

Comment **concevoir** un **gameplay secondaire** pour un MMO de stratégie sur mobile ?

Félix Belthoise



Mémoire de stage
Licence Professionnelle Métiers du Jeu Vidéo - 2018 / 2019



REMERCIEMENTS

Maëlle Holtzer
Marie Muller
Darius Guernet

Bruno Belthoise
Philippe Laurens
Franck Scharff
Tap4Fun et toute son équipe
Ainsi que toute l'équipe pédagogique de la LP MJV



ABSTRACT

“[There are] two ways of looking at consistency: the first says that a system is consistent with the external world if every theorem comes **true**; the second says that a system is internally consistent if all theorems come out **mutually compatible** [...]”.

In this statement by Douglas Hofstadter in the fourth chapter of his masterpiece *Gödel, Escher, Bach* lies the central point of my researches on the design of a secondary gameplay. As my main task at Tap4Fun Paris, I worked on it during the entirety of my internship. Starting with the notion of consistency, I will try to explain how I designed a secondary gameplay for a mobile MMO strategy game.

The goal of this document is to provide a useful insight into a design work involving the coexistence of two different gameplay systems. First, you will find a presentation of the company I worked for and my role as an intern Game Designer. Then, I detail the process and methods I used to design a secondary gameplay. The third and last part is a summary of the knowledge acquired during this internship and my studies.

SOMMAIRE

1 INTRODUCTION

2 I/ POSITIONNEMENT À TAP4FUN

3 A) Présentation de Tap4Fun

4 B) Félix à Tap4Fun

6 C) Quotidien à Tap4Fun

7 II/ CONCEPTION D'UN GAMEPLAY SECONDAIRE : DRIVE THE ROCKET

8 A) Recherche du gameplay secondaire

8 1) Ape o'calypse et les Strategy Live Games

9 a. Principe de base d'un SLG

10 b. Exemples de SLG

11 c. *Ape o'calypse*

16 2) L'insertion d'un gameplay secondaire

17 a. Objectif d'un gameplay secondaire

19 b. Exemples de gameplay secondaires

21 c. Contraintes d'un gameplay secondaire

22 3) Recherches préparatoires

23 a. Le lien gameplay

24 b. Le lien des valeurs de *Ape o'calypse*

25 c. Perspective narrative

26 Le choix : Drive the Rocket

26 1) Pitch

27 2) Explication du choix : Franck et les monstres
souterrains

28 B) Intégration du gameplay secondaire

28 1) Level Design : itération 1

29 a. Présentation

30 b. Les Chunks

32 c. Adaptation à la vitesse

33 2) Level Design : itération 2

33 a. Présentation

34 b. Les différentes zones

36 c. Adaptation à la vitesse

38 3) L'intégration au jeu principal

39 a. L'accès à Drive the Rocket : une
manière diégétique

41 b. Le sentiment d'obligation

42 c. La compétition

43 III/ SYNTHÈSE

44 A) Tap4fun : un cadre professionnel

45 B) Tap4Fun et l'enseignement de Bobigny

46 C) L'intérêt de l'expérience personnelle

47 D) Félix Belthoise : Game Designer

48 CONCLUSION

50 ANNEXES

51 Bibliographie et Webographie

52 Lexique

53 Pitches

58 Génération procédurale

62 Carnet de notes

INTRODUCTION

Pour son prochain MMO de stratégie, Tap4Fun Paris a souhaité engager un Game Designer Junior pour la réalisation d'un gameplay secondaire.

Je suis **Félix Belthoise** et j'ai répondu à cet appel, dans le cadre de mon stage de fin d'études de la Licence Professionnelle Métiers du Jeu Vidéo : Level Designer et Game Designer.

Un gameplay secondaire est un mini-jeu optionnel, **inclut** et **lié** à un jeu plus large. Mais quelles sont donc les **spécificités** d'un tel gameplay ? Comment le **cadre** du MMO de stratégie **influence**-t-il sa conception ?

Plus simplement, **comment concevoir un gameplay secondaire pour un MMO de stratégie sur mobile ?**

Mon approche pour répondre à cette problématique se fait à travers la notion de **consistance**.

D'après la définition de Douglas Hofstadter dans son livre *Gödel, Escher, Bach* (chapitre 4), un système est dit **consistant avec son environnement** s'il est en harmonie avec les éléments externes, et il est **consistant avec lui-même** si tous ses éléments tendent vers le même effet d'ensemble.

On parle alors de **consistance interne et externe**.

Ces caractéristiques, une fois appliquées au game design, ont guidé mes recherches dans la conception de ce gameplay secondaire.

La première partie de ce document présente Tap4Fun, ma place dans l'équipe et le contexte dans lequel mon travail a été mené.

Dans une seconde partie, je développe le processus de création d'un gameplay secondaire à la fois consistant avec le jeu principal et consistant intérieurement.

La troisième partie revient sur les relations entre la Licence Professionnelle et mon travail à Tap4Fun, tout en effectuant une rétrospective sur chacun des deux et sur mon apprentissage.

Enfin, une conclusion synthétise les éléments de réponse à la problématique explorés dans l'ensemble de ce mémoire.

I/ POSITIONNEMENT À TAP4FUN



A - Présentation de Tap4Fun

Spécialisé dans le développement de **MMO* free-to-play* sur mobile**, Tap4Fun est à l'origine de plusieurs jeux à succès tel que **Brutal Age** et **Sea Game**.

Fondé en Chine en 2008, Tap4Fun fait ses débuts dans le développement d'applications pour iPhone.

Ce n'est qu'en 2011 que l'entreprise s'est tournée vers le marché du **jeu mobile** en sortant une série de trois MMO de stratégie : Island Empire, Galaxy Empire et King's Empire.

Leur objectif ? **Créer des jeux en ligne compétitifs qui connectent les joueurs du monde entier.**

Aujourd'hui, Tap4Fun est une entreprise mondiale dont le siège et studio central est situé à **Chengdu** (Chine), qui a fondé deux autres studios : **Tap4Fun Paris** et **Tap4Fun Beijing**.

Toujours dans un objectif d'extension, l'entreprise souhaite ouvrir prochainement un studio à Bangkok.

L'organisation de l'entreprise est faite pour que chaque studio puisse **développer un jeu simultanément.**

Tap4Fun Paris possède ainsi son propre projet en se chargeant de la **partie créative** (game design et éléments graphiques), développée dans le même temps à Chengdu par une équipe de programmeurs.

La communication entre Paris et Chengdu est donc quotidienne et les allers-retours fréquents !



Brutal Age et Sea Game



Island Empire, Galaxy Empire et King's Empire

B - Félix à Tap4Fun

Mon entrée comme stagiaire à Tap4Fun Paris s'est faite grâce à une annonce relayée par **Maëlle Holtzer**, Game Designer à Tap4Fun que j'avais précédemment rencontrée lors de plusieurs événements, et **ancienne étudiante à la Licence Professionnelle Métiers du Jeu Vidéo : Level Designer et Game Designer de Bobigny**.

Après une candidature et un entretien, me voilà Game Designer Junior à Tap4Fun pour 3 mois et demi !

Tap4Fun Paris est composé de 12 personnes incluant deux **Game Designers**, quatre **Artistes 2D**, trois **Artistes 3D** et un **programmeur**.

Mes principaux interlocuteurs furent donc **Maëlle Holtzer** (également ma tutrice d'entreprise), **Franck Scharff** (Game Designer) avec lequel j'ai partagé mon travail et plusieurs réflexions de Game Design, et **Philippe Laurens** (Studio Manager) pour les prises de décisions importantes.

De plus, j'ai eu l'occasion de travailler en partie avec **Fabien Rioli**, le programmeur liant le studio Parisien et les développeurs de Chengdu, qui avait commencé à prototyper le gameplay secondaire que j'avais développé dans le cadre de mon stage.

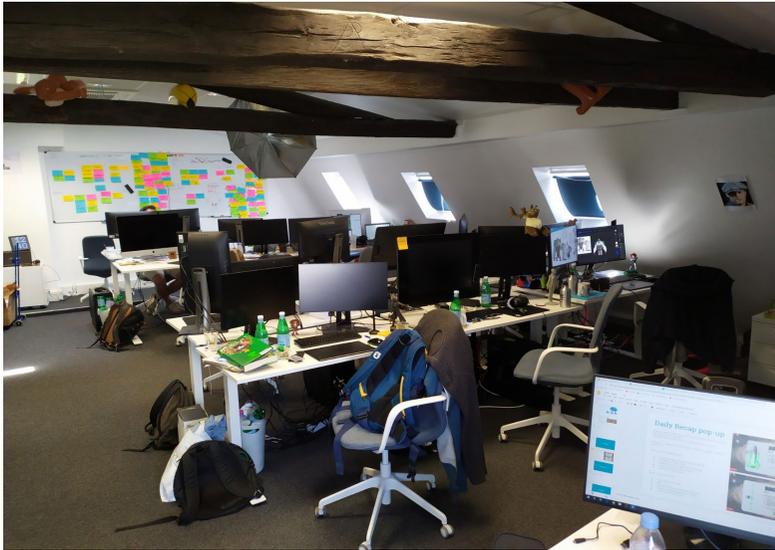
Plusieurs de mes travaux sur l'Interface Utilisateur m'ont également orienté vers **Nicolas Rivet**, le Lead Artist du projet.



*Philippe Laurens
(Studio Manager)*



*Maëlle Holtzer
(Game Designer)*



Bureaux en Open Space



Salle à manger



Espace canapé



... et Super Smash Bros !

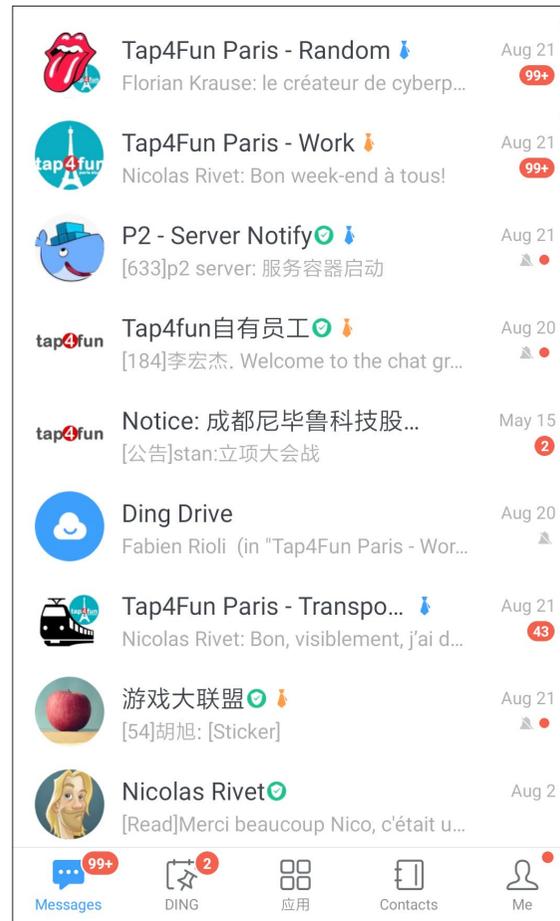
C - Quotidien à Tap4Fun

Le principal outil de communication de Tap4Fun entre les membres de l'équipe et avec Chengdu était **DingTalk**, une application bureautique et mobile permettant de discuter à la fois en privé et en groupe.

Le partage de fichiers se faisait par **Google Drive**, et l'entreprise a mis à disposition plusieurs **tablettes tactiles** pour tester les jeux.

Les développeurs, quant à eux, ont utilisé **Unity** et **Git** pour collaborer sur le projet.

Concernant mes outils personnels, j'ai utilisé en majorité **Google Slides**, **Google Sheet** et **Google Docs** pour la documentation, ainsi que **Photoshop** et **Illustrator** pour tous les graphiques et mockups que j'ai réalisés.



DingTalk en pleine activité !

Cependant, l'outil s'étant révélé le plus utile dans mon travail était un **cahier de notes** personnel, présent durant toutes les étapes (quelques photos de son contenu sont disponibles dans les Annexes p.62).

L'équipe m'a laissé en **autonomie** la plupart du temps, ce qui m'a permis d'organiser mon temps et mes missions comme je le souhaitais.

Toutefois, des points étaient faits régulièrement pour orienter mon travail et vérifier que je n'étais pas bloqué.

Plusieurs missions m'ont été assignées durant le stage : **aide à la conception du jeu principal**, conception d'**Interfaces Utilisateurs (UI)**, **documentation** en anglais et, bien entendu, **conception d'un gameplay secondaire**.

**II/ CONCEPTION D'UN
GAMEPLAY SECONDAIRE :**
DRIVE THE ROCKET

A - Recherche du gameplay secondaire

1. Ape o'calypse et Tap4Fun Paris

Avant d'aborder le gameplay secondaire, il est essentiel d'étudier le contexte dans lequel il s'inscrit afin de connaître au mieux son environnement.

Le jeu que conçoit actuellement Tap4Fun Paris entre dans la lignée d'un genre de free-to-play très populaire, existant depuis quelques années : les **jeux massivement multijoueur de stratégie en temps réel** (également appelés **SLG** pour **Strategy Live Game**).

