

# MAKE ME A WORLD

The title 'MAKE ME A WORLD' is written in a bold, dark green, rounded font. A green vine with a red apple and a green leaf is integrated into the text. The vine starts under 'MAKE', goes up under 'ME', and then loops down under 'A WORLD'.

## Game Design Document

Elliott Blacha  
Lise Calmelet  
Lena Ogulluk  
Charles Panet

2023–2024





# Sommaire

## OVERVIEW

### 7 – 10 Introduction

- Equipe
- Fiche d'Identité
- Intentions

### 11 – 30 Game Design

- **Processus Créatif**
- **Références Game Design**
- **3C**
  - Caméra
  - Controls
  - Character
- **Noyau System**
  - Noyau
  - Tendance
- **Situations de Jeu et Boucles de Prédictions**
- **Game System**
  - OCR
  - Boucle de Gameplay
  - Mécaniques
  - Schéma de Ventrice
  - Signs et Feedbacks
- **Flowcharts**

## 36 – 45 ● Game Design Document

### 37–41 3C

- Character
- Caméra
- Controller

### 42 – 43 ANALYSE

- Typologie de joueurs
- Matrice de Caillois

### 44 – 45 LEVEL DESIGN

## 46–59 ● Charte Technique

### 47 Challenge Technique

### 48–49 Algorithmes Utiles

- Algorithme de Terrain
- Algorithme de Caméra

### 50–59 Difficultés à surpasser

- Taille du Terrain
- Modification des liaisons du Terrain
- Optimisation



## 60–76 Charte Graphique

### 61–64 Inspirations générales

- Intentions
- Moodboard

### 65–67 Terrain

- Intentions
- Moodboard
- Processus Créatif

### 68–72 Balle & Objects

- Intentions
- Moodboard
- Processus Créatif

### 73–76 Menus

- Intentions
- Moodboard
- Processus Créatif

## 77 – 84 Charte Sonore

### 78–79 Intentions

### 80 References

### 81–84 Asset List

- UI
- Balle
- Univers

85	 <b>Organisation</b>
86	<b>Methode Agile et Trello</b>
67-88	<b>Communication &amp; Versionning</b>
89	 <b>Annexes</b>
90 - 93	<b>QA</b>
94	<b>Améliorations</b>
95	<b>Remerciements</b>



## Introduction

Make Me A World est un jouet développé dans le cadre de notre 2ème année à l'ICAN. Nous avons pour consigne de créer un jouet à partir d'un système noyau et d'une image générée par IA.



Un sisyphé jonglant avec des tigres-cubes. Giger Art Style.

## Equipe



**Charles Panet**

Lead Game Programming  
Game Design  
Direction Artistique



**Elliott Blacha**

Lead Game Art  
Game Programming  
Game Design  
Direction Artistique



**Lise Calmelet**

Lead Sound Design  
Game Programming  
Game Design  
Direction Artistique



**Lena Ogulluk**

Lead Game Design  
UI Design  
Level Design  
Direction Artistique



## Fiche d'identité

### Pitch

Déformez un monde de pâte à modeler et observez votre balle et son environnement évoluer dans ce jouet en 3D envoûtant et harmonieux qui vous replongera en enfance !

### Univers

L'univers de *Make Me A World* est un univers enfantin. Le joueur est directement transporté dans la tête d'un enfant jouant à la pâte à modeler grâce à son imagination.

C'est un univers rond, joyeux et envoûtant qui évolue au fur et à mesure que le joueur évolue et déforme celui-ci.

### Type de Jeu

Jouet basé sur l'exploration et la déformation de terrain

### Cible

Mid core Gamer appréciant les jeux d'exploration

### Support

PC (clavier et souris)

### Unique Selling Point

Déformer un terrain sur lequel se trouvent des objets affectés par la physique.

---

### Game Concept

*Make Me A World* est un jouet d'exploration et de déformation en 3D dans lequel le joueur sera immergé dans un monde enfantin cartoonesque empruntant un style «pâte à modeler» qu'il pourra déformer à sa guise. Le joueur pourra y apprendre que ses actions sur le monde ont des conséquences, notamment grâce à une balle roulant sur le terrain et à la faune et à la flore environnante.

## Intentions

Notre première intention a été d'imaginer un jouet dans lequel le joueur se sente libre, qu'il puisse faire ce qu'il veut avec les outils qu'on lui donne et cela de façon accessible, tout en ressentant une satisfaction visuelle et sensorielle élevée.

Nous nous sommes tournés vers les plaisirs ludiques des enfants et la déformation nous a semblé intéressante grâce à la forte sensation de liberté et de satisfaction donnée aux joueurs. Elle permet en effet au joueur d'exprimer sa créativité et l'ajout de la balle permet de gagner la sensation de glisse et de rapidité, apportant un autre aspect "satisfaisant" au jouet. Le joueur doit pouvoir se perdre, se détendre et rester «absorbé» dans le jeu, comme envoûté.

Pour se faire, nous avons opté pour un direction artistique douce et agréable, ronde et enfantine avec des couleurs peu saturées. Nous avons souhaité développer une direction artistique en rapport direct avec notre gameplay principal, et le style «pâte à modeler» nous a semblé parfait. Le principe de la déformation est donc mis en avant directement dans le visuel du jouet qui se complémente bien avec le gameplay très simpliste et facile à maîtriser de notre jouet.

Make Me A World est un jouet mettant en exergue la beauté de la déformation, montrant au joueur qu'il peut influencer plus que ce dont il a le contrôle direct.



# Game Design

## Processus Créatif

### Recherches du Système noyau

Partant de la consigne de faire un jouet à partir d'un système noyau, nous nous sommes penchés sur la création d'un système. Nos premières idées tournaient autour d'un système de tiles ou d'objets sur lesquels coulerait de l'eau et que le joueur placerait afin de créer une rivière tout en jouant avec la physique de l'eau.

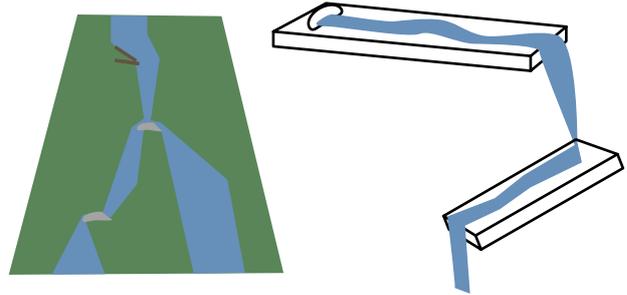


Schéma de nos deux premières idées de prototype

Nous avons véritablement été charmés par cette idée d'utiliser la physique pour déplacer des objets sans avoir un effet direct sur eux, et après plusieurs itérations de concept, nous nous sommes penchés sur l'archétype du circuit de bille.



Circuit à Billes en plastique

L'idée de construire tout un terrain et de laisser ensuite un objet libre sur celui-ci nous a plu et nous avons décidé de développer cette idée.

Nous nous sommes vite rendus compte que nous souhaitions donner au joueur le plus de liberté possible face au jouet et le fait d'avoir des morceaux pré-construits de terrain que le joueur allait emboîter nuisait grandement à ce sentiment de liberté.

L'utilisation d'une Brush modifiant directement le terrain pendant que l'objet est déjà dessus s'est révélé notre meilleure solution à ce problème.

Une fois le système principal trouvé, nous avons choisi d'itérer une première fois grâce au Terrain Tool, un outil permettant, tout comme notre concept, de déformer des terrains. Cette étape nous a permis de s'assurer que toute notre équipe portait la même vision du projet et de s'assurer de pouvoir commencer le prototypage dans la même optique.



## Processus Creatif

### Variations et développement du Concept

Nous avons ensuite cherché des variations de ce concept tel qu'avoir plusieurs balles au lieu d'une seule, un terrain qui se retourne quand la balle s'apprête à tomber ou encore des modifications de tailles et de formes de la balle.

Ces variations nous ont permis d'établir les points forts et les points faibles de notre système noyau, de les améliorer et les réduire.

Nous avons déterminé 3 points forts : La Liberté, la Simplicité et la Satisfaction et 3 points faibles : la Redondance, la Rejouabilité et la Lisibilité.

Plusieurs de ces variations ont finalement été gardées. Par exemple, le terrain que nous avons avant les variations était un simple terrain carré. Lorsque la balle tombait de celui-ci, elle réapparaissait en son centre et le joueur continuait de jouer.

Ce terrain limitait trop le joueur selon nous et réduisait cette sensation de liberté que nous recherchions dans notre concept. Nous avons donc itérer deux variations à ce problème : un terrain à deux faces qui se retournerait lorsque la balle tombe, permettant ainsi au joueur de continuer de jouer en voyant son terrain sous un autre angle et une version dans laquelle le joueur se trouve sur un terrain «semi-infini» composé d'un terrain carré entouré

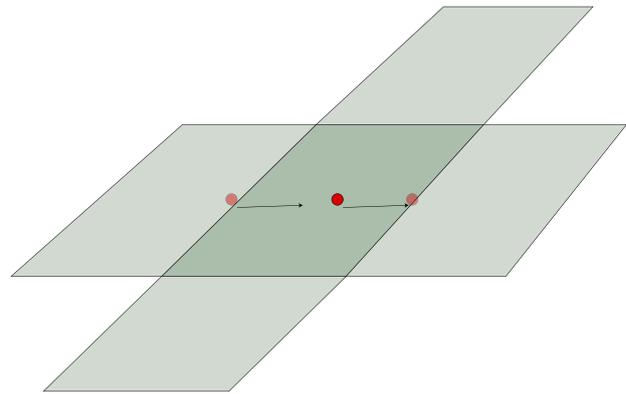


Schéma du principe du terrain «semi-infini»

d'autres terrain. Lorsque le joueur arrive au bout du terrain de base, il se retrouve téléporté de l'autre côté de celui-ci, lui donnant ainsi l'impression d'avoir continué son chemin et d'être dans un terrain infini alors qu'il retourne sans cesse au même point. Cette variation nous a ainsi permis d'améliorer la jouabilité de notre jeu ainsi que la sensation de liberté du joueur et le côté satisfaisant grâce aux montagnes se créant à l'horizon et offrant au joueur un terrain évolutif.

Un autre problème auquel

## Processus Creatif

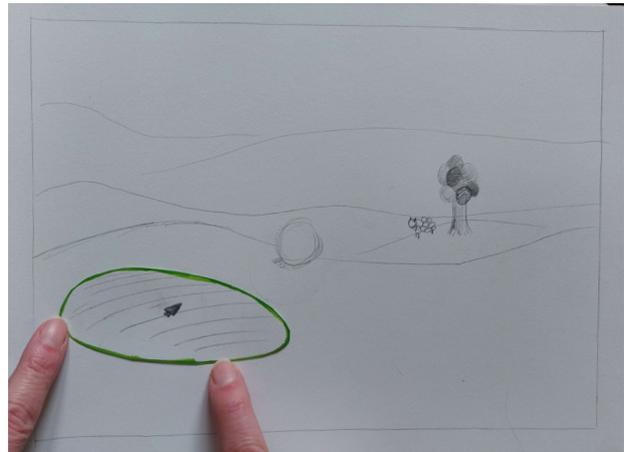
nous avons dû faire face concerne la redondance et la rejouabilité de notre jouet. En effet, lorsque les joueurs avaient modifié tout le terrain, ils avaient tendance à ne plus vouloir trop le modifier et à rester coincés.

Nous avons alors essayé une variation consistant à ajouter au terrain un système d'érosion, le faisant revenir à une hauteur proche de celle de base en un certain temps. Grâce à cette variation, nous avons pu diminuer la redondance et améliorer la rejouabilité mais également la sensation de vitesse. Le terrain s'affaissant tout seul, il force les joueurs à continuer d'avancer pour garder les «collines» qu'il crée autour de lui.

De nombreuses variations ont suivi et finalement les variations sur la brush sont arrivés.

Nous souhaitions tester la liberté pour le joueur de changer directement les metrics de son propre character de deux façons : La Force et la Taille.

Après de nombreux playtests, nous nous sommes rendus compte qu'augmenter la taille de la Brush permettait d'obtenir une vitesse plus élevée de la balle et ce de façon plus rapide qu'avec un petit diamètre de Brush; tandis qu'avoir un diamètre de Brush plus petit permettait d'avoir un



Prototypage Papier du feedback de taille de la brush

meilleur contrôle direct de la trajectoire de celle-ci.

Cette variation nous a permis de donner au joueur plus de liberté sur sa façon de jouer mais également d'éviter trop de redondance en lui offrant la possibilité de jouer comme il le souhaite.

Pour la Force de la Brush, nous nous sommes vite rendus compte que les joueurs l'utilisaient trop peu pour qu'elle soit pertinente et n'apportait que très peu de changement dans notre design. Certes, lorsque la force est plus élevée, la balle va plus vite, mais cela faisait perdre trop de contrôle sur celle-ci et les joueurs n'arrivaient plus à la contrôler comme ils le souhaitaient. Cette variation nous éloignait d'un des principes même de *Make Me A World*, l'accessibilité. Nous avons donc gardé le changement de taille de la Brush mais pas de Force.

---

Toutes ces variations nous ont permis d'améliorer notre concept en soutenant ses points forts et minimisant ses points faibles. Ainsi, nous avons souhaité que *Make Me A World* soit un jeu accessible et mettant en avant la beauté de la déformation et l'impact indirect qu'elle peut provoquer. *Make Me A World* est une métaphore de l'effet papillon. Un moindre geste peut entraîner la balle à rouler et à déranger les objets sur le terrain, mais chacun de ses gestes permet également de créer un monde merveilleux et unique que les joueurs peuvent admirer.



## Références Game Design



Notre première référence fut les parcours de billes. Il s'agit d'un jouet pour enfant grâce auquel le joueur possède plusieurs morceaux de chemin qu'il va positionner afin de créer un parcours sur lequel il va laisser une bille dégringoler jusqu'en bas.

Nous avons repris cette idée d'agir sur un terrain tout en ayant un objet rond central sur lequel le joueur n'agit pas directement.

Après avoir effectué des playtests sur notre premier prototype, nous nous sommes penchés sur «EXO ONE», un jeu d'aventure dans lequel le joueur contrôle une balle soumise à différents types de terrains et de gravité en fonction de la planète sélectionnée par le joueur.

Ce qui nous a le plus intéressé fut la gestion de la caméra. L'environnement entourant la balle est grandement mis en avant par la caméra du jeu, une simple caméra à la troisième personne visant la direction dans laquelle la balle va et étant suffisamment éloignée pour permettre de voir ce qui entoure la balle. Dans Exo One, la caméra met en avant l'environnement de la balle plus que celle-ci.



Dans Make Me A World, nous souhaitons également mettre en avant l'environnement autour de la balle, évoluant au fur et à mesure de l'action du joueur sur le terrain mais tout en permettant à la balle de rester un élément important. Nous avons donc gardé ce principe de caméra à la 3ème personne en la centrant légèrement plus sur la balle pour donner au joueur une meilleure sensation de contrôle de celle-ci.

## Références Game Design

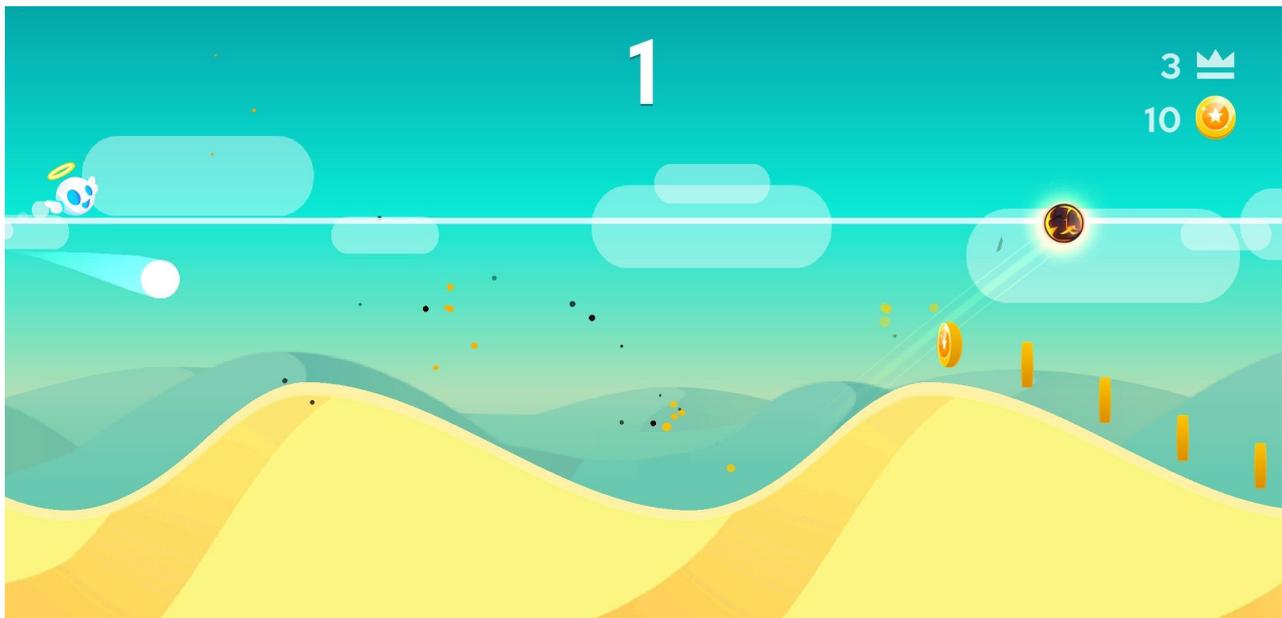
«Journey» est un jeu d'exploration dans lequel le joueur contrôle un personnage glissant sur le sable.

Ce qui nous a intéressé dans son design est son côté contemplatif et sa façon de perdre le joueur par les contrôles et les mouvements de celui-ci sur le terrain. Nous nous sommes grandement inspirés du game feel de ce jeu, les moyens de designs utilisés afin de faire croire au joueur qu'il glisse sur le sable (sons, mouvement de la caméra, particules,...)



Afin de comprendre les sensations de vitesse et les feedbacks d'un jeu utilisant la physique, nous nous sommes tournés vers «Dune!!», un jeu mobile dans lequel le joueur contrôle une balle en lui appliquant plus de masse en appuyant dessus, lui permettant ainsi de prendre de la vitesse grâce aux courbes du terrain.

Les effets sur la balle (particules, changements de sprites) et en dehors (modification du fonds en fonction de la vitesse, changements de la musique, sons du vent) nous ont grandement influencé dans nos choix de design afin de faire ressentir au mieux la vitesse au joueur.





## Références Game Design

Les «Sims 4» est un jeu de gestion et de simulation de vie dans lequel nous nous sommes inspirés d'un outil. Il s'agit du Terrain Tool, permettant de modifier directement le terrain de sa parcelle.

Lorsque nous cherchions un moyen de modifier un terrain de façon lisse, cet outil nous a directement intéressé ainsi qu'un autre outil similaire du même nom : le Terrain Tool utilisable directement sur Unity. Leur façon de «déformer» le terrain plutôt que de le modifier nous a grandement plu et nous nous sommes inspirés de ces deux outils afin de parfaire notre brush.



Après tous les playtests effectués, de nombreux joueurs nous ont fait remarquer qu'ils trouvaient leur expérience frustrante par le manque de vitesse de la balle. Afin d'ajouter un élément de surprise et de permettre une meilleure sensation de la vitesse, nous nous sommes penchés vers la mécanique des Arches, permettant de booster la balle en terme de vitesse tout en donnant au joueur un aspect de surprise détonnant avec le côté calme et tranquille du jouet.

Cet élément de «Just Cause 4» a été notre plus grosse inspiration dans le design de ces arches grâce à leur mécanique de cercle lors des sessions de WingSuit dans les airs. Lorsque le joueur passe dans un cercle, il prend de la vitesse, l'influençant ainsi à tenter de passer dans tous les cercles qu'il aperçoit.



## 3C : Caméra

Make Me A World est un jouet contemplatif comportant deux aspects importants : la déformation du terrain et le contrôle de la bille grâce à cette mécanique de déformation. Il était donc important pour nous d'offrir au joueur une expérience mettant en avant cet environnement de la meilleure façon tout en lui donnant la meilleure lisibilité sur la bille et ainsi lui permettre de le contrôler le plus facilement possible.

Dans ce but d'offrir au joueur la meilleure lisibilité possible sur les mouvements de la bille mais également sur son environnement, nous avons désigné 3 comportements de caméra que nous expliquerons de façon plus détaillée plus tard : Un comportement «classique» avec une caméra à la 3ème personne pointant dans la direction dans laquelle part la

balle; un comportement «joueur» qui permet au joueur de tourner la caméra autour de la bille grâce aux Inputs Q et D et un comportement «vitesse» qui fait s'éloigner la caméra de la bille plus celle-ci prend de la vitesse.

