des ressources et rémunère le joueur en expérience de métier.

Le taux de réussite.

Dans certains jeux, nous trouvons un taux de réussite des produits, taux qui peut dépendre ou non du niveau du métiers. Nous ne nous étendrons pas sur cette contrainte car elle est vraiment simple à comprendre : plus le taux de réussite est fort, plus la contrainte est faible. Un taux de 100% rend cette contrainte nulle.

Sachez juste que cette contrainte est démultipliée pour peu que la majorité des recettes produisant des objets à hautes valeurs ajoutées demandent des ressources secondaires. En effet, les taux d'échecs vont s'additionner et ainsi augmenter sensiblement la consommation de ressources et donc faire grimper la contrainte.

Un petit calcul simple pour vous donner une idée du chiffrage. On raisonne ici en moyenne, et n'oublions pas qu'avec un peu de malchance, certains joueurs vont galérer beaucoup plus que ça.



Soit une recette permettant de fabriquer « l'épée de la mort qui tue » (taux de réussite à 50%), qui demande 4 lingots d'alliage mythril-adamantium et deux gemmes rouges. L'alliage de mythril-adamantium est issue d'une recette (taux de réussite à 50%) réclamant 2 minerais de mythril (taux de réussite à la récolte à 20%) et 1 minerai d'adamantium (taux de réussite à 25%). La gemme rouge est issue d'un drop à 5% sur un monstre.

Pour 1 gemme, il faudra en moyenne tuer 20 monstres.

Pour 1 minerai de mythril, il faudra en moyenne récolter 5 fois. Pour 1 minerai d'adamantium, il faudra en moyenne récolter 4 fois. Pour 1 lingot d'alliage, il faut en moyenne 4 minerais de mythril et 2 minerais d'adamantium (on rate une fois sur deux), soit 20 récoltes de mythril et 8 récoltes d'adamantium.

Pour l'épée enfin, on rate 1 fois sur 2, il nous faut donc 8 lingots d'alliage et 4 gemmes, soit 160 récoltes de mythril, 64 récoltes d'adamantium et 80 monstres.

Vous voyez à quel point tous les taux de réussite (ou plutôt tous les taux d'échecs) se cumulent très rapidement. Sur des jeux où les actions sont limitées par jour, la contrainte se fait encore plus sentir (idem sur les jeux où il faut par exemple attendre une heure pour forger ledit « équipement de la mort qui tue »).

La quantité d'ingrédients.

C'est une des contraintes les plus importantes et, malheureusement, l'une des plus souvent négligées.

En elle-même, la contrainte est simple : plus il faut d'ingrédients (que ce soit en diversité ou en quantité), plus la contrainte est forte.

Cependant, elle est rarement bien évaluée. Plus de la moitié des jeux auxquels j'ai joué avec un système d'alchimie proposaient un système décourageant et démotivant, simplement à cause de la quantité.

L'exemple-type est celui du jeu où pour faire une potion, on vous demande 10 champignons, 5 eaux et 13 fioles vides. A partir du moment où le PJ risque de se prendre des dégâts sur le temps de la récolte des 10 champignons et des 5 eaux, les potions devront rapporter plus de PV que la quantité de dégâts maximale lors de la récolte. C'est idiot, mais c'est rare, très rare. À tel point que les level designers créent des quêtes nécessitant les produits de la recette pour justifier l'existence même de cette recette (c'est une erreur classique que j'ai moi-même faite).

Dans le même genre, il y a les armures qui demandent 500 ressources de 4 types (pour un total de 5000 à 6000 monstres). « Oui, mais il suffit de coopérer ». Ne rêvez pas, mesdames et messieurs les game designers. A part quelques guildes super-organisées, ça ne se passe jamais comme ça et ça dégoute généralement le créateur des objets — qui passe finalement tout son temps à farmer pour tenter de monter son métier pour obtenir l'armure du niveau haut dessus — et les récolteurs (qui n'obtiennent que rarement leur armure).

