

# Rapport de la premiere soutenance

- PROJET S2 -

Tom CASCHERA

Felix CAPEL GOUDARD

Alexis MERLE

Hedi MOKRANI

Mars 2022

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Avancement du projet</b>	<b>3</b>
<b>3</b>	<b>Tom Caschera</b>	<b>4</b>
3.1	Generation Procedurales . . . . .	4
3.2	Pour la prochaine soutenance . . . . .	10
<b>4</b>	<b>Alexis Merle</b>	<b>11</b>
4.1	Travail réalisé . . . . .	11
4.2	Système de faim et vie . . . . .	11
4.3	Menu . . . . .	12
4.4	Inventaire . . . . .	13
4.5	Difficultés rencontrés . . . . .	14
<b>5</b>	<b>Hedi Mokrani</b>	<b>15</b>
5.1	Deplacements . . . . .	15
5.2	IA . . . . .	16
5.2.1	Se promener : . . . . .	17
5.2.2	Attaquer : . . . . .	17
<b>6</b>	<b>Felix Capel-Goudart</b>	<b>19</b>
6.1	Multijoueur . . . . .	19
6.2	Site web . . . . .	20
6.3	Prochainement . . . . .	21
6.4	Difficultés rencontrés . . . . .	21
<b>7</b>	<b>Conclusion</b>	<b>22</b>

# Chapitre 1

## Introduction

Le but pour cette première soutenance était d'avoir un environnement jouable avec un inventaire simple mais fonctionnelle et une map quasiment complète. En ayant pour objectif de pouvoir rajouter par la suite sur cette base les nouvelles fonctionnalités facilement. Le code devait donc être le mieux structuré possible pour être facilement modifiable par la suite tout en restant le plus clair et concis. Aussi, on a dû apprendre à travailler en groupe et s'organiser pour se répartir les différentes tâches entre nous.

# Chapitre 2

## Avancement du projet

Nous avons développé un algorithme de avec une IA de sanglier qui se déplace aléatoirement ainsi qu'une map avec génération procédurale et un inventaire. Le code est très structuré et divisé en de nombreuses classes et fonctions différentes pour le rendre compréhensible et efficace. Nous avons eu un problème lors de l'implémentation du code c dans Unity mais une bonne partie du travail a été fait. Le site Web a aussi été réalisé.

Une description plus détaillée est présenté ci dessous avec chaque tâche réalisée par chaque membre du groupe :

# Chapitre 3

## Tom Caschera

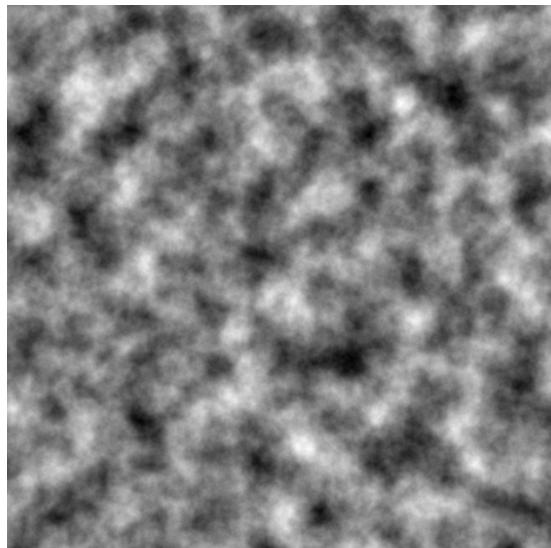
### 3.1 Generation Procedurales

Pour ma part, j'ai réalisé la génération procédurale de la map.

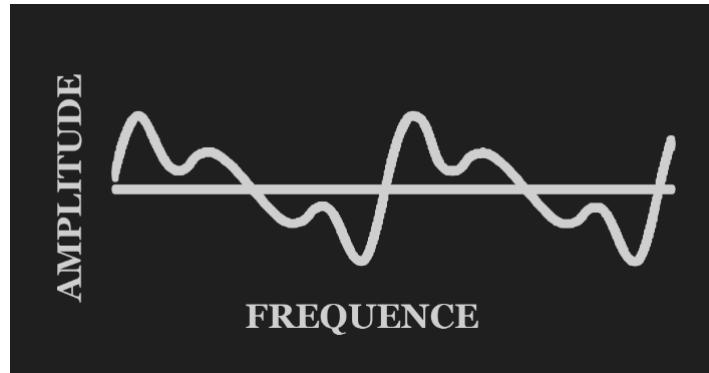
Pour ce faire, j'ai commencé par me documenter sur le fonctionnement d'une génération procédurale.

Le premier point à aborder pour cela est la noise map.