La difficulté dans la conception d'un tel jeu réside dans son équilibre entre les mécaniques classiques du genre et les mécaniques originales.

La volonté de Tap4Fun Paris d'apporter sa touche personnelle l'a amené à se fixer la **règle des 80/20** : 80% de mécaniques classiques de SLG pour 20% de nouveautés.

À travers la présentation des bases du SLG et plusieurs exemples de jeux populaires du même genre, nous allons étudier par quelle perspective *Ape o'calypse* (**nom provisoire**) apporte son originalité au genre du SLG.

a. Fonctionnement d'un SLG

Les **SLG** sur mobile sont nombreux et partagent le plus souvent une base de **mécaniques communes** :

- Chaque joueur gère sa propre **ville**.
- Chaque ville est constituée de différents **bâtiments** à débloquent puis à **améliorer**.
- Chaque bâtiment débloquent de nouvelles **possibilités d'actions**, d'**informations** et d'**améliorations** des caractéristiques du joueur.
- Chaque ville est disposée sur une **carte**, commune à un large ensemble de joueurs.
- Des **objectifs** sont présents sur cette carte, amenant les joueurs à **s'affronter** à l'aide de leurs **armées**.
- Les joueurs peuvent se grouper en **alliances** permettant de s'entraider pour la capture des **plus grands objectifs**.
- Des **quêtes** principales et journalières, concernant l'évolution du joueur, le récompensent à mesure de leur accomplissement.



Carte d'un serveur de
Rise of Kingdoms

b. Trois exemples de SLG

Les trois SLG ci-dessous font partie des plus populaires du moment et proposent chacun un univers très différent. Afin d'observer l'étendue des possibilités d'un SLG, en voici une brève présentation :

Rise of Kingdoms (Lilith Games)

Dans Rise of Kingdoms, les joueurs représentent différentes grandes **civilisations** qui traversent l'**histoire**, de l'âge de pierre à l'ère féodale. À l'aide de leurs **troupes** et de leurs **Alliances**, les joueurs devront asseoir leur **domination** par la capture de différents **Autels** et **Sanctuaires** au fur et à mesure du jeu.



Ville dans Rise of Kingdoms

Mobile Royale (IGG)

Mobile Royale est un SLG **médiéval fantastique** dans lequel chaque joueur incarne le **seigneur** d'un château. Grâce à la **croissance** de leur royaume, l'**élevage** de leur dragon et l'**entraînement** de leurs troupes, ils combattent pour la possession et la garde des **Forts** disposés sur la carte.



Ville dans Mobile Royale

Guns of Glory (Funplus)

Situé dans un univers mélangeant **mousquetaires, zombies** et **créatures fantastiques**, Guns of Glory place chaque joueur en tant que **seigneur** d'un grand domaine. Chacun devra **affirmer sa puissance** grâce à l'**exploration** de la carte, la défense de son domaine et les **combats d'Alliance**.



Ville dans Guns of Glory

c. *Ape o'calypse*

Ape o'calypse (nom provisoire) est un SLG sur mobile en cours de développement par Tap4Fun Paris depuis Novembre 2018.

Au début du projet, Maëlle Holtzer (tutrice de stage, Game Designer), Franck Scharff (Game Designer) et Philippe Laurens (Studio Manager) ont constaté que, dans la plupart des SLG, **l'objectif final n'est pas clair** pour un nouveau joueur.

Communiqué de manière indirecte ou incomplète, l'accomplissement de cet objectif doit régulièrement passer par un grand nombre d'**objectifs intermédiaires** dont la **relation** avec l'objectif final n'est **pas explicite**.

Certains joueurs en viennent rapidement à se demander pour quelles raisons ils combattent ...

Ainsi, Tap4Fun souhaite créer un SLG dont l'objectif final est **clair** et communiqué **dès le départ** aux nouveaux joueurs. Il s'agit également de **limiter** le nombre d'**étapes intermédiaires** et de les relier directement et **explicitement** à l'objectif final.

Le thème du jeu, quant à lui, provient des **tests marketings** effectués par Tap4Fun Chengdu (la maison mère) : plusieurs **vidéos** présentant chacune une thématique différente ont été mises en ligne sur les réseaux sociaux, afin d'enregistrer les plus **populaires**.

Tap4Fun a ainsi eu le choix du thème parmi un ensemble de quelques vidéos ayant eu les meilleures réactions de la part du public.

Parmi ces vidéos, plusieurs d'entre elles présentant des **singes humanoïdes armés** dans un **monde post-apocalyptique** ont emballé l'équipe de Tap4Fun Paris, en particulier grâce aux valeurs ressortant de ces vidéos et devenant ensuite les valeurs-clés du jeu : les **Armes**, le **Dynamisme** et l'**Hu-mour**.



Image tirée d'une des vidéos marketing

Ape o'calypse, ou la nouvelle conquête spatiale :

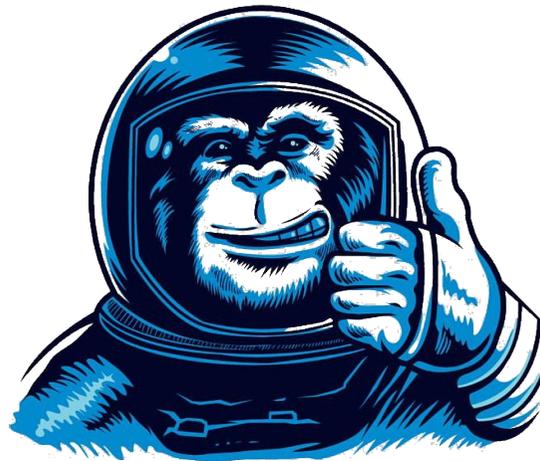
Being stuck on a ruined planet to fight for dominance isn't always fun and guns. Sometimes, you realize that you are stuck on a ruined planet with **NO MORE BANANAS** on it!

This is unacceptable, in fact you barely remember what bananas are. You only remember one thing: **bananas are important.**

You must find the mythical bananas! And if there isn't any on earth, the only option left is... SPACE!

Be the first to launch your space rocket and explore the unknown, TO BANANAS AND BEYOND!

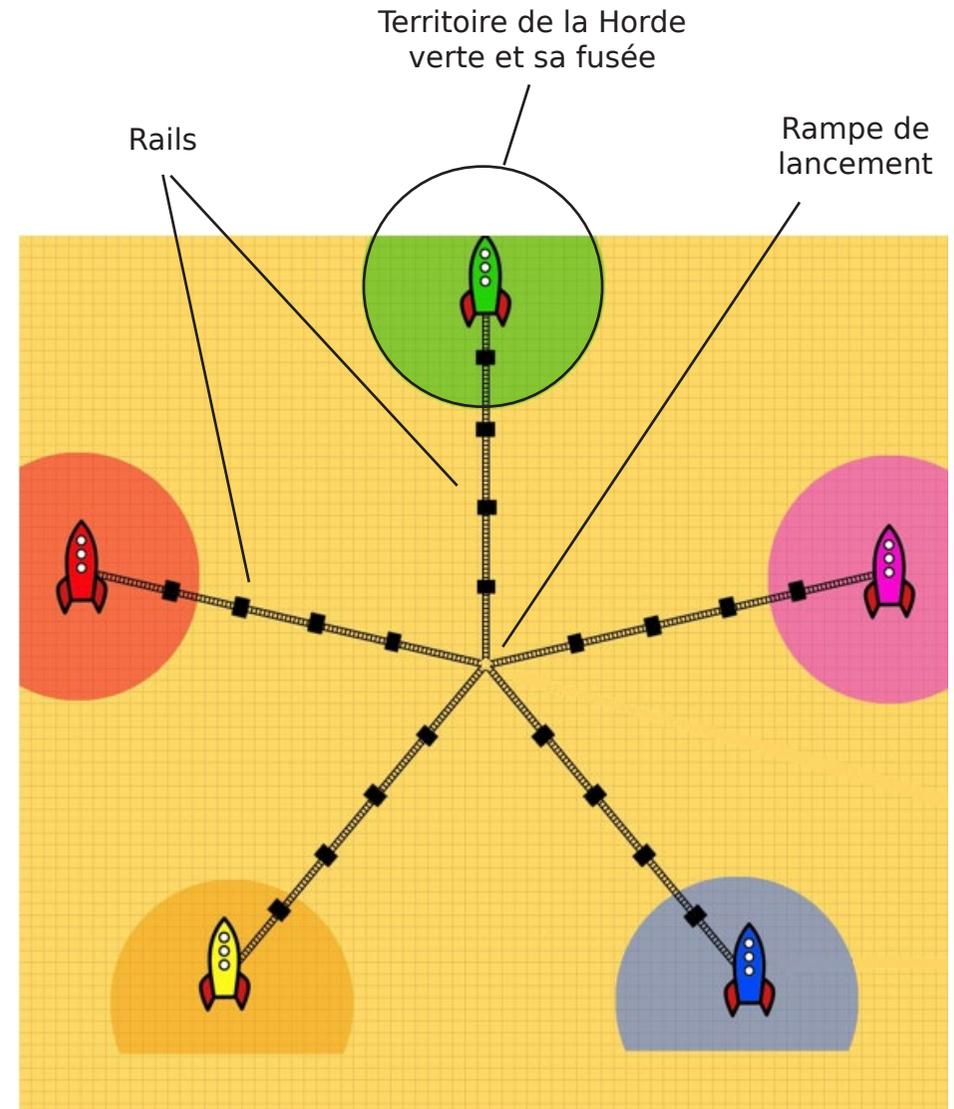
(écrit par Maëlle Holtzer)



Le gameplay, en quelques lignes :

Chaque **Horde** construit une **fusée spatiale** puis la **transporte** jusqu'à la rampe de lancement au **centre** de la carte. La **première fusée** à atteindre l'espace **gagne**, et envoie quelques souvenirs spatiaux à la Horde.

- Il y a **5 Hordes** de singes, chacune possédant son propre territoire
- Chaque joueur fait partie d'une Horde (environ 1000 joueurs par Horde au lancement d'un serveur)
- Chaque Horde possède une **fusée** posée sur un **train**, au cœur de son territoire
- Des **rails** partent de chaque territoire et mènent au **centre** de la carte
- **4 relais** sont disposés le long de chaque rails
- Le jeu se termine lorsqu'une **fusée arrive au centre** de la carte
- Les joueurs de chaque Horde peuvent se grouper en **Alliance**, ce qui permet à ses membres de profiter de nombreux avantages



Premier mockup de la carte de Ape o'calypse

Le but est d'être la première Horde à **envoyer un singe dans l'espace** ! Pour cela, chaque Horde doit :

1/ Remplir sa fusée de carburant

De l'**Uranium** peut être collecté sur la carte.

Les joueurs de chaque Horde doivent unir leurs efforts pour **remplir la fusée d'Uranium**, ce qui prendra plusieurs jours.

Une fois celle-ci pleine, le train démarre automatiquement et s'arrête au premier relai rencontré.

2/ Remplir chaque relai d'Uranium afin d'alimenter le train en carburant

Chaque fois que le train s'arrête à un relai, il y passe un certain temps avant de repartir.

Les joueurs peuvent **diminuer le temps d'arrêt** en **remplissant les relais d'Uranium**.

Les joueurs peuvent également **attaquer les relais ennemis** afin de collecter de plus grandes quantités d'Uranium tout en ralentissant le train ennemi.

Une fois le dernier relai quitté, le train ne s'arrête qu'une fois la rampe de lancement atteinte.



Mockup d'un écran de jeu de Ape o'calypse

2. Place du gameplay secondaire

Maintenant que nous percevons mieux la nature du projet de Tap4Fun, nous pouvons nous demander **pourquoi**, **où** et **comment** le gameplay secondaire s'inscrit dans le jeu principal.

La plupart des SLG possèdent déjà un gameplay secondaire.

Nous allons donc nous tourner vers eux pour comprendre leur **objectif**, **analyser** quelques un d'entre eux et aborder les **contraintes** nécessaires à sa conception.

a. Objectif d'un gameplay secondaire

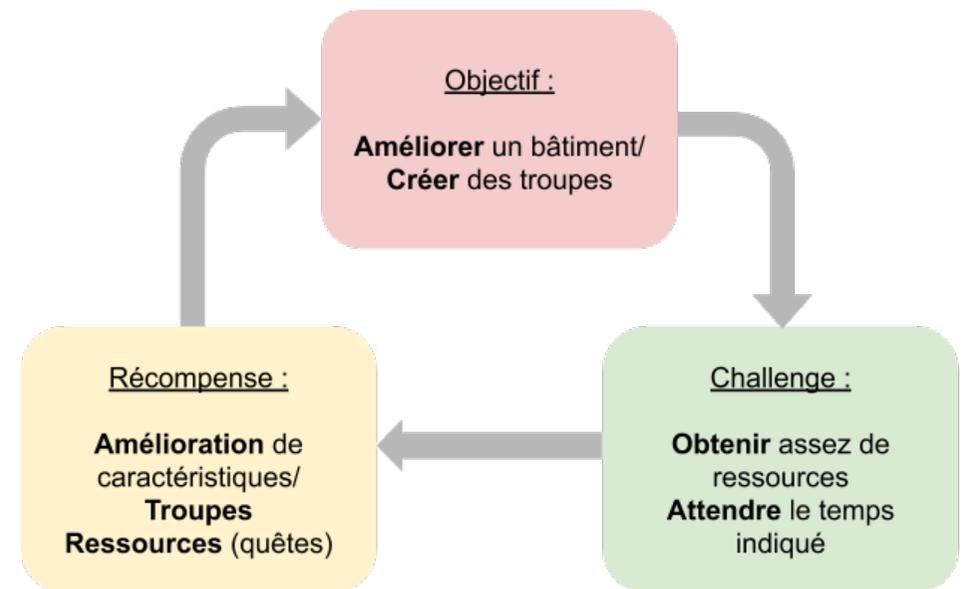
Pour devenir de **plus en plus puissant** dans un SLG, il faut constamment **construire** et **améliorer** ses bâtiments et ses troupes.