Dans les jeux à un seul joueur, c'est encore pire, vu que la coopération n'existe pas. Monter son métier va prendre du temps, beaucoup de temps, souvent trop de temps pour être intéressant. À l'inverse, il y a les quantités trop faibles cumulées à des ingrédients très faciles d'accès et très disponibles, qui créent une contrainte nulle.

Les recettes.

C'est la dernière grande contrainte : la façon dont on crée le produit.

Il existe globalement trois grandes écoles :

- Les ingrédients sont associés en fonction d'une caractéristique commune (dans *Oblivion*, les ingrédients doivent être liés par un même effet).
- Les ingrédients doivent entrer dans une recette (connue ou non), leur ordre est généralement sans importance.
- Les ingrédients doivent entrer dans une matrice, leur ordre a une importance et détermine la nature de l'objet (c'est le système de *Minecraft*).

Bien sûr, chaque école a ses propres contraintes.

Dans le premier système, la contrainte dépendra essentiellement de la rareté de l'effet associé à l'ingrédient. Plus l'ingrédient peut être associé à d'autres ingrédients, moins la contrainte est forte.

Dans le deuxième système, tout dépend de la façon d'obtenir une recette. Si elles sont données, la contrainte est réduite ; s'il faut les obtenir au bout de longues quêtes, la contrainte sera plus forte ; s'il faut la deviner, la contrainte peut devenir bloquante si rien n'est fait pour aider le joueur (indiquer le nombre de recettes faisables, le nombre d'ingrédients, etc.).

À titre d'exemple, le système d'alchimie de *Heroes' Chronicles* tenait de l'impossible quand j'y ai joué (il y a quelques années déjà, le système a peut-être changé depuis). Trente plantes, cinq emplacements maximum (il existait des recettes à 2, 3, 4 et 5 ingrédients), possibilité de devoir utiliser plusieurs fois la même plante, aucune aide. Les ingrédients étaient perdus en cas d'échec. Pour vous donner une idée, ça fait quand même quelques 25 137 900 ($30^2 + 30^3 + 30^4 + 30^5$) combinaisons possibles. Là-dessus, il devait y avoir 1000 ou 2000 combinaisons gagnantes. Autant dire que le système pouvait éventuellement intéresser les hardcore gamers, mais dégoutait très rapidement tous les autres qui se hâtaient de fouiller le net à la recherche de la solution...

Le troisième système est très proche du deuxième. Il s'agit en réalité d'un système dérivé où la recette a l'avantage d'être visuelle, ce qui constitue une aide en soi, mais complexifie le problème en augmentant la quantité d'ingrédients possibles.

Conclusion.

Encore une fois en matière de game design et de level design, la création d'objets n'est jamais qu'une question d'équilibre.

Cet équilibre joue en réalité sur deux variables :

- **Le produit** : où l'équilibre se fera entre l'« inutile » (objet — trop — courant, accessible plus facilement par d'autres moyens, etc.) et le « cheaté » (objet nettement plus puissant que tout ce qu'on peut obtenir par d'autres moyens, objet surpassant les objets achetés avec de l'argent réel, etc.).
- **Les contraintes de production** : où l'équilibre se fera entre le « trop simple/trop rapide » (contraintes quasiment nulles) et le « trop complexe/trop aléatoire/trop long » (contraintes trop fortes).

Et enfin, l'équilibre d'interaction entre ces deux variables, la comparaison entre le produit et les contraintes qui ont permis de le réaliser (toujours l'histoire de la potion de soin coûtant plus de PV qu'elle n'en rapporte).

Combien de fins pour mon jeu ?

Par [Pierre Guyot](#) | le 2 avril 2013 22:46:59 | Catégories : [Ernest Adams](#), [Game Design](#), [Scénario](#)

Cet article est une traduction de [Pierre Guyot](#).

L'[article original](#) est écrit par Ernest Adams publié par Gamasutra (décembre 2004).

L'auteur original n'est en aucun cas responsable de la précision de cette traduction.