Une noise map est une texture avec des nuances de noir et de blancs que l'on va représenter par des valeurs allant de 0 à 1 (0 pour le blanc et 1 pour le noir).

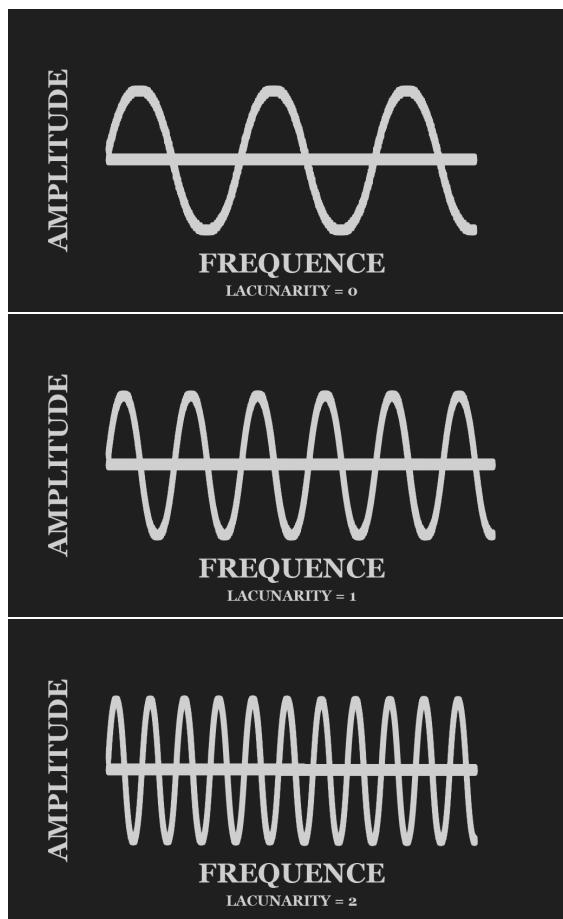


La génération de cette Noise Map est gérée par deux variables, l'amplitude et la fréquence.

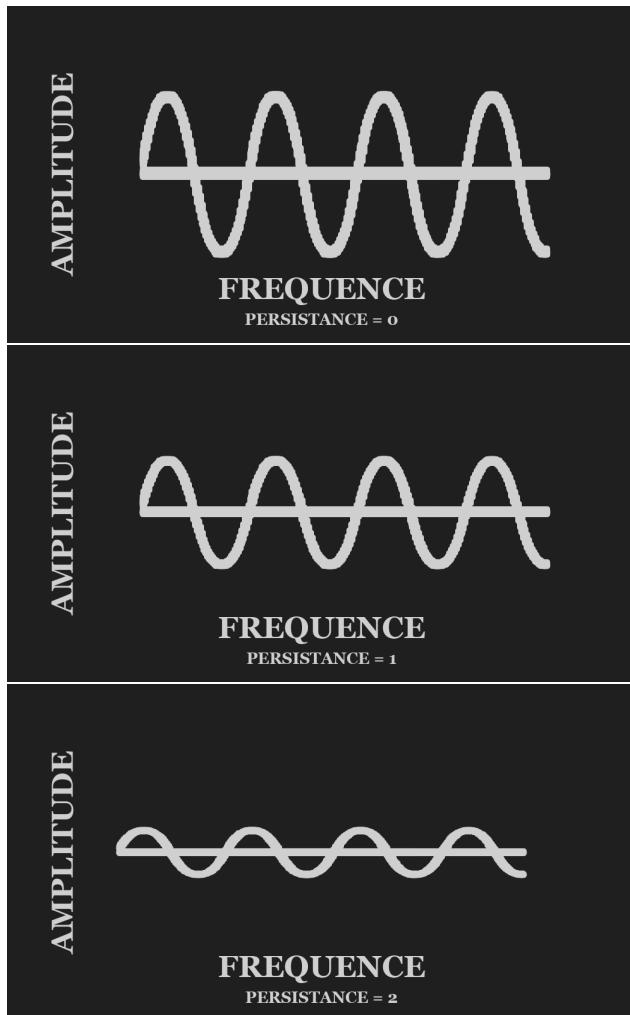


Par la suite, il faut donc gérer chaque région de cette noise map avec différents paramètres qui modifie le taux de noir, la répartition etc... .

Le premier paramètre : La lacunarité, permet d'accentuer la proportion de petit endroit.