La caméra est suffisamment éloignée de la bille afin que les joueurs puissent observer le monde autour d'eux se modifier petit à petit, tout en gardant un œil sur la bille, et ainsi améliorer le contrôle qu'ont les joueurs sur celle-ci.

Nous souhaitions réellement mettre en avant le côté contemplatif et changeant du monde tout en gardant le plus de lisibilité possible sur les mouvements de la bille et ainsi faciliter le plus possible son contrôle indirect.

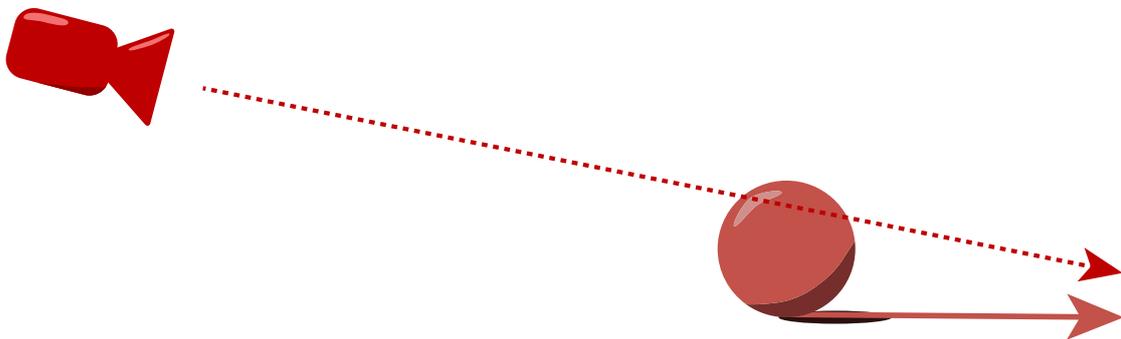


Schéma du Comportement «Classique» de la caméra



## 3C : Contrôle

Nos contrôles ont été pensés dans le but d'être le plus facilement pris en main par les joueurs, l'accessibilité de notre jouet faisant partie de nos priorités de design. Nous avons donc choisis des contrôles minimes à la souris en plaçant nos deux principales features, la montée et l'abaissement du terrain, sur les deux clics principaux de celle-ci.

Nous avons effectué de nombreux play test suite auxquels nous nous sommes rendus compte que la majorité des joueurs utilisait plus souvent la montée de terrain que l'abaissement et qu'il leur était plus évident que la montée se trouvait sur le clic gauche.

Pour donner plus de liberté au joueur, nous avons donné aux joueurs la possibilité de modifier la brush d'une façon : La taille.

Pour garder le plus possible de simplicité et d'accessibilité dans

nos contrôles, nous avons choisis de permettre au joueur de modifier la taille par un simple scroll de la souris, lui permettant de la modifier très rapidement et sans utiliser sa deuxième main.

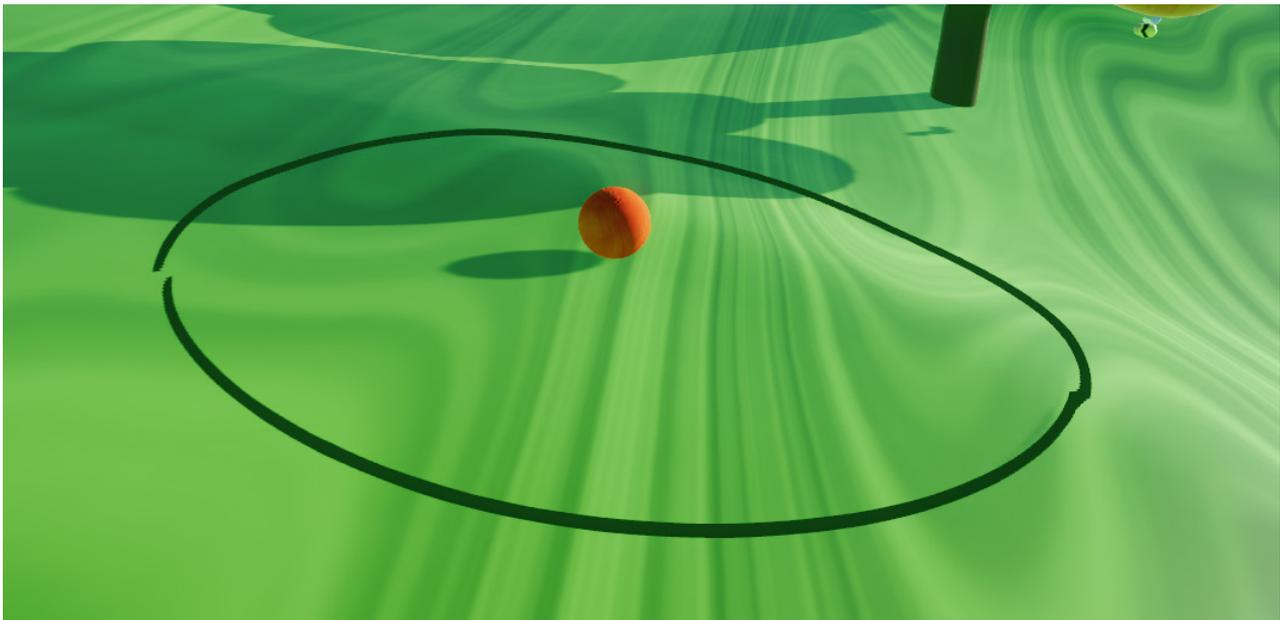
La caméra, quant à elle, est contrôlable grâce aux touches Q et D permettant d'améliorer la vision du joueur sur le terrain et ainsi la lisibilité des mouvements de la bille.

## 3C : Character

Le character de Make Me A World est la brush. Elle est l'outil du joueur et agit directement sur le terrain afin que le joueur puisse déformer sa topologie, créant ainsi son propre monde unique et agissant indirectement sur son environnement.

permettant au joueur d'apercevoir facilement et rapidement ses métriques, métriques que le joueur peut modifier grâce à un input modifiant la taille de la Brush.

Le seul visuel que le joueur aperçoit de celle-ci est son feedback de taille, cercle





## Noyau Système

Notre noyau système est basé sur la déformation d'un espace de jeu dans lequel des objets sont affectés par la physique.

Les diverses variations sur le système de base (la déformation d'un terrain) nous ont permis d'améliorer notre design. Nous avons ainsi pu ajouter le fait d'avoir une bille, roulant sur le terrain et des objets statiques devenant mobiles lors de collisions avec la bille.

C'est un système qui plonge le joueur dans un état de satisfaction et d'immersion lourde, l'envoyant complètement dans une bulle d'immersion.

C'est cette immersion systémique que nous avons cherché à développer en continuant de rechercher un moyen d'utiliser le principe de déformation couplé de la physique.

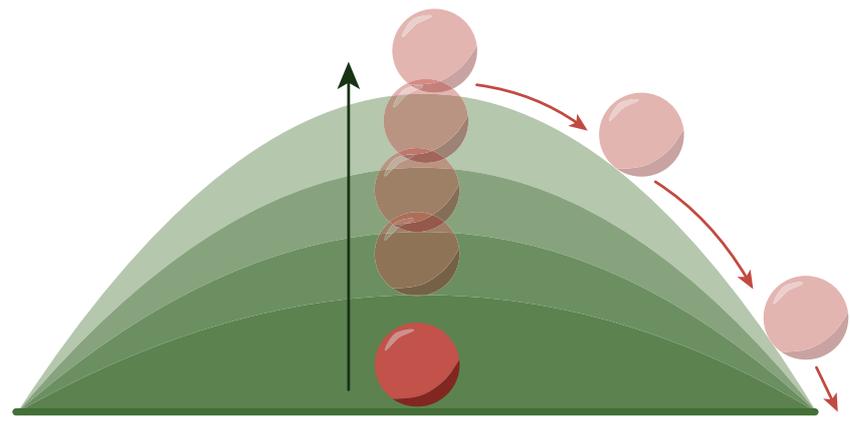


Schéma du principe du terrain déformable et de la bille

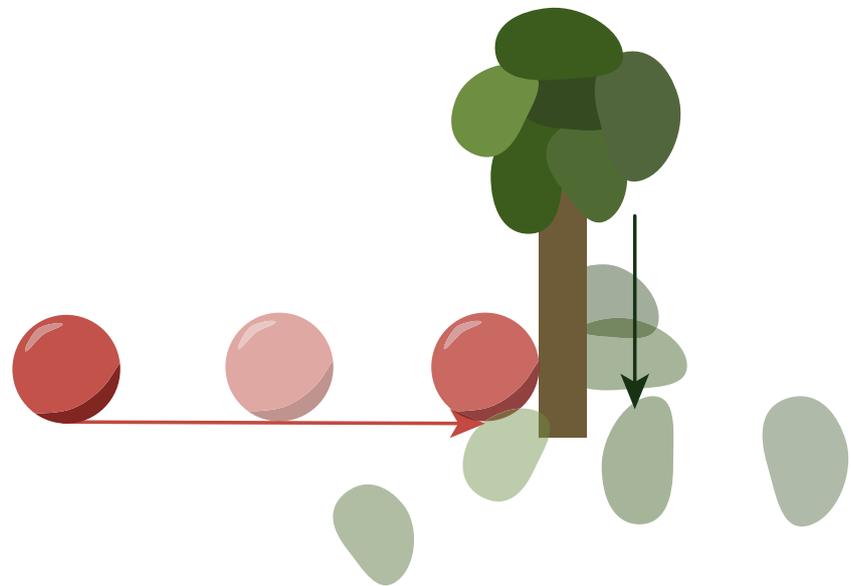


Schéma du principe de collision avec des objets

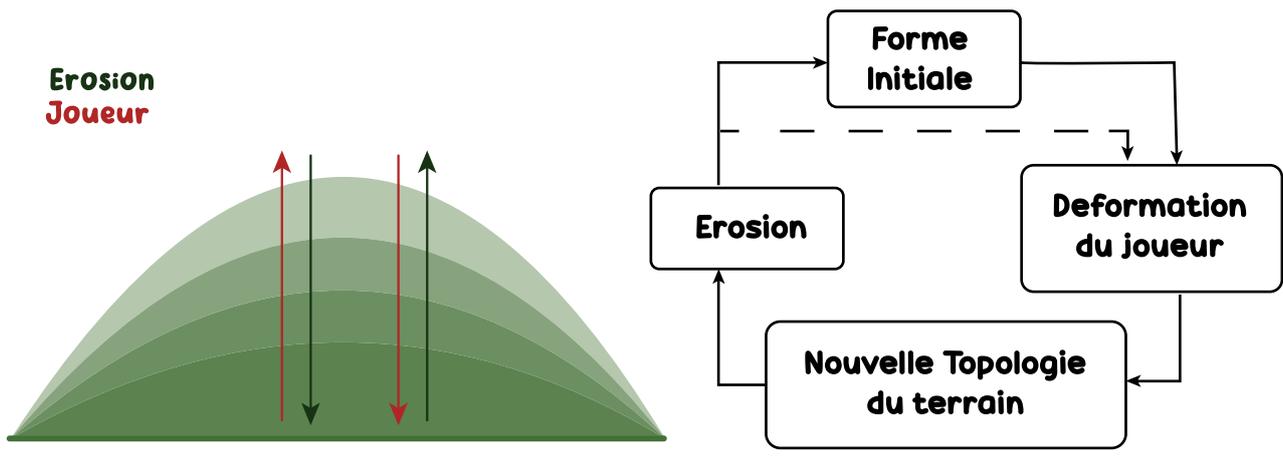
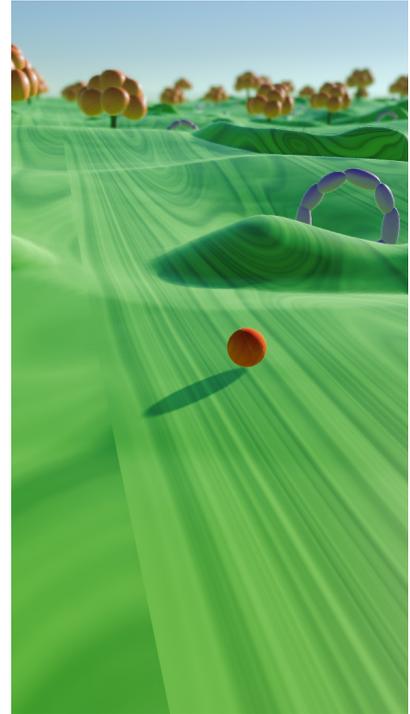
## Noyau Système

### Tendances et Tension

Dans *Make Me A World*, nous pouvons observer que la tendance du jeu (ou plutôt du Terrain) va être de revenir à sa forme initiale. Cette tendance est due à l'érosion, mécanique forçant le terrain à reprendre sa forme de départ après avoir été déformé par le joueur.

Cette tendance entre en conflit direct avec la tendance du joueur qui est de modifier la typologie de ce terrain dans le but de faire bouger la balle, d'améliorer le terrain ou tout autre but que le joueur se donne.

Ces deux tendances créent alors de la tension et les joueurs doivent donc lutter contre la tendance du terrain de retourner à son état d'origine en le modifiant sans cesse sous peine de le voir retourner à sa forme initiale.



Schémas de la Tension entre la Tendance du Jeu et la Tendance du Joueur



## Situations de Jeu et Boucles de Prédiction

La balle est lâchée par l'enfant au dessus d'une crevasse. On peut observer une arche suivie d'un mouton. La balle, affectée par la gravité se met à rouler jusqu'au point le plus bas de la crevasse.

### Prédiction

Le joueur aperçoit une arche sur sa droite et un mouton un peu plus loin

### Décision

Il décide de rentrer dans le mouton

### Action

Il passe sous l'arche, sa vitesse est multipliée et il rentre dans le mouton

### Régulation

La balle perd de la vitesse en rentrant dans le mouton et celui-ci perd sa laine et tombe

### Apprentissage

Passer sous une arche multiplie la vitesse de la balle et la fait aller dans la même direction que l'arche

Rentrer dans un mouton le fait tomber et fait tomber sa laine

## Situations de Jeu et Boucles de Prédiction

### Prédiction

Le joueur n'a pas modifié le terrain depuis un moment et celui-ci est retourné à son état d'origine à cause de l'érosion

### Décision

Il décide de monter le terrain afin de créer une montagne sous la balle pour la faire reprendre de la vitesse

### Action

Le joueur augmente (clic gauche) le terrain sous la balle

### Régulation

La balle prend de la vitesse en tombant de la montagne. Des montagnes sont créées à l'horizon et après quelques secondes, la montagne créée commence à s'affaisser à cause de l'érosion.

### Apprentissage

Le joueur peut utiliser la brush pour faire prendre de la vitesse à la balle. Utiliser la brush fait évoluer l'environnement du jouet. Les actions du joueur avec la brush sur le terrain sont affectées par l'érosion et reprennent leur état d'origine après quelques minutes.



## Game System

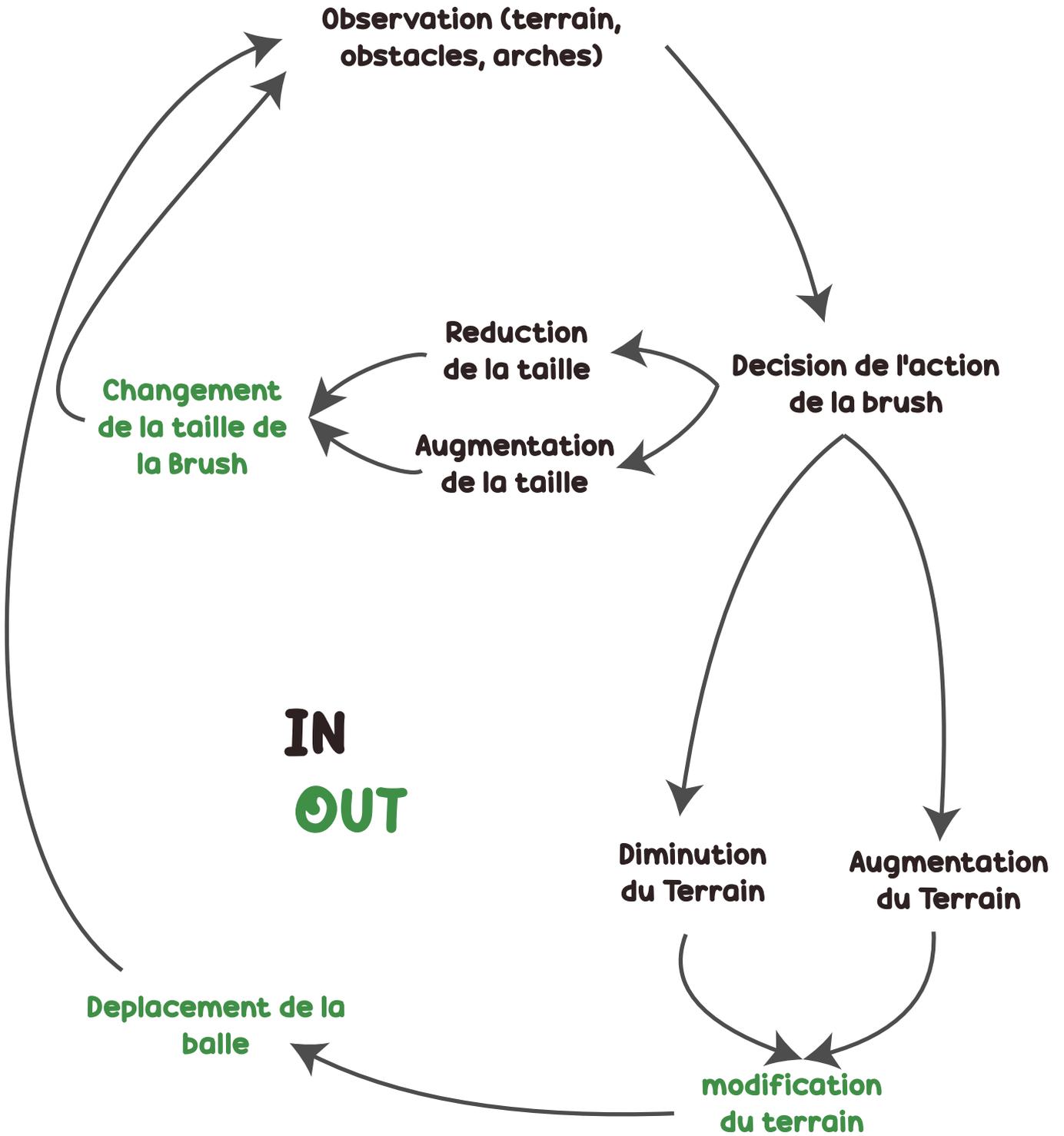
### OCR

Make Me A World étant un jouet, il est impossible de lui donner d'OCR Macro, n'ayant aucun but. Voici quelques OCRs fait à partir des retours de nos playtests :

	Objectif	Challenge	Reward
Micro term	Prendre de la vitesse	Passer dans une arche ou utiliser la brush et la gravité pour faire prendre de la vitesse à la balle	Feedbacks sonore, feedbacks visuels (particules), prise de vitesse
Mid term	Rentrer dans un mouton	Modifier le terrain afin de diriger la balle vers le mouton tout en lui faisant prendre de la vitesse	Feedbacks sonore, feedbacks visuels (particules et perte des boules de laine), diminution de la vitesse de la balle due à la collision
Mid term	Passer dans une arche	Déformer le terrain afin de diriger la balle sous l'arche.	Augmentation de la Vitesse de la balle

# Game System

## Boucle de Gameplay





## Game System

### Mécaniques

#### Déformer le terrain

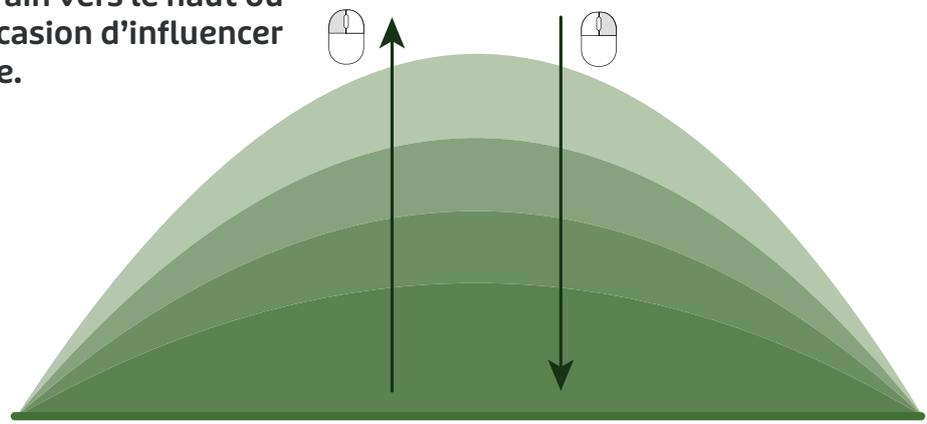


Monter le terrain



Descendre le terrain

Permet de gonfler le terrain vers le haut ou le bas et par la même occasion d'influencer le mouvement de la balle.



