

Introduction au développement de jeu vidéo

GDE-101 - 16.03.2021



Changelog

- Vous avez débloqué la partie Game Design sur Moodle !
 - Et sur Discord aussi. Des ressources vont arriver.



A propos des ressources

- Les ressources partagées sont souvent des oeuvres derrière paywall ou sous copyright.
- Elles peuvent cependant en général être obtenues avec vos accès [EPFL](#) ou [UNIL](#).
 - D'ailleurs, le saviez-vous ? Tant que vous êtes étudiant·e, vous avez la possibilité de télécharger pas mal d'articles scientifique gratuitement ! Profitez-en !
- Une autre solution qui fonctionne, c'est de faire un petit courriel aux scientifiques concernés, et de leur demander gentiment.



Dr. Holly Witteman
@hwitteman

That \$35 that scientific journals charge you to read a paper goes 100% to the publisher, 0% to the authors. If you just email us to ask for our papers, we are allowed to send them to you for free, and we will be genuinely delighted to do so.

Louie  @Mantia

What's something that seems obvious within your profession, but the general public seems to misunderstand?

Show this thread

6:46 PM · 05 Jul 18



A propos des ressources

- Il existe également d'autres moyens d'obtenir les articles et les ressources de manière illégale.
- **Game* ne cautionne pas l'utilisation de sites illégaux**, tels que Sci-Hub ou Library Genesis (pour les articles et les livres), TPB (pour les films ou les jeux), BookFI (pour les livres).
 - Si une oeuvre indé vous plaît, ça reste chouette de la payer !





Le Game Design



Le Lion, la Sorcière blanche et l'Armoire magique





Plan

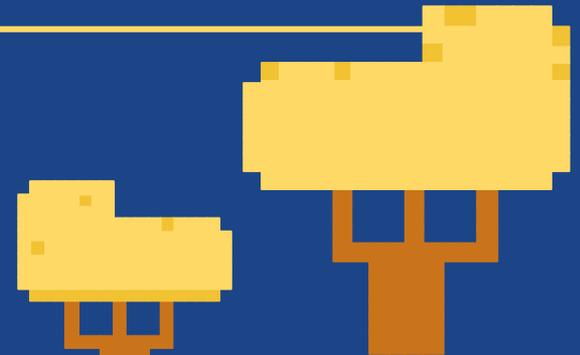
C'est quoi le game design ?

Le cahier des charges

Tester et retester

Projet !

Prototyper





C'est quoi le game design ?

C'est gentil de demander.



Question complexe, s'il en est ?

- “Game design is the act of deciding what a game should be” (Jesse Schell, The Art of Game Design).
- “Game design is the art of applying design and aesthetics to create a game” (Wikipédia).
- “[G]ame design is the process of conceiving, planning and directing the creation of an electronic game” (WhatIs.com).
- “Le game design [...] est le processus de conception préalable des mécaniques d'un jeu avant son élaboration” (WikiBooks).
- “Le game designer est « le chef d'orchestre » de l'équipe de création d'un jeu vidéo” (Strate, École de Design).

Le game
design est
l'art de
concevoir et
d'assembler
de manière
logique les
éléments
d'un jeu.



Un peu de contexte

- C'est super intéressant et y'a des cours complets dessus !
- L'histoire du Game Design moderne est fondamentalement lié à l'histoire du jeu de manière générale.
 - Depuis la naissance de l'humanité^[citation needed], les humains jouent, c'est un fait.
 - Les jeux évoluent en fonction des époques et des cultures, mais suivent (presque) toujours des règles.
 - Ces règles sont parfois fixées, mais surtout changent en fonction des gens (Uno controversy)
 - Avec le jeu vidéo, c'est un peu différent...