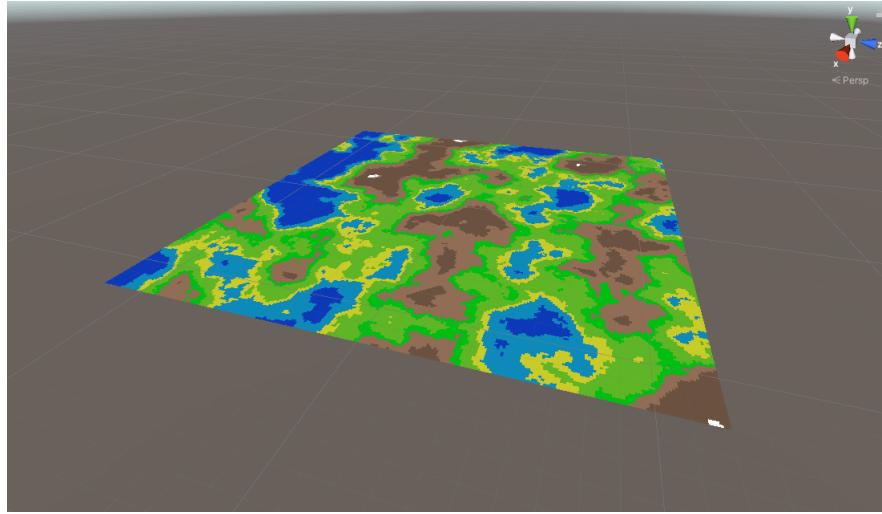
Le second paramètre : La persistance qui à elle seule permet de changer l'influence des petits endroits.



Une fois que la Noise Map a été créée, la partie la plus complexe commença.

L'attribution des couleurs suivant le niveau de la noise map, puis la gestion des reliefs.

Pour ce qui est des couleurs, j'ai créé un script qui permet de choisir la couleur à partir d'une hauteur définie (voir image ci-dessous).



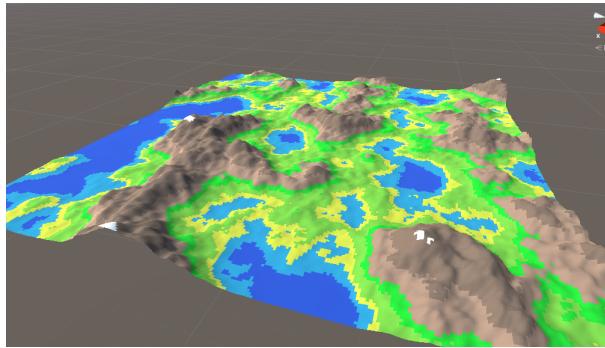
Colour Map

-\ Water	Name	Water
	Height	0.3
	Colour	
-\ ater 2	Name	ater 2
	Height	0.4
	Colour	
-\ sand	Name	sand
	Height	0.45
	Colour	
-\ grass	Name	grass
	Height	0.55
	Colour	
-\ HERBE	Name	HERBE
	Height	0.6
	Colour	
-\ TERRE	Name	TERRE
	Height	0.7
	Colour	
-\ CAILLOUX	Name	CAILLOUX
	Height	0.9
	Colour	
-\ NEIGE	Name	NEIGE
	Height	1
	Colour	

Gestion des différents niveau

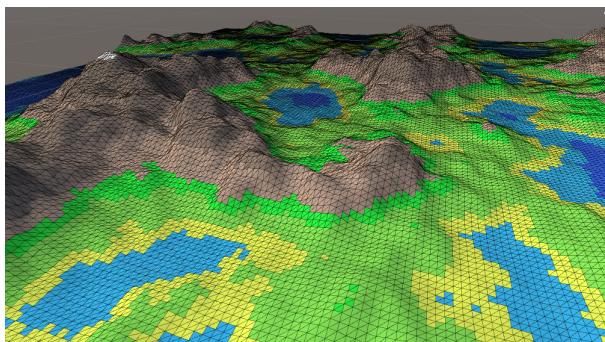
Pour la partie des couleurs j'ai rencontré de nombreux problèmes, le principal étant que la map prenait seulement la dernière couleur, la plus haute, et la mettait de partout, un problème que j'ai donc réussi à corriger.

Après cela, il fallait donc gérer les différents reliefs de la map. Pour cela il suffisait juste d'appliquer un multiplicateur suivant la hauteur définie pour les couleurs. Ce qui donna donc une map de ce style.

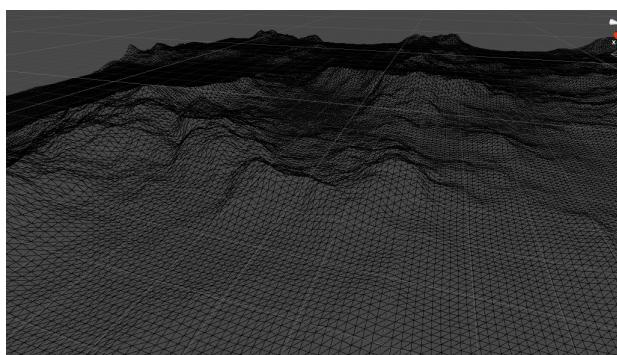


Map avec relief

Pour faire cela, il fallait donc créer le mesh, qui se représente par des triangles rectangles (vertex) pour former la forme de la carte. Cette notion sera importante pour la suite du projet.



