

# Créer le document conceptuel d'un jeu (« Game design document »)

## Introduction

L'étude des jeux et de leurs règles n'est pas née avec les premiers jeux vidéo. Si aujourd'hui on parle de game design essentiellement pour les jeux vidéo, le terme peut s'appliquer aussi à des jeux plus anciens. Des types de jeux qui ont clairement inspiré le jeu vidéo, comme le flipper, les jeux de rôle, de stratégie et de simulation.

Les jeux les plus anciens sont sans doute les jeux de plateaux traditionnels, dont certains trouvent leur origine dans l'antiquité et d'autres sont plus récents, comme le jeu d'échecs et le jeu de dames. A partir de règles très simples, ils travaillent avant tout l'esprit du joueur, stimulent ses capacités logiques et stratégiques. Plus généralement, les types de jeux abordés précédemment implémentent déjà deux éléments fondamentaux du game design : l'espace et les mécaniques de jeu.

- L'espace de jeu est limité par ses frontières et la liste de déplacements possibles.
- Les mécaniques de jeu sont les règles, les actions réalisables dans l'aire de jeu.

Le **concepteur de jeux** (traduction de l'expression anglaise *Game Designer*) est la personne qui conçoit un jeu, qui en fait le *game design*.

Le terme est très utilisé dans le jeu vidéo où il s'agit d'un poste de responsabilité.

Un *designer* de jeu vidéo développe la forme, le concept et la jouabilité (*gameplay* en anglais). Cela inclut de concevoir le champ du jeu, les spécifications, les propriétés des objets, les règles etc...

Le **Game Design Document** (en français : document de design ou document conceptuel de jeu vidéo, parfois on l'appelle aussi le cahier des charges), souvent abrégé GDD, est un document qui présente en détail tous les éléments qui feront partie d'un jeu donné : *Gameplay*, univers, règles, effets audio, vidéo, modélisation, programmation, storyboard, etc.

Le game design est un domaine large et flou, contrairement à la programmation ou au graphisme et le rôle d'un game designer est presque toujours différent selon le projet pour lequel il travaille. Cependant, dans le monde amateur, il en est tout autrement et le game designer cumule aussi souvent plusieurs fonctions dans ce milieu; à côté du game design, il est graphiste ou codeur et fréquemment c'est aussi le chef du projet; en effet, c'est lui qui décide de l'univers, de la faune et de la flore... Dans ce monde amateur tout est extrêmement flexible. Parfois, il n'y a pas de game designer; toute l'équipe prend des décisions ensemble. Nous allons donc plutôt traiter le game designer au sens « pro » du terme, il est toujours plus facile après d'assouplir les concepts que nous aborderons

## Structure

Un Game Design Document peut être fait de textes, images, diagrammes, concepts artistiques ou de tout autre média pour permettre d'illustrer les décisions de *design*.

Bien que considéré comme une exigence par plusieurs entreprises, le GDD n'est pas standardisé dans l'industrie vidéoludique.

Par exemple, des développeurs peuvent choisir de rédiger et de conserver le document avec un logiciel de traitement de texte, ou bien de le partager par l'intermédiaire d'un *groupware*.

En principe, un GDD contient par exemple environ une trentaine de pages et regroupe tout ou partie des éléments suivants :

- Introduction
- Un code projet
- Genre / Plateformes cibles
- Contexte ou background
- Histoire ou Scenario / Personnages
- Jouabilité / Intelligence artificielle / Interface utilisateur / contrôles du jeu
- Concepts artistiques (Images, sons, musique)

Rentrons dans le détail d'un fichier de spécifications de notre Game Design Document, en anglais on les appelle *Design Specifications* ou SPEC.