Note : pour plus de facilité, nous utiliserons le terme « construction » pour désigner à la fois la construction des bâtiments, leur amélioration et l'entraînement des troupes.

Une construction requiert systématiquement **2 éléments importants** afin d'être achevée : des **ressources** et du **temps**.

- Les **ressources** sont générées automatiquement en petites quantités, mais le joueur peut en obtenir plus rapidement en les **récoltant** sur la carte ou en accomplissant des **quêtes**.
- Le **temps de construction** est, quant à lui, un **temps d'attente** nécessaire avant la finalisation de la construction. Ces temps d'attente (allant de quelques secondes à plusieurs jours) **augmentent** en fonction du niveau de la construction et, de ce fait, en fonction de la **progression** du joueur.

Par conséquent, plus le joueur est **puissant**, plus le temps de complétion d'une boucle de construction **s'allonge**.



Boucle de gameplay pour une construction dans un SLG*

L'objectif du gameplay secondaire est ainsi de proposer une **boucle de gameplay courte** dont le **temps de complétion reste le même** tout au long du jeu afin de contrebalancer les autres temps de complétion croissants.

Le joueur a donc **toujours accès** à un **objectif rapide à accomplir**, quelle que soit sa progression.

Note : En général, les joueurs ont également le choix de dépenser de l'argent réel sous forme de Hard Currency afin de diminuer les temps de construction. Cependant, le gameplay secondaire étant destiné à tous les joueurs, nous prenons en compte en priorité les joueurs confrontés à ces temps de construction dans toute leur longueur.*

b. Exemples de gameplay secondaire dans les SLG

Avant de procéder à la recherche du gameplay secondaire pour Ape o'calypse, j'ai analysé ceux des SLG présentés plus tôt (*Rise of Kingdoms*, *Mobile Royale* et *Guns of Glory*) :

Rise of Kingdoms :

Le mode Exploration de *Rise of Kingdoms* propose au joueur de **combattre** une **intelligence artificielle** dans une suite ordonnée de **niveaux** afin d'obtenir des **récompenses**. Chaque niveau est une bataille contre une ou plusieurs **armées** ennemies. Comme dans le jeu principal, celles-ci combattent **automatiquement** après que le joueur ait choisi la **composition** de son armée.



Mode Exploration de
Rise of Kingdoms

Mobile Royale :

Le gameplay secondaire de *Mobile Royale* propose également des **combats** contre une **intelligence artificielle**. Les armées peuvent être accompagnées par plusieurs **Héros**, ceux-ci possédant des **pouvoirs spéciaux** que le joueur peut activer durant la bataille. Il est également possible de sélectionner les différents **types de troupes** composant son armée.



Mode Campagne de
Mobile Royale

Guns of Glory :

Le gameplay secondaire de *Guns of Glory* propose quant à lui un **Dungeon Crawler***, dans lequel le joueur contrôle une **créature** à travers des **souterrains** peuplés de **monstres** et de **trésors**. La créature peut ensuite être améliorée grâce à des **points de compétences**.



Mode Catacombes de
Guns of Glory

Plusieurs conclusions ont été retenues lors des analyses, permettant de mettre en lumière les principaux **pièges à éviter** lors de la conception d'un gameplay secondaire :

- **Le gameplay doit pouvoir se diversifier facilement**

Dans les trois gameplays secondaires présentés ci-dessus, le gameplay peine à se renouveler.

Le joueur se retrouve souvent dans les mêmes situations en boucle, rendant le tout laborieux.

- **Rendre les phases automatiques (s'il y en a) les moins longues possibles**

Les combats de Rise of Kingdoms sont automatique : le seul moment où le joueur a une véritable influence dans le gameplay secondaire est lorsqu'il compose son armée avant de l'envoyer.

Les combats durent en moyenne une minute entière, le joueur est donc inactif durant toute la durée du combat, ce qui devient rapidement ennuyeux.

- **L'influence du gameplay secondaire sur le jeu principal (et vice-versa) doit être justement équilibrée**

Le mode *Exploration* de *Rise of Kingdoms* nous demande d'utiliser **nos propres troupes du jeu principal** pour combattre.

Comme les niveaux deviennent de plus en plus difficiles et nécessitent de plus en plus de troupes, cela limite donc le joueur à jouer selon sa progression et non selon le moment où il souhaite jouer.

À l'inverse, le Dungeon Crawler de *Guns of Glory* ne possède que les récompenses comme lien gameplay avec le jeu principal, ce qui tend à réduire sa consistance par rapport au jeu global.

c. Les contraintes

Plusieurs contraintes ont donc été posées par mes propres analyses, ainsi que par ma tutrice Maëlle afin d'avoir un **cadre de recherche adapté** et **cohérent** avec le jeu principal.

La dernière étape avant la recherche du gameplay secondaire a donc été de lister ces dernières contraintes :

- **Bas coût de production** : étant un gameplay secondaire, l'équipe n'a pas beaucoup de moyens à lui consacrer. Il doit donc être peu coûteux à concevoir et à développer.
- **Facilité et rapidité de compréhension** : la boucle de gameplay étant courte, le coût d'entrées doit rester bas afin de proposer un jeu léger à prendre en main.
- **S'intègre à l'univers du jeu** : le gameplay secondaire doit pouvoir s'intégrer de manière harmonieuse à l'univers préétabli.
- **Correspond au public visé** : pour remplir son rôle, le gameplay secondaire doit intéresser le public déjà présent sur le jeu principal.
- **Courtes sessions de jeu** : comme vu plus haut, le temps de complétion d'une boucle de gameplay doit rester courte afin de contrebalancer les temps de complétion croissants des autres boucles de gameplay du jeu principal.
- **Ne doit pas être obligatoire** : les joueurs ne doivent pas avoir le sentiment de perdre quelque chose en n'y jouant pas. Il doit donc rester totalement optionnel dans l'esprit des joueurs afin de ne pas contrarier le jeu principal ou pénaliser le public n'y jouant pas.
- **Adapté au jeu principal** : le gameplay secondaire doit être cohérent avec le jeu principal en termes de gameplay, graphismes et narration.

3. Recherches préparatoires

Il est enfin temps de s'atteler à la recherche du gameplay secondaire !

Pour cela, un grand nombre de pitches* ont été proposés à l'équipe. Chacun possède un lien avec le jeu principal afin de rendre le gameplay secondaire le plus consistant possible avec les éléments externe.

Trois catégories de liens se sont rapidement imposées durant mes recherches : les **liens gameplay**, les **liens des valeurs de Ape o'calypse** et les **liens narratifs**, qui permettent de renforcer la cohérence entre le gameplay secondaire et le jeu principal.

a. Le lien gameplay

Mon premier lien d'accroche avec le jeu principal fut le **lien gameplay**, qui consiste à rechercher un **gameplay du même type** que celui du jeu principal (la **stratégie**).

Il m'a semblé qu'un jeu abordant la thématique du combat d'une manière différente que dans le jeu principal intéresserait facilement les joueurs.

Puisant dans ma culture vidéoludique personnelle en ce qui concerne les jeux de stratégie, l'influence de **jeux grand public** comme *Clash of Clans* ou de **jeux « indépendants »** comme *Bad North* ou *Into the Breach* a été déterminante durant la conception des différents pitches.

Mentionné par Philippe Laurens (Studio Manager), un jeu édité en 1990 par Atari nommé *Rempart* m'a ainsi guidé vers le concept de **Defend** (présenté ci-dessous).

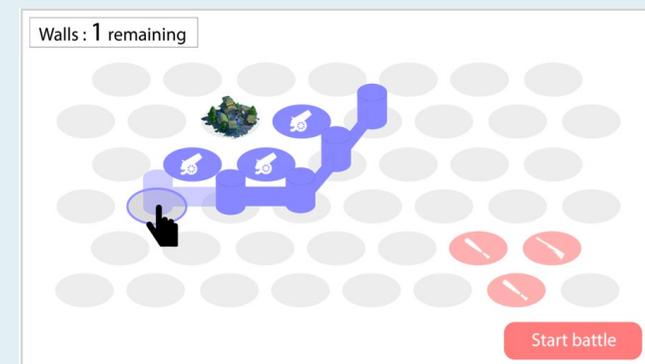
Tous les autres pitches présentés à l'équipe sont en *Annexes p53*.

Defend : *Build and maintain your defenses around your city !*

Dans ce jeu de défense, le joueur alterne entre 2 phases : une phase de **construction**, durant laquelle il peut **créer des remparts** et des **tourelles**, et une phase de **défense** lors de laquelle des **troupes ennemies** contrôlées par une intelligence artificielle attaquent le joueur. Ce dernier peut **contrôler** en temps réel la **cible** de ses tourelles.

Inspiration : *Rempart* (Atari, 1990)

Cependant, bien que l'équipe ait trouvé les pitches intéressants, ils se révélaient à la fois **trop complexe à comprendre et à produire** pour un gameplay secondaire. En effet, tous les concepts auraient demandé un travail de **Level Design très gourmand** tout en étant **trop longs à prendre en main**.



b. Le lien des valeurs de *Ape o'calypse*

Le second lien d'accroche prend comme pivot les valeurs du jeu principal, mentionnées précédemment : les **Armes**, le **Dynamisme** et l'**Humour**.

Il s'agissait donc de mettre une ou plusieurs de ces valeurs au centre du processus de recherche.

Partant de la valeur humoristique, j'ai surtout cherché des **situations amusantes** qui mettent en scène les singes de manière apparemment ridicule. Par exemple, que penserait un singe de la fonction d'un iPod, ne connaissant pas son utilité première ? Selon le jeu *Mutant Year Zero : Road to Eden* ([The Bearded Ladies](#), 2018), il pourrait y voir un « testeur de murissement de fruits » à en juger l'apparence ! (voir le [Hipster Fruit Tester](#))

Une des pistes les plus prolifiques a été inspiré par le gameplay de *Burrito Bison : Launcha Libre*, mentionné par Maëlle durant mes recherches (présenté ci-dessous).

Throw the Astronaut : *“Look, Sam, if our astronaut can't bounce back from this, there's no way he can bounce back from a moon crash ! That's definitely how science works !”*

Après **propulsion** d'un astronaute par un **canon**, le joueur doit diriger l'avatar le plus loin possible en le faisant **rebondir** de la manière la plus efficace.

Appuyer sur l'écran fait **descendre** l'astronaute **plus rapidement** au sol, sa force de rebondissement dépendant de ce qu'il touche lors de l'impact. Le jeu **s'arrête** lorsque l'astronaute **ne rebondit plus**.

Inspiration : *Burrito Bison* ([Juicy Beast](#), 2016)

Ce pitch combine à la fois la valeur des **Armes** (la propulsion par le canon), du **Dynamisme** (les rebondissements de l'astronaute) et de l'**Humour** (l'entraînement d'un astronaute à rebondir proprement et efficacement), ce qui en faisait un candidat de choix pour le gameplay secondaire.



c. Le lien narratif

Le dernier lien d'accroche sur lequel j'ai travaillé concerne le lien narratif, le point de départ de cette perspective étant ici lié à un élément de l'univers du jeu.

Beaucoup d'éléments étaient ici intéressants : l'espace, la fusée, le train, le post-apocalyptique, l'uranium, l'entraînement des troupes ... Je me suis donc appliqué à **combiner ces éléments** avec différents **genres de jeux**.

Entre le jeu de rythme guidant les convois d'Uranium et le Tap Tap game relaxant dans lequel la recherche du génome de banane se transforme en exposition de fruits étranges, on peut citer *Liana Runner* (présenté ci-dessous) inspiré de certains niveaux de *Donkey Kong Country - Tropical Freeze*.

Le principal avantage du lien narratif concerne la **quantité** et la **diversité** des pitches qu'il m'a été possible de proposer. En combinant des genres de jeux avec des éléments de narration, j'ai pu aboutir à plusieurs **idées intéressantes** et **originales** tout en restant **en lien** avec le gameplay principal.

Ces perspectives de travail m'ont donc permis de proposer un grand nombre de pitches différents à l'équipe, et ont guidé les décisions jusqu'au choix du pitch final : **Drive the Rocket !**

Liana Runner : "Train you monkeys in a deadly playground !"