### Micro- Challenges :

**Tactique :** Réfléchir à la direction à prendre et comment

(Je vais faire une bosse un peu plus loin devant la balle avant de lui faire prendre de la vitesse afin qu'elle saute haut)

**Précision :** Le joueur doit faire gonfler le terrain au bon endroit afin de déplacer la balle là où il le souhaite.

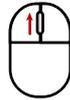
**Mesure :** Savoir quand s'arrêter et quand commencer à gonfler le terrain.

**Observation :** Analyser le terrain pour choisir la direction à prendre.

## Game System

### Mécaniques

### Augmenter / Rétrécir la taille de la Brush

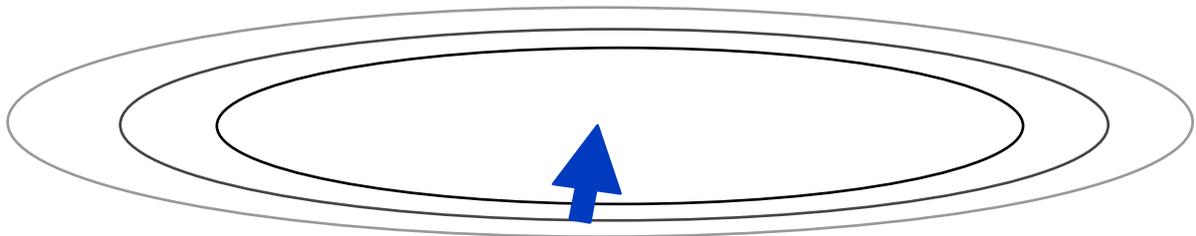


Scroll vers le haut  
pour augmenter



Scroll vers le bas  
pour réduire

Permet d'augmenter ou de rétrécir le diamètre de la brush.



### Micro- Challenges :

**Tactique :** Savoir quelle taille de brush utiliser pour les mouvements voulus (je vais utiliser une taille de brush fine pour mieux contrôler la bille)

**Précision :** Modifier la brush à la taille exacte souhaitée ou nécessaire.

**Mesure :** Savoir quand s'arrêter de scroller pour avoir la parfaite taille.

**Observation :** Analyser le terrain pour savoir quelle taille est la plus adaptée à une certaine situation.



# Game System

## Schéma de Ventrice

### CONCEPT

Déformation d'un terrain sur lequel se trouve une bille et des objets affectés par la physique

### Paradigm / Skills

Conscience Spatiale

Manipulation d'un espace de jeu

### FEATURE/CHALLENGE

déformer le terrain

Controler la bille

### Méchanic

Le terrain reprend sa forme initiale apres un certain temps sans avoir été modifié

Le terrain peut etre modifié jusqu'à une limite de hauteur (haut et bas)

La bille peut rentrer en collision avec d'autres objets

La bille est affectée par la physique

### Interface

Clic gauche pour augmenter le terrain

Clic Droit pour creuser le terrain

Scroll pour augmenter ou diminuer la taille de la Brush

Q & D pour faire tourner la caméra

## Game System

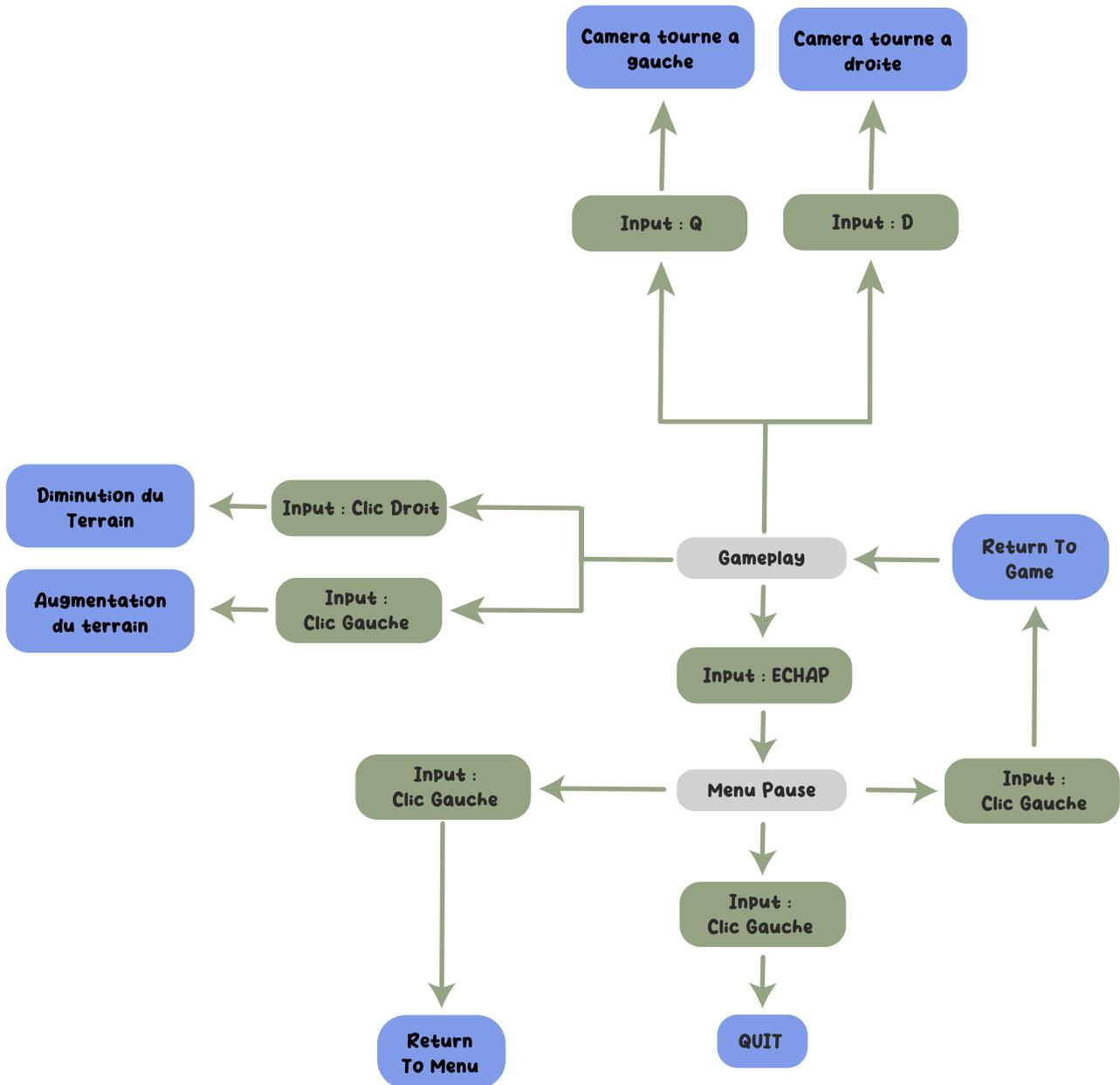
### Signs et feedbacks

Signs	Action	Feedback(s)
Son d'un mouton + Visuel du mouton	Le joueur envoie la balle sur le mouton	Feedback sonore du mouton + Particules + le mouton perd sa laine
Curseur de la souris + Feedback visuel de la taille de la brush	Input Clic Droit / Clic Gauche	Feedback sonore (son «gonflage» / Son «dégonflage») + Modification du terrain
Feedback Visuel de la taille actuelle de la Brush	Input Scroll	Feedback sonore (son augmentation/ diminution de la brush) + Modification du feedback visuel
Visuel d'un arbre feuillu	Le joueur envoie la balle sur l'arbre	Feedback sonore de l'impact sur le tronc et des feuilles qui tombent + Particules + meshes des feuilles qui tombent et roulent sur le sol
Mesh d'un objet	La balle entre en collision violemment contre un objet	Feedback sonore de l'impact + Particules (en fonction de l'objet) + Screenshake



## FlowCharts

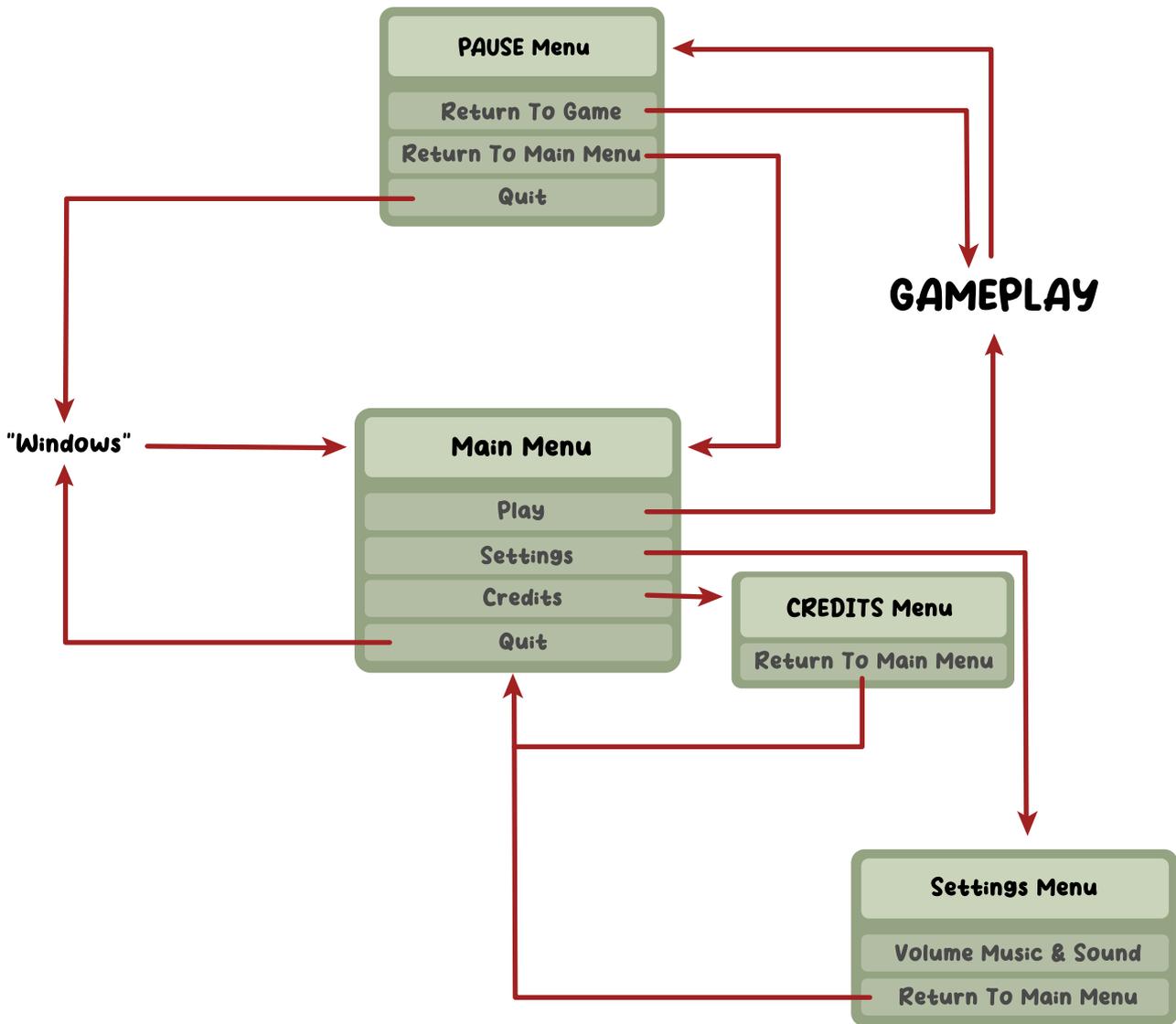
### Gameplay





# FlowCharts

## Menus



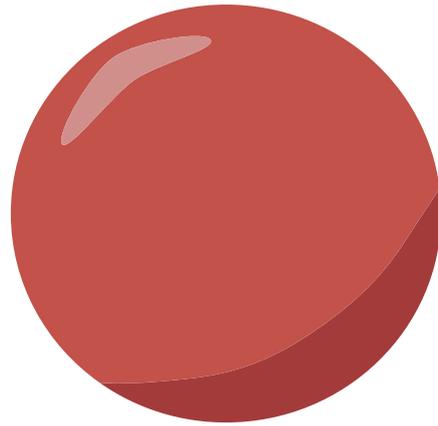
# PAUSE

RETURN TO GAME

RETURN TO MAIN MENU

QUIT





# GAME DESIGN DOCUMENT



## Caméra

### Comportement «Classique»

Le comportement «Classique» de la caméra consiste à la faire suivre automatiquement la balle et à pointer la direction dans laquelle celle-ci se dirige.

Cela permet au joueur de pouvoir plus facilement contrôler la balle sans s'occuper de la caméra.

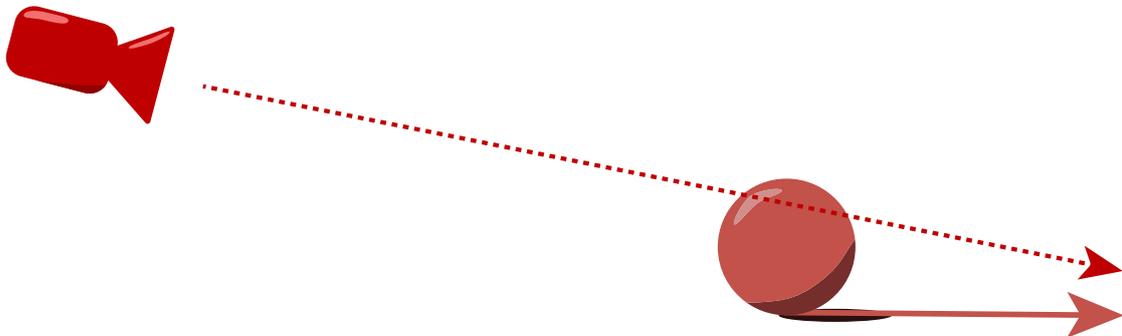


Schéma du Comportement «Classique» de la caméra

## Caméra

### Comportement «Joueurs»

Le comportement «Joueurs» se déclenche lorsque le joueur utilise un des deux Inputs de la caméra, Q et D.

Ce comportement permet au joueur de reprendre la main sur la caméra automatique pour améliorer sa vision de la balle ou pour observer l'environnement autour de celle-ci.

Son comportement est très simple : en appuyant sur Q, la balle tourne autour de la bille en partant vers la gauche tandis qu'en appuyant sur D, la caméra part vers la droite.

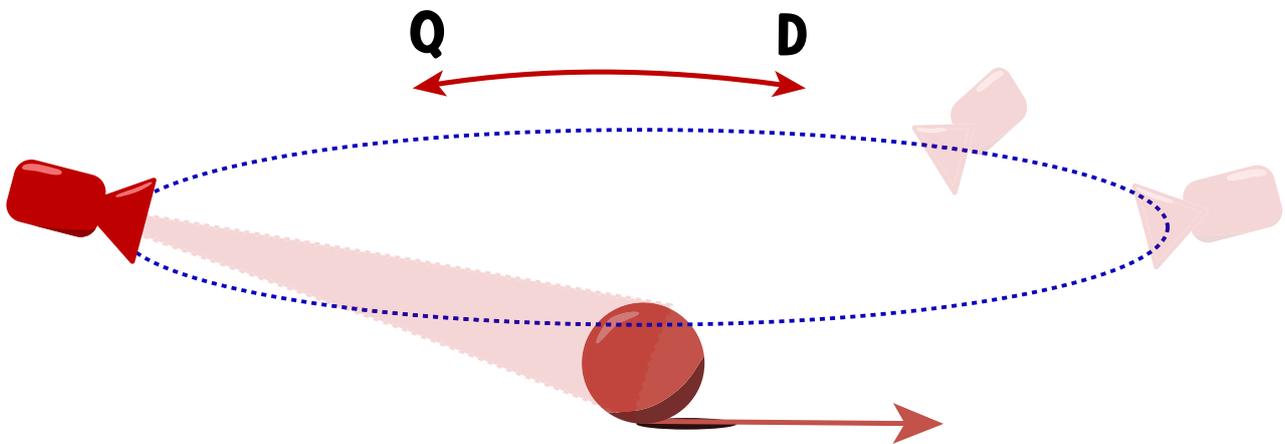


Schéma du Comportement «Joueur» de la caméra

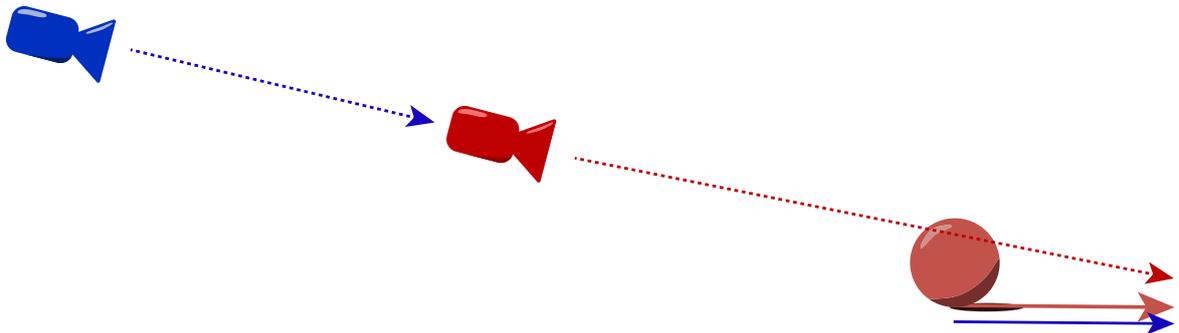
Après un certain temps sans toucher à l'un de ces deux Inputs, la caméra reprend son comportement classique et revient donc pointer dans la direction où va la bille.



## Caméra

### Comportement «Vitesse»

Le comportement «Vitesse» fait reculer la caméra à une position plus éloignée de la bille (entre deux valeurs fixes) en fonction de la vitesse de la bille. Plus celle-ci est rapide et plus la caméra reculera afin de donner au joueur une meilleure sensation de vitesse.



Vitesse «Lente» →

Vitesse «Rapide» →

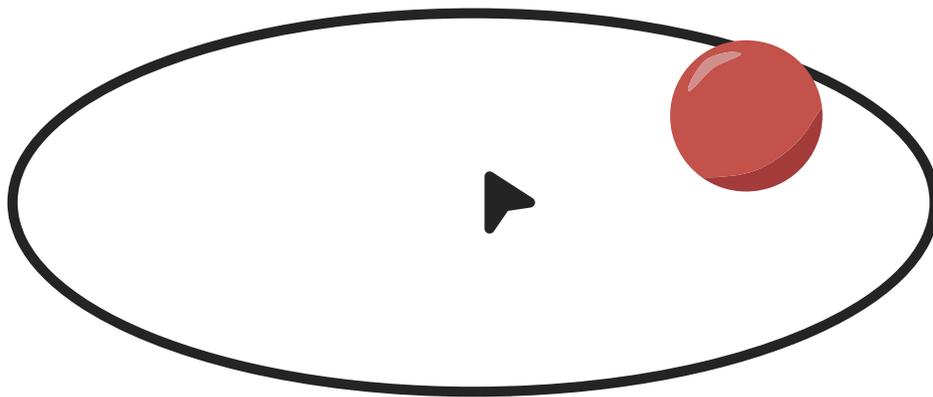
Schéma du Comportement «Vitesse» de la caméra

## Character

Le character de *Make Me A World* est la Brush utilisée par le joueur pour déformer le terrain. Celle-ci lui permet de mettre en oeuvre sa créativité afin de s'amuser avec le terrain mais également avec l'écosystème autour de lui et la bille.