L'évolution

- Au départ, les jeux vidéos héritent des codes de jeux de plateau ou de jeux de sports.
- Également au départ, les jeux sont en général réalisés par peu de personnes.
 - Certains sont d'ailleurs des références et des vétérans de l'industrie, comme John Carmack ou John Romero (oui).
- Au fur et à mesure, les compétences (et donc le nombre de personnes) augmentent, et un besoin apparaît pour des gens pour coordonner le tout !
 - De nouvelles célébrités, comme Sid Meier ou Hideo Kojima.



Théorie du jeu, mathématiques

- Le Game Design moderne repose sur BEAUCOUP de domaines.
 - Théorie des jeux.
 - Sciences du jeu.
 - Design et IU.
 - Logique.
 - Etc.
- Des mathématiciens comme John Von Neumann (papa des modèles d'ordinateur) montrent que l'on peut quantifier les actions des joueur·euse·s.
- Il faut également prendre en compte le confort et l'intérêt des humains dans le jeu.





Dans le cadre de ce cours...

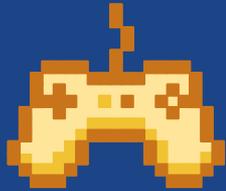
- On ne va pas (trop) partir en théorie.
- On va tenter de rester sur les axes du cours (design, prog, art), avec la contrainte de notre plateforme (PICO-8).
- On ne va probablement^[citation needed] pas révolutionner l'industrie.





Des questions ?

discord.gg/8tVCZJG



Par écrit : canal #en-direct

Par oral : demander la parole
dans #en-direct





Tester et retester

Avoir des retours, c'est important





Un peu controversé de commencer avec ça ?

- Tant pis 🙄.

This slide intentionally left blank.

Pourquoi tester ?

- Il est vraiment facile de se perdre dans une création.
 - Vous la connaissez dans tous les sens, vous savez exactement où et quoi, vous avez pensé à tout.
 - ... Vraiment ?
- Ce n'est pas parce que *ça vous semble* logique que c'est logique.
 - *Human is weird.*
- L'expérience prime.
- Attention : ce n'est pas une session de debug ou une étude de marché.



^ tw gunshot

Le testing
permet de
voir votre jeu
avec les
yeux d'un·e
joueur·euse.





Les grandes étapes

1) Protocole

Écrire un protocole de test pour déterminer les aspects à tester.

2) Recrutement

Rechercher des gens pour participer aux tests.

3) Session

Laisser les gens casser des trucs.

4) Observer

Se placer à l'extérieur pour recueillir des informations.

5) Discuter

Échanger avec les gens ayant réalisé les tests.

6) Implémenter

Sélectionner et implémenter les idées et retours.



Écrire un protocole

- Déterminer quoi tester.
 - Tout le jeu ? Une mécanique particulière ? Un aspect précis ?
- Pour de petites créations, on peut “tout tester”, mais, de manière générale, c’est intéressant de savoir ce que l’on attend.
- Déterminer le public cible.
 - Viser plutôt l’étudiante en informatique ou le conducteur de taxi ?
Plutôt les militaires ou plutôt les cuisiniers ?
- (Bonus) Pour une expérience scientifique, choisir ses variables d’étude.



A qui demander ?

- Un maximum de gens !
 - C'est souvent une idée de base de demander à des gens "qui s'y connaissent"... Mais cela introduit un biais.
 - Essayez d'avoir autant de sources que possible (familles, ami·e·s, âges, genres, métiers, expérience de jeu, ...).
 - Visez votre public cible (jeu éducatif ? multijoueur ? formation ?)
- Plusieurs "niveaux" :
 - Les proches pour obtenir des avis "rassurants".
 - Les expert·e·s pour obtenir des avis "tranchants".
 - Les étranger·ère·s pour obtenir des avis "diversifiés".





Comment playtester ?

En vous taisant.





Comment playtester ?

- Toute interaction humaine est une influence.
 - “Le but du jeu c’est...” → Si le jeu ne peut pas être compris sans vous, il y a une amélioration à faire.
 - “Qu’est-ce que tu penses de ceci” → La pensée de la personne va être dirigée vers le “ceci”.
 - “Je trouve que ceci pourrait être amélioré, qu’en penses-tu ?” → Biais sur la qualité du “ceci”, la personne va chercher un défaut.
 - “Qu’est-ce que tu irais ajouter ?” → La personne va chercher quelque chose à ajouter.
 - Etc.
- Je vous recommande de toujours donner le **minimum** d’informations sur le jeu.