L'écran **défile horizontalement** alors qu'un **singe** se fait poursuivre par une **scie gigantesque**. Des **lianes** de différentes longueurs se **balancent** depuis le plafond : si l'avatar en **touche** une, il la **saisit** automatiquement. Le joueur indique au singe de **lâcher sa liane** par un « **tap** » sur l'écran. Si l'avatar **sort** de l'écran, **rate** une liane ou touche un **obstacle**, c'est **game over** !

Inspiration : Donkey Kong Country : Tropical Freeze ([Retro Studio](#), 2018)



Le choix : Drive the Rocket

1. Pitch

In order to choose which monkey is going into space with the Rocket, all the players of the Horde try to actively train an Astronaut.

Drive the Rocket est un jeu à **défilement horizontal**, dont le but est de parcourir **la plus longue distance** avec une fusée miniature qui avance **de plus en plus rapidement** au fil du temps.

Le joueur contrôle le **mouvement vertical** de la fusée : **appuyer** sur l'écran permet de la faire monter et **relâcher** l'écran la laisse retomber.

En **longeant les murs**, la fusée gagne un **boost** qui augmente sa **vitesse** et **multiplie son score** ; mais si elle **touche un mur** ou un obstacle, elle s'écrase et c'est **game over** !

La vitesse de défilement augmentant au fur et à mesure, ce concept challenge principalement la **vitesse de réaction** du joueur (il a de moins en moins de temps pour réagir aux obstacles).

Le lien entre Drive the Rocket et le jeu principal repose donc en premier lieu sur le **lien narratif** : l'entraînement d'un Astronaute.

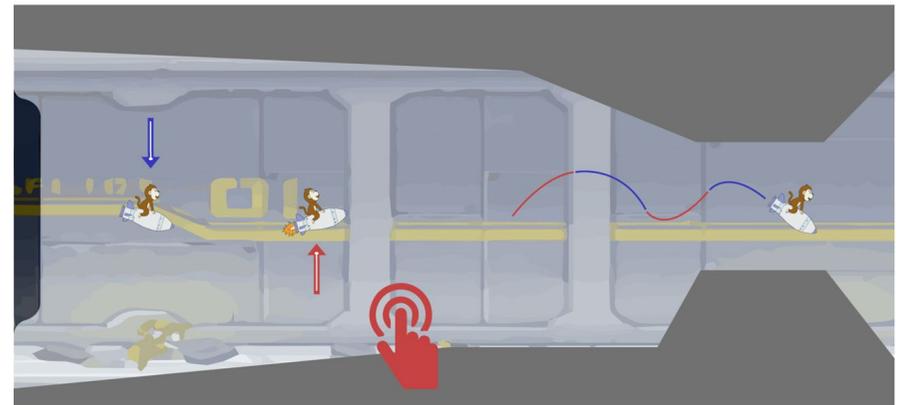


Schéma des contrôles pour Drive the Rocket

2. Explication du choix : Franck & les monstres souterrains

Après plusieurs étapes de sélection, *Throw the Astronaut* et *Drive the Rocket* étaient les deux derniers concepts en concurrence.

Répondant chacun à leur manière aux problématiques de design, la décision de s'orienter vers *Drive the Rocket* vint d'un **élément narratif** conçu par **Franck** (Game Designer) concernant des **monstres** présents sur la carte du jeu principal :

Autrefois appartenant à une 6e Horde, les monstres environnementaux sont des singes sauvages et agressifs s'étant exilés dans de gigantesques souterrains.

Ils accèdent à la surface durant la nuit, à l'aide de trappes présentes sur toute la carte.

Franck a alors proposé d'exploiter les souterrains comme **terrain d'entraînement** pour *Drive the Rocket* et d'utiliser les **trappes** comme moyen d'**accès au gameplay secondaire**.

Cette idée renforçant significativement la cohérence entre le jeu principal et le gameplay secondaire, la team a unanimement accepté de continuer avec *Drive the Rocket*.



Concept de monstre souterrain

B - Intégration d'un gameplay secondaire

1. Level Design : itération 1

La nécessité de garder un bas coût de production a rapidement orienté le Level Design vers de la **génération procédurale***.

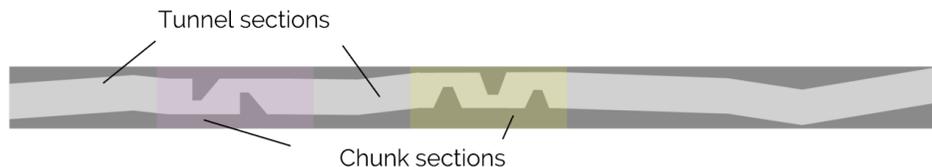
Solution à la fois pratique pour le renouvellement du gameplay et dangereuse par le risque de redondance, c'est également une des techniques les plus adaptées pour les jeux à défilement horizontal infini (*Jetpack Joyride*, *SF Cave* ...).

N'ayant jamais conçu de système procédural, c'était une expérience que j'attendais impatiemment et c'est avec une grande curiosité que je me suis lancé dans une **première itération** de Level Design pour *Drive the Rocket*.

a. Présentation

Mon premier **point de départ** pour la génération procédurale de Drive the Rocket fut la conception un système permettant une **gestion facile** de la **difficulté**.

Me basant sur mes connaissances de **design rationnel***, j'ai imaginé qu'un niveau se génère par **cases égales** d'environ 20 mètres. Deux types de sections pourront apparaître dans ces cases : des sections « **Tunnel** » (ne contenant que des murs plus ou moins droits, cette section étant présente par défaut) et des sections « **Chunk** » (contenant les obstacles).



Mockup de Level Design, itération 1

Chaque case possède une **probabilité**, croissante au fur et à mesure du jeu, de contenir une **section Chunk**. Sinon, c'est une section Tunnel qui sera générée.



Schéma des probabilités d'apparition d'un Chunk pour les 10 première sections

Les **Chunk** étant des sections **plus difficiles** que les Tunnels, ce système permet de **modifier facilement** la difficulté générale et la difficulté croissante en modifiant les **probabilités d'apparition**.

b. Les Chunks

Les sections « Chunks » sont **préconçues** et **choisies aléatoirement** par le jeu en fonction de leur **difficulté** (plus le jeu avance, plus la probabilité d'apparition d'un chunk difficile augmente).

Afin de **calculer** la difficulté de chaque chunk, toujours à l'aide de **design rationnel**, j'ai imaginé que chacun puisse être divisé en **7 parties égales** selon le schéma suivant :



Schéma de la composition d'un Chunk

Divers **obstacles** peuvent être présents sur les **Slots** mais **aucun sur les Gates**, nous assurant que que deux Chunks mis côte à côte soient **toujours jouables**.



Décomposition d'un Chunk en 7 parties

Chaque **obstacle** possède finalement une **valeur** en fonction de sa **difficulté propre**.

Reste à calculer la **difficulté totale** du Chunk en tenant compte de divers **paramètres** comme le **nombre** d'obstacles, le nombre de **spots vides** entre deux obstacles et la **difficulté** de chaque obstacle.

Chaque paramètre aura un **poids différent** dans l'équation en fonction de son importance (à modifier en fonction des tests).

Exemple



Chunk A



Chunk B

On devine ici que le Chunk A est plus difficile à passer que le Chunk B. Voyons quel résultat nous donne l'équation suivante, en supposant que chaque paramètre ait la même importance :

Valeur de difficulté = (Nombre d'obstacles) - (Nombre maximum de spots vides entre 2 obstacles) + (Difficulté de chaque obstacle)

Note : on suppose que les obstacles présent dans les 2 chunks possèdent chacun une difficulté propre de 1

Paramètre \ Chunk	Chunk A	Chunk B
Nombre d'obstacles	3	2
Nombre maximum de spots vides entre deux obstacles	1	2
Difficulté des obstacles	1+1+1 = 3	1+1 = 2
Total	(3) - (1) + (3) = 5	(2) - (2) + (2) = 3

Le résultat confirme notre prédiction : le Chunk A, possédant une difficulté totale de valeur 5, est plus difficile à passer que le Chunk B de valeur 3.

Si nous voulions créer un Chunk C de valeur 4, nous pourrions également nous aider de l'équation pour équilibrer sa conception.

c. Adaptation à la difficulté croissante

La **vitesse** du défilement augmentant petit à petit, je me suis finalement retrouvé face au problème suivant : comment s'assurer que le jeu reste toujours **théoriquement faisable** alors que la vitesse peut augmenter indéfiniment ?

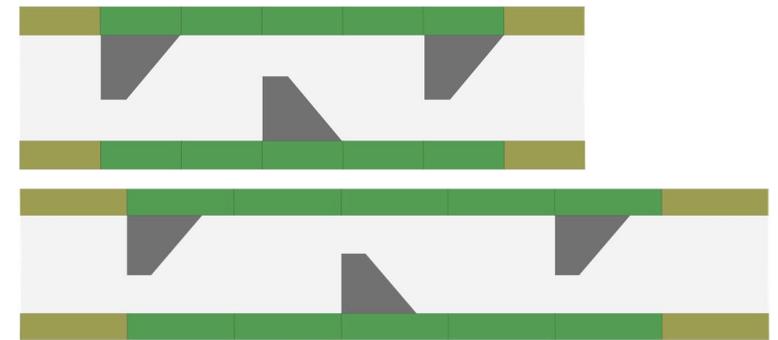
C'est en utilisant à nouveau le système d'évaluation de la difficulté d'un chunk que j'ai trouvé une solution. En **augmentant l'espace horizontal** de chaque partie petit à petit en fonction de la vitesse globale du joueur, **l'espace** entre les obstacles **s'agrandit** et adapte automatiquement le chunk à la vitesse :

*Note : le jeu challengeant principalement la **vitesse de réaction** du joueur face à l'arrivée de plus en plus rapide des obstacles, ce système conserve bien le challenge intact.*

Conclusion sur l'itération 1 :

Cette première proposition de Level Design pour Drive the Rocket permet ainsi un grand contrôle sur la **difficulté** du jeu. Il est rapide à concevoir, prototyper et tester car il requiert un **nombre limité d'éléments** pour fonctionner.

Cependant, l'équipe m'a fait remarquer l'aspect très **mécanique** de l'alternance entre les Chunks et les Tunnels. Ce système ayant le défaut d'être trop simple et donc **répétitif**, sa consistance interne s'appauvrit. L'objectif de concevoir à un système **plus organique** m'a emmené dans une **seconde recherche** de Level Design pour *Drive the Rocket*.



Comparaison entre un chunk au départ et le même chunk plus tard dans le jeu.

En étirant simplement les parties horizontales, l'espace entre les obstacles s'agrandit.

2. Level Design : itération 2

a. Présentation

Cette seconde recherche de Level Design pour Drive the Rocket fut guidée, en plus de l'aspect **organique**, par l'idée d'introduire des **embranchements** durant le niveau, donnant un choix au joueur entre plusieurs ramifications.

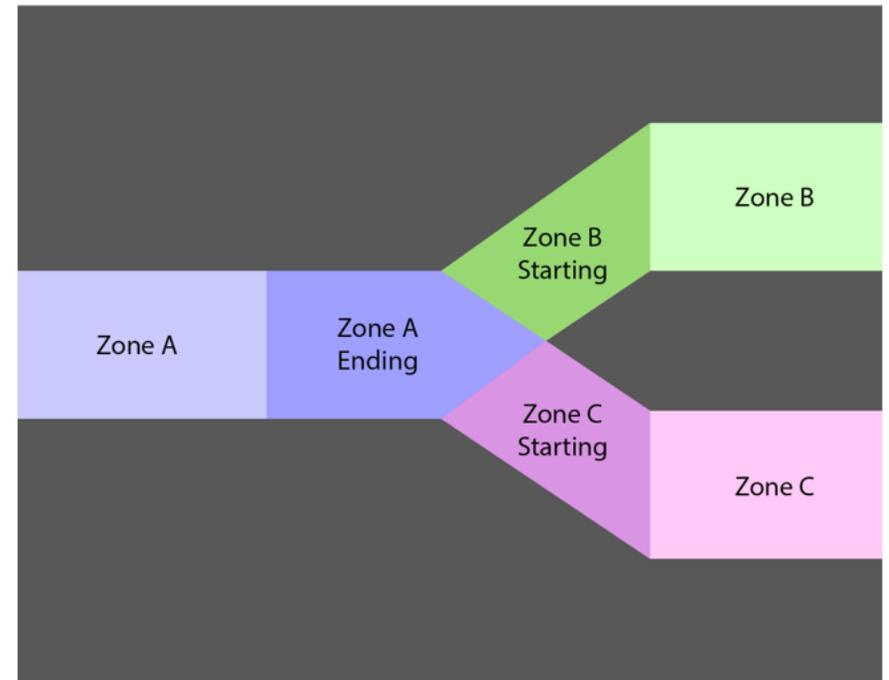
Introduire un nouveau type de choix dans un jeu appelle systématiquement une autre question : comment rendre tous ces choix **équivalents** et **intéressants** ?

Partant de cette problématique, j'ai abouti à un système dans lequel chaque embranchement propose un **type de challenge** différent. Le joueur choisit celui qu'il souhaite affronter en s'engageant dans l'un ou dans l'autre.