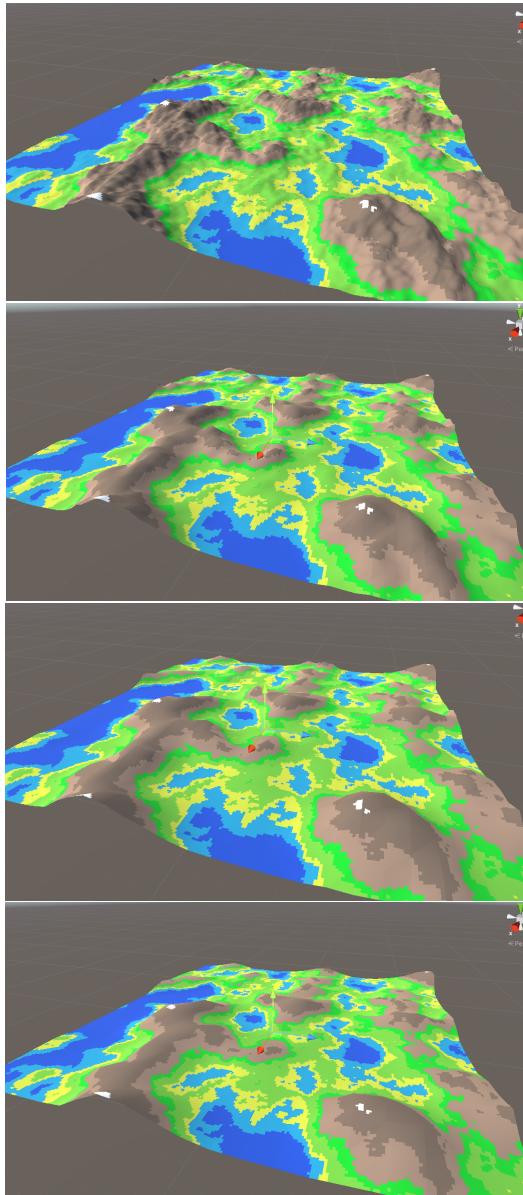
Map wireframe



Map wireframe

Par la suite, il fallait donc penser à l'optimisation de la map, puisque si la carte est immense, il faut d'énormes ressources pour la générer, c'est pourquoi j'ai mis en place un système de Chunk.

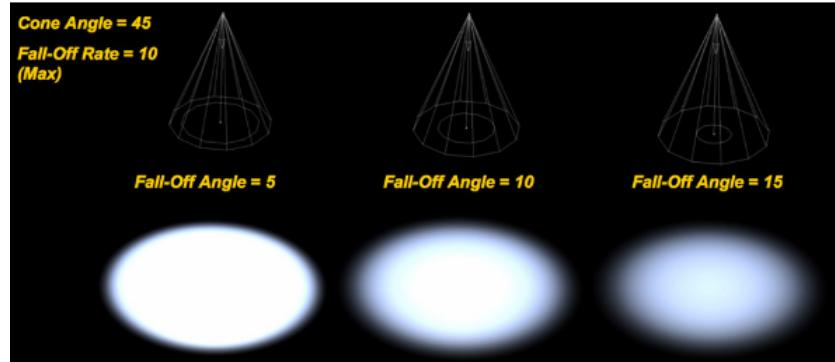
Les chunks lointains ne seraient donc pas charger et permettraient un coût plus léger. Ainsi que les chunks les plus loins du joueurs mais tout de même visible serait chargé de manières plus légères.



Plus on s'éloigne plus la generation est rapide et lisse.

Par la suite il fallait donc faire en sorte que les bordures de la map soient continues et non pas coupées comme cela.

Pour cela j'ai utilisé un Fall Of qui permettait de mettre de l'eau dans la partie extérieure permettant de mieux percevoir une île.



### 3.2 Pour la prochaine soutenance

Pour la prochaine soutenance il faut que je finissent d'implémenter la map entièrement, avec le spawn aléatoire des arbres, cailloux et autres bâtiments ainsi que le spawn des monstres.

De plus j'implémenterais le système de minage pour ce crafter des items.