Le·La
joueur·euse
va se
tromper,
faire des
erreurs,
casser des
choses.



Prenez des notes !

- Observez comment les joueur·euse·s abordent le jeu, qu'est ce qui est recherché en premier, comment les règles sont assimilées, etc.
- Prenez note des actions :
 - Est-ce que l'ordre des actions est respecté ?
 - Est-ce que les règles sont (volontairement ?) ignorées ? Oubliées ?
 - Est-ce qu'il y a une surcharge mentale ? Pas assez d'éléments ?
- Notez également les réactions apparentes !
 - Les gens se parlent à eux-mêmes. Profitez-en.
 - Observez le visage, les mains, les muscles. Tension ? Hésitation ? Intérêt ? C'est observable !
- On dérive doucement vers des sciences sociales...

Savoir prendre du recul

- Après avoir passé des heures, des jours, des années sur une création, recevoir une critique négative peut faire mal...



- Il faut savoir accepter la critique, la décortiquer, en garder l'essence, et jeter l'inutile.
 - Toute critique est bonne à prendre... Mais toute critique n'est pas bonne à accepter.

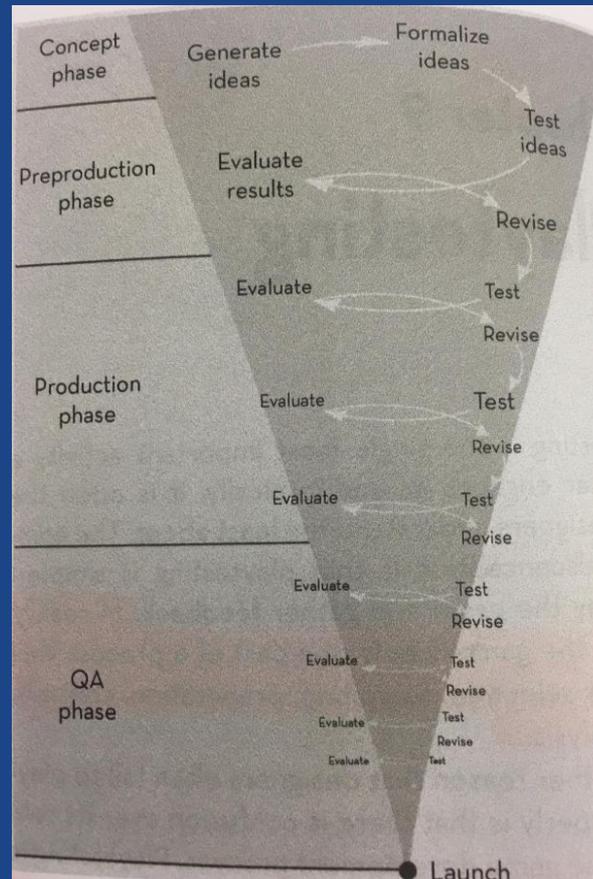


Keep in mind

A Game Design is not *“set in stone”*.

Creativity is all about **breaking boundaries** and
challenging ideas!

Le
playtesting,
c'est régulier
! Des
sessions
régulières
sont
nécessaires.





Des questions ?

discord.gg/8tVCZJG



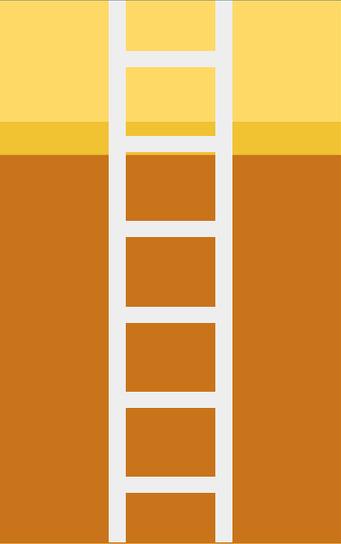
Par écrit : canal #en-direct

Par oral : demander la parole
dans #en-direct





Prototyper



Pour tester il faut un support après tout





C'est quoi un prototype ?