Les niveaux sont divisés en différentes zones. Tous les 400 mètres, un embranchement présente au joueur **deux nouvelles zones**. Chaque zone propose un **type d'obstacles** bien spécifique durant toute sa traversée : aucune zone n'est plus simple ou plus difficile qu'une autre, ce sont simplement différents challenges.

Ce découpage du niveau en zones possède également un fort **potentiel narratif**.

En attribuant à chaque zone une **thématique** liée aux obstacles qu'elle propose, on peut étendre l'univers du jeu global par l'ajout de **narration environnementale***.



Mockup d'un embranchement dans un niveau de Drive the Rocket

b. Conception des différentes zones

Afin de proposer des challenges différents lors de la conception de chaque zone, j'ai développé un outil simple groupant deux aspects importants des niveaux de *Drive the Rocket* : **l'anticipation des obstacles** et **l'utilisation du boost**.

- Anticipation des obstacles : des obstacles **aléatoires** sont **plus difficiles** à anticiper que des obstacles réguliers ou de simples murs
- Utilisation du boost : plus le niveau est en **ligne droite** et les obstacles sont **petits**, plus le joueur peut **profiter du boost** (déclenché lorsque l'avatar longe un mur) sans risque ni interruption

Niveau	Favorable au boost	Peu favorable au boost
Anticipation facile		Challenge de type 1
Anticipation difficile	Challenge de type 2	

Nous pouvons ainsi observer en bleu deux **combinaisons** de challenges différents et **équilibrés**, nommés « type 1 » et « type 2 » (les cases proposant une combinaison de challenges trop faciles ou trop difficiles sont grisées). Cela m'a permis de concevoir **trois zones** grâce à ces combinaisons.

Zone de Laboratoire :

- Cette zone est relativement **droite**, les obstacles sont principalement situés **sur les murs**
- Le joueur est encouragé à y **utiliser le boost**, les obstacles étant simples à éviter bien que difficiles à anticiper (challenge de type 2)
- Cette zone est présentée comme étant un **ancien laboratoire**, dans lequel le joueur peut voir certains éléments narratifs (par exemple des tags) concernant l'histoire des différentes Hordes



Schéma de la zone de Laboratoire

Zone de Tunnel :

- Zone à forme très **erratique**, elle ne présente cependant **pas d'obstacles** autres que les murs
- Le joueur est encouragé à rester **proche du centre** car les murs changent rapidement d'angles, ce qui les rend difficile à longer proprement (challenge de type 1)
- L'aspect narratif présente la zone comme étant un **tunnel**, probablement creusé par les monstres souterrains



Schéma de la zone de Tunnel

Zone des Portails :

- Zone relativement **droite** possédant des **obstacles réguliers** en forme de porte
- Le joueur n'est pas encouragé à prendre le boost, les portails étant d'autant plus **difficiles à éviter** lorsque le joueur longe le mur (challenge de type 2)
- Les obstacles sont ici présentés comme de gigantesques portails, protégeant probablement quelque chose ...

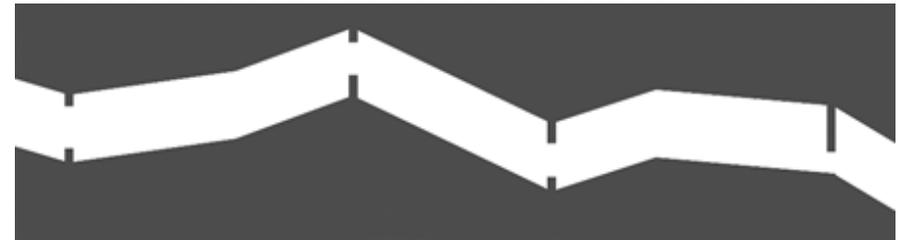


Schéma de la zone des Portails

Ces trois zones proposent ainsi différents challenges au joueur. Face à un embranchement, celui-ci pourra faire son choix sur ceux qu'il souhaite affronter.

c. Adaptation à la vitesse

La question d'adaptation du jeu à la vitesse croissante reste encore d'actualité concernant ce nouveau système de Level Design.

Durant la création des **paramètres** nécessaires à la génération procédurale, je me suis rendu compte que ceux-ci pouvaient justement nous servir à **adapter le jeu** en fonction de la vitesse.

Regardons tout d'abord comment une zone de Tunnel est générée :

- La zone est créée par succession de **tronçons**
- Chaque tronçon est défini par 3 paramètres : la **direction**, l'**angle** et la **longueur**

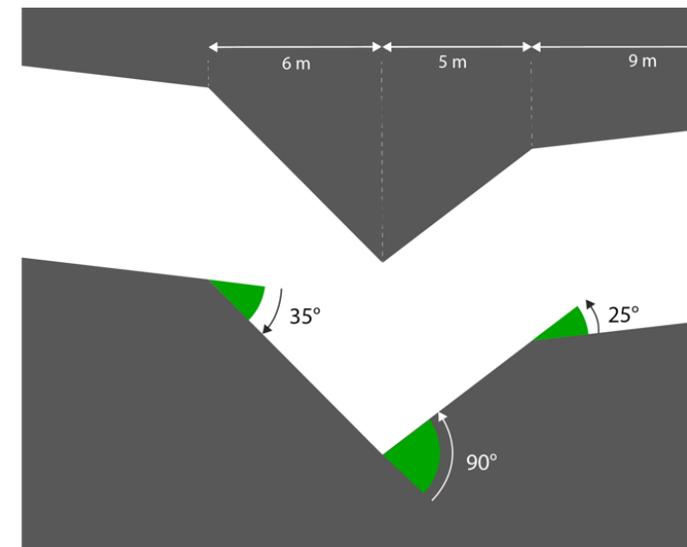
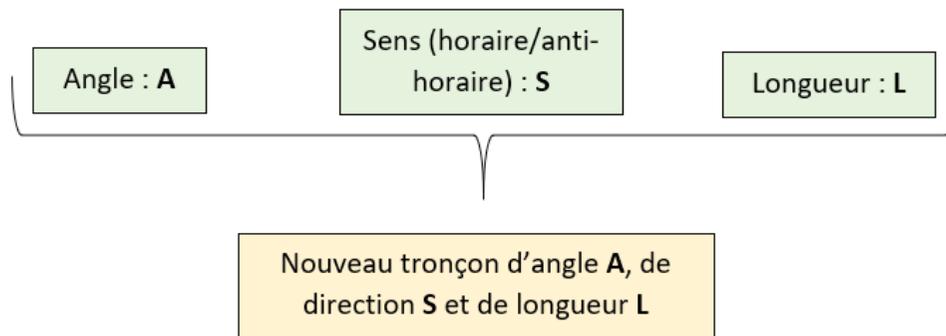


Schéma représentant les différents paramètres lors de la génération d'une zone de tunnel

L'angle et la longueur sont choisis aléatoirement dans des **ensembles** tel que ceux-ci :

Paramètre	Valeur
Angle	[25°, 50°]
Longueur	[3m, 10m]

En **modifiant** la valeur de ces ensembles, nous pouvons **accentuer les angles** et **allonger les tronçons** durant de jeu. Pour adapter le Level Design à la vitesse croissante, il suffit donc de **modifier les paramètres** rendant le jeu impossible à grande vitesse.

Par exemple, concernant la longueur des tronçons, elle peut **s'agrandir peu à peu** suivant les valeurs suivantes.

Paramètre	Valeur à 0m	Valeur à 1000m
Longueur	[3m, 10m]	[8m, 15m]

C'est donc par l'intermédiaire de ces paramètres que nous pouvons adapter le jeu à l'augmentation de la vitesse.

Conclusion sur l'itération 2 :

Ce nouveau système, bien que **plus complexe** à mettre en place, permet cependant de répondre aux problématiques en termes de **coûts de production**.

L'offre d'un **choix** concernant les différents challenges rend cette itération plus **variée** que la précédente ainsi que bien plus consistante intérieurement.

L'ajout d'une **thématique** pour chaque zone permet également de **renforcer l'intégration** de *Drive the Rocket* à l'univers du jeu principal, ce qui participe à sa consistance externe.

Après quelques modifications mineures, l'équipe a validé ce modèle.

3. L'intégration au jeu principal

Dernière étape dans la conception de *Drive the Rocket* : les **points de liaison** entre le gameplay secondaire et le jeu principal.

Afin de rendre *Drive the Rocket* **harmonieux** dans le paysage de *Ape o'calypse*, ces points sont une partie importante dans notre quête de consistance pour le gameplay secondaire.

Après la question de l'**accès** au gameplay secondaire par le jeu principal, nous verrons les solutions possibles pour gérer le **sentiment d'obligation** dans *Drive the Rocket* une nouvelle manière pour *Drive the Rocket* et *Ape o'calypse* de **s'influencer mutuellement**.

a. L'accès au gameplay secondaire

La question de l'accès au gameplay secondaire fut déjà partiellement résolue lors du choix du concept et de l'apport narratif de Franck.

Les **trappes**, servant narrativement de porte d'entrée et de sortie aux monstres des souterrains, sont un excellent moyen d'**ancrer** le gameplay secondaire dans le jeu principal : le **passage** de l'un à l'autre se faisant par un élément **diégétique***, le lien entre les deux est renforcé.

Afin de rendre la **recherche** des trappes intéressantes, celles-ci sont placées de manière **aléatoire** sur la carte pendant une **durée déterminée**, ce qui permet de changer leur position régulièrement.

Leur quantité sera en fonction de la carte, l'objectif étant que le joueur puisse en trouver une en maximum **quelques secondes**.

De cette manière, la recherche de trappe fait partie du plaisir de jeu, sans rendre le tout laborieux ou pénible.



*Mockup d'un écran d'interface
lorsque le joueur clique sur une trappe*

Néanmoins, il s'est également trouvé nécessaire d'avoir un moyen d'**accéder** au gameplay secondaire directement **depuis la ville** du joueur, ce pour des raisons extérieures.

J'ai effectué une première proposition : un bouton « Trouver une trappe » est présent dans l'**interface** d'un des bâtiments de la ville (par exemple, le Laboratoire).

En cliquant dessus, la caméra se **déplace directement** vers la trappe la plus proche et ouvre l'interface qui lui est associée.

Une telle solution rendrait cependant le processus de recherche de trappe **caduque**, le joueur ayant accès à une solution plus simple pour accéder à *Drive the Rocket*.

Une seconde proposition a alors été acceptée : une trappe est **déjà présente** dans la ville du joueur. Par contre, le **challenge** qu'elle propose est **plus simple** et les récompenses y sont moins importantes.

Cela permet ainsi d'encourager le joueur à chercher une trappe à l'extérieur tout en lui laissant la possibilité d'avoir un accès direct au gameplay secondaire.

Localisation de la Trappe/ Paramètre	Ville	Extérieur
Difficulté	Simple	Normale
Récompense	Qualité basse	Qualité normale

b. Le sentiment d'obligation

Gérer le sentiment d'obligation dans un SLG n'est pas chose facile lorsque les joueurs sont récompensés constamment.

Afin de limiter les avantages que peuvent procurer le gameplay secondaire, j'ai tout d'abord proposé de **ne pas récompenser** le joueur par des ressources à la fin d'un niveau de *Drive the Rocket*.

Cependant, c'est une **solution impossible** compte tenu du format du jeu mobile free-to-play et du SLG, et il m'a fallu tout de même intégrer un système de récompenses.

J'ai alors proposé une mécanique supplémentaire pour **encourager** les joueurs à arrêter *Drive the Rocket* au-delà de leur envie, sous les conseils de Maëlle.

Une trappe à l'**extérieur de la ville** ne permet de jouer au gameplay secondaire que **trois fois maximum**, après quoi elle disparaît et le joueur doit en **chercher une autre** s'il souhaite continuer (la trappe à l'intérieur de la ville n'a pas besoin de cette limite puisque les récompenses y sont par défaut moins importantes).

L'avantage de cette mécanique, en poussant le joueur hors du jeu, est de lui **demander implicitement** s'il souhaite continuer ou non à jouer (par l'intermédiaire de la **nouvelle recherche** de trappe).

De plus, comme le joueur sera à nouveau dans le jeu principal, sa situation aura peut-être évoluée : la réception un message personnel ou une nouvelle construction à lancer.

c. La compétition : un enjeu supplémentaire

Jusque-là, l'enjeu du gameplay secondaire reste très limité puisque sa réussite n'a aucune autre conséquence sur le jeu principal que les récompenses que recevra le joueur.

La volonté d'ajouter un **lien gameplay** entre les deux jeux m'a amené à concevoir un système de **compétition**, ne donnant aucun nouvel avantage aux joueurs mais donnant un intérêt supplémentaire au gameplay secondaire.

En effet, le **meilleur joueur** de chaque Horde à Drive the Rocket verra son Astronaute **partir dans la fusée** lors de son décollage !

Déroulement de la compétition :

- Les joueurs sont classés en fonction de leur **meilleur score**.
- Les joueurs participent d'abord à une **pré-sélection** à l'intérieur de leur **Alliance** (un joueur sans Alliance ne peut donc pas participer).
- Lorsque qu'un des cinq trains entre dans le **dernier relais**, le meilleur joueur de chaque Alliance est sélectionné pour la **finale**.
- Lorsque la fusée **décolle**, le meilleur joueur de tous les finalistes verra son Astronaute **aller dans l'espace** !