# Chapitre 4

## Alexis Merle

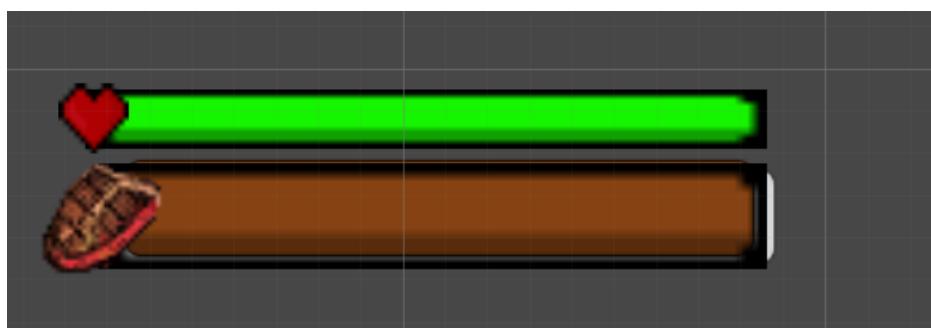
### 4.1 Travail réalisé

Personnellement, je me suis occupé dans cette première partie de projet des interfaces dont un inventaire, un menu ,un système de vie et de faim permettant de créer un réalisme sur le jeu.

Toutes ces interfaces m'ont permis de me familiariser avec Unity et les UI de celui ci.Je n'avais jusqu'ici très peu utilisé pour ma part, j'ai pu assez rapidement les utiliser car Unity est un logiciel accessible pour les débutants qui souhaitent se lancer dans le développement d'un jeu.

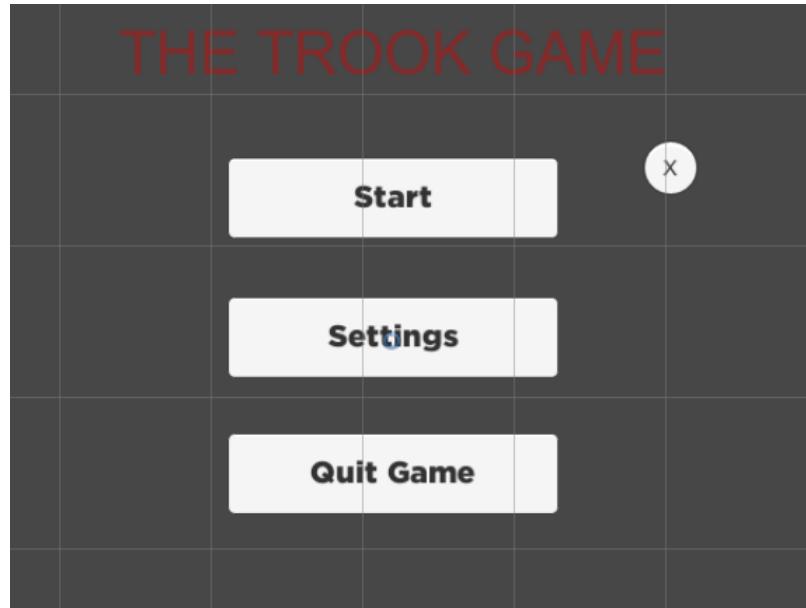
### 4.2 Système de faim et vie

J'ai commencé par ce qui me semblait le plus simple pour pouvoir me familiariser avec les UI notamment ce qui est beaucoup utilisé pour l'inventaire et le menu. //J'ai donc créer des barres qui permettent le suivi de la vie et de la nourriture par l'utilisateur, ces 2 barres sont reliés entre elles. // La barre de faim diminue lorsque les touches de mouvements sont pressés sinon elle reste stable, une fois que la barre de faim a atteint son minimum la vie baisse petit à petit jusqu'à atteindre 0 en attendant que le joueur mange. Ces barres de vie ont des designs simples et sont basés sur les UI slider.