- “[...] un modèle original qui possède toutes les qualités techniques et toutes les caractéristiques de fonctionnement d'un nouveau produit [...]” (source).
- Meh.
- On va partir sur “un moyen de tester une idée”.



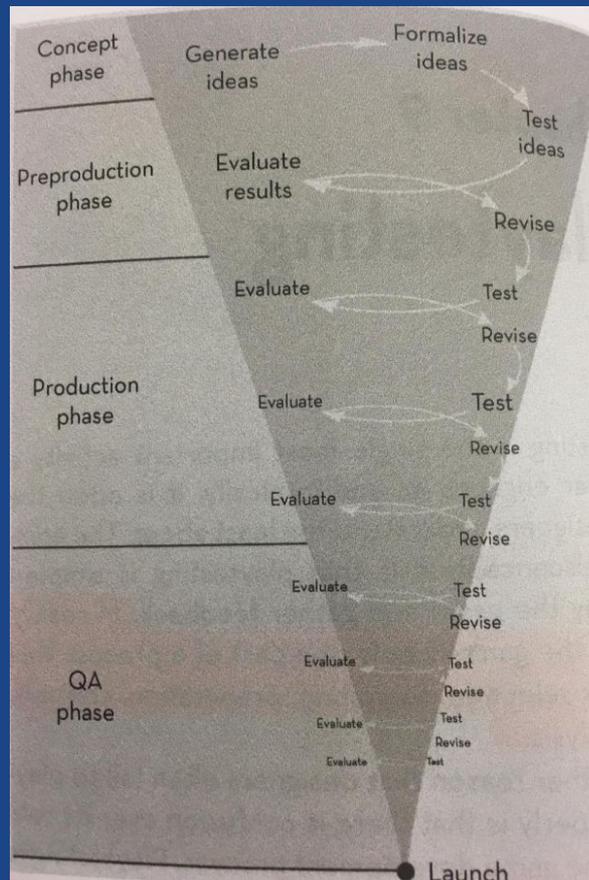


Pourquoi ?

- Les humains, ça aime bien tester des trucs.
 - L'idée c'est d'exploiter ce côté pour faire évoluer son jeu dans la bonne direction.
- Prototyper, c'est améliorer.
 - Chaque prototype permet de voir ce qui va, ce qui ne va pas.
 - Cela permet de "moins regretter" en cas d'échec.

Le ~~playtesting~~ prototyping, c'est régulier ! Des sessions régulières sont

nécessaires blablabla vous avez
l'idée.





C'est la vie

- Faire des prototypes, c'est bon, mangez-en.
- Faire *plusieurs* prototypes, c'est encore meilleur.
 - Plusieurs prototypes sur un aspect spécifique, une mécanique précise.
 - Si ça fonctionne → On garde ; Si ça ne fonctionne pas → On enlève.



Le prototype virtuel (ou mental)

- Fondamentalement pas un prototype en soit, mais peut aider.
- Un papier, un stylo, un pote, et l'idée en quelques mots.
 - Cela permet de confronter rapidement les points de vue, de voir si l'idée est viable, et surtout si elle est existante ("Ah ouais c'est un peu comme Fortnite non ?").



Le prototype paper (ou physique)

- Utilisez du papier, du crayon, des ciseaux, des pions, ... Pas besoin de coder pour créer des jeux vidéo !
 - En tout cas au début.



Le prototype digital (ou numérique)

- Une fois les mécaniques et les grandes lignes fixées, il est temps d'avancer.
- Créer un (plusieurs) prototype(s) digital(taux).
 - Pas obligé de faire un truc bien codé dès le départ !
 - Utiliser des outils de prototyping/de faux jeux (vidéo, slide, ...).