« Ce sont nos choix, Harry, bien plus que nos capacités, qui déterminent qui nous sommes vraiment. »

-- Albus Dumbledore, *Harry Potter et la chambre des secrets*.



Il y a quelques temps, j'ai été engagé par un studio pour travailler comme consultant sur un jeu de tir qui proposait plusieurs fins. A la lecture des documents de game design, je me suis rendu compte que celles-ci étaient toutes aussi sombres les unes que les autres. J'ai dit aux designers que les jeux de tir grand public ne sont pas des romans de Kafka et que les joueurs qui parviendraient à la fin d'un tel jeu estimeraient avoir droit à au moins une fin modérément positive. Et ils en ont tenu compte.

D'un point de vue artistique, mon argument n'était pas valable : les designers devraient pouvoir créer l'équivalent vidéoludique des romans de Kafka, s'ils le souhaitent. Mais d'un point de vue commercial, je pense que c'est une décision saine. Ce studio m'a demandé mon avis sur un jeu commercial et je pense que le public n'est pas encore prêt à dépenser 50 dollars [NDT : !!!] pour un jeu de tir triple A qui ne peut que mal finir. Gagner et perdre a beau être une question de définition, une mauvaise fin donne toujours la sensation de perdre et un *happy ending* donne toujours un sentiment de victoire. Au fil des années, nous avons entraîné les joueurs à penser qu'ils finiront toujours par gagner à la fin d'un long jeu, même s'ils doivent n'y arriver qu'après de multiples tentatives. Plus le jeu est long, plus le joueur s'est investi dedans et moins il acceptera une mauvaise fin, comme je l'ai expliqué à mes clients. Les jeux d'aventure ou d'énigmes peuvent s'en tirer, mais si un *survival* se solde par la mort du joueur, celui-ci risque de ne pas apprécier.

Néanmoins, cette situation m'a fait réfléchir sur le problème des fins multiples. C'est une question essentielle pour quelqu'un qui conçoit un jeu vidéo : le jeu doit-il proposer une ou plusieurs fins ? Comment devraient-elles se dérouler et comment en décider ?

Fins et types de jeux

Si vous faites une partie de billard avec quelqu'un dans un bar, deux issues sont possibles : vous gagnez ou vous perdez. Si vous vous asseyez pour faire un poker, quand vous quittez la table, soit vous avez plus d'argent qu'au début de la partie, soit moins : ce score final vous permet de définir simplement victoire et défaite. Sans plus d'analyse, on pourrait conclure que les jeux vidéo doivent proposer ou bien une façon de gagner et une façon de perdre, ou bien un score final. Néanmoins, beaucoup de jeux vidéo – et même pour la majorité d'entre eux – ne sont pas semblables au billard ou au poker. Comme il y a une immense variété de jeux possibles, il est difficile de définir des règles de conception qui s'appliquent à tous. Je vais essayer de traiter le problème en séparant les jeux en différents types et pour certains d'entre eux, la réponse sera assez immédiate.

On peut déjà répondre simplement à la question des jeux en équipes et des jeux compétitifs multi-joueurs, qu'il s'agisse d'une simulation de billard ou de *Counter-Strike*. Les jeux compétitifs doivent évidemment proposer les fins standards mentionnées plus haut : c'est ce que les joueurs attendent lorsqu'ils jouent l'un contre l'autre. Pour les jeux multijoueurs en coopération, où les joueurs jouent ensemble contre l'ordinateur, c'est un peu moins clair : dans certains cas, ils gagnent tous ensemble, dans d'autres, ils perdent tous ensemble. Parfois, un joueur abandonne les autres en cours de partie. En fin de compte, trois types de fins sont possibles : aucun joueur ne gagne, certains joueurs gagnent, tous les joueurs gagnent (souvent, les deux dernières options sont confondues).