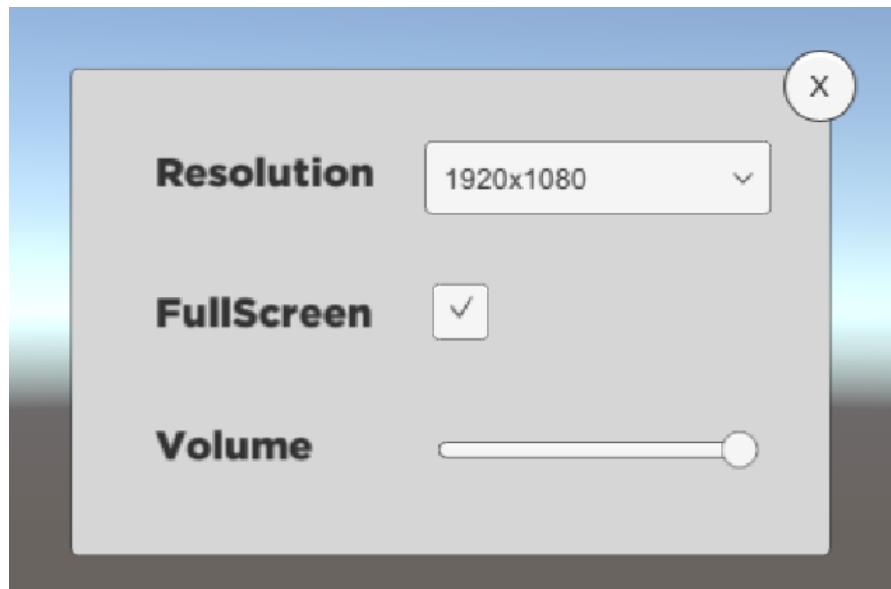


### 4.3 Menu

J'ai donc ensuite implémenté 3 menus : Le premier, le menu d'accueil contient trois boutons permettant de lancer le jeu et le quitter puis un dernier qui permet de charger le menu contenant les paramètres.



Un second menu, le menu de paramètre permet de mettre en plein écran ou non le jeu, de personnaliser sa résolution et de modifier le volume du jeu, par la suite des éléments permettant de personnaliser encore plus son jeu pourront être rajouter.



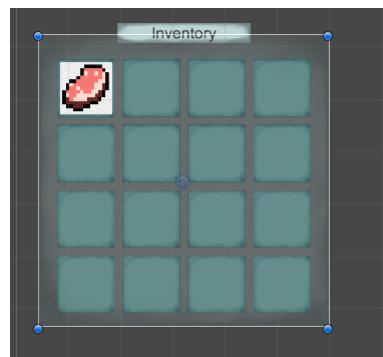
Un dernier menu qui ressemble fortement au premier mais qui apparaît lors de la mort du joueur. Lorsqu'il meurt il peut soit rejouer, retourner au menu principal ou bien quitter le jeu.



Etant débutant sur Unity le design de mes menus n'était pas ma priorité, l'objectif était d'arriver à faire des menus fonctionnels et prendre en main les UI proposés par celui-ci. Le design des menus sera traité dans un second temps lorsque notre jeu sera opérationnel.

#### 4.4 Inventaire

J'ai également implémenté un inventaire fonctionnel avec des items permettant de gagner de la vie et de ne plus avoir faim. Cet inventaire combine de nombreuses fonctionnalités de Unity ce qui rend rapidement la création de l'inventaire difficile grâce à mes précédentes créations j'avais quelques bases qui m'ont permis de me repérer. Le design de cet inventaire et des slots a été récupéré dans des assets Unity ce qui m'a facilité la tâche pour ne pas passer par la tâche design, sans cela mon inventaire aurait été complètement blanc.



## 4.5 Difficultés rencontrés

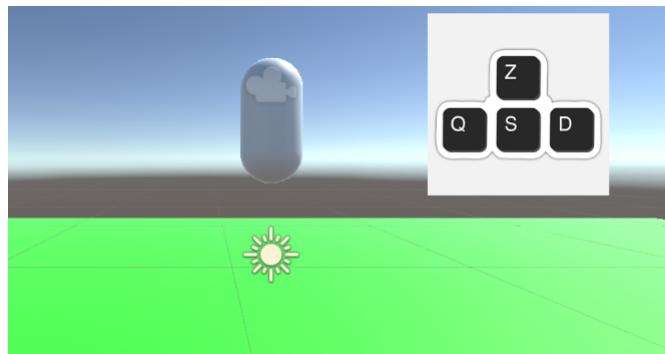
Les principales difficultés que j'ai rencontré lors de cette première partie de projet sont la prise en main d'un nouveau logiciel que je n'ai jamais utilisé auparavant mais aussi des nouvelles fonctionnalités en C et de nouveaux éléments à manipuler avec le script. Ce jeu étant mon projet je ne savais pas vraiment comment commencer ou bien quoi créer en premier (un objet, une scène un script ?). Au final, en fouillant sur internet j'ai pu résoudre ces problèmes assez rapidement et avancer petit à petit dans la création de l'interface graphique.

# Chapitre 5

## Hedi Mokrani

### 5.1 Déplacements

De mon côté j'ai initialisé le projet. Tout d'abord avec la première scène pour pouvoir y implémenter un personnage basique, sans animation et coder les premiers déplacements.



Il est possible de faire bouger la caméra à l'aide de la souris. La petite caméra visible sur la photo ci-dessus est l'emplacement nommé FPS dans les jeux. Il est possible d'effectuer des tours à 360° autour du joueur.

L'un des problèmes rencontrés lors de l'implémentation des déplacements de la vue à l'aide de la souris était que la vue ne se bloquait pas lorsque le joueur regardait vers le ciel et il était ainsi possible d'effectuer une rotation totale en regardant vers le haut.