Cette méthode permet de donner un **premier objectif concret** pour chaque joueur : ils doivent d'abord **affronter leur Alliance** avant d'affronter les autres joueurs.

De plus, cela permet de confronter sur le même plan des joueurs plus ou moins bons (toutes les Alliances n'ayant pas la même puissance).

Note : afin de garantir une équité entre tous les joueurs, le score d'un niveau de Drive the Rocket lancé depuis la ville n'est pas pris en compte puisque la difficulté du jeu y est réduite.

III/ SYNTHÈSE

A - Tap4Fun : un cadre professionnel

Mon travail en tant que stagiaire à Tap4Fun Paris m'a plongé dans un cadre et une exigence professionnelle que j'avais hâte d'expérimenter, étant ainsi au plus proche du métier de Game Designer.

J'ai pu y découvrir la vie en entreprise et entretenir de véritables rapports professionnels avec mes collègues.

La petite taille du studio m'a permis d'observer quotidiennement tous les corps de métiers en activité et d'être au cœur de la création d'un jeu vidéo de cette envergure.

L'accompagnement de ma tutrice, Maëlle Holtzer, fut déterminant durant mon apprentissage.

Me guidant dans mes réflexions et questionnements de design, elle m'a notamment appris à me projeter plus en amont dans l'esprit des joueurs afin d'y trouver les solutions.

Bien que mon travail n'ait pas pu réellement se poursuivre jusqu'au prototypage de Drive the Rocket, mon expérience chez Tap4Fun s'est révélée véritablement enrichissante.

B - Tap4Fun et l'enseignement de Bobigny

Si mon stage à Tap4Fun était le boss final d'un jeu vidéo, la Licence Professionnelle de Game Design et Level Design en serait l'armurerie.

Les connaissances acquises en cours d'années m'ont chacune aidées lors de la conception d'un gameplay secondaire pour *Ape o'calypse*.

Les cours de Design et Monétisation m'ont permis d'appréhender plus rapidement les mécaniques d'un SLG ainsi que son système économique, ce qui a facilité mon intégration au projet.

Les connaissances apportées par les cours de Rational Design et de Level Design ont guidé la conception d'un système de génération procédurale, tout en me permettant d'être plus créatif.

Enfin, mes schémas et mockups ont été réalisés à l'aide d'outils découverts en cours d'Infographie, sans lesquels mes idées auraient eu des difficultés à être représentés graphiquement.

Les compétences acquises grâce à l'université se sont donc révélées déterminantes dans la conception d'un gameplay secondaire pour *Ape o'calypse*.

C - L'intérêt de l'expérience personnelle

Un des bagages qui m'a été le plus efficace durant mon travail sur Drive the Rocket fut mon expérience personnelle.

Mes trois années en tant que musicien professionnel avant d'entrer à l'IUT de Bobigny m'ont aidé à respecter les exigences professionnelles et les deadlines, ainsi qu'à accomplir mes missions en autonomie.

Les nombreuses game jams auxquelles j'ai participé ont quant à elles prouvées leur utilité en ce qui concerne la communication, le travail d'équipe et l'anticipation des difficultés de programmation.

Ces apprentissages se sont révélés d'une grande aide, en particulier lors de la recherche des différents pitches.

Les connaissances en psychologie cognitive exposées par Daniel Kahneman dans *Think Fast and Slow*, ainsi que les processus de recherches décrits par Derek Yu dans *Spelunky* ont également eu leur place lors des phases de recherche et de conception de Drive the Rocket.

Enfin, le visionnage de nombreuses rediffusions de la Game Developers Conference a peu à peu enrichi ma culture vidéoludique et a participé au développement de mon intuition de game designer, se révélant essentielle tout au long du processus.

D - Félix Belthoise : Game Designer

La combinaison de l'université et du stage en entreprise m'a donné une large perspective sur métier de Game Designer.

Entre la prise en main des outils essentiels et la conception de projets étudiants puis professionnels, ces multiples expériences m'ont permis de concrétiser mon désir de mettre mes connaissances en pratique.

La confrontation aux difficultés et problématiques du métier de Game Designer en situation a ainsi confirmé ma volonté de continuer sur la cette lancée : j'ai donc décidé de poursuivre ma route en tant que Game Designer professionnel dès l'année prochaine !

CONCLUSION

Au fur et à mesure de la conception de Drive the Rocket, j'ai cherché à rendre ce gameplay secondaire consistant avec lui-même et avec le jeu principal en puisant dans mes acquis universitaires et mon expérience personnelle.

Le design d'un gameplay secondaire ne peut se faire en-dehors du jeu principal.

Plonger le joueur dans une nouvelle expérience sans le sortir de la précédente nécessite d'approfondir le gameplay principal par le gameplay secondaire, et non d'être une conception indépendante.

Cependant, ce gameplay secondaire doit également rester consistant intérioritément sous risque de s'effiloche, se perdre et s'oublier dans le jeu principal.

Ainsi, mon travail s'est concentré sur la recherche de lien d'accroche entre les deux jeux sans les rendre dépendants l'un de l'autre.

Mais le travail effectué n'est que la première partie de la mise en forme de Drive the Rocket.

Un prototypage est encore nécessaire et renouvellera toutes les questions de conception tout en en posant de nouvelles.

Le challenge est-il assez varié ?

Le système de génération procédurale suffisamment organique ?

Comment les phases de tests influent-elles sur la conception d'un jeu vidéo ?

Cette dernière question trouvera sa réponse lors de l'intégration de Drive the Rocket dans Ape o'calypse, dont la sortie n'est plus pour longtemps !

MERCI !

ANNEXES

Bibliographie et Webographie

- ***Gödel, Escher, Bach*** (Douglas Hofstadter, 1979)
- ***The Design of Everyday Things*** (Don Norman, 1988)
- ***Thinking, Fast and Slow*** (Daniel Kahneman, 2011)
- ***Spelunky*** (Derek Yu, 2016)
- ***Rules of Play*** (Katie Salen & Eric Zimmerman, 2003)
- ***Gamasutra*** (<https://www.gamasutra.com/>)
- ***Rock Paper Shotgun*** (<https://www.rockpapershotgun.com/>)
- ***Retransmissions GDC*** (https://www.youtube.com/channel/UC0JB7TSe49lg56u6qH8y_MQ)
- ***Fandom Wikis*** (<https://www.fandom.com/explore>)

Lexique

MMO : abbréviation de «Massively Multiplayer Online Game» (jeu en ligne massivement multijoueur), désignant un genre de jeu faisant participer un très grand nombre de joueurs simultanément.

Free to play : type de jeu vidéo à l'utilisation gratuit, incluant souvent une boutique virtuelle avec options d'achats afin de se rentabiliser

Boucle de gameplay : schéma d'actions répétées par le joueur durant une partie ou la totalité du jeu

Hard Currency : monnaie virtuelle faisant un lien entre l'argent réel et les éléments du jeu disponibles à l'achat

Dungeon Crawler : genre de jeu vidéo de rôle dont le gameplay met l'exploration de donjons en avant

Pitch : présentation brève d'un jeu et de ses éléments les plus caractéristiques

Génération procédurale : création automatique de contenu suivant un ensemble de règles (une « procédure »)

Design Rationnel : Décomposition logique d'un principe macro pour en maîtriser efficacement les rouages micro et optimiser méthodiquement leur conception et intégration (définition : Anthony Deutsch)

Narration Environnementale : narration racontée par l'environnement ou par les éléments que le joueur rencontre

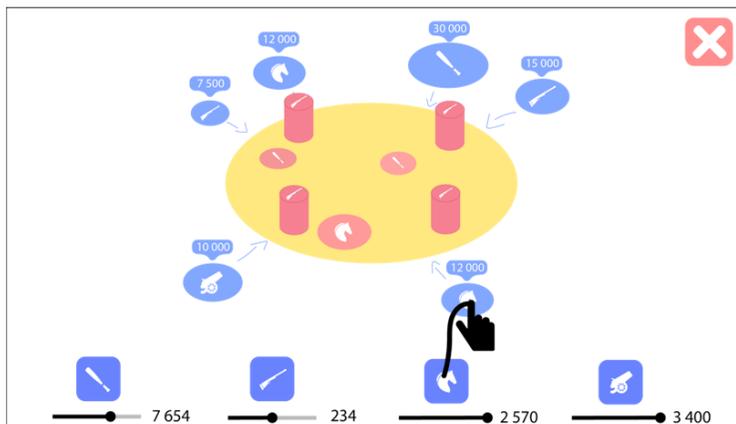
Diégétique : un élément diégétique est un élément présent physiquement dans l'univers auquel il se rattache. S'oppose aux éléments extradiégétiques.

Pitches - Lien gameplay

Strategy : *Attack barbarian cities in real-time !*

Le joueur doit détruire un village barbare. Avant que le combat ne commence, il peut placer ses troupes en-dehors de la ville. Lors de l'attaque, les troupes avancent vers la ville et attaquent ce qu'il y a directement en face. Le joueur peut toujours ajouter des troupes supplémentaires durant le combat, mais s'il les perd toutes, c'est game over !

Inspiration : *Clash of Clans* ([Supercell](#), 2012)

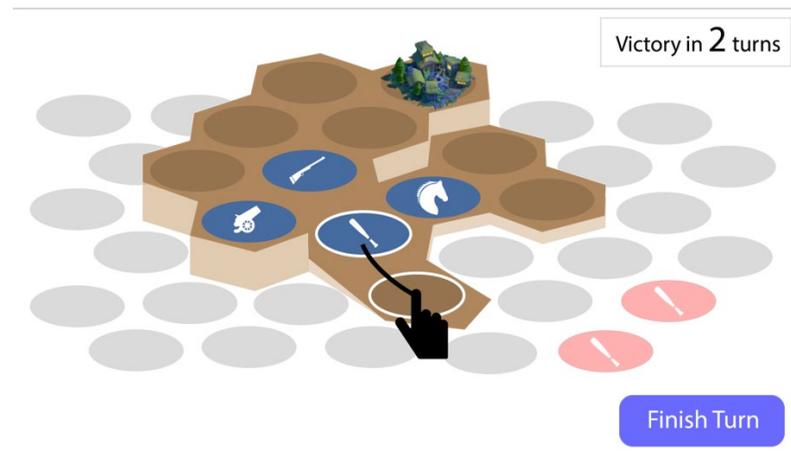


Mockup pour Strategy

Bad Breach / Into the North : *Defend villages against attackers !*

Dans ce jeu de stratégie en tour par tour, le joueur doit protéger un village des attaques barbares durant un certain nombre de tours. Pour ce faire, il n'a que quelques troupes avec lesquelles il va devoir se défendre ...

Inspiration : *Bad North* ([Plausible Concept](#), 2018)
Into the Breach ([Subset Games](#), 2018)

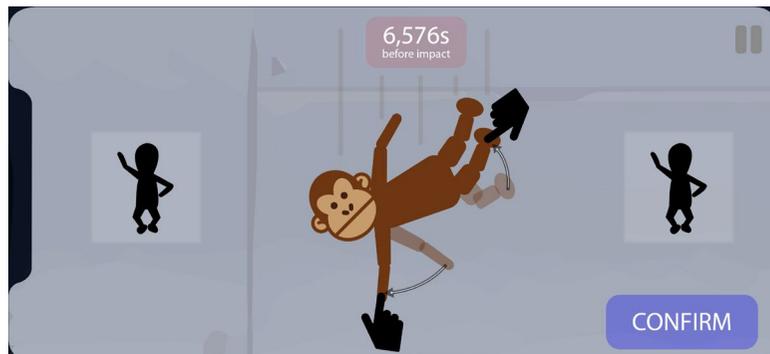


Mockup pour Bad Breach

Pitches - Lien des valeurs de *Ape o'calypse*

Monkey See, Monkey Do : *Inside an aerodium-like room, the astronaut trains to move in a low-gravity environment. With style, it's always better.*

À l'intérieur d'un Aérodiom, un astronaute tombe lentement mais sûrement vers le sol ... Pour reprendre de la hauteur, le joueur peut bouger les membres du singe pour lui donner une pose spécifique, indiquée par des ombres sur les côtés de l'écran. Plus la précision est grande, plus l'astronaute reprendra de la hauteur !