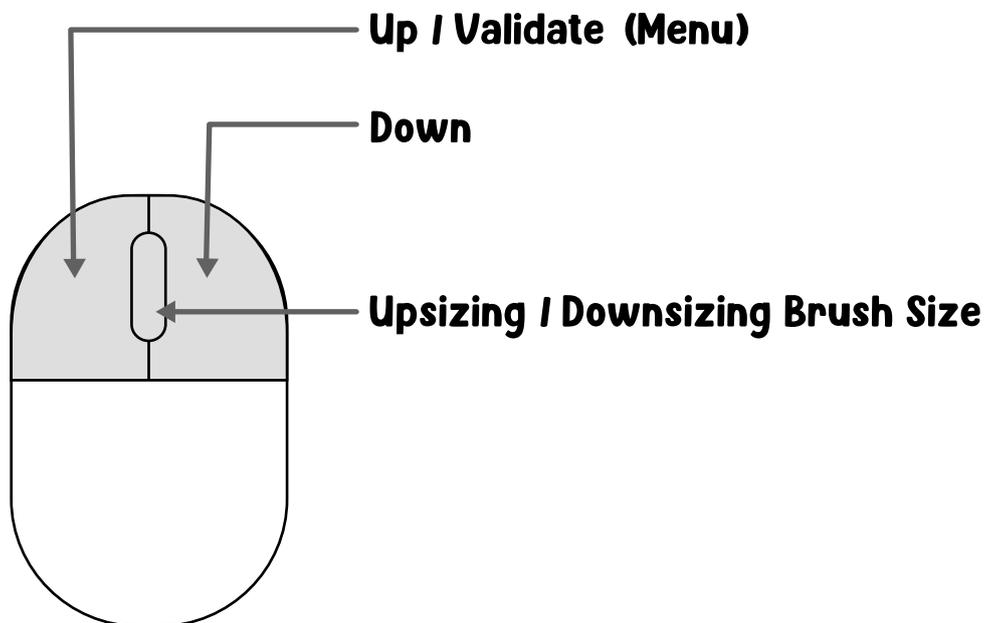
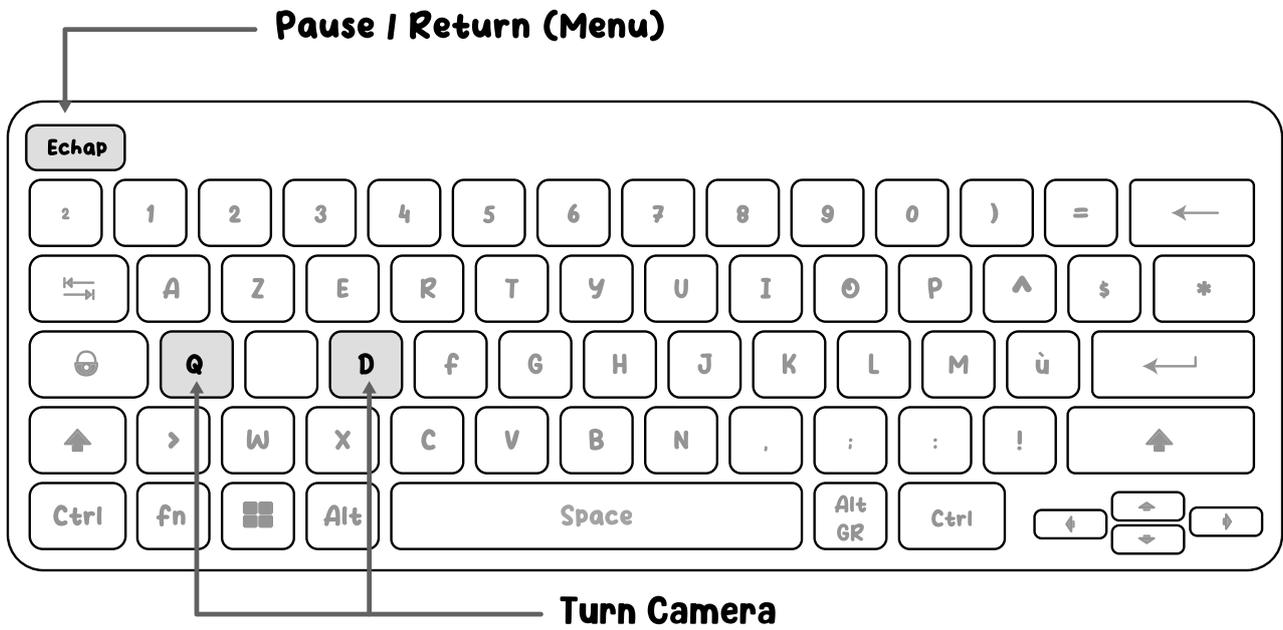
Le déplacement de la bille se fait donc uniquement par le biais de l'utilisation de la brush sur le terrain. Les metrics de celle-ci ont donc un impact direct sur le contrôle de la bille et nous avons

souhaité laisser au joueur le contrôle de sa taille afin de lui permettre de choisir la façon dont il souhaite contrôler la bille et le terrain (par exemple en utilisant une brush fine, le contrôle de la bille est plus précis mais plus difficile et prendre de la vitesse avec une brush fine est plus compliqué ; tandis qu'avec une brush large, contrôler précisément la bille est bien plus dur tandis que lui faire prendre de la vitesse devient très facile).





## Controls

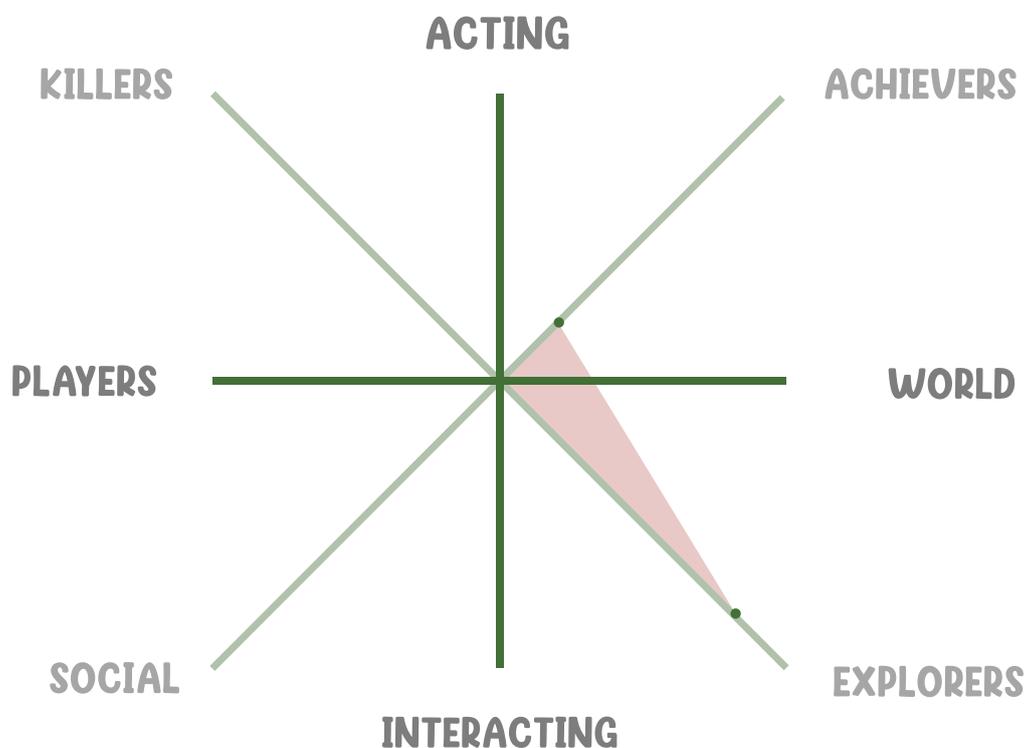


Make Me A World est un jeu qui se veut le plus accessible possible. Les contrôles de la brush sont principalement à la souris, donnant au joueur l'opportunité de se concentrer uniquement sur le contrôle de la trajectoire de la balle qui est l'élément le plus difficile du jeu.

Les joueurs de Make Me A World sont donc des joueurs mid-core, le contrôle de la trajectoire de la balle restant relativement complexe à maîtriser. Ceci étant, les contrôles de la brush étant très simples, le jeu peut plaire à des joueurs un peu plus casuels qui apprécient le côté contemplatif du jeu.

## Modele de Bartle

Make Me A World ne comporte pas d'élément incitant les joueurs à socialiser entre eux, étant un jouet ne comportant qu'un seul joueur. Le jouet étant axé sur l'exploration et l'évolution de l'environnement, poussant le joueur à parcourir plus de terrain dans le but de ne pas laisser le monde tomber «à plat», la cible est principalement le joueur de type «explorer» bien qu'il puisse également intéresser le joueur «acheiver», notamment par le nombre de challenge que les joueurs peuvent se donner dans le jeu (ex : sauter au dessus d'une arche).



Modele de Bartle

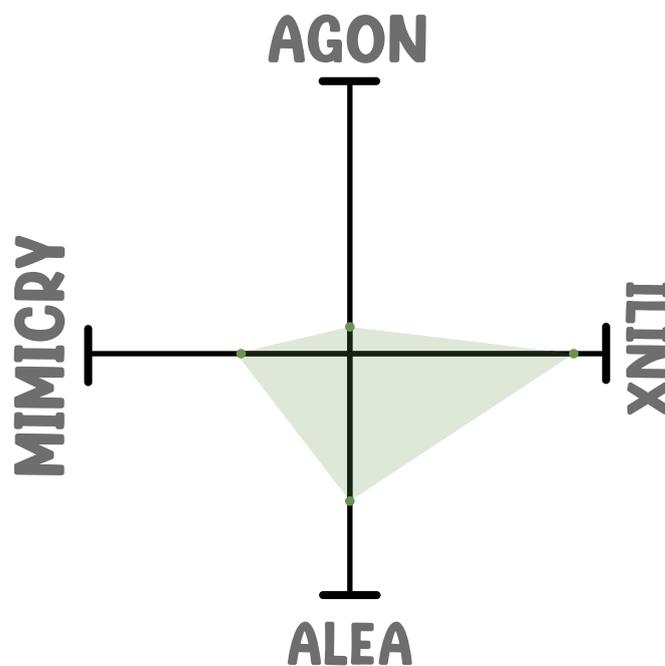


## Matrice de Caillois

Le modèle de Caillois nous permet de classer *Make Me A World* comme un jeu basé principalement sur l'Ilnx, c'est -à -dire sur le vertige et l'immersion du joueur dans le monde et le gameplay du jeu. Le joueur est plongé dans un univers très stylisé et qu'il peut lui-même modeler à sa façon. Le jouet comporte également une grosse part d'Aléa de part le simple fait de ne pas savoir sur quoi le joueur va tomber ni où vont tomber certains objets. Le Mimicry est ensuite un peu présent. Le joueur joue à déformer le terrain, il simule les gestes d'un enfant jouant avec de la pâte à modeler ou s'investit comme une forme divine supérieure, simulant alors le modelage d'un monde à son image. Enfin, il ne contient que très peu d'Agon. Étant un joueur, *Make Me A*

*World* est principalement du côté du Paida mais pas totalement : le joueur est libre de jouer comme il le souhaite mais il reste contraint dans son jeu par certaines règles telles que la physique ne pouvant donc pas le mettre entièrement du côté du Paida.

**Paida** —●— **Ludus**

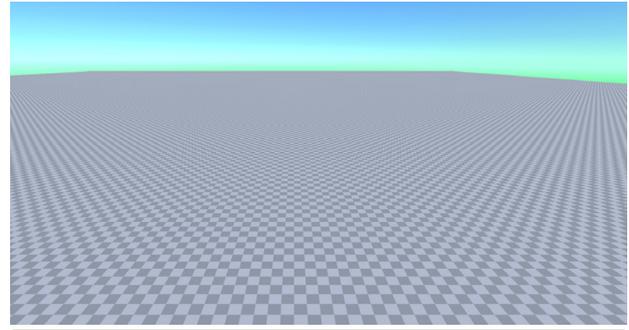


Matrice de Caillois

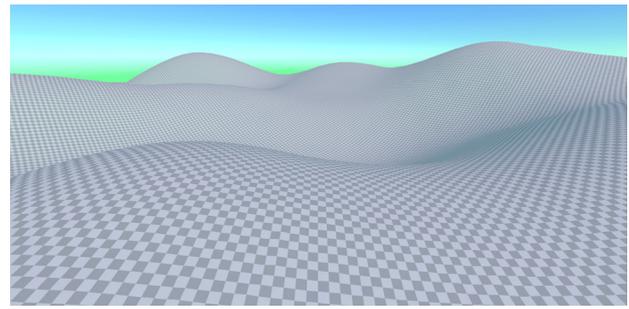
## Level Design

Nous souhaitons offrir au joueur une expérience de jeu à forte rejouabilité et lui conférant une impression d'évolution.

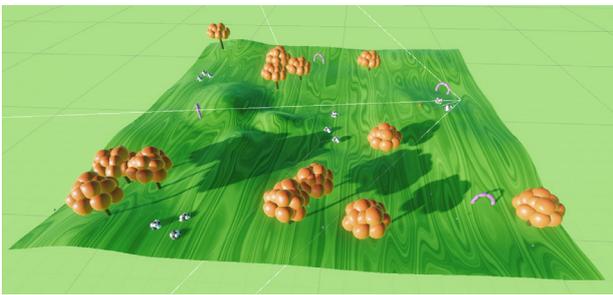
Nos premières intentions ont été de laisser le terrain plat et vide, permettant au joueur de façonner son propre monde sans notre influence. Nous nous sommes rendus compte qu'avec cette approche, les joueurs restaient «timides» face à l'utilisation de la brush, ils essayaient de faire se déplacer la balle mais ne s'intéressaient pas assez à l'environnement et à la forme du terrain. Pour forcer l'immersion des joueurs et leur montrer le potentiel de la brush, nous avons choisis de proposer un terrain déjà modifié, propulsant le joueur dans un véritable paysage de collines ayant son propre écosystème. Avec cette nouvelle approche, les joueurs comprennent immédiatement le pouvoir du character du jeu et poussent leur créativité bien plus loin.



Terrain plat



Terrain déjà modifié



Terrain Final

Les objets du terrain sont placés de deux façons : seuls ou en «clusters». Trouvant les arbres et les moutons difficiles à atteindre, nous avons choisis de faire des «clusters», petites zones avec plusieurs objets au même endroit pour faciliter leur interaction avec la balle. Sur le terrain, nous trouvons donc des zones avec plus ou moins d'objets, créant ainsi plus ou moins de difficultés pour le joueur dans ses challenges.



Afin de nous assurer une meilleure rejouabilité, nous avons développé un système afin de permettre au joueur de jouer autant qu'il veut, sans être stoppé par le jeu, grâce au terrain semi-infini.

Grâce à un système de terrain «Mère» entouré de terrains «Imposteurs», reproduisant les actions du joueur sur chaque terrain tout en gardant le joueur captif sur le terrain Mère, le joueur a l'impression de se trouver sur un terrain infini dont l'horizon évolue et se déforme alors que ces déformations qu'il aperçoit ne sont que des copies de celles que le joueur effectue sur le terrain Mère.

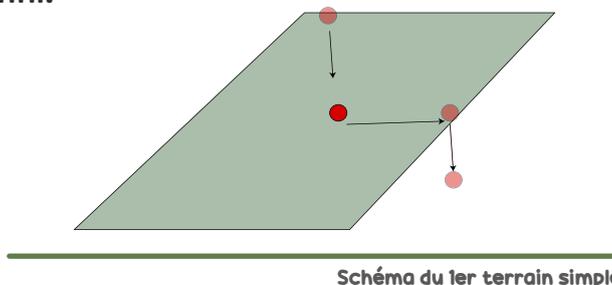


Schéma du 1er terrain simple

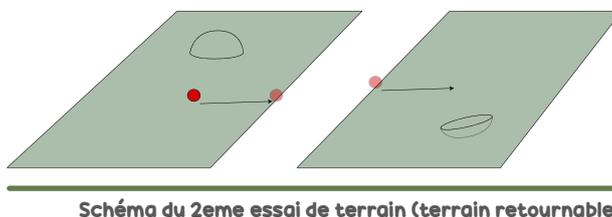


Schéma du 2eme essai de terrain (terrain retournable)

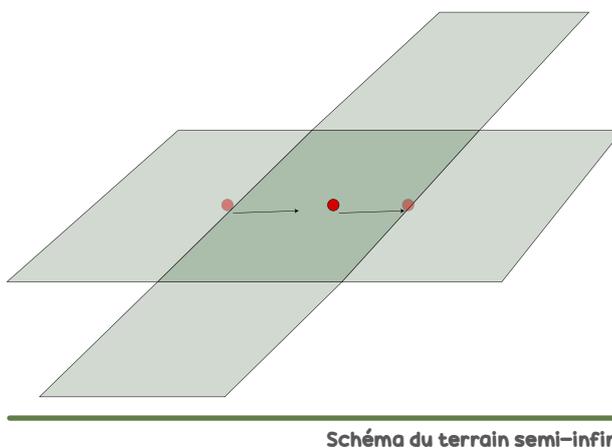
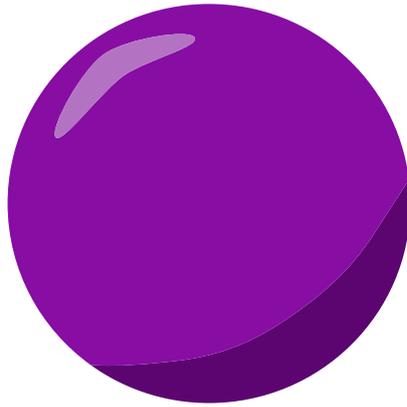


Schéma du terrain semi-infini

Enfin, afin de donner au joueur l'impression d'avancer, nous avons également choisis de disposer nos objets en zones «d'intérêt». Ainsi, le joueur peut se balader et trouver des arbres et animaux seuls mais également des troupeaux d'animaux et plusieurs arbres côte à côte, le but étant de le pousser à se balader pour en découvrir plus mais également d'attirer son œil sur l'environnement disponible.



# **Charte Technique**



## Challenge Technique

Notre jouet permet au joueur de modeler à sa guise son propre terrain de jeu. C'est pourquoi nous avons utilisé l'outil Terrain de Unity comme base de travail.

L'outil Terrain permet aux Game Designers de créer du relief dans un environnement de jeu, simplement à l'aide de multiples brush. Les brush sont utilisés pour élever ou abaisser le terrain en épousant une forme prédéfinie.

Notre premier challenge lors de la création de notre jouet a été de donner au joueur l'accès à cet outil terrain en mode Play. En effet, le terrain étant à l'origine un outil de développement, il n'est pas fait pour pouvoir être utilisé en mode Play.

Rapidement, nous avons conclu qu'il serait impossible de directement donner au joueur l'accès aux outils donnés normalement par l'outil. Nous avons donc créé les nôtres, simulant le fonctionnement des outils d'origine.

Nous avons donc étudié en profondeur le fonctionnement et la structure de l'outil Terrain (comment fonctionne-t-il et comment est-il structuré?). C'est alors que, partant de ces nouvelles connaissances, nous avons créé manuellement une brush simulant le comportement de la brush circulaire de l'outil. Avec cette brush qui définit un cercle autour de l'endroit que pointe le curseur, nous déplaçons vers le haut ou le bas les différents points/vertex de la heightmap se situant dans les coordonnées définies par notre brush.





## Algorithmes Utiles

### Algo de Terrain:

La technique principale permettant la modification du terrain est basée sur la HeightMap de celui-ci :

Lorsque le joueur clique sur une des parties du terrain , le point trouvé est converti vers la heightmap et on calcule le brush du terrain à partir de ce point-ci.

La conversion du point vers les coordonnées de la heightmap du terrain se fait en plusieurs étapes:

1. Tout d'abord, le point est transformé en une position locale par rapport au terrain, en soustrayant la position du terrain à celle du point :

```
Vector3 terrainLocalPos = point - _terrain.transform.position;
```

2. Ensuite, cette position locale est normalisée en divisant chaque composante x et z par la taille correspondante du terrain :

Cela donne une position normalisée où (0,0) représente le coin inférieur gauche du terrain et (1,1) représente le coin supérieur droit.

```
Vector3 normalizedPos = new Vector3(terrainLocalPos.x / _terrainData.size.x, 0, terrainLocalPos.z / _terrainData.size.z);
```

3. Enfin, cette position normalisée est convertie en coordonnées de la heightmap, en la multipliant par la largeur et la hauteur de la heightmap (moins 1 afin de tenir compte de l'indexation à partir de zéro) :

```
int mouseX = (int)(normalizedPos.x * (_heightmapWidth - 1));  
int mouseZ = (int)(normalizedPos.z * (_heightmapHeight - 1));
```



## Algorithmes Utiles

### Algo de Caméra:

La caméra , comme le montre le schéma ci-dessous, regarde en permanence la balle , cependant elle est rattachée en Follow à un empty qui suit la balle grâce à un Lerp. Cela permet à la caméra de suivre la balle de manière fluide, même quand elle se cogne et que sa vitesse est réduite instantanément à zéro.