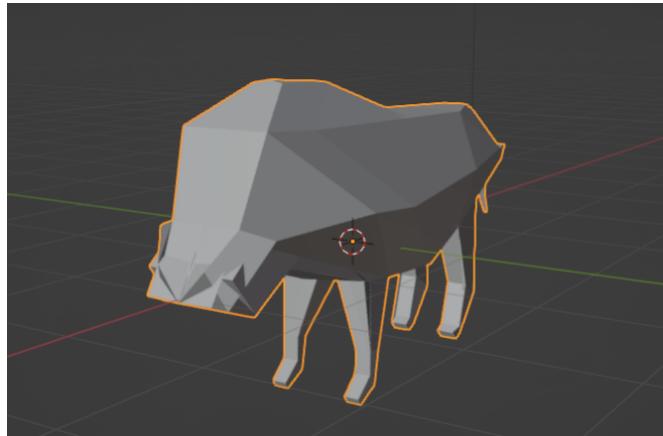
Il est maintenant corrigé et plus aucun bug n'est à noter vis-à-vis des déplacements. La dernière implémentation est le saut. Le joueur peut effectuer un saut grâce à de simples formules de physique concernant la gravité reproduite dans le moteur Unity. En bref, la physique de base est fonctionnelle sur notre jeu.

## 5.2 IA

La deuxième partie dont je me suis occupé est l'implémentation du premier monstre agressif notre jeu qui sera un des monstres de base : le phacochère.

J'ai dans un premier temps dû en trouver un pour pouvoir l'implémenter à notre jeu, je me suis donc rendu sur Blender, un outil nous permettant de créer des objets en 3D, pour pouvoir avoir une première base de monstre.

Voici ce que j'ai réussi à faire pour notre phacochère (après de nombreux essais...) :



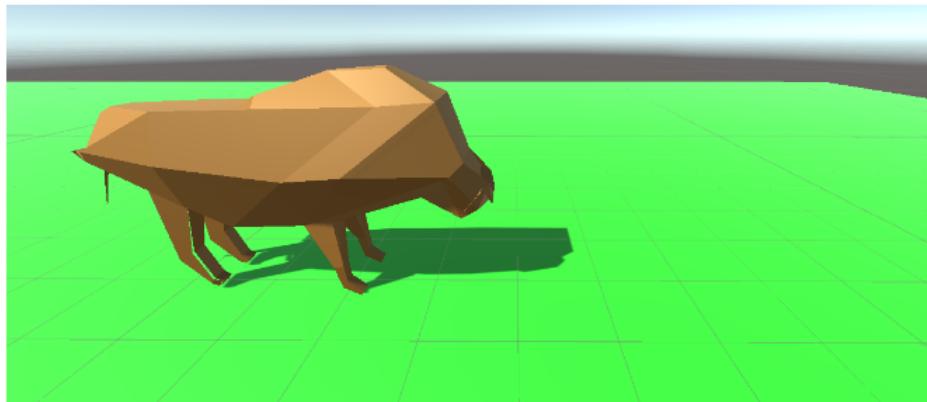
Cette première version du phacochère en “Low Poly” sera donc une base que j'utiliserai pour coder l'IA de ce premier monstre.

Pour tout ce qui concerne l'IA, le phacochère a pour l'instant deux modes :

-Le premier est “se promener”. Il se balade tranquillement en tournant de temps en temps sans trop s'éloigner de sa position d'origine (pour rester proche du troupeau)

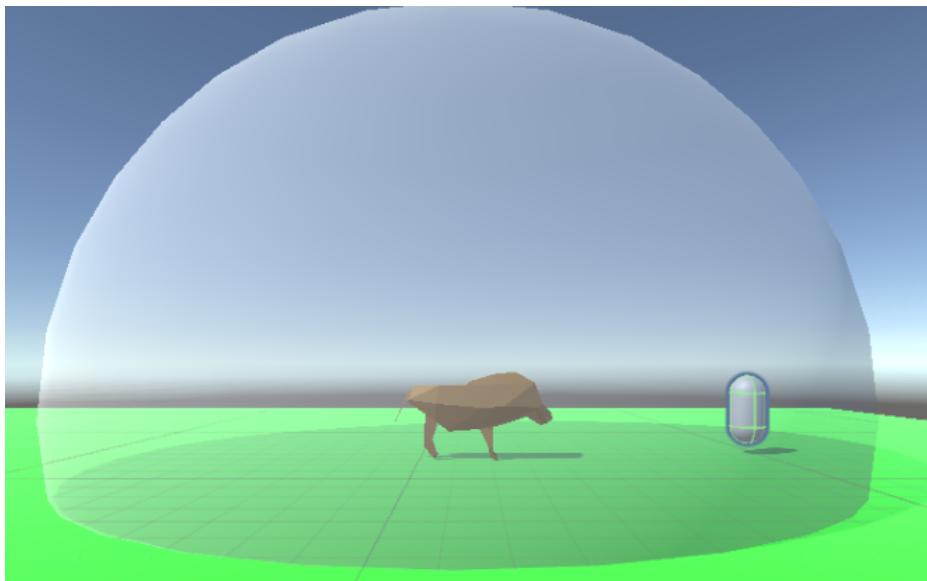
-Le second mode est “Attaquer” qui se déclenche quand le joueur décide d'un peu trop s'approcher.