On peut aussi répondre simplement pour la plupart des jeux online massivement multijoueurs, en particulier pour les univers persistants et les MMORPG. En général, ils ne proposent pas de fin du tout. L'intérêt de ce genre de jeux est précisément que les joueurs peuvent continuer à jouer indéfiniment. Les joueurs essayent de ne pas « perdre » – à savoir mourir – mais la mort n'est en général pas permanente et dans tous les cas, le jeu continue pour les autres joueurs. Il n'y a pas de « victoire » à proprement parler dans un monde persistant, à moins qu'il n'y ait une fin et une condition de victoire proprement définies, ce dont je n'ai jamais entendu parler [NDT : *Curiosity* ?]. Dans *SimCity* et dans la plupart des *citybuilders*, vous ne pouvez ni perdre ni gagner, le jeu n'ayant donc qu'une fin – et comme dans le cas des MMORPG, la « victoire » ne constitue pas le but du jeu. Les jeux d'arcade classiques comme *Space Invaders* ne proposent pas non plus de condition de victoire : le jeu se contente de devenir de plus en plus difficile jusqu'à ce que vous perdiez. La seule façon de « gagner » est de mourir avec un score plus élevé que quiconque a jamais joué. Mais qu'on définisse ceci comme une victoire ou une défaite, le jeu ne propose en fin de compte qu'une seule fin.

Les jeux courts à un seul joueur (disons d'une durée allant de quelques minutes à trois ou quatre heures) sont aussi relativement simples de ce point de vue. La plupart ne sont pas très complexes et rentrent dans un schéma victoire/défaite comme le solitaire ou bien donnent un score final qui permet au joueur de décider s'il a gagné ou perdu.

Les jeux longs à un seul joueur

Il nous reste maintenant le cas des jeux longs à un seul joueur et ceux-ci sont en général plus complexes à traiter. Tout d'abord, intéressons nous à ce qui se passe quand l'avatar du joueur meurt au milieu d'un tel jeu. Il s'agit évidemment d'une « défaite », mais je crois qu'il ne s'agit pas à proprement d'une « fin ». Évidemment, cette défaite met fin à l'*expérience de jeu* mais elle ne constitue pas une *véritable* conclusion au jeu. Le joueur sait qu'il n'a pas été au bout du jeu et qu'il devrait réessayer. Il n'y a d'ailleurs pas beaucoup d'intérêt à créer beaucoup de contenu pour accompagner la mort de l'avatar du joueur, parce que le joueur risque de le voir souvent et d'en être lassé. Pour les besoins de la discussion, je vais maintenant laisser de côté ce type de fins prématurées.

Alors, si on laisse de côté le *game over*, combien de fins un jeu long pour un seul joueur devrait-il proposer ? La réponse a à voir avec l'esprit du gameplay et avec les buts que ces différentes fins devraient servir. Tout d'abord, prenons le cas de jeux longs à un seul joueur parmi les plus vieux du genre, les *side-scrollers*. Dans ces jeux, le joueur est confronté à un flot continu d'obstacles et doit lutter pour sa survie. La plupart du temps, ils ne proposent qu'une fin, conçue de sorte à donner une sensation de triomphe au joueur. C'est la récompense du joueur pour avoir réussi à se frayer un chemin jusqu'au bout sans mourir. Des fins

multiples seraient ici inutiles : elles auraient pour effet de réduire ce sentiment de victoire chez le joueur. Si l'une des fins constitue une plus grande récompense que les autres, le joueur va la voir comme la « bonne » fin – la meilleure fin – et verra les autres comme inférieurs, même si elles ne sont pas présentées comme des défaites. Ça na pas tellement d'intérêt, à moins de vouloir établir une distinction spécifique entre une « bonne » partie et une « excellente » partie, par exemple avec un niveau ou une cinématique bonus.

Les jeux longs à histoire

Considérons maintenant un jeu long, à un seul joueur, mais avec une histoire. *Starcraft*, par exemple, propose une histoire linéaire avec une seule fin. La manière de jouer ne change pas l'histoire. De la même façon que pour les *side-scrollers*, le gameplay se concentre sur la survie et la victoire, et lorsqu'il gagne, le joueur est récompensé en découvrant la suite de l'histoire. Le jeu ne donne pas la possibilité au joueur de changer l'histoire et en fait, l'histoire est essentiellement utilisée pour lier entre elles différentes missions. Il est possible de faire le jeu d'un bout à l'autre sans s'intéresser le moins du monde à l'histoire : finir les missions est une récompense en soi [NDT : pour avoir fini le jeu quand j'étais plus jeune sans comprendre un mot d'anglais, je confirme pleinement les dires de l'auteur.].