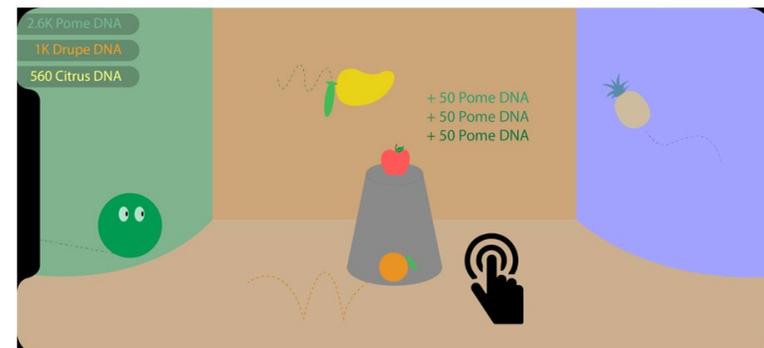


Mockup pour Monkey See, Monkey Do

Tap Tap Fruit : *Scientists are secretly searching a way to discover Bananas by mixing fruit DNA ...*

Dans ce Tap Tap game, le joueur peut produire de différents ADNs de fruits en « tapant » sur le générateur d'ADN. Il peut ainsi créer de nouveaux fruits en mixant les ADNs pour les exposer dans sa collection. Mais que se passe-t-il si on introduit un peu d'Uranium dans un fragment d'ADN ? Les fruits commencent à se comporter de manière étrange ...

Inspiration : *Tap Tap Fish (FLERO Games)*

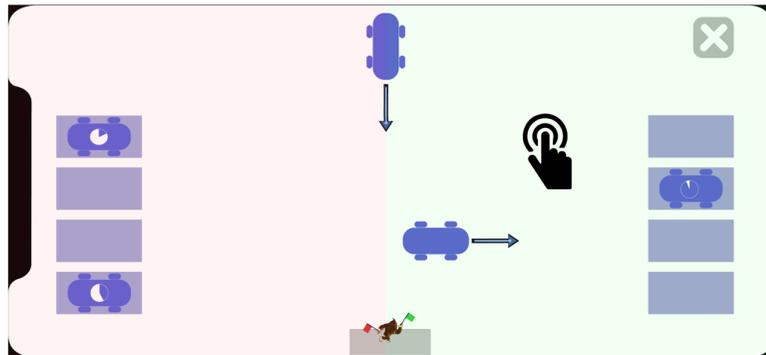


Mockup pour Tap Tap Fruit

Rhythm game : *Help convoys unloading their cargo by showing them where to go.*

Le joueur doit guider chaque convoi à une place vide : appuyer sur un côté de l'écran indiquera au prochain convoi dans quelle direction il doit se rendre.

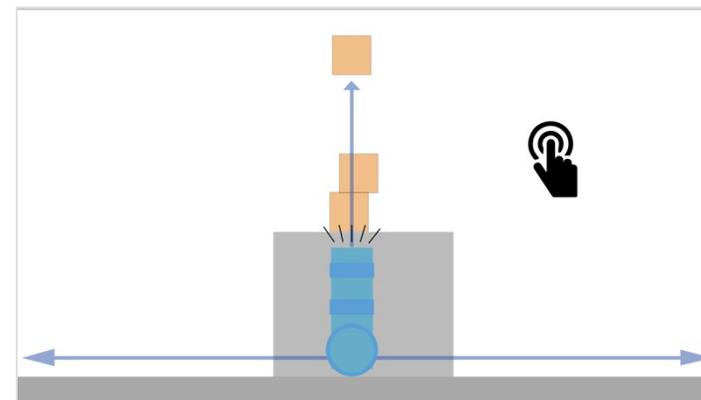
Les convois partent et libèrent leur place au bout de quelques secondes. Si des convois se touchent, ils s'écrasent et c'est game over.



Mockup pour Rhythm Game

Tower construct : *To test how materials behave in high altitudes (and fall down from it), monkeys throw washing machines in the air to build the highest constructions possible*

Un canon se déplace automatiquement de gauche à droite. Un simple « tap » sur l'écran permet de lui faire tirer une machine à laver : le joueur doit les empiler sans les faire tomber pour aller le plus haut possible. Si la tour s'écroule, c'est game over !



Mockup pour Tower Construct

Pitches - Lien narratif

Scrap! : *Recycle the wasted tech and create useful weapons for your soldiers !*

Dans ce Linker, le joueur crée des objets en liant les matériaux du même type grâce à un simple tracé du doigt. En liant plus de matériaux, il crée des objets de meilleure qualité. Cependant, il faut construire tous les objets requis dans le nombre de tours indiqués !



Mockup pour Scrap!

Tchoo Tchoo : *Build your test locomotive and race others to show you engineering prowess !*

Le joueur construit une locomotive à partir d'objets très divers. Il peut ensuite la faire entrer en compétition dans des courses contre les créations des autres joueurs. Certains objets spéciaux peuvent être ajoutés et utilisés durant la course !

Inspiration : *C.A.T.S (ZeptoLab)*
King of Thieves (ZeptoLab)



Mockup pour Tchoo Tchoo

Push the Lever : *Tap to get energy to develop your city*

Appuyer sur l'écran booste le générateur, produisant des unités d'énergie que le joueur peut collecter. Les composants du générateur peuvent être améliorés avec des unités d'énergie afin d'augmenter sa production d'énergie. Le joueur peut également dépenser l'énergie pour produire une Impulsion, boostant tous les bâtiments de sa ville proches du générateur !

Inspiration : *Tap Titans* (Game Hive Corporation, 2014)

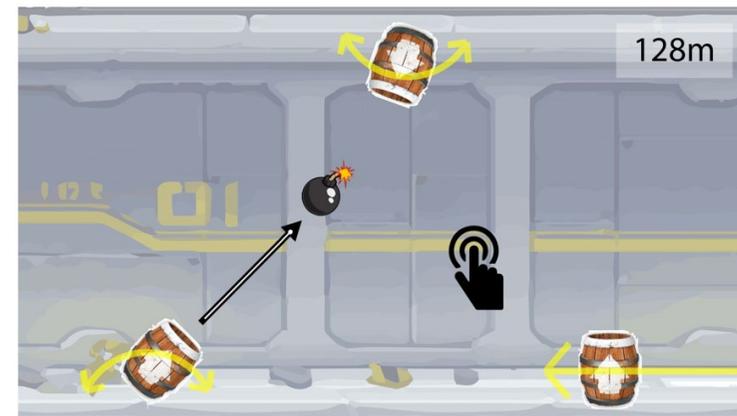


Mockup pour Push the Lever

Boomball : *A bomb has been lighted by mistake, the emergency system needs to get it out of the way !*

Le jeu défile automatiquement vers la droite. Le joueur doit transporter une bombe le plus loin possible à l'aide de tonneaux en mouvement et en rotation. Appuyer sur l'écran envoie la bombe dans la direction que pointe le tonneau ; en touchant un nouveau tonneau, elle rentre directement dedans. Si la bombe sort de l'écran ou touche les murs, elle explose et c'est game over !

Inspiration : *Donkey Kong Country* (Rare, 1994)

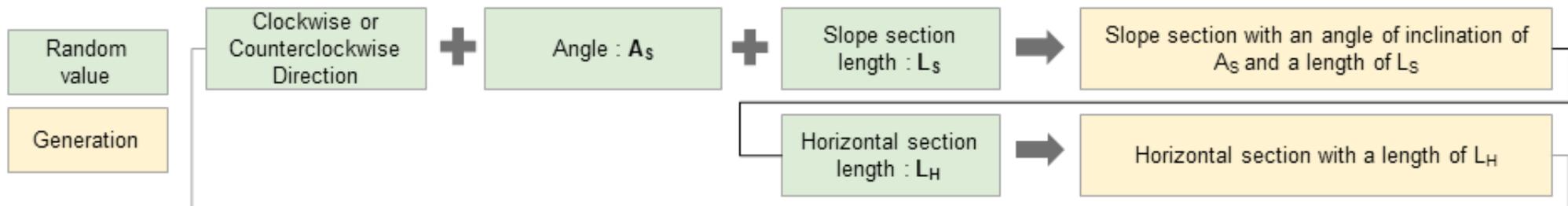


Mockup pour Boomball

Génération procédurale - Lab Zone

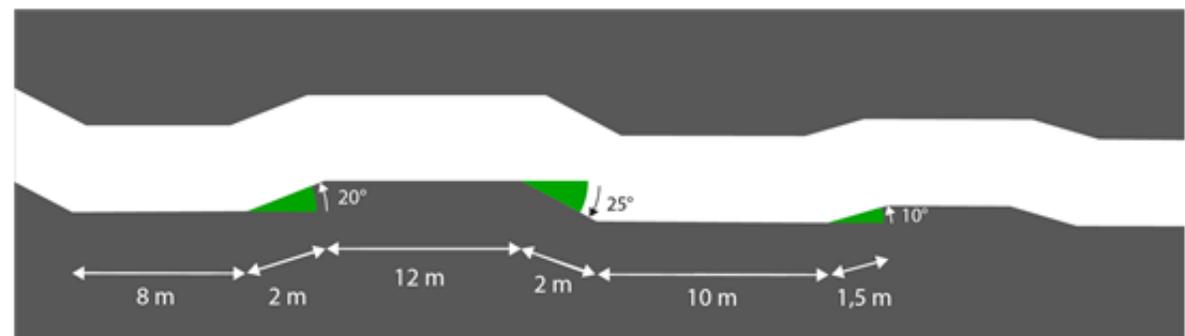
Voici un extrait du document de génération procédurale pour le Level Design de *Drive the Rocket* :

- The Lab zone is generated in 2 steps : the walls layout and the obstacles placement.
- Step 1** : the walls are defined by 3 characteristics : Direction, Angle and Length



- This way, the tunnel alternates between short slope sections and longer horizontal sections.

Angle range	A_S	$[10^\circ, 35^\circ]$
Slope section length range	L_S	$[1.5\text{m}, 3\text{m}]$
Horizontal section length range	L_H	$[8\text{m}, 14\text{m}]$

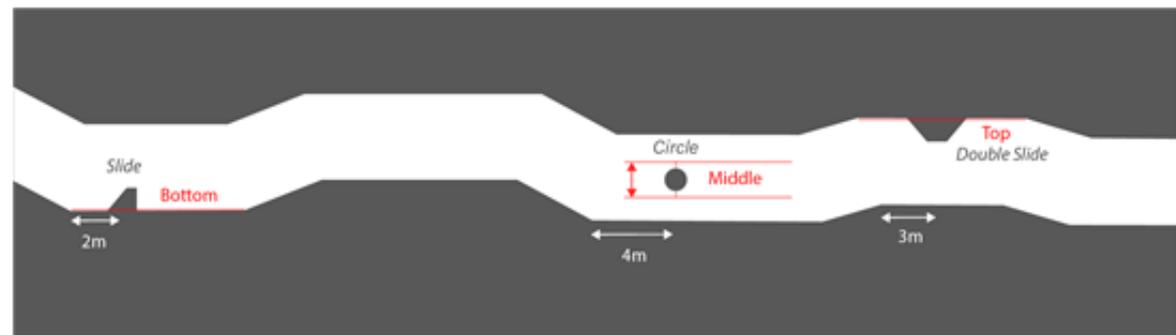


For information purposes only, exact values
TBD

- **Step 2** : for each **horizontal section**, there is a chance an obstacle appears on the bottom, the middle or the top of this section (different probability for each line). This probability can change over time (more and more obstacles)
- If an obstacle appears, it has **less chance** to be on the same line (bottom/middle/top) as the preceding obstacle
- There can be different kind of obstacles for each line (in the image : a **Slide**, a **Circle** and a **Double Slide**). They each have a different probability to be the generated obstacle.
- Obstacles are placed at a random distance from the beginning of the section

Starting appearance probability	30% for each horizontal section
Bottom & Top	30% each
Middle	40%

For information purposes only, exact values
TBD



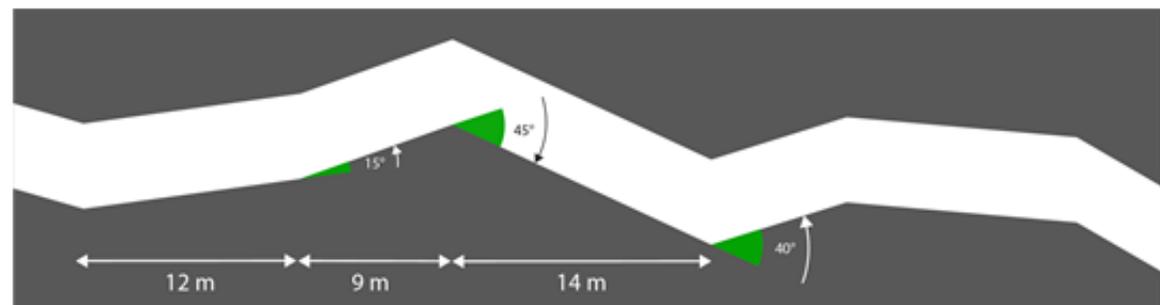
Génération procédurale - Gates Zone

- This zone is generated in 2 step : the walls layout and the gates placement.
- Like the Lab zone, the walls are defined by 3 characteristics : Direction, Angle and Length



- There is a **minimum and a maximum angle** for the sections, as well as a **maximum angle difference** between a previous and a new section.