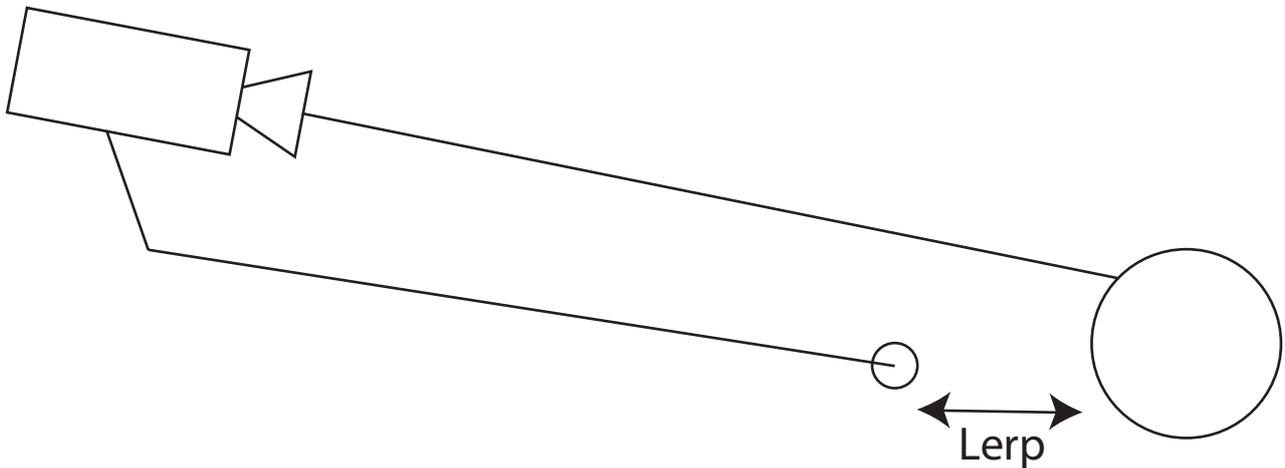


Schéma Fonctionnement Caméra



## Difficultés à surpasser

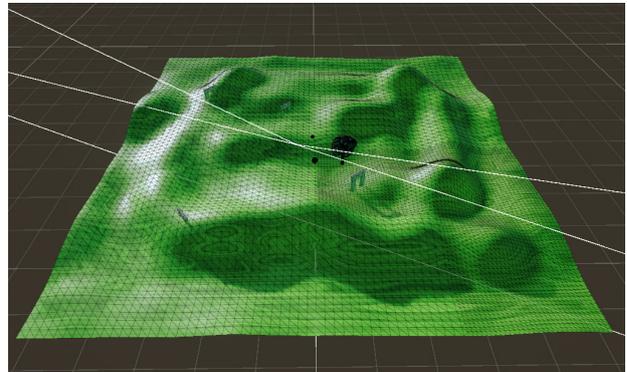
Après la difficulté initiale due à notre choix de mécanique, nous avons eu 3 importantes difficultés relativement liées les unes aux autres.

### Difficulté 1 : La taille du terrain

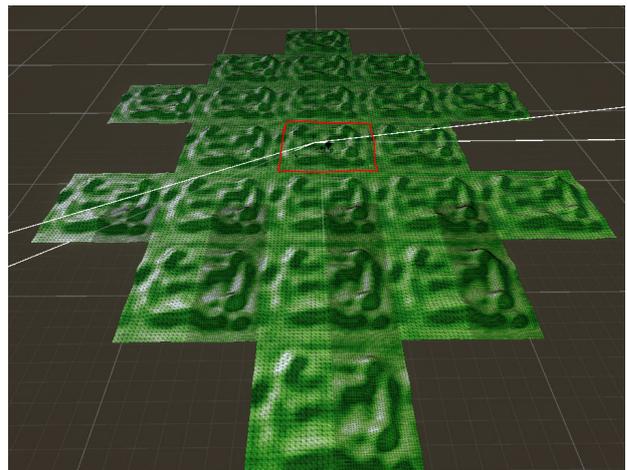
Notre jouet, de par sa nature, doit si possible éviter de confiner le joueur dans un espace trop petit. Cependant, augmenter la taille du terrain revient aussi à en augmenter la résolution et donc le nombre de points stockés dans sa heightmap. Le tout, couplé à la modification constante de la topologie du terrain, risquait de provoquer des chutes de FPS.

La solution que nous avons choisie pour pallier à ce problème a été de créer une multitude de terrains imposteurs tout autour du terrain d'origine.

Le terrain imposteur est un terrain qui reproduit toutes les modifications de topologie du terrain d'origine, mais qui se situe sur la périphérie de ce dernier. Ainsi, lorsque le joueur crée une colline sur son terrain, tous les terrains autour de lui créent la même colline.

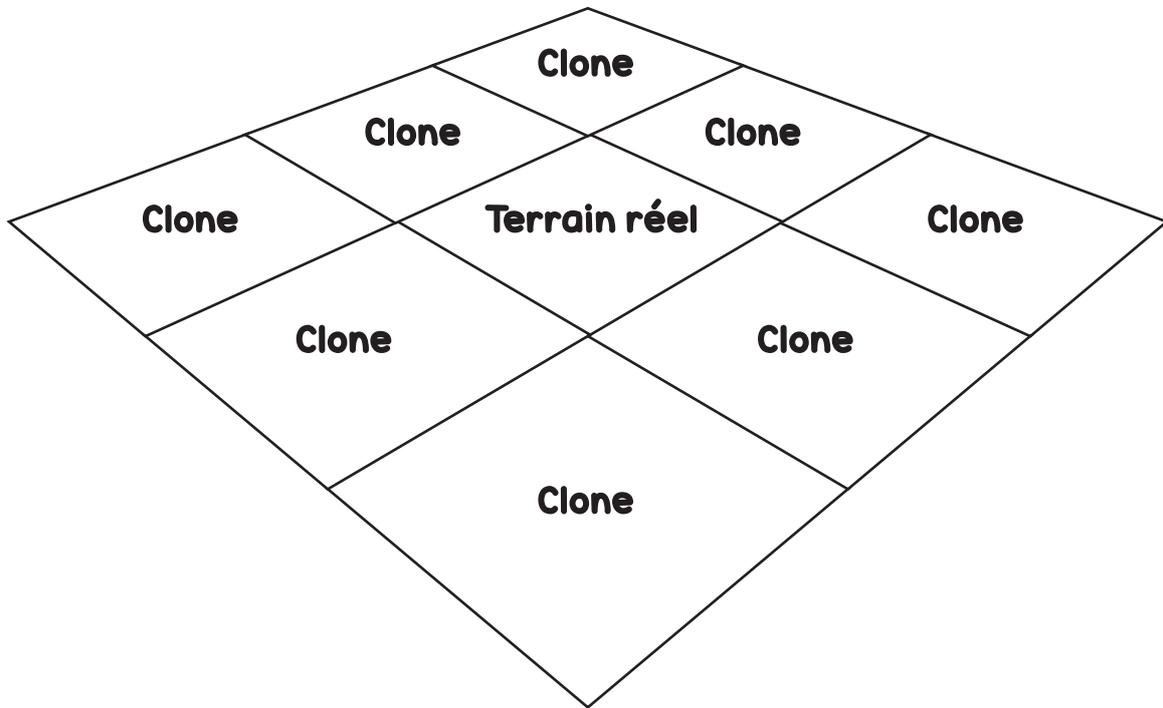


Ici le terrain principal, avant que la partie ne soit lancée, il est seul.

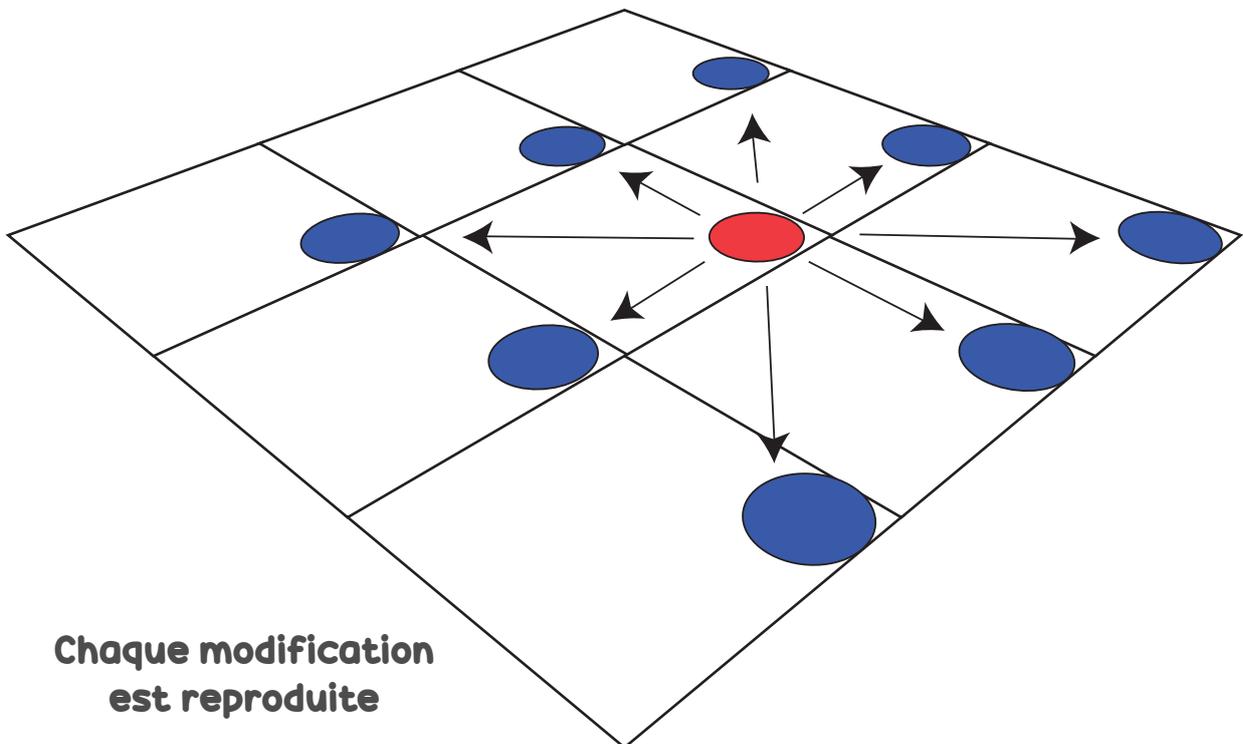




## Difficultés à surpasser



Ici, la partie est lancée : le terrain encadré en rouge est le terrain principal tandis que les autres sont les terrains imposteurs. Le joueur ne peut pas les atteindre, il ne peut que les observer.



Chaque modification est reproduite

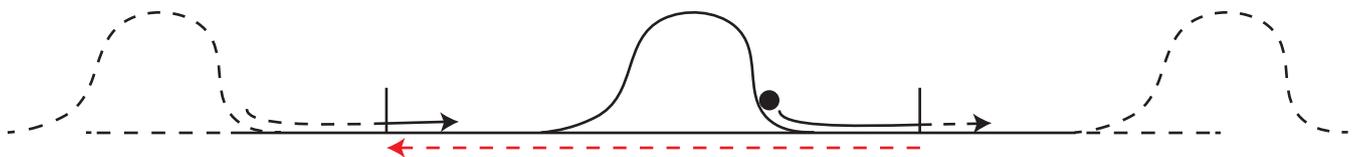
## Difficultés à surpasser

Cette solution est avantageuse pour plusieurs raisons. Dans un premier temps, elle permet d'ajouter un décor au loin, donnant au joueur une sensation de circuler dans un monde plus grand. De plus, l'effet de modifications de tous les terrains simultanément est un effet agréable à constater. Cela rend le monde de jeu plus vivant et évolutif car le joueur ne se rend pas compte à première vue que le terrain est dupliqué plusieurs fois.

Cependant, l'ajout des terrains imposteurs devait obligatoirement se coupler à une méthode pour confiner le

joueur, sinon nous aurions dû continuer à générer des terrains indéfiniment, rendant alors le problème de FPS plus important qu'il ne l'aurait été avec un grand terrain. C'est pour cela que nous avons choisi de faire une téléportation de type «Pacman» qui confine le joueur dans le terrain d'origine.

Une téléportation de type «Pacman» permet au joueur de se retrouver confiné dans une zone définie. Lorsqu'il dépasse le confinement d'un côté de sa zone, il se retrouve téléporté du côté opposé.



Teleportation de la balle  
en sortant du terrain réel

Ces deux solutions couplées l'une à l'autre permettent au joueur d'avoir la même impression que lorsqu'on monte l'escalier infini dans «Mario 64», étant donné que le joueur, lorsqu'il se téléporte, se trouve devant une version identique à son terrain d'origine.

Nous avons d'ailleurs utilisé la fonction [OnTargetObjectWarped] pour rendre la téléportation invisible à la caméra. Cette fonction permet à un objet et sa caméra de se téléporter ensemble sans que la caméra ne soit perturbée par ce déplacement.



## Difficultés à surpasser

### Difficulté 2 : Les modifications des liaisons de terrain

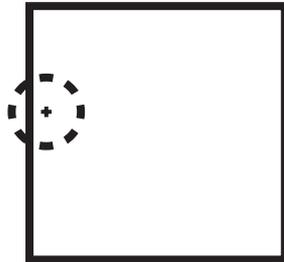
Dans la situation où le joueur cherche à modifier le terrain, à l'endroit ou le terrain d'origine et le terrain imposteur se rejoignent, nous nous sommes rapidement rendus compte

que le script de modification devait non seulement affecter le terrain d'origine mais également l'imposteur qui le côtoie.

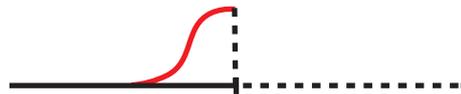
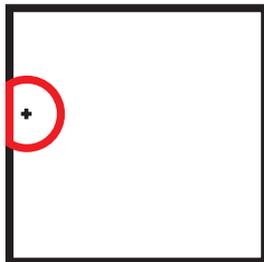
Vue de dessus

Vue de coté

Ce qu'il doit se passer



Ce qu'il se passe réellement



Click de la souris

+

Zone censé être affecté



Zone réellement affecté





## Difficultés à surpasser

Cette fonctionnalité a été implémentée en permettant au joueur de cliquer n'importe où sur la carte pour modifier le terrain. Le script de modification convertit alors la position du joueur en la position relative au terrain. Pour cela, il suffit de déplacer les valeurs

correspondantes à l'opposé si elles étaient en dehors des limites du terrain. Ensuite, étant donné que les terrains ont tous la même terraindata (ils partagent tous les mêmes données), ces modifications sont répliquées sur les imposteurs.

---

Voici comment cela fonctionne :

1. Si la coordonnée est inférieure à zéro, cela signifie que le joueur a dépassé le bord gauche ou inférieur du terrain. Dans ce cas, on ajoute la taille du terrain à la coordonnée pour la ramener à la limite opposée du terrain.
2. Si la coordonnée est supérieure ou égale à la taille du terrain, cela signifie que le joueur a dépassé le bord droit ou supérieur du terrain. Dans ce cas, on soustrait la taille du terrain de la coordonnée pour la ramener à la limite opposée du terrain.
3. Enfin, la coordonnée modifiée est renvoyée.

2 références

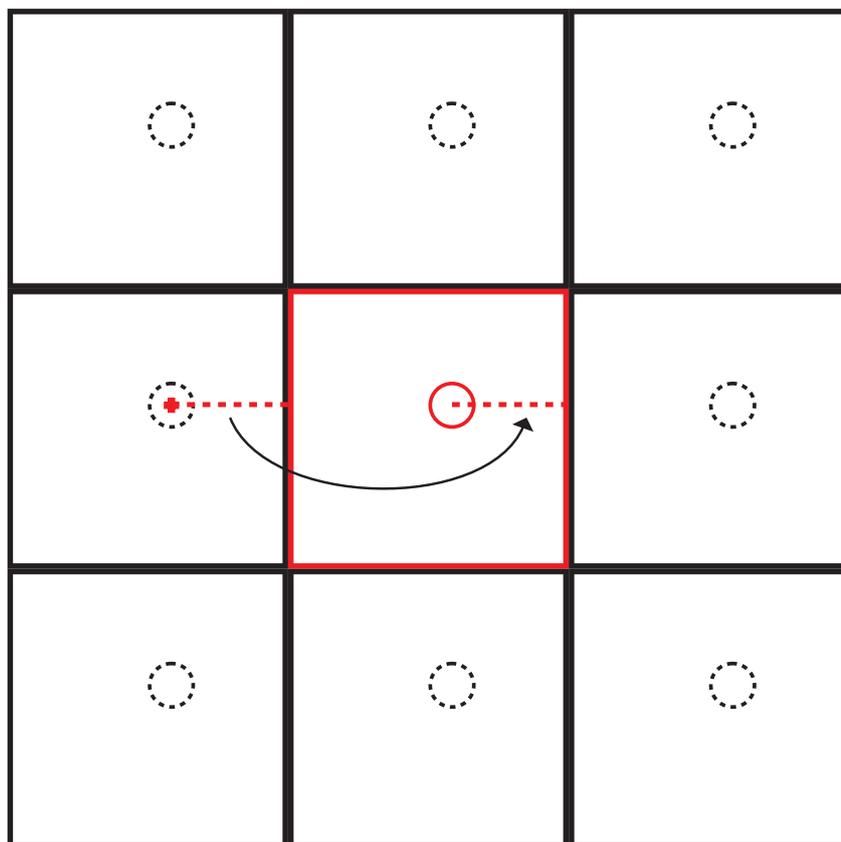
```
private static int LoopCoordinate(int coord, int terrainSize)
{
    if (coord < 0)
    {
        coord += terrainSize;
    }
    else if (coord >= terrainSize)
    {
        coord -= terrainSize;
    }

    return coord;
}
```

Concrètement, cela donne donc l'impression que le joueur peut modifier les terrains imposteurs et permet d'avoir une illusion parfaite de terrain infini.



## Difficultés à surpasser



Clonage de la modification de terrain



Zone de modification de terrain



Click de la souris



Terrain d'origine



Distance reporté sur le terrain d'origine



Grâce à cet ajout, le terrain crée une illusion parfaite de terrain infini.



## Difficultés à surpasser

### Difficulté 3 : L'optimisation des FPS

Modification du Terrain:

Nous avons rencontré la plupart des problèmes de FPS sur le script permettant la modification InGame du terrain qui a dû recevoir beaucoup d'optimisations au fur et à mesure du développement du jeu. Ce script prend en référence un point du terrain obtenu grâce à un raycast entre

la position de la souris du joueur vers l'avant.

Cette position sur la HeightMap du terrain sert ensuite à construire un cercle qui permet la modification des différents points de cette map de la manière suivante:

1. On commence par définir les limites du cercle en calculant les valeurs minimales et maximales pour x et z sur la base du centre et du rayon du cercle.

```
int minX = centerX - radius;  
int maxX = centerX + radius;  
int minZ = centerZ - radius;  
int maxZ = centerZ + radius;
```

2. Ensuite, on parcourt chaque point dans le carré défini par ces limites.
3. Pour chaque point, il faut calculer la distance entre le point et le centre du cercle. Si cette distance est inférieure ou égale au rayon du cercle, cela signifie que le point est à l'intérieur du cercle.

```
for (var z = minZ; z <= maxZ; z++)  
{  
    float distanceZ = z - centerZ;  
    float distanceZSqr = distanceZ * distanceZ;  
  
    for (var x = minX; x <= maxX; x++)  
    {  
        float distanceX = x - centerX;  
        float distanceSqr = distanceZSqr + distanceX * distanceX;  
        if (distanceSqr <= radius * radius)  
        {
```



## Difficultés à surpasser

4. On normalise ensuite cette distance en la divisant par le rayon du cercle pour obtenir une variable de proportion allant de 0 à 1.

```
float normalizedDistance = Mathf.Sqrt(distanceSqr) / radius;
```

5. On stocke ensuite dans une variable la hauteur actuelle du terrain à ces coordonnées.

6. Puis on modifie cette variable de hauteur en ajoutant la force de modification, multipliée par la variable de proportion obtenue à l'étape 4 et par le temps écoulé depuis la dernière image.