Attention : petit spoiler pour *Half-Life* dans le paragraphe suivant.

Je pense que des jeux comme *Starcraft* n'ont besoin que d'une fin et que celle-ci devrait être présentée comme une victoire (bien qu'on puisse évidemment laisser certains éléments en suspens pour une éventuelle suite). Faire suivre au joueur une histoire linéaire jusqu'au bout pour tout d'un coup lui présenter différentes fins est amusant mais en fin de compte superflu. Je pense qu'il s'agit par exemple d'une (toute petite) faiblesse de *Half-Life*. La proposition que fait G-Man à Gordon s'avère être une offre que le joueur ne peut pas refuser, alors à quoi bon ? C'est un choix bancal et une perte de temps pour le jeu. Et s'il s'était agi d'un choix légitime, il aurait alors fallu deux suites différentes, une pour chaque alternative.

Néanmoins, certains joueurs sont vraiment intéressés par l'intrigue et aiment à jouer un rôle dedans : ces joueurs voudront ressentir l'impact et les implications de leurs actions. Pour eux, leurs actions devraient impacter les événements de l'histoire et ceci passe éventuellement par un système de fins multiples. *Wing Commander* constitue un bon exemple de ce type de jeu. Il est connu pour proposer une intrigue véritablement non-linéaire. L'échec d'une mission ne se solde pas nécessairement par une fin prématurée, mais change simplement le tour que prend l'intrigue. La réussite ou l'échec des missions devient un élément narratif et en effet, *Wing Commander* propose différentes fins.

Néanmoins, je ne crois pas que ceci soit absolument nécessaire. Ça dépend de la nature de l'intrigue elle-même. Un seul soldat a très peu de chance de faire une différence significative dans une guerre. Il peut apporter une série de contributions, modestes mais cumulatives, et ses choix affecteront profondément son expérience *personnelle*, mais lui-même ne changera pas nécessairement l'issue du conflit. L'objectif du jeu est de survivre à la guerre et différents joueurs s'attaqueront au problème de façon différente et expérimenteront des histoires différentes, mais en fin de compte, la fin est la même : le joueur a réussi à survivre et la guerre s'achève (peut-être). On peut donc légitimement concilier une intrigue non-linéaire et une fin unique, si la fin se focalise sur les événements à grande échelle plutôt que sur les expériences personnelles.

Challenges et choix

Quand vous essayez de déterminer combien votre jeu devrait proposer de fins, tenez compte des différents types d'actions dont le joueur dispose pour impacter l'histoire. Si c'est la réussite ou l'échec face à des *challenges* qu'on lui propose qui permettent au joueur de changer le déroulement de l'histoire, alors celle-ci devrait se construire autour de la performance du joueur, avec des fins qui diffèrent en fonction de celle-ci – des fins bonnes ou mauvaises, ou peut-être un score final avec différentes conséquences. Les joueurs ont légitimement le droit de s'attendre à une forme de récompense lorsqu'ils réussissent un challenge, qu'il

s'agisse de *loot* comme dans les RPG ou d'une fin positive à l'histoire, et inversement. C'est le cas de *Wing Commander*, où ce sont les challenges (à savoir réussir les différentes missions) qui contrôlent principalement le déroulement de l'histoire.



L'autre grand type d'actions qui permet de changer l'intrigue, ce sont les *choix*. La différence entre challenge et choix est que le choix de requiert pas que le joueur accomplisse quelque chose (même si ledit choix peut-être difficile à faire). En conséquence, il n'y a pas de présupposé selon lequel un bon choix entrainerait une récompense ou même qu'il *existe* un bon choix. Bien sûr, il est possible de proposer des bons et des mauvais choix menant à de bonnes et à de mauvaises fins, mais attention : si c'est trop appuyé, le jeu risque de devenir manichéen voire moralisateur. Il est plus intéressant de proposer au joueur des choix moralement ambigus et d'explorer certaines de leurs conséquences.