Stades de développement

Stades de développement [\[modifier \]](#) [modifier le code \]](#)

Pré-alpha [\[modifier \]](#) [modifier le code \]](#)

La terminologie "pré-alpha" fait référence aux étapes du développement du produit logiciel et précédant les tests formels, comme l'[analyse des exigences](#), la conception du logiciel, le développement et les tests unitaires. Dans le développement open source, il existe plusieurs types de versions pré-alpha.

Alpha [\[modifier \]](#) [modifier le code \]](#)

La phase alpha du cycle de vie du logiciel est souvent la phase dans laquelle on commence [les tests logiciels](#) (alpha étant la première lettre de l'[alphabet grec](#), utilisée comme chiffre 1). Dans cette phase, les [développeurs](#) testent généralement le logiciel en utilisant des techniques de [boîte blanche](#). Une validation supplémentaire est ensuite effectuée par une autre équipe de test, en utilisant des techniques de [boîte noire](#) ou de [boîte grise](#). Au sein d'une équipe, la version qui précède le test en boîte noire est communément appelée *version alpha*.

Bêta [\[modifier \]](#) [modifier le code \]](#)

La bêta, nommée d'après la [deuxième lettre de l'alphabet grec](#), est la phase de développement logiciel suivant l'alpha. La phase bêta commence généralement lorsque le logiciel est assez complet, mais susceptible de contenir un certain nombre de bugs. Les bugs des logiciels en phase bêta impactent souvent la vitesse ou la [performance du logiciel](#), et peuvent provoquer des [plantages](#) ou des pertes de données. L'objectif des tests bêta est de réduire l'impact des bugs sur l'expérience utilisateur, en incorporant souvent des [tests d'utilisabilité](#). Le processus de livraison d'une version bêta aux utilisateurs est appelé *publication bêta* et c'est généralement la première fois que le logiciel est disponible en dehors de l'organisation qui l'a développé. Les versions bêta du logiciel peuvent être publiques ou privées, selon la clientèle ciblée. La version bêta est souvent utilisée pour les démonstrations au sein d'une organisation, mais aussi pour attirer des clients potentiels.

Le rôle des [bêta-testeurs](#) est de signaler activement les problèmes de la version bêta d'un logiciel. Ce sont généralement des clients ou des représentants de clients travaillant au sein de l'équipe. Les bêta-testeurs ont tendance à proposer leurs services gratuitement.

Version stable [\[modifier \]](#) [modifier le code \]](#)

Également appelée *version de production*, la *version stable* est la dernière version qui a passé toutes les vérifications et tests. Les bugs restants sont considérés comme acceptables et n'affectent généralement pas l'expérience utilisateur. La version stable est mise en production. Dans certains domaines (par exemple dans les [distributions Linux](#)), on retrouve deux types de versions stables: les versions normales (ou stables) et les versions de [support à long terme](#) (LTS) qui sont maintenues pendant une plus longue durée.



Des questions ?

discord.gg/8tVCZJG



Par écrit : canal #en-direct

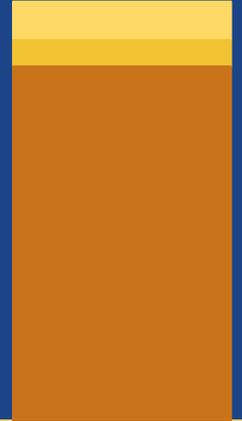
Par oral : demander la parole
dans #en-direct



Le cahier des charges



La feuille de route !