Raycast de la souris



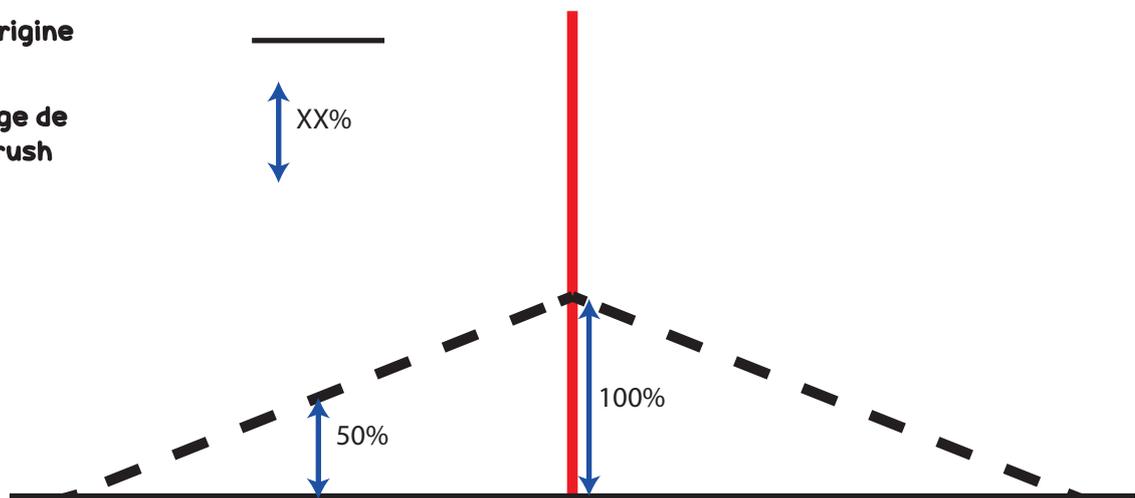
Modification du terrain



Terrain d'origine



Pourcentage de force de brush



## Difficultés à surpasser

Toutes ces étapes permettent à la modification d'être plus forte au centre du cercle et de diminuer ensuite progressivement vers l'extrémité de ce dernier.

```
float height = _heights[loopZ, loopX];  
height += strength * Time.deltaTime * (1 - normalizedDistance);  
height = Mathf.Clamp01(height);  
  
_heights[loopZ, loopX] = height;
```

La majorité de la puissance de calcul était utilisée pour faire des appels / assignations récurrentes à l'entièreté de la heightmap du terrain, ce qui, pendant le prototypage, avait pour effet de grandement réduire les FPS, même sur des configurations de PC Haut de gamme.

De nombreuses itérations d'optimisation ont dû être faites afin d'arriver à une solution convenable, l'objectif étant alors de rendre le jeu accessible au plus de configurations possibles. Notre objectif était 30 FPS stables.

L'objectif était donc de réduire au maximum le nombre d'opérations mathématiques nécessaires au

fonctionnement du brush mais aussi de les rendre les plus efficaces possible pour éviter un maximum de calculs superflus.

Au final, nous avons regroupé des opérations étant effectuées plusieurs fois, comme le montre le screen ci dessous: il montre le script principal actuel ayant été optimisé.



## Difficultés à surpasser

```
private void ModifyCircularBrush(int centerX, int centerZ, int radius, float strength)
{
    int terrainSize = _heightmapWidth;
    bool isLoopXZero = false;
    bool isLoopZZero = false;

    int minX = centerX - radius;
    int maxX = centerX + radius;
    int minZ = centerZ - radius;
    int maxZ = centerZ + radius;

    for (var z = minZ; z <= maxZ; z++)
    {
        float distanceZ = z - centerZ;
        float distanceZSqr = distanceZ * distanceZ;

        for (var x = minX; x <= maxX; x++)
        {
            float distanceX = x - centerX;
            float distanceSqr = distanceZSqr + distanceX * distanceX;
            if (distanceSqr <= radius * radius)
            {
                float normalizedDistance = Mathf.Sqrt(distanceSqr) / radius;

                int loopZ = LoopCoordinate(z, terrainSize);
                int loopX = LoopCoordinate(x, terrainSize);

                float height = _heights[loopZ, loopX];
                height += strength * Time.deltaTime * (1 - normalizedDistance);
                height = Mathf.Clamp01(height);

                _heights[loopZ, loopX] = height;

                // Check si loopZ or loopX est à 0
                if (loopZ == 0)
                {
                    isLoopZZero = true;
                }
                if (loopX == 0)
                {
                    isLoopXZero = true;
                }
            }
        }
    }

    // Copy à la bordure opposée seulement si nécessaire
    if (isLoopZZero)
    {
        for (int x = 0; x < terrainSize; x++)
        {
            _heights[terrainSize - 1, x] = _heights[0, x];
        }
    }
    if (isLoopXZero)
    {
        for (int z = 0; z < terrainSize; z++)
        {
            _heights[z, terrainSize - 1] = _heights[z, 0];
        }
    }

    _terrainData.SetHeights(0, 0, _heights);
}
```

Méthode principale actuelle



# **Charte Graphique**



## Intentions et Inspirations

L'Univers de Make Me a World est basé sur l'imagination d'un enfant s'amusant à déformer de la pâte à modeler. Le joueur doit être ainsi replongé dans son enfance et l'aspect «imaginaire» de l'environnement est très présent notamment par l'évolution de celui-ci grâce aux actions du joueur.

C'est un jeu facile à prendre en main mais qui demande un peu de challenge quant au contrôle de la balle. L'environnement et ses détails ne doivent donc pas entrer en conflit avec la lisibilité des mouvements de la balle, sans pour autant que celle-ci ne devienne le centre du jeu, qui se veut contemplatif.

Nous souhaitons donc mettre en avant un monde enfantin, coloré, qui soit harmonieux, doux, et collant parfaitement avec l'aspect satisfaisant.

En faisant des recherches sur les différentes productions utilisant déjà de la pâte à modeler, nous sommes tombés sur «Wallace et Gromit» et «Chicken Run», films célèbres relatant les aventures de personnages évoluant dans un monde de pâte à modeler. Nous avons pu remarquer les formes très rondes et déformables des personnages mais également de leur environnement, donnant immédiatement le ressenti de pouvoir être déformé et donnant un aspect innocent et enfantin.



Wallace & Gromit



Il était également important pour nous de rester le plus lisible et minimaliste quant aux informations disponibles sur l'écran du joueur. Il doit pouvoir voir en priorité l'environnement et la balle, sans être dérangé.



Chicken Run

Les adjectifs qualifiant notre direction artistiques sont :

Doux  
Rond  
Innocent  
Enfantin  
Lisse



## Moodboard Général



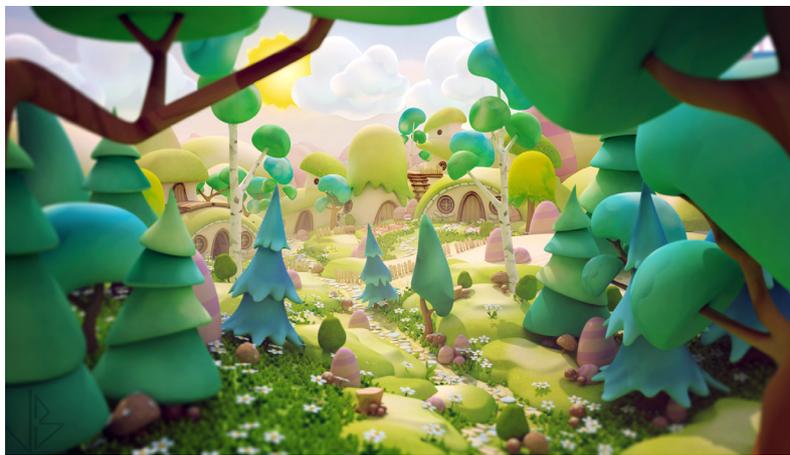
3d landscape with fairytale elements, Freepik



Motion Land Segment, William Foster



3d rendering of ecosystem, Freepik



Margaritka, Samuel St



La Terre Paysage De Style Mignon, 3dimension2000



TeplicaWorkshop



La Terre Paysage De Style Mignon, style pate à modeler, 3dimension2000



3D Forest Pack, LomelRueca



## Terrain

### Intentions

L'aspect esthétique le plus important de *Make Me A World* est le terrain. Le joueur peut le déformer à sa guise et il était important pour nous de garder cet aspect "pâte à modeler" et grand espace de verdure lisse. Il était donc crucial pour nous que l'immersion du joueur soit optimale, sans pour autant nuire à la lisibilité de ses actions et de son environnement, notamment les mouvements de la balle.

Pour rester sur nos intentions de direction artistique, le terrain du jouet

doit être d'une couleur douce et peu saturée, une couleur non-agressive qui n'attire pas trop l'attention.

Enfin, la texture et la couleur du terrain doivent permettre au joueur de bien apercevoir les dénivelés, qu'il puisse comprendre la topologie du terrain, sans effort.

### Référence principale



Beautiful green hills and canyons in iceland aerial river, StudioWorkStock

Nous avons tiré nos premières inspirations directement des grands espaces de collines verdoyantes d'Islande.

Avec leur aspect lisse, rond et à l'état naturel, sans que l'homme n'y touche, elles nous ont permis de déterminer immédiatement les couleurs de notre terrain : des tonalités de vert peu saturé.



# Terrain

## Moodboard



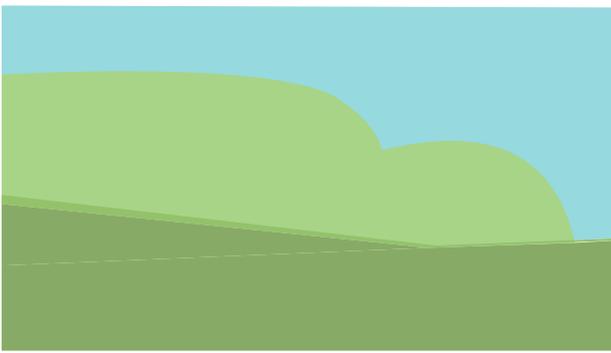
The Shire, Concept art de David Ortiz



Texture des collines, Super Mario 3D World



collines et herbe, Flower (2009)



Chicken Run



## Terrain

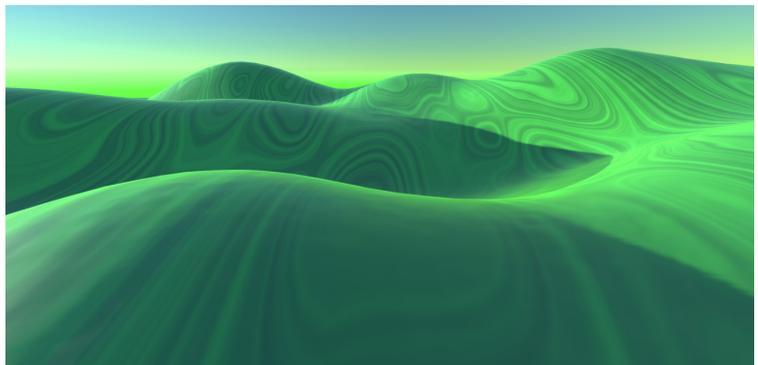
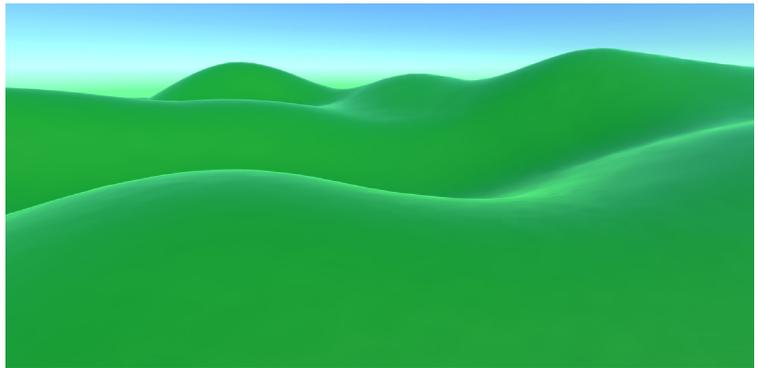
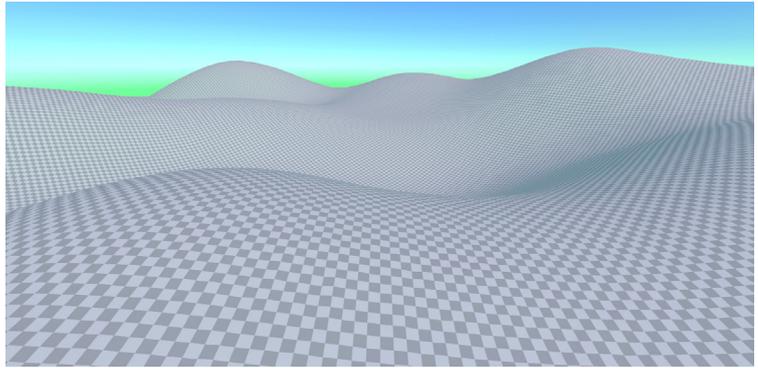
### Processus créatif

Nous avons commencé la création du terrain en lui appliquant un aplat de couleur vert, tout simple.

Nous nous sommes vite rendus compte qu'un simple aplat de couleur ne suffirait pas à donner l'aspect «pâte à modeler» à ce terrain, mais également que cela nuirait grandement à la lisibilité de la topologie de celui-ci. Les joueurs avaient du mal à se repérer et à apercevoir les dénivelés.

Nous avons donc fait un shader que nous avons appliqué sur le terrain, lui permettant d'avoir des formes rondes sur ses zones hautes ou basses semblables aux couleurs obtenues lorsque l'on mélange plusieurs couleurs de pâte à modeler.

Ce shader nous a permis de rendre le terrain bien plus lisible tout en lui permettant de coller bien plus avec nos intentions d'obtenir un résultat ressemblant à de la pâte à modeler.





## Balle et objets

### Intentions

La balle est l'un des éléments central de Make Me A World. Elle est le guide du joueur et il est important pour nous que sa lisibilité soit impeccable. Il s'agit également du seul élément de notre jouet qui ne soit pas fait de pâte à modeler. Son apparence doit donc se défaire de l'aspect mou et modelable du reste des éléments.

Le terrain étant vert, nous avons choisi de colorer la Balle en un Rouge Vif, couleur complémentaire du Vert, lui permettant de détonner avec les tons de l'environnement peu saturé et permettant une lisibilité immédiate, et ce peu importe la situation. Afin de l'éloigner encore plus de l'aspect pâte à modeler, nous lui avons ajouté un aspect métallique, contrastant avec la texture mate du reste des éléments.

Nous avons 4 objets différents sur le terrain : les arbres, les moutons, les arches et les abeilles. Ces quatre objets doivent être ronds, amusants et doivent avoir un aspect friendly et malléable. Leurs couleurs doivent rester dans les mêmes tons peu saturés voir pastels du terrain afin de garder l'aspect pâte à modeler.

Les animaux choisis sont des animaux communs que chaque enfant a déjà rencontré ne serait-ce que dans des livres. Ce sont des animaux faciles à reproduire en pâte à modeler et ils peuvent ainsi donner un sentiment de nostalgie.

Afin de contraster avec la couleur du sol, les feuilles des arbres ne sont pas vertes mais orange-rouge, donnant un effet Automne sur l'environnement du jouet.

---

### Choix du Format

Afin de mettre en valeur l'environnement du jouet, nous avons opté pour un format paysage standard. C'est un format permettant d'avoir le meilleur angle de vue sur le terrain tout en gardant en focus notre balle.

## Balle et objets

### Références



Earth day illustration, Freepik



TeplicaWorkshop



Image du film Le Lorax



Angry Bee, Amanda Suita



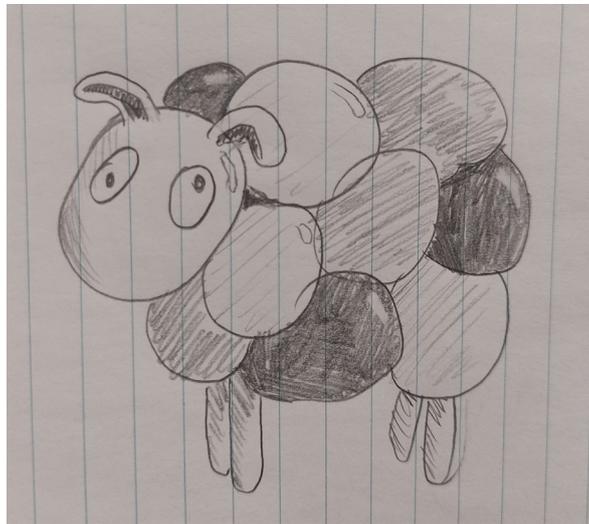
Personnage du film Shaun Le Mouton



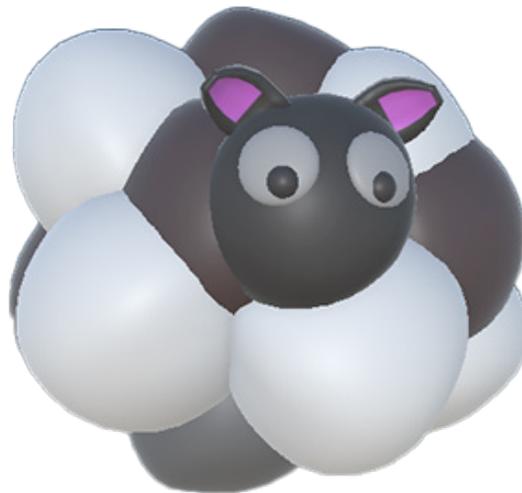
## Balle et objets

### Processus créatif

#### Premières itérations



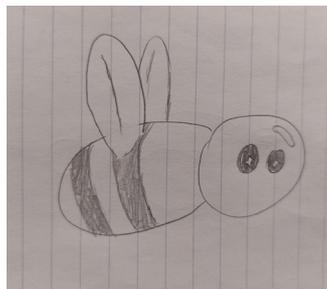
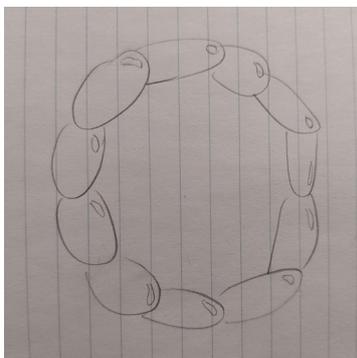
#### Résultat Final



## Balle et objets

### Processus créatif

#### Premières itérations



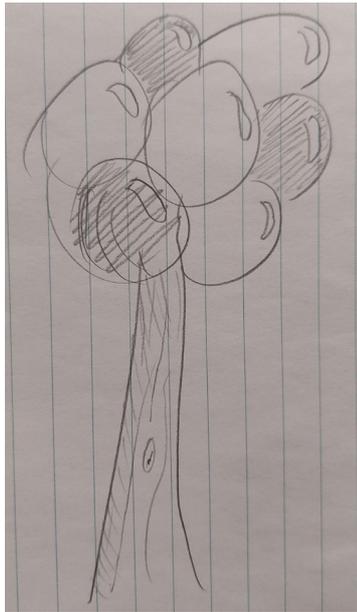
#### Résultats Finaux





## Balle et objets

### Processus créatif





## Menus

### Intentions

La seule UI de *Make Me A World* est le feedback intrinsèque de la taille de la Brush. Nous avons souhaité faire en sorte que celui-ci soit parfaitement lisible pour que le joueur sache exactement la taille de sa brush à n'importe quel moment.

Pour les menus du jouet, nous avons voulu créer des menus aux boutons très ronds avec des couleurs peu saturées, rappelant directement les couleurs de l'environnement du jeu. Le fond des menus permet toujours d'apercevoir le terrain, que ce soit grâce à une illustration dans les menus principaux ou à un fond transparent dans le menu pause.

Les boutons sont toujours de couleur rouge pour se détacher du vert, couleur du Terrain.