1. Introduction : C'est un aperçu général du jeu qui indique à quoi il ressemble, qui en explique l'idée et l'intérêt en commençant par l'idée originale, la suite permettant à quiconque de comprendre aisément de quoi l'on parle.
2. Un code projet : Chaque jeu doit avoir un code pour pouvoir le designer durant la phase de production. En règle générale, ce n'est pas le nom définitif du jeu pour les gros projets car ce nom final est choisi par le service marketing. Pour nous qui ne disposons pas d'un service marketing et puisque nous voulons coder un pacman, le code projet sera : Pacman
3. Le contexte ou background : Dans cette section, vous devez décrire dans ses moindres détails le monde dans lequel se situe votre jeu: villes, économie, faune, flore, religions, nourriture, différentes civilisations, histoire, géographie ou simplement plateau de jeu...  
Par exemple, pour un jeu se situant en France en 2006, pas besoin de décrire tout cela, mais il est bien essentiel de regrouper les points importants. De plus, écrivez en fonction de ce dont vous aurez besoin, il est inutile de faire un essai de 8 pages sur une civilisation perdue si cette civilisation n'est mentionnée qu'une fois pendant 30 secondes dans le jeu.

4. \* Le scénario : Détaillez l'histoire du jeu depuis la situation initiale jusqu'à son dénouement. Il n'est pas obligatoire de parler des quêtes secondaires, mais c'est toujours cela de pris. Cette partie peut s'enrichir au fil du travail sur le jeu  
  
\* Les personnages : Décrivez les personnages clés du jeu. Plus le personnage est important, plus sa personnalité doit être soignée. Il peut être pratique de faire des dessins (artworks) des personnages pour aller avec. Mentionnez bien le rôle qu'auront les personnages dans l'histoire, leurs capacités,...
5. Jouabilité ou gameplay : Ici, vous couvrirez l'interface de jeu, les touches à utiliser, mais aussi comment les combats et déplacements sur la carte sont gérés, les différentes caractéristiques du joueur, le système de dégâts, que se passe-t-il en cas de mort, les différents mouvements possibles...attention, ne mettez pas par exemple que le joueur peut ouvrir la porte bleue du 8ème monde avec un coup de tête: ce genre de détails est réservé aux level designers. Cependant, précisez bien dans les mouvements que le joueur peut donner des coups de têtes dans les objets et ennemis.
6. Les concepts artistiques (Images, sons, musique) : Cette partie pour documenter le type de graphismes qui seront employés: couleur chaudes ou froides, anatomie des personnages,... le type de sons attendus et l'ambiance générale que devront dégager les musiques employées dans le jeu.

Comme vous le voyez, pour chaque partie j'ai fait une petite liste de ce que vous pouvez indiquer, mais elle n'est pas exhaustive; n'oubliez pas que lorsque vous créez un jeu vidéo, votre seule limite est votre imagination et que vous devrez détailler dans votre documents tous les concepts originaux ou complexes. Le cahier vous servira d'encyclopédie tout au long de la production et reflétera votre potentiel de game designer.

Idéalement, un cahier de game design fait une quarantaine de pages. Cependant, il se peut très bien être plus petit pour un jeu simple, et beaucoup plus long si le jeu est très complexe.

## Pourquoi écrire un Game Design Document ?

Certains veulent se plonger tout de suite dans la modélisation ou le code du jeu et voient le travail sur papier comme une perte de temps.

Les joueurs ne le verront jamais, alors à quoi bon ?

Je connais cette façon de penser : c'est souvent la méthode que j'utilise pour créer des jeux simples avec Panoramic. Je me revois en train de commencer à coder le début de mon jeu alors que je n'avais pas encore une idée claire de certaines parties de mon jeu.

Une des objections les plus courantes à l'écriture de GDD est que personne ne les lit.

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 développeurs et autres intervenants sur le projet (artistes, testeurs, ...) 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, ces derniers peuvent constituer la base du jeu et l'équipe peut y ajouter de nouvelles fonctionnalités. Alors pourquoi écrire un GDD ? Mais parce que ce n'est pas pour ça que le prototype existe. 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. Il est toujours dangereux de transformer le code d'un prototype en code final. De plus, les prototypes ne contiennent presque jamais tous 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 afin de réaliser le jeu. Un prototype ne remplace pas les documents papier — diagrammes, cartes, listes d'objets, etc. — nécessaires à la construction de ces niveaux.