L'idée est d'orchestrer une expérience





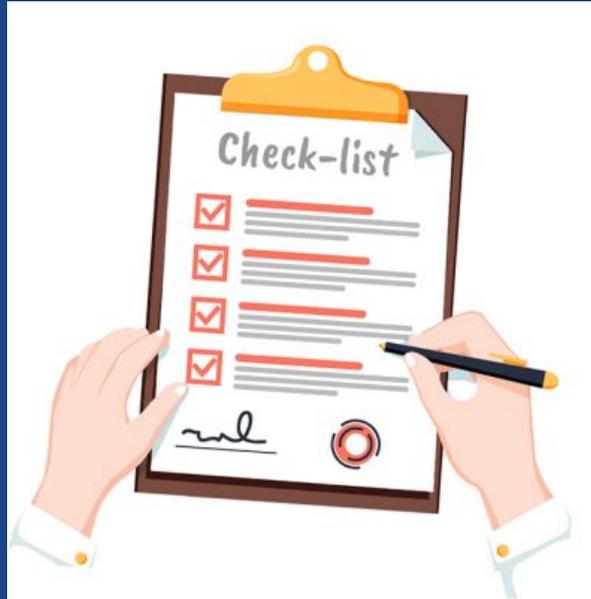
Comment trouver des bonnes idées ?

- Trois axes
 - Observer.
 - *Brainstormer*.
 - Cadavre exquis.
- Des idées, c'est facile à trouver, mais éviter de chercher trop loin.
 - "Je vais faire un shooter FPS en 3D avec des parties plateforme et un bullet time sauf qu'il est limité et consomme de l'énergie magique, qu'on peut récupérer avec des mini-jeux de karting et..."
 - Sim-pli-fier.
- ... Et faut tester (;.



C'est pas tout d'avoir des idées...

- Réaliser un cahier des charges permet d'avoir une ligne directrice claire - et de mettre tout le monde au clair.





Le game design document

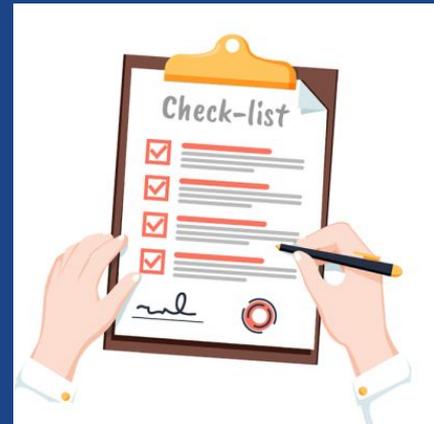
- Contient en gros tous les éléments permettant de diriger son jeu.
 - D'excellents exemples sont en ligne !
- Personnalisé pour *votre* jeu, n'est pas obligé de tout contenir.
- Parmi les éléments susceptibles d'y apparaître :
 - Thème du jeu, "mood", ...
 - Mécaniques principales, mécaniques tierces.
 - Influences - relativement intéressant pour chercher vos inspirations.
 - Elevator pitch (#Startup) : le jeu en trois phrases et/ou 90s, c'est quoi ?
 - Histoire, lore, éléments de l'univers.
 - Explication du gameplay.
 - Graphismes, style, musique, ...
 - etc.

C'est
important
d'avoir un
document
plus ou
moins
formel qui
décrit votre
jeu.



Sur quoi je me concentre ?

- Évidemment, cela va dépendre de votre jeu...
- Une bonne base dans le cadre de cours :
 - Une ou deux phrases pouvant décrire votre jeu.
 - Une ou deux phrases pouvant raconter l'histoire (s'il y en a une...).
 - Une ou deux phrases pouvant expliquer le gameplay.
 - Une ou deux phrases *par mécanique*.
 - Condition de victoire, condition de défaite.
- Bonus :
 - Public cible.
 - Inspirations.



L'originalité
c'est voler
depuis
assez de
sources
pour ne pas
arriver à
pointer le
plagiat.





Des questions ?

discord.gg/8tVCZJG



Par écrit : canal #en-direct

Par oral : demander la parole
dans #en-direct



Merci de votre attention !

discord.gg/8tVCZJG

Par écrit : canal #en-direct

Par oral : demander la parole
dans #en-direct

CREDITS: This presentation template was created by Slidesgo,
including icons by Flaticon, and infographics & images by Freepik.