### Références



All of You (2020)



Dorf Romantik (2021)

## Menus

### Processus créatif

Premières itérations



Résultat Final

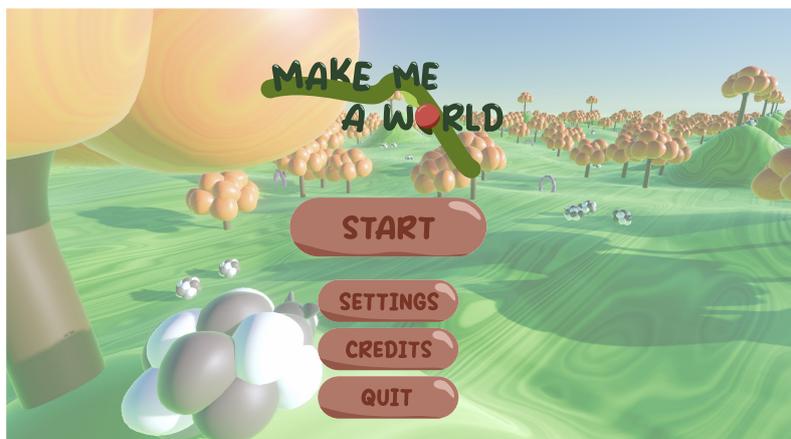




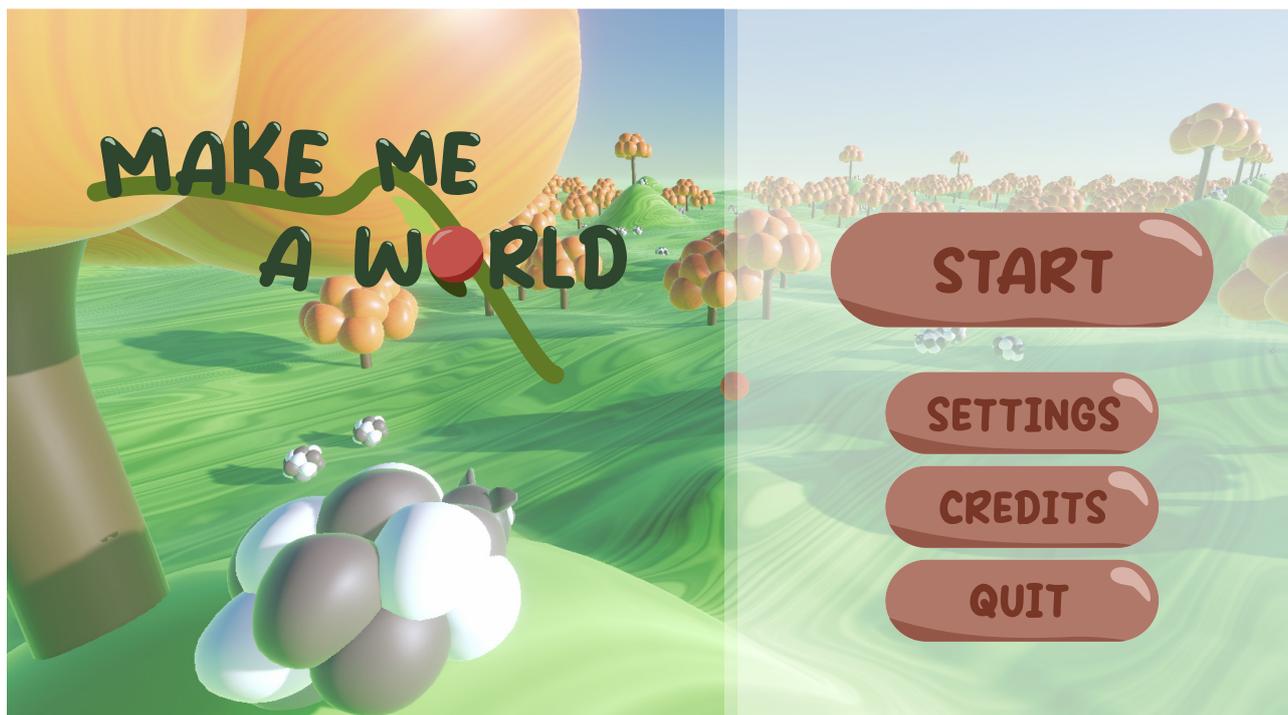
# Menus

## Processus créatif

Premières itérations



Résultat Final

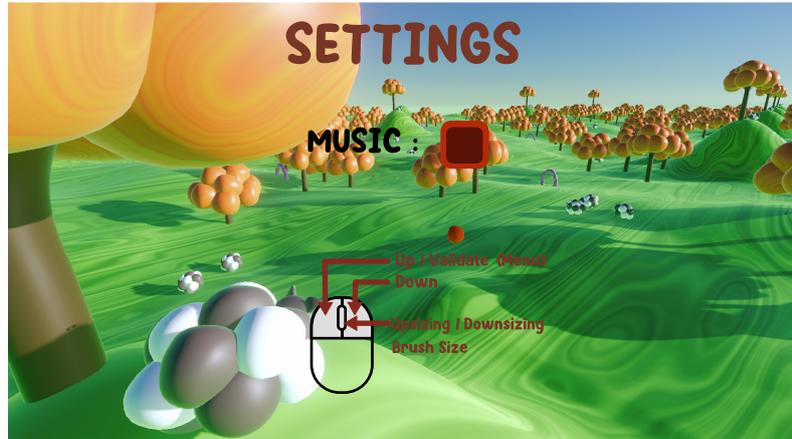




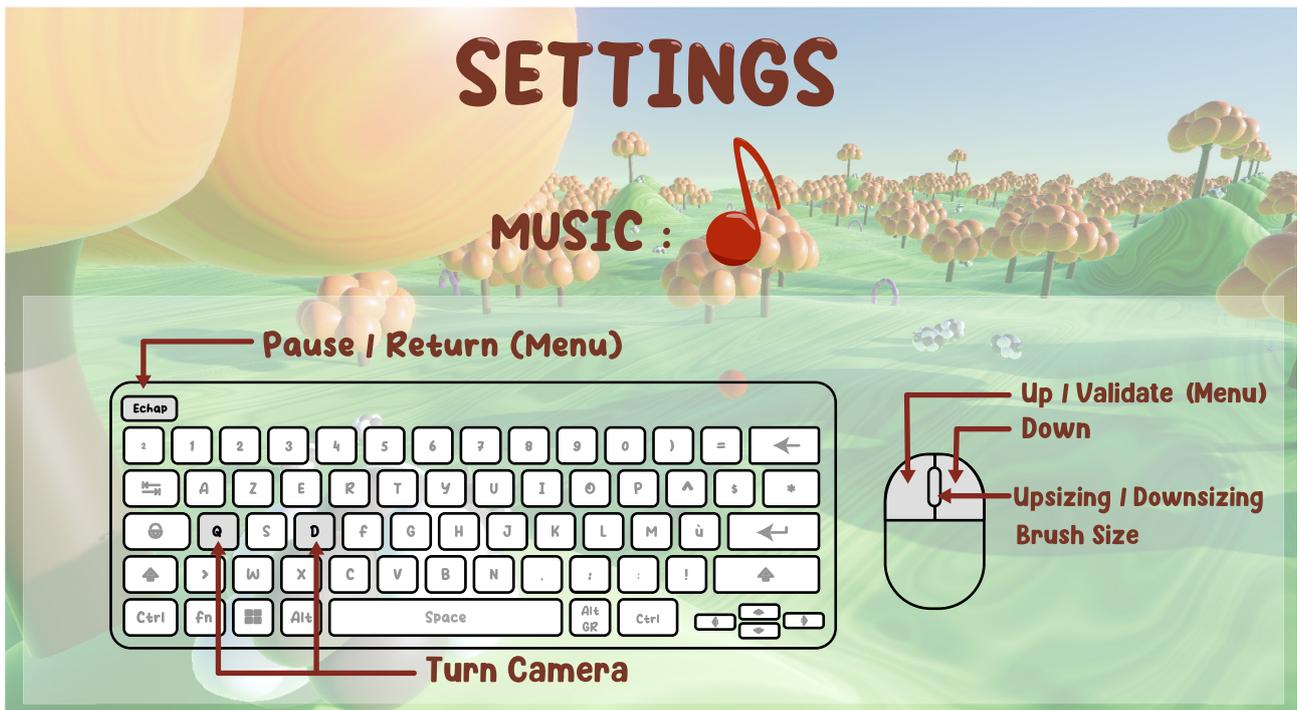
## Menus

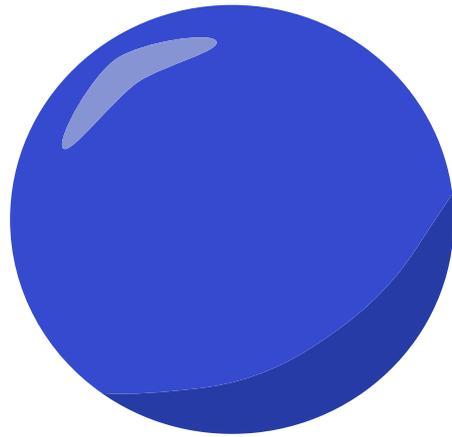
### Processus créatif

#### Premières itérations



#### Résultat Final





# Charte Sonore



## Intentions :

Dans *Make Me A World*, nous avons souhaité utiliser le sound design comme un moyen de contribuer à développer l'immersion, mais aussi d'améliorer le game feel, qui est un aspect central de notre jouet.

C'est un jouet contemplatif, calme et satisfaisant et l'immersion sensorielle nous était donc très importante. Nous voulions créer un sentiment d'apaisement, qui encouragerait le joueur à prendre le temps d'explorer la map sans se sentir pressé par le temps ni en danger. Nous souhaitions le faire voyager en enfance et lui permettre, pendant un moment, d'oublier où il se trouve et de se laisser emporter par l'ambiance de notre œuvre.

Le Sound Design est donc un grand point de notre design et il nous a permis un développement du Game Feel optimal, notamment grâce à un design évolutif pour aider et guider le joueur, par exemple dans notre recherche d'un

moyen d'attiser la curiosité du joueur pour l'encourager à se diriger vers les points d'intérêts de la map ou encore pour lui donner des informations sur ses inputs.

La direction artistique visuelle étant centrée autour de la matière de la pâte à modeler, nous souhaitions utiliser des sons doux faisant appel à l'imagination, aux souvenirs de l'enfance et incitant à la rêverie. Certaines des actions proposées au joueur étant semblables à ce qu'il pourrait expérimenter dans un rêve (comme la modification du terrain), nous avons souhaité les distinguer de façon claire de ceux faisant davantage appel à des registres familiers comme les bruits d'animaux et l'ambiance sonore générale par la création de sons s'éloignant de tout sens de réalisme.

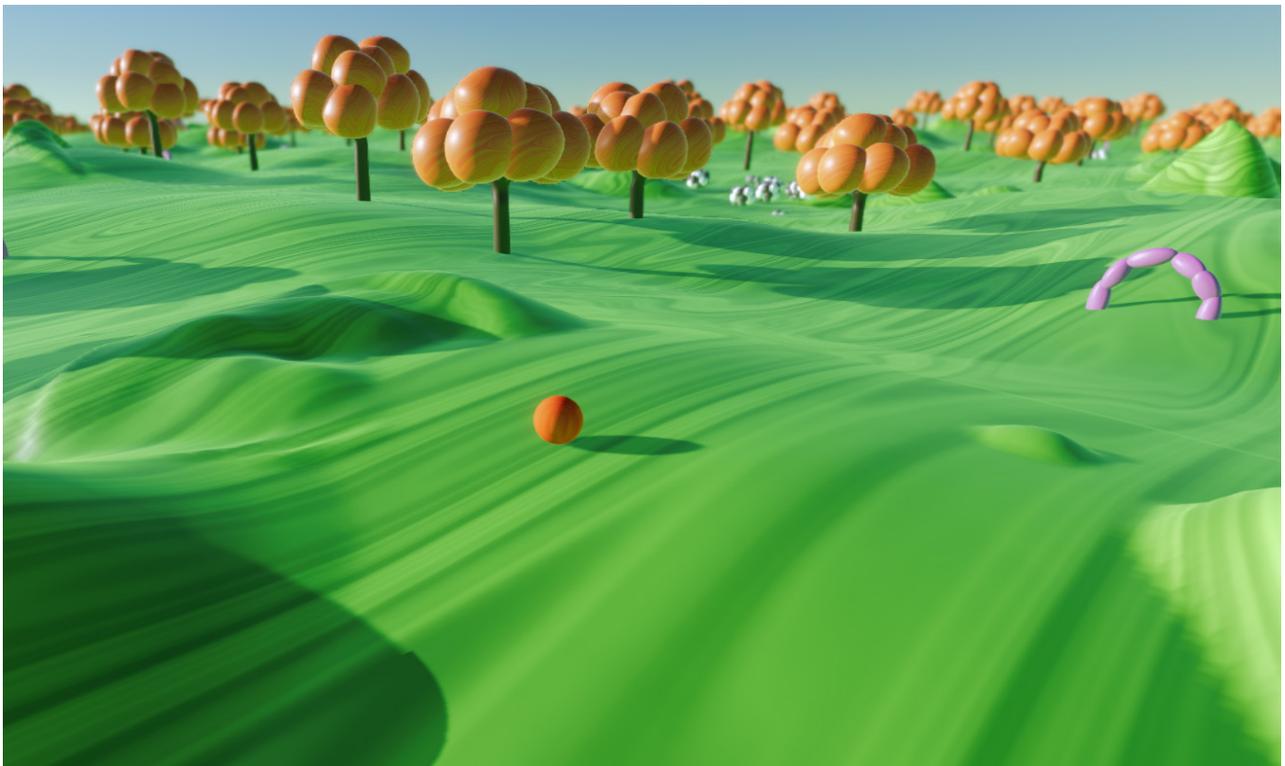


## Intentions :

L'utilisation de musique nous a semblé particulièrement pertinente pour accentuer les aspects de douceur, d'imagination et de rêverie, tout en y ajoutant une touche de poésie. Notre objectif était de créer une ambiance réconfortante et de donner une identité forte à notre jouet. Elle contribue également de manière centrale à faire passer des émotions et façonner le game feel.

Pour la musique, nous avons fait le choix d'une composition originale, car cela nous a permis de nous assurer que celle-ci reste au plus près de nos intentions tout au long du processus de création du jouet.

Nous avons décidé de l'utiliser comme un outil pour suggérer une histoire d'aventure, par la présence d'une mélodie qui, bien que présente, semble lointaine et donne la sensation au joueur qu'elle est encore balbutiante et incomplète. Il s'agit d'une invitation pour le joueur à écrire sa propre histoire et lui laisser l'espace nécessaire pour s'approprier son expérience. L'utilisation d'une gamme majeure et d'accords consonants nous a semblé approprié pour faire passer un sentiment d'harmonie et de calme.



## Références

L'inspiration principale que nous avons au niveau des bruitages liés à la balle et aux collisions sont les films d'animation de Nick Park réalisés en pâte à modeler, notamment «Wallace et Gromit».

Le personnage du chien (Gromit) est celui qui nous intéresse tout particulièrement, car il est l'un des seuls à ne pas porter de chaussures. Les bruits de ses pieds et le sol sont donc représentatifs du bruit que fait la pâte à modeler lorsqu'elle entre en collision avec un objet.

Nous avons également repris l'esprit de certains bruits "cartoonesques" utilisés lors du gonflement de certains objets dans les films.



Wallace & Gromit



Everything

Comme mentionné précédemment, la musique est une composition originale créée pour le jouet. Elle se situe à la frontière entre des bandes sons comme celle du jeu «Everything», dont l'effet mystique et grandiose donne au joueur l'envie d'explorer grâce à l'utilisation de notes longues et de beaucoup de réverbération, et un jeu comme «Marble Saga : Kororinpa» qui associe des accords majeurs et des mélodies entraînantes que l'on pourrait qualifier d'enfantines.



Marble Saga : Kororinpa



## Asset List

Nos sons peuvent être regroupés en trois catégories : Les sons relatifs à l'UI, les sons liés au comportement de la balle, et les sons liés à l'environnement. Nous avons utilisé une asset list afin de faire l'inventaire des sons nécessaires, leur description, s'ils doivent être loop, s'ils sont spatialisés, s'ils ont été intégrés ou encore leur priorité.

### UI

**Hover souris :** Nous avons choisi d'utiliser un son musical assez discret, qui rappelle un marimba. Nous avons coupé l'attaque du son pour le rendre plus doux et ainsi coller à l'ambiance enfantine et rêveuse, tout en respectant l'aspect doux produit par la pâte à modeler.

**Clic souris :** Pour rendre le feedback du clic, nous avons utilisé un son ayant davantage d'attaque que celui du hover souris. Il s'agit d'un "pop" musical dont le pitch est plus bas (d'une quarte) lorsqu'on effectue une action d'annulation ou de retour en arrière.



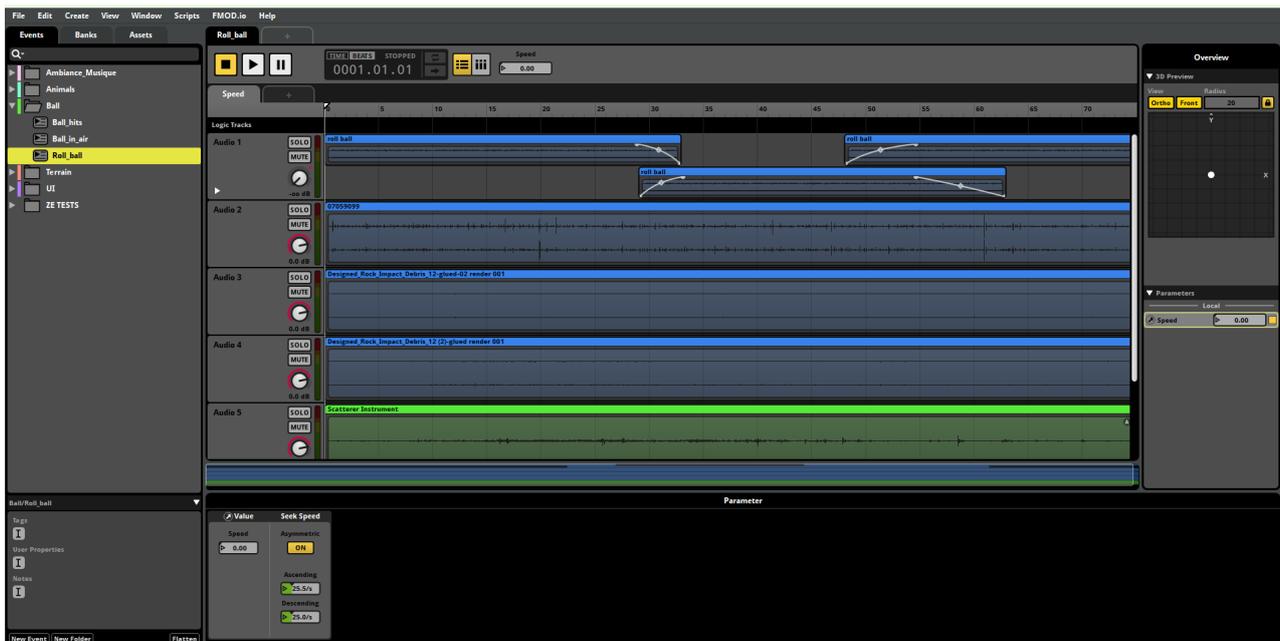


## Asset List

### Balle

**Sons de collision :** Les objets étant tous en pâte à modeler, le son de collision de la balle avec les objets de l'environnement est un bruit d'impact sourd semblable aux pas du chien dans «Wallace et Gromit».

**Son de la balle qui roule :** Le bruit de roulement de la balle est composé de nombreux bruits différents que l'on peut séparer en deux catégories : des bruits continus qui servent à reproduire l'action de roulement de la balle sur la surface, et des bruits plus saccadés et humides pour reproduire l'effet de décollement de la balle de la surface collante de la pâte à modeler.



Screen fmod Balle qui roule

**Son de la balle en l'air :** Lorsque la balle décolle de manière significative du terrain et s'élève, un bruit de vent se joue lorsqu'elle est en l'air. Cela accentue la sensation de mouvement et de hauteur.