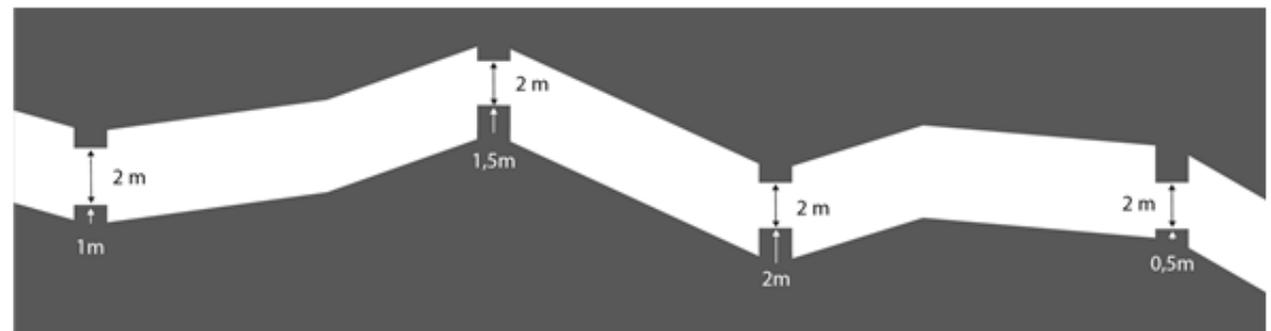
Angle range	[10°, 35]
Max angle difference with previous section (game start)	45°
Section length range	[8m, 14m]



For information purposes only, exact values
TBD

- **Step 2** : between each sections, there is a chance a gate-shaped obstacle appears. This probability **doesn't change** over the course of the game (as this would be impossible to use the boost after some time ; however, other things will change to higher the difficulty).
- Each gate has an "open space" which size is fixed (= not random) but **reduced over time**, until a theoretical minimum.
- The vertical position of this "open space" is randomly chosen in a range.

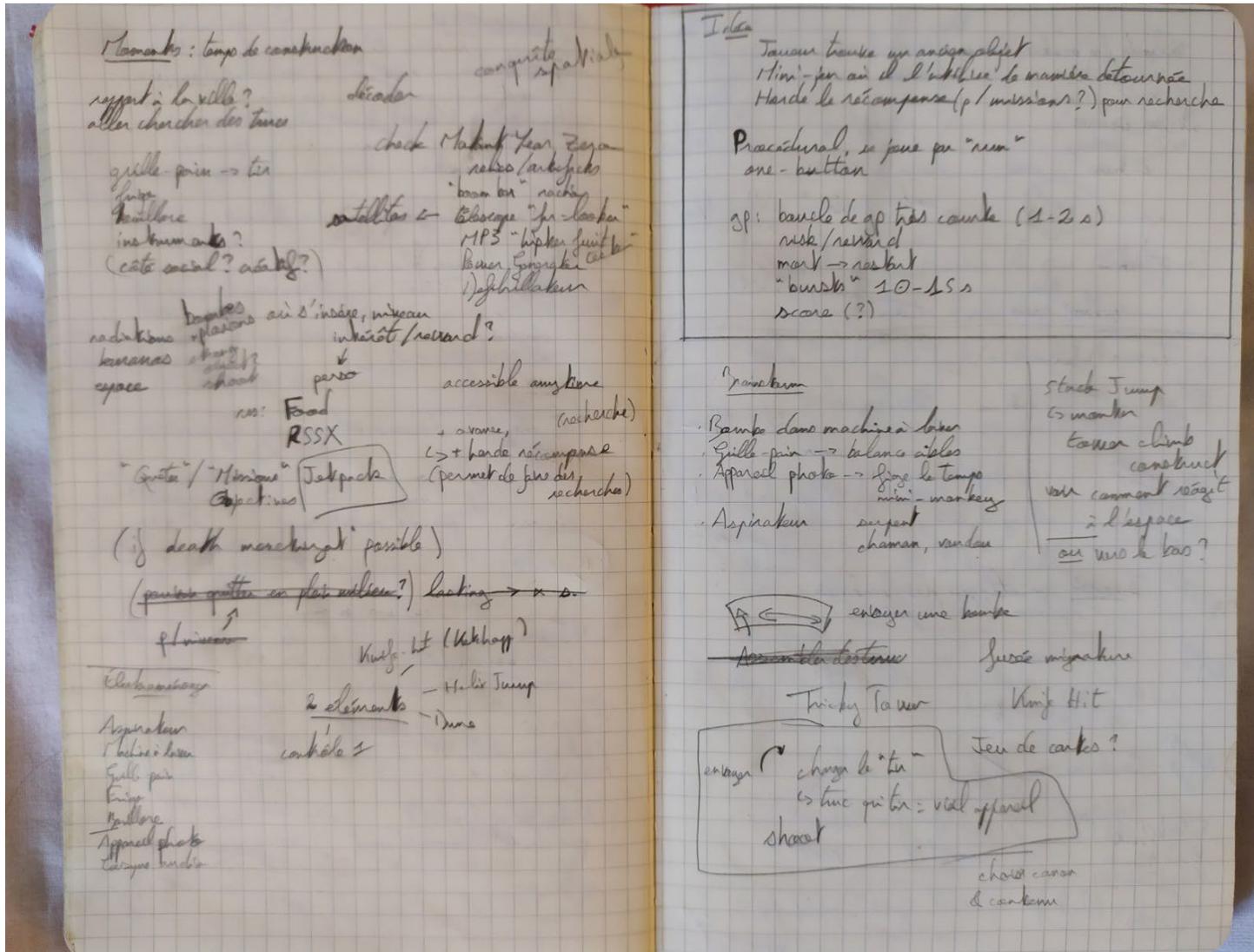
Appearance probability	50%
Starting open space size	2,5m
Vertical position range	[1m, ceiling - open space size]



For information purposes only, exact values
TBD

Carnet de notes

Voici un extrait de mon carnet de notes durant mes recherches à Tap4Fun :



Jap Titans:

Skills & mana
Arche de compétences

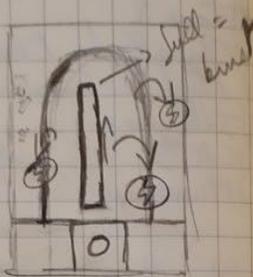
récompense
obéissance

basement de villa (généralisme)
↳ lettres / kanten

+ passer

meubles

force?



bass = bloquer

xp? topologie

catch?

pour rails

changer ~~jeux~~ du juel (baf) → pas infini

Entraînement → Δ false
Singes se croisent

relier à l'espace?

Joue monter un assemblage



rep, infini
& tout + bas
qu'une
juste

donner le + d'énergie à la rampe

shoot weapon

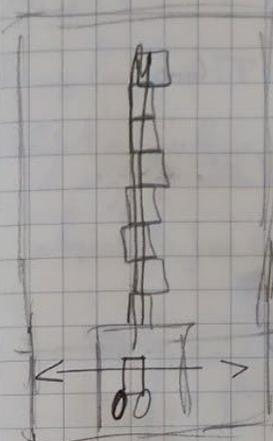
bombes dans machines à l'axe
tester la rétro force des matériaux
gros scrap metal



⊖ les XK etc...

Tower Control

≠ former des objets?



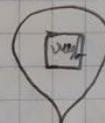
power-up au lancement
barré top or start →
check ticking tower objects

catch mécanique?
objets → effets

au canon au laser entrés
en "rythme", bouger la tour?



assemblage → doit bien se combiner



only the
top highest block
count

pile de garbage

passers?

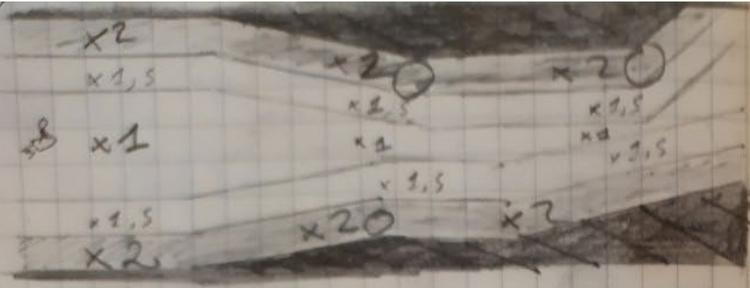


items centered

le joueur choisit le
bloc?

- o Stabilization
- o Drop la tour
- o Multiple bricks

timer
resp objet
tous les
blocs
bien placés

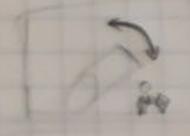


look of TCA

simulat^o? traverser un peu le côté fun

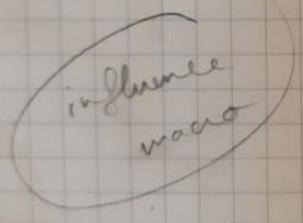
pas de la ville

food: backgrounds
+ villes 2D



by good?

chambers
jelka



with
more
sh
livable (1)

o what make the f blu keeping or losing speed boost?
o there influence

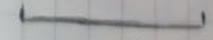
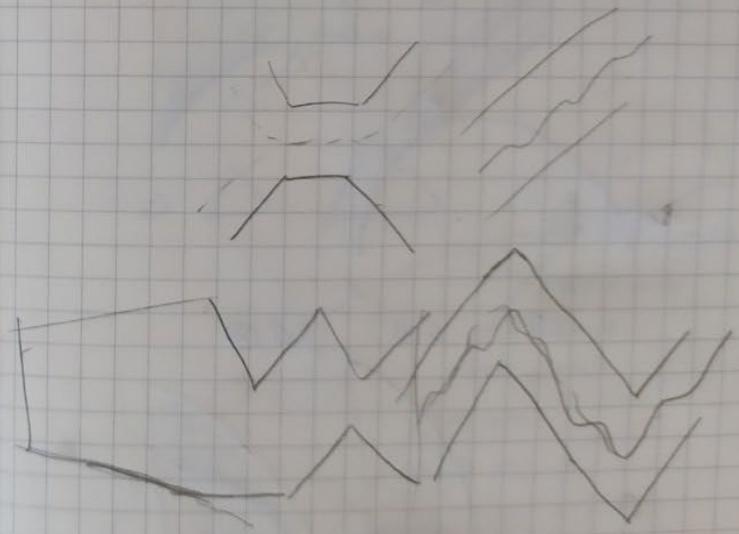
P1

P2

Speed boost

- + couple ^{lost}
- spreading
- more thrilling?
- short-term
- easy to learn / hard to master
- (when it begins to be too risky, allow to come back to a lower speed)

kept
long term



given
an input means
rather constant / change

boost? well
proximity

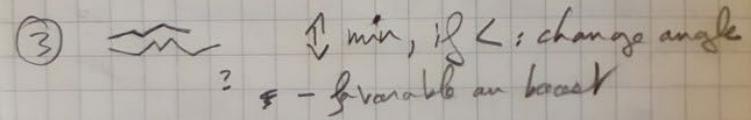
≠ ! proche en compte espacement
(chunks) mixer chunks &

quels types peuvent challenger le joueur?
favoriser le boss? → lignes droites

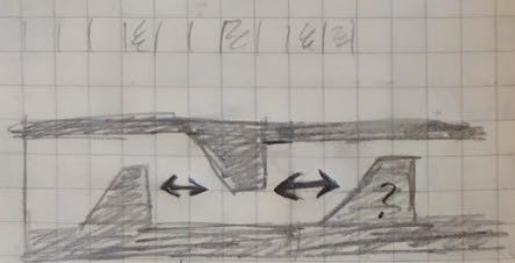
chunks: de + on + espaces → rythme



→ points



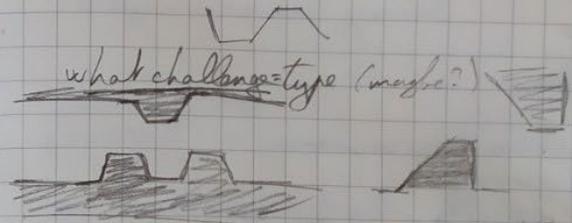
→ progressive difficulty



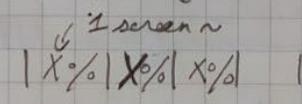
difficulty? number of obstacles

type → how to define?

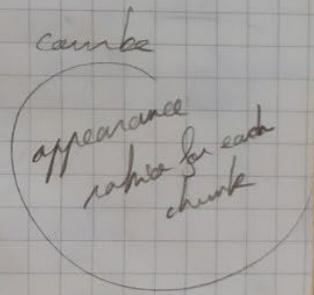
mana (LFO) rest/switch off



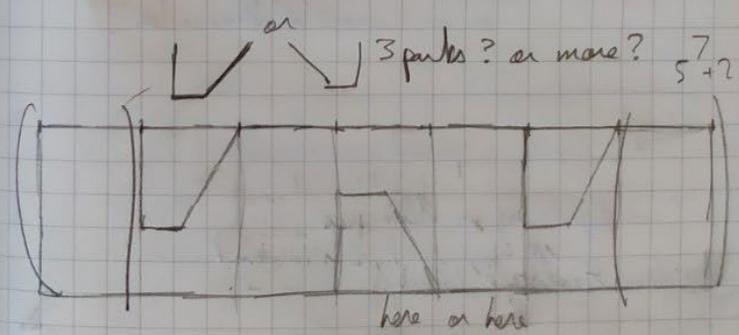
- o progressive
- o Chunks variations
- o Chunks difficulty



+ how to choose the chunk?



When there is a chunk:



- o each "part"/obstacle has a difficulty score (+ increasing /obscuring) multiplier
- separable (1/2 part)
- number of obstacles
- diff value