Le *choix simple* est le plus facile à implémenter : le joueur prend une décision précise et l'intrigue suit une direction qui dépend de l'alternative choisie. Au choix suivant, on a un nouvel embranchement de l'histoire. Si les conséquences sont immédiates, ceci peut paraître mécanique : le joueur voit se dessiner l'arbre des choix qui lui sont proposés. Bien que ce soit indispensable pour les RPG [NDT : occidentaux], il s'agit d'une des faiblesses des dialogues basés sur un arbre de décision : on voit vraiment l'arbre se dessiner à mesure qu'on avance dans les dialogues. On peut atténuer cet aspect mécanique en retardant les effets d'un choix (le jeu prendre note de la décision du joueur sur le moment, mais l'histoire ne prend un tour différent que plus tard). Néanmoins, cette façon de procéder a aussi ses inconvénients. Si le joueur regrette sa décision, il doit revenir loin en arrière dans le jeu – en supposant qu'il ait encore une sauvegarde suffisamment ancienne. *Le Guide du Voyageur Intergalactique* a été maudit par nombre de joueurs, parce qu'il propose un choix apparemment insignifiant au tout début du jeu qui empêche le joueur de gagner une fois celui-ci arrivé à la toute fin. En règle générale, un choix ne devrait pas être insignifiant, ses conséquences ne devraient pas intervenir avec un trop grand décalage et le joueur devrait en connaître (ou être capable de déduire) les conséquences. Et bien sûr, un choix ne produit pas nécessairement une seule conséquence : choisir un alignement mauvais dans *Donjons et Dragons* a par exemple des conséquences tout au long du jeu.

Une autre forme de choix est le *choix cumulatif*, dans lequel les actions cumulées du joueur affectent l'intrigue, pas ses actions prises individuellement. C'est ce qu'on trouve dans les jeux qui donnent la possibilité d'être bon ou mauvais, sans vous demander spécifiquement de choisir en début de partie. *Black and White* en est un bon exemple : l'accumulation de vos décisions affecte le déroulement du jeu et l'apparence de votre créature (il ne s'agit pas exactement d'un avatar).

Ce dont on peut être sûr, c'est que si l'histoire change avec les décisions du joueur, il voudra revenir en arrière pour voir ce qui ce serait passé s'il avait pris une autre décision. C'est dans la nature humaine de s'intéresser à ce qui aurait pu se produire dans d'autre circonstances. Soyez cléments avec vos joueurs : donnez leur la possibilité de sauvegarder avant les choix importants – tout particulièrement si leur décision peut avoir des répercussions catastrophiques qu'ils n'auraient pas pu prédire.

Intrigues multiples et fins multiples

Dans un jeu avec beaucoup de choix, l'histoire peut produire un nombre d'embranchements qui croît exponentiellement et, en théorie, un nombre énorme de fins possibles. Dans la pratique néanmoins, je ne crois pas que ce soit très intéressant. Tout d'abord, il est très coûteux de créer autant de contenu, comme s'en sont rendu compte les développeurs de *Wing Commander* : pendant des années, *Wing Commander IV* a été le jeu le plus coûteux à développer de tous les temps. Mais au delà de ça, je n'ai pas l'impression que ce soit vraiment nécessaire. Un joueur intéressé par l'intrigue aura rarement envie de disposer d'une liberté totale pour explorer chaque embranchement. Un joueur intéressé par une liberté totale jouera à un *Grand Theft Auto*, laissera de côté les missions et l'intrigue et ira simplement explorer la ville et la mettre à feu et à sang.