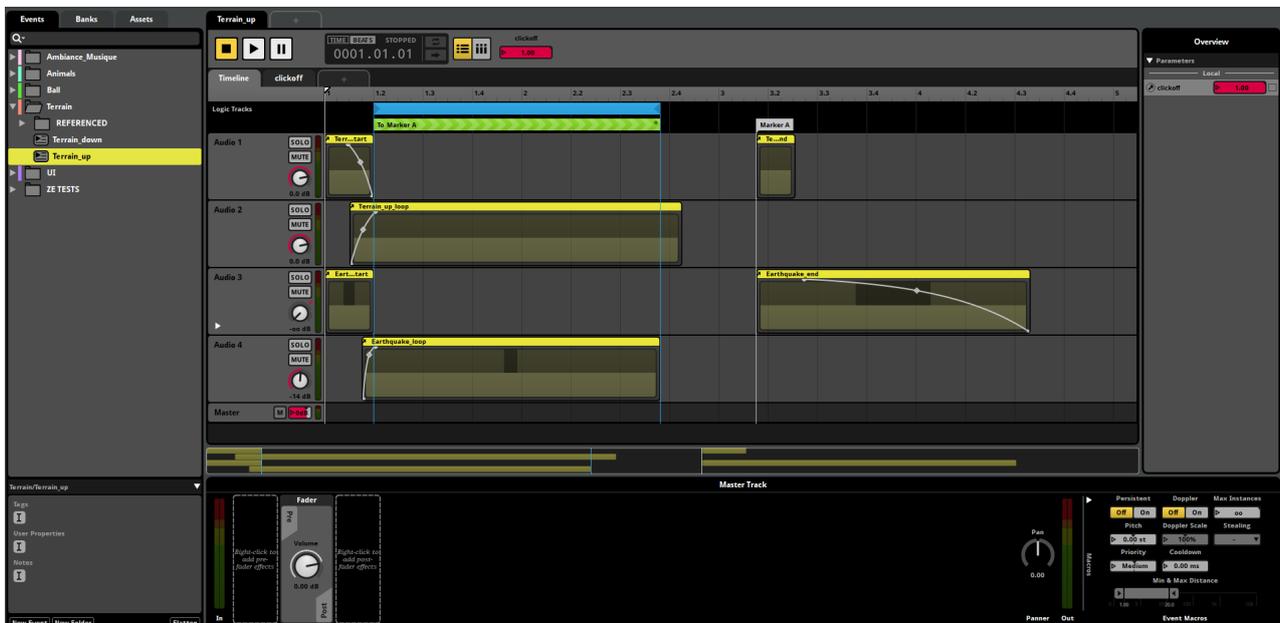


## Asset List

### Univers :

**Animaux :** Les trois animaux présents dans le jouet (mouton, abeille et grenouille) émettent des bruits qui leur permettent d'être localisés par le joueur. Cela lui permet de prendre conscience de leur présence et de savoir où ils se trouvent par rapport à la position de la balle s'il souhaite la diriger vers eux. A la différence des autres animaux, le mouton a comme particularité d'émettre un cri lorsque la balle entre en collision avec lui. Cela a pour but d'accentuer la surprise provoquée par le détachement des boules de laine du mouton et de renforcer l'aspect comique de la situation.

**Terrain :** Le son que produit le terrain lorsqu'il monte ou descend est composé d'un ballon qui se gonfle et d'un bruit de tremblement de terre. Il est lié au clic du joueur et comporte trois parties : un début, une boucle et une fin. Lorsque le joueur clique, le début du son se joue. Ensuite, la boucle centrale se joue tant que le joueur reste appuyé sur le bouton. Lorsqu'il relâche le clic souris, la troisième partie du son se lance, ce qui le termine. Le pitch augmente lorsque le terrain monte, et est réduit lorsque le terrain descend. Ce son plutôt "cartoonesque" est un clin d'œil aux bruits liés aux vêtements qui se gonflent ou qui craquent dans «Wallace et Gromit».



Screen Fmod Terrain

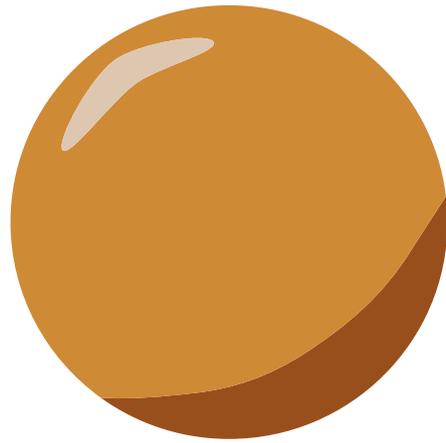


## Asset List

### Univers :

**Ambiance :** L'ambiance est composée du son du vent dans les arbres, de chants d'oiseaux et de grillons. Le but est de donner l'impression d'un monde regorgeant de vie. Cela augmente la crédibilité de notre univers et rend plus vraisemblable la présence des animaux. L'utilisation de sons familiers à tous et liés aux saisons chaudes du printemps et de l'été a pour objectif de renforcer le sentiment de bien-être et de relaxation et faire appel à des souvenirs d'enfance (par exemple de voyages ou de vacances).

**Musique :** La musique est composée de plusieurs pistes qui se superposent de façon différente selon les événements qui ont lieu en jeu. Lorsque la balle arrive sur le terrain et/ou tant qu'elle est immobile (c'est à dire que sa vitesse est égale à zéro), une seule note se joue, pour donner l'impression que le temps est en suspens. Dès que la vitesse de la balle est supérieure à zéro, une nouvelle piste se lance comprenant des accords majeurs. Cela a pour but de marquer le début de l'action du joueur et accentuer sa sensation d'avoir un impact sur le monde qu'il explore. Ce changement marque le début de son aventure. Ensuite, lorsque la balle se rapproche de zones qui contiennent des points d'intérêt, d'autres pistes s'ajoutent, incitant ainsi le joueur à se diriger vers ces espaces et à interagir avec les objets qui les composent.



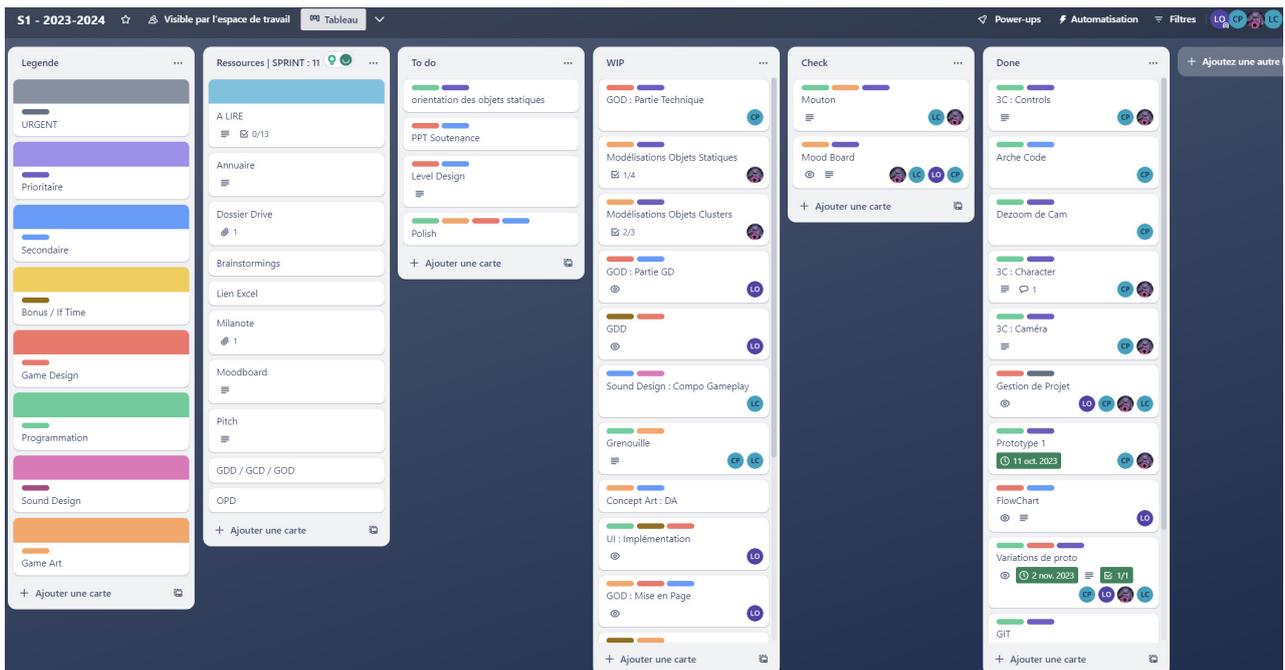
# Organisation

## Méthode Agile et Trello

Nous avons choisi d'utiliser la méthode Agile afin de s'assurer une communication utile et efficace. Nous avons organisé des sprints de deux semaines durant lesquels nous nous sommes donnés des objectifs à atteindre avant une journée de réunion en présentiel (à l'exception de 3 réunions en distanciel) tous les mardis pour contrôler les avancements, discuter des améliorations et diverses questions que nous avons et des objectifs du sprints suivant.

Afin de s'assurer des objectifs et de leurs avancées durant les sprints, nous avons utilisé Trello, site permettant de faire des checklists rapides et à plusieurs.

Nous avons donc créé une légende afin de séparer les tâches par types (Game Art, Game Design, Game Programming et Sound Design) et surtout par priorité afin d'être le plus efficace dans notre planification et dans notre travail.



Matrice de Caillols

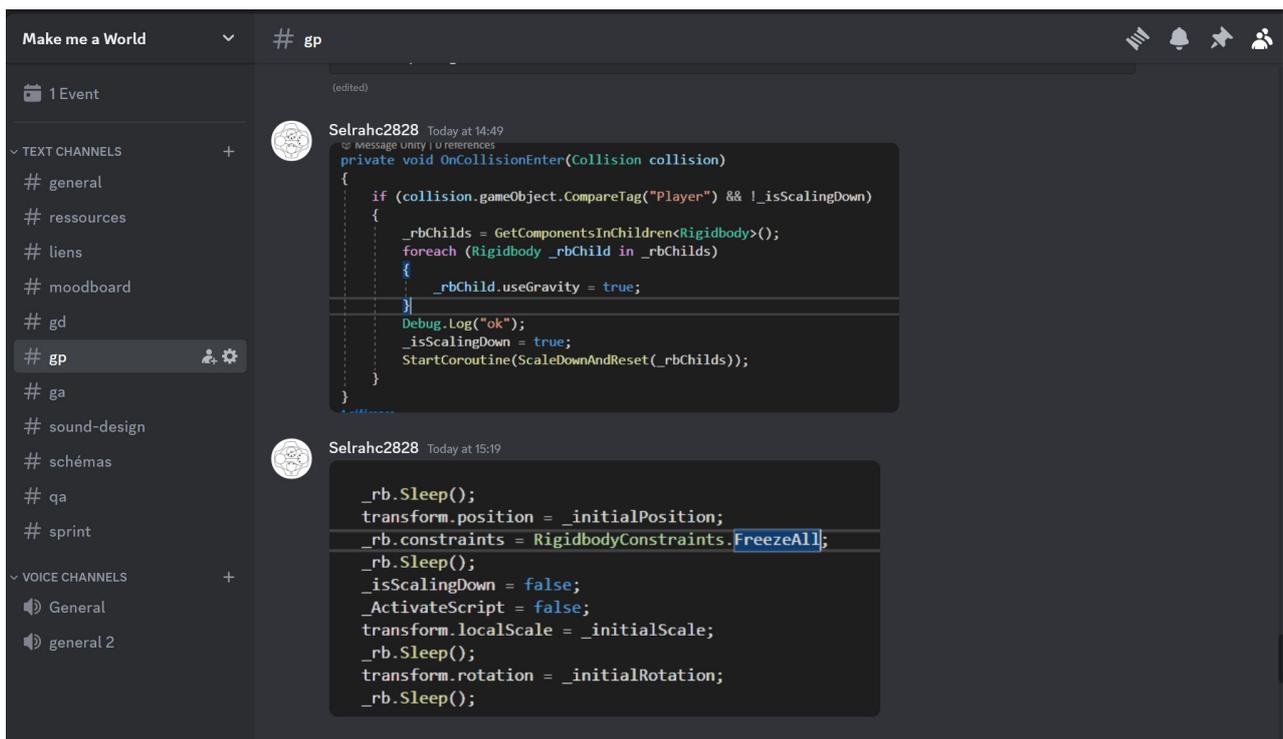


## Communication

### Discord

Pour assurer une communication directe et permanente, nous avons créé un serveur sur discord, divisé en plusieurs Channels, pour pouvoir retrouver les discussions plus facilement. Chaque membre du serveur possède deux à trois rôles permettant au reste du groupe d'envoyer des

notifications directement aux personnes concernées et 2 channels vocaux nous permettant de discuter directement de plusieurs sujets en même temps.



Matrice de Caillóis



## Communication

### Google Drive

En plus du server Discord, nous avons créé un Google Drive divisé en 4 grandes catégories : Backups (pour les différentes versions du projet UNITY sauvegardées), Direction Artistique (pour garder tous les sprites, modèles et textures mais également la documentation et le brainstorming de direction artistique), Game Design (pour garder le document dans lequel

se trouve tout notre brainstorming depuis le début du projet, des backups du GOD et GDD, des images d'anciennes variations, les backups du fichier illustrator contenant tous nos Schémas et nos références de Game Design) et Sound Design (pour les différents fichiers sons, la documentation sonore, le brainstorming de direction sonore et les images de références).

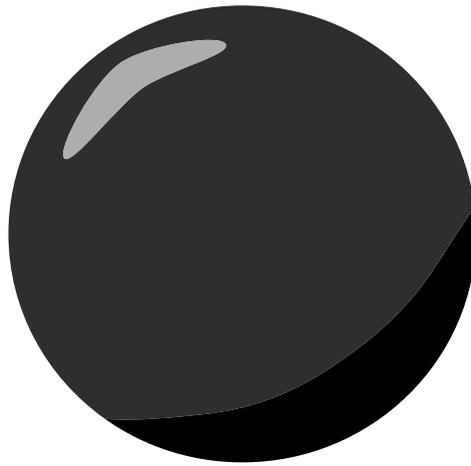


Matrice de Caillois

### GIT HUB

Nous nous sommes assurés de toujours garder le Git à jours et avons utilisé 4 branches afin d'être tranquilles : Une branche MAIN, sur laquelle ne se trouve que des versions «milestones» stables du projet, une branche DEV, sur laquelle toutes les features et les

visuels du jeu ont été directement implémentés, et une branche SOUND, nous permettant de faire l'intégration du son sans se poser la question de si quelqu'un travaille déjà sur le projet ou non.



# ANNEXES

## QA

## Testeu.r.se : A

Age : 25 ans

Type : Mid Core

- Le.a joueu.r.se n'a pas bien compris comment contrôler la caméra avec la barre espace + bouger la souris. Iel essayait de bouger la Brush et la caméra tournait en meme temps. Cela a provoqué de la confusion et une difficulté à sortir de la situation dans laquelle se trouvait la balle.

- Iel n'a pas cherché à modifier la force de la Brush mais à utiliser plusieurs tailles de Brush différente.

- Iel n'arrivait pas à prendre de la vitesse

- Iel voulait faire entrer la balle en collision avec les objets et à réussi (difficilement). Iel a essayé de passer dans une arche mais n'a pas réussi.

**Changements suite au test :**

- Augmentation de la taille des troncs des arbres, des moutons, des arches.
- Changement des contrôles de la caméra «joueur» de Space+souris aux touches Q & D.
- Modification de la Balle (friction, poids, gravité) pour lui permettre de prendre de la vitesse plus facilement

## Testeu.r.se : B

Age : 21 ans

Type : Mid Core

- Le.a joueu.r.se a compris tous les contrôles mais a eu du mal à faire ralentir la balle.

- En faisant prendre trop de vitesse à la balle, la caméra «vitesse» c'est mise trop loin de la balle et le joueur pouvait voir les terrains imposteurs et la téléportation.

- Le.a joueu.r.se souhaitait se cogner contre les Moutons. Il a réussi après 5min d'essais.

- Le.a joueu.r.se a voulu passer sous une arche. Il a réussi facilement mais la balle a pris trop de vitesse.

- Aucune hésitation de la part du joueu.r.se quant aux contrôles mais le.a joueu.r.se a fait remarquer qu'il serait pratique d'avoir un rappel des touches quelque part.

**Changements suite au test :**

- Augmentation de la taille des Moutons
- Ajout d'un menu «CONTROLES»
- Modification de la Balle (friction, poids, gravité) pour lui permettre de ralentir au lieu de garder la même vitesse en permanence
- Modification de l'éloignement maximal de la caméra «Vitesse» pour éviter que les joueurs ne puissent voir la téléportation et les terrains imposteurs.



## QA

## Testeur.r.se : C

Age : 20 ans

Type : Mid Core

- Le.a joueu.r.se a eu des difficultés à appréhender les commandes, surtout pour ce qui est des virages.
- Le.a joueu.r.se a fait rentrer la balle dans le sol par accident en voulant lui faire prendre trop de vitesse. Iel a en suite tenté de reproduire cela.
- Le.a joueu.r.se a été projeté dans la direction opposée lorsqu'iel a percuté un mouton.
- Le.a joueu.r.se ne savait pas qu'iel pouvait modifier l'angle de caméra, une fois qu'iel l'a appris. Iel a eu des difficultés à le faire tout en jouant.

**Changements suite au test :**

- Augmenter la friction de la balle pour permettre un meilleur ralentissement dans les virages.
- Se questionner sur la possibilité d'inclure la balle qui rentre dans le terrain comme option de gameplay ou empêcher que cela se produise.
- Les interactions avec les moutons sont intéressantes, et le côté semi-aléatoire qui en résulte rend la chose plus fun.
- Le contrôle de la camera ne doit pas être sur la souris.

## Testeur.r.se : D

Age : 33 ans

Type : Casual

Le.a joueu.r.se a eu de grandes difficultés à contrôler la balle.

Le.a joueu.r.se est passé à travers un arbre en voulant le percuter.

Le.a joueu.r.se a constaté la téléportation.

Le.a joueu.r.se a pris trop de vitesse en passant par les arches et a fini par s'envoler.

**Changements suite au test :**

- Améliorer le contrôle de la balle.
- Donner aux arbres le même comportement à la collision que les moutons.
- Corriger les problèmes de téléportation.
- Réduire la propulsion des arches.

## QA

### Testeur.r.se : E

Age : 20 ans

Type : Mid Core

- Le.a joueu.r.se a mis du temps à contrôler la balle.
- Le.a joueu.r.se se plaint du peu de FPS en jeu.
- Le.a joueu.r.se n'a pas constaté la téléportation.
- Le.a joueu.r.se a ne prend plus assez de vitesse pour se balader tranquillement.

#### **Changements suite au test :**

- Optimiser afin d'améliorer les FPS du jeu.
- La téléportation marche bien.
- Augmenter un peu la propulsion des arches.

---

### Testeur.r.se : F

Age : 19 ans

Type : Mid Core

- Le.a joueu.r.se a du mal à contrôler la balle .
- Le.a joueu.r.se a trouvé que la collision avec les moutons est très satisfaisante .
- Le joueur a trouvé dommage de ne pas pouvoir interagir avec les abeilles .
- Le caméra pourrait être améliorée .

#### **Changements suite au test :**

- Correction du comportement des moutons.
- Rajouter un peu plus d'animaux pour garder cette sensation de vie.
- Améliorer la caméra



## QA

**Testeur.r.se : G**

Age : 18 ans

Type : Casual

- Le.a joueu.r.se a trouvé qu'il y avait trop d'obstacles sur le terrain.
- Le.a joueu.r.se est passé à travers une abeille en voulant la percuter.
- Le.a joueu.r.se ne s'est pas rendu compte de la téléportation.
- Le.a joueu.r.se s'est amusé à foncer dans les moutons à répétition.

**Changements suite au test :**

- Réduire le nombre d'obstacle sur le terrain pour rendre ce dernier plus navigable.
- Rendre plus clair le fait que les abeilles sont «non solides».
- Continuer dans le sens de l'aspect amusant des moutons.

---

**Testeur.r.se : H**

Age : 25 ans

Type : Hardcore

- Le.a joueu.r.se a rapidement compris les commandes mais à trouvé les virages compliqués, ainsi que les ralentissements.
- Le.a joueu.r.se a repéré la téléportation.
- Malgré sa maîtrise des commandes, le.a joueu.r.se a eu des difficultés à percuter un arbre.

**Changements suite au test :**

- Augmenter la friction de la balle pour permettre son contrôle.
- rendre la téléportation encore moins visible.
- Agrandir la taille des arbres.

## Améliorations

Dans le futur, nous souhaiterions que l'environnement dans *Make Me A World* devienne encore plus évolutif, notamment grâce à l'ajout de nouveaux animaux avec différents comportements, ou encore de briques de Gameplay faisant varier les situations, tel qu'un Bouncer faisant rebondir la balle.

Nous souhaiterions également faire en sorte que les objets du terrain *Mère* changent lors des téléportations et de l'aggrandir, permettant ainsi de ressembler encore plus à un terrain infini et à un monde à explorer et évoluant petit à petit.

Nous ajouterions également plus de cohérence et de réalisme dans les mouvements des animaux afin d'optimiser l'immersion du joueur.

Enfin, un meilleur comportement de la caméra lorsqu'elle rentre dans les obstacles est nécessaire. Nous permettrions à la caméra de faire disparaître l'objet dans lequel elle rentre.

## Remerciements

Le Corps Enseignant de l'ICAN Paris

Nos Testeurs QA

Les élèves de l'ICAN Paris

Nos Proches

Merci

**MAKE ME  
A WORLD**

The text "MAKE ME A WORLD" is rendered in a bold, dark green, rounded font. A green vine-like line starts under the 'M' of "MAKE", curves up to pass under the 'E' of "ME", then curves down to pass under the 'O' of "WORLD". A small green leaf is attached to the vine between "ME" and "WORLD", and a red berry is attached to the vine under the 'O' of "WORLD".