Il y a plusieurs raisons positives également que j'évoquerai ci-dessous en commençant par la moins importante :

Beaucoup de gens dans l'industrie voudraient avoir l'argent d'abord et se pencher sur les détails après — évidemment, qui ne le voudrait pas ? 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 cela ne fonctionnera certainement pas avec un vrai professionnel qui voudra avoir une trace écrite afin de pouvoir vérifier la viabilité technique et économique du projet. Pas de GDD, pas d'argent donc si vous êtes autofinancé, ce n'est pas un problème, mais il y a plusieurs autres bonnes raisons pour écrire un GDD.

La plupart des contrats de développements comportent un calendrier des différentes étapes qui définit entre autre quand le développeur devra fournir certains livrables et quand l'éditeur devra avancer plus d'argent pour le développement.

Vous ne pouvez pas créer ce calendrier sans avoir conçu au moins une partie détaillée des concepts du jeu pour savoir quelles fonctionnalités seront dans le jeu.

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 le faire également pour expliquer qu'ils diffèrent le paiement et exigent plus de travail. Si vous savez ce que vous vous engagez à fournir et à quelle date, vous savez 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.

Ecrire un GDD sert principalement à transmettre des informations aux autres personnes. Si par exemple, vous voulez que votre jeu inclue 45 types de créatures, les artistes de l'équipe devront modéliser, texturer et animer chacune d'entre elles. 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. Personne ne pourrait garder tout ça en tête. Les jeux de sport par exemple 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.

Le processus d'écriture transforme également 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.

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 » mais la plus grande partie du design consiste à bien penser aux détails et à essayer de les retranscrire sur papier le mieux possible. 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 schémas et en plans concrets et même si personne d'autre que vous ne lit votre document, écrire une idée, c'est prendre une décision, atteindre une conclusion.

En tant que développeurs, nous tolérons les longs horaires parce que nous avons la chance d'apporter une contribution créative et si cela nous était retiré, il n'y aurait plus d'intérêt à développer. En conséquence, une énorme partie des décisions de conception ne sont pas prises à votre bureau, mais lors de réunions, sur un forum, autour d'une tasse de café ou d'un repas... Certaines d'entre elles doivent être mises de côté avec l'aide du « lead designer » et du reste de l'équipe. Elles peuvent sembler bonnes mais poser des problèmes dans l'histoire ou dans la réalisation du jeu. Dans tous les cas, quand une décision de game design finie par être prise et vous devez poser vos conclusions sur le papier — même si vous savez déjà que vous allez les modifier plus tard. La raison en est que vous avez besoin d'une trace papier, un enregistrement de ce que vous avez déjà décidé. C'est comme pour toutes les réunions. Tout le monde parle et on prend des décisions, mais les réunions ne servent à rien sans un « compte-rendu de réunion » car après quelques jours plus personnes ne se rappelleraient des décisions qui y ont été prise et 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. Si les gens ne se souviennent pas de la même chose, vous pouvez vérifier les informations dans le GDD.

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 lorsque les différents membres de l'équipe ont une idée différente de ce que vous attendiez. Il est bien plus facile et moins cher de corriger une erreur de design avant que le code ne soit écrit ou que les éléments graphiques ou sonores ne soient réalisés.

Le point le plus important est donc que les *design documents* sont des outils permettant de s'organiser. Il ne s'agit pas d'histoires à lire, mais d'une liste d'informations et de choses à mettre en œuvre pouvant prendre de nombreuses formes : diagrammes, *concept art*, références graphiques, textes explicatifs, tableaux de statistiques ou d'attributs, listes de toutes sortes, storyboards, notes de réunion...

## 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 mais j'espère avoir répondu à certaines questions sur ces documents.

Et en effet, pour 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, vous n'écrirez sans doute 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 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, une répartition du travail et donc suivre leur avancée, sans savoir ce qui doit être produit, et cette information doit exister sous une forme organisée, écrite et agrémentée de diagrammes, tableaux et listes. C'est ça la conception. Vous vous en passez, mais à vos risques et périls, les risques augmentant de plus en plus avec la taille du projet.