Ce qu'un joueur intéressé par l'histoire recherche, c'est une conclusion narrative satisfaisante et crédible à l'expérience qu'il vient de vivre. Une fin qui permet de faire retomber la tension dramatique de l'histoire d'une façon cohérente. Ceci ne requiert pas nécessairement un grand nombre de fins. Souvenez-vous de vos cours de littérature [NDT : pour les Anglo-Saxons, du moins] : une histoire devrait comporter un thème, une thématique. Le thème est le cœur de l'œuvre et il devrait toucher le joueur, intellectuellement et émotionnellement. La présence d'un thème riche est caractéristique des grandes œuvres de la littérature et c'est ce qui différencie une œuvre d'une simple distraction. La thématique d'une histoire peut être ambiguë, mais elle ne devrait pas être contradictoire ou illogique. Une part essentielle de cette thématique passe par la façon dont l'histoire se conclut : heureuse, triste, dérangeante, ambiguë. Ceci reste vrai, même pour une histoire interactive. Si les fins et les alternatives offertes par l'histoire sont trop nombreuses, la thématique de l'histoire risque d'être confuse, voire perdue.



Planescape : Torment, un jeu que j'adore par ailleurs, propose si je ne me trompe pas huit fins possibles, chacune basée sur une décision que le joueur prend à la toute fin du jeu. J'ai sauvegardé et rechargé jusqu'à les avoir toutes vues (je crois). Je comprends ce que les designers ont cherché à faire en incluant toutes ces fins : le jeu est plein de choix difficiles et ils ont voulu que le jeu s'achève sur un dernier choix particulièrement difficile et que le joueur en voit les conséquences qui en découlent. Et pourtant, je n'ai pas eu l'impression que c'était vraiment *nécessaire*. L'une des fins donnait le sentiment d'être la « bonne » fin et les autres – bien que n'étant pas à proprement parler des défaites – étaient moins satisfaisantes. La thématique de l'histoire devenait plus floue à ce moment du jeu. Je pense qu'il aurait été plus fort de proposer une fin unique ou un choix qui découle du cumul des actions du joueur tout au cours du jeu, plutôt qu'un unique choix mécanique à la fin.

Conclusion

Pour faire court, ne proposez jamais de fins multiples simplement parce que vous le pouvez : c'est couteux et cela n'améliorera sans doute pas le jeu. Créez des fins multiples quand elles s'inscrivent dans un but plus large. Ce but peut être très simple : récompenser ou punir le joueur en fonction de ce qu'il a accompli, juger de la « qualité » de la partie qu'il a jouée. Victoire et défaite standards constituent des fins importantes pour beaucoup de types de jeux.

Si votre jeu inclut une histoire interactive, des fins multiples peuvent permettre de refléter les actions du joueur au cours du jeu, actions dont il attend qu'elles aient des conséquences significatives. Si l'histoire est dirigée principalement par des challenges, les différentes fins devraient refléter les échecs et les succès du joueur. Si ce sont les choix qui dirigent l'histoire, vous pouvez explorer beaucoup plus de directions, mais soyez prudent : la puissance émotionnelle d'un événement important peut être fortement diminuée si le joueur peut revenir en arrière et empêcher cet événement. *Starcraft* se termine par le sacrifice d'un personnage héroïque qui choisit de mourir pour le bien de la galaxie et le joueur ne peut rien faire pour l'empêcher. Ce moment aurait été bien moins fort si le joueur pouvait simplement recharger et choisir un dénouement plus heureux. Parfois, le joueur ne devrait *pas* avoir le choix, car cela réduit l'impact de l'histoire.

Assurez vous que chaque fin soit cohérente avec la ligne narrative qui y mène, même si celle-ci a été extrêmement variée. Faites en sorte que chaque fin réussisse à créer un sentiment de satisfaction narrative, ce « aaaahh » de contentement quand on arrive à la fin d'un bon livre ou d'un bon film. Et faites attention à ce que chacune de vos fins renforce la thématique de l'histoire, plutôt que de la brouiller ou de la distordre.

Je ne dis pas qu'il ne faut pas proposer de fins multiples. Je dis juste qu'il faut le faire avec soin et dans un but précis.