### **5.2.1 Se promener :**



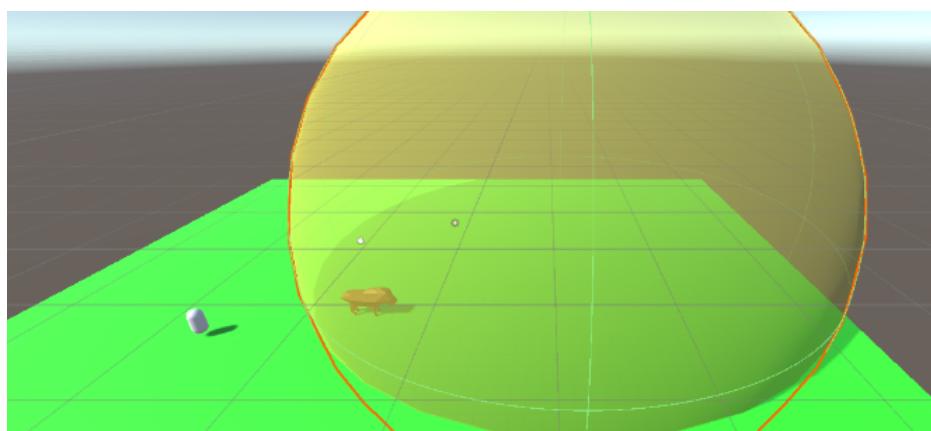
Aucun joueur à l'horizon, il n'y a aucune menace, le phacochère va donc se déplacer normalement dans son environnement avec son troupeau. Ses déplacements sont simples, il va simplement avancer de manière aléatoire sans trop s'éloigner de son point de spawn et donc de ses congénères.

### **5.2.2 Attaquer :**



Il n'y a plus d'autre solution, le phacochère est farouche et se sent menacé par le joueur qui est rentré dans son espace vitale. L'animal charge donc dans la direction du joueur pour pouvoir lui faire des dégâts.

Le script décrit juste avant execute maintenant un code permettant de récupérer la position du joueur et celle du phacochère et de faire en sorte que ce dernier se rue vers le joueur en effectuant des déplacements vers l'avant. Infatigables, il chargerà jusqu'à ce qu'une zone plus grande délimitant son espace de vie sera dépassée et à ce moment il retournera au centre de celle-ci.



# Chapitre 6

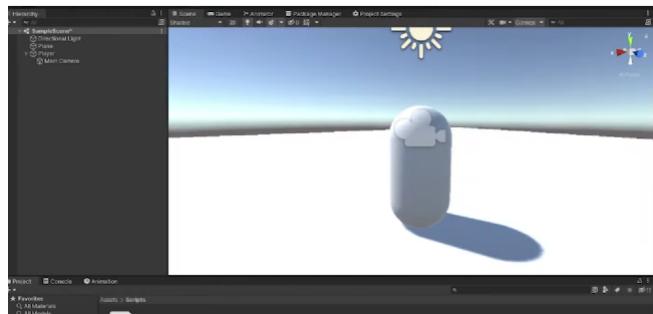
## Felix Capel-Goudart

### 6.1 Multijoueur

Pour ma part j'ai opté pour la manière de jouer en multijoueur de pouvoir héberger en local un serveur afin que des personnes puissent se connecter. Pour cela j'ai utilisé un package inclus dans Unity nommé Mirror qui permet de se connecter en Peer To Peer. Les appareils du réseau sont directement connectés. Il s'agit d'une architecture distribuée. Il n'y a pas de concept de serveur centralisé. Il établit une communication directe entre les joueurs.

Bien sûr, on économise beaucoup en n'ayant pas de serveur dédié.

La première étape est de créer une scène basique et d'appliquer les scripts créés par Hedi afin de bouger et déplacer un personnage.



Dans un deuxième temps on va pouvoir implémenter Mirror dans le projet de test afin de faire une partie en "localhost", c'est-à-dire hébergé sur un même ordinateur ou réseau . Par la suite, on "build" le projet afin de pouvoir l'exécuter avec deux joueurs en local.

Dans un premier temps il faudra mettre dans le script de désactiver la caméra du joueur qui n'est pas en local afin de ne pas afficher sur les deux écrans la même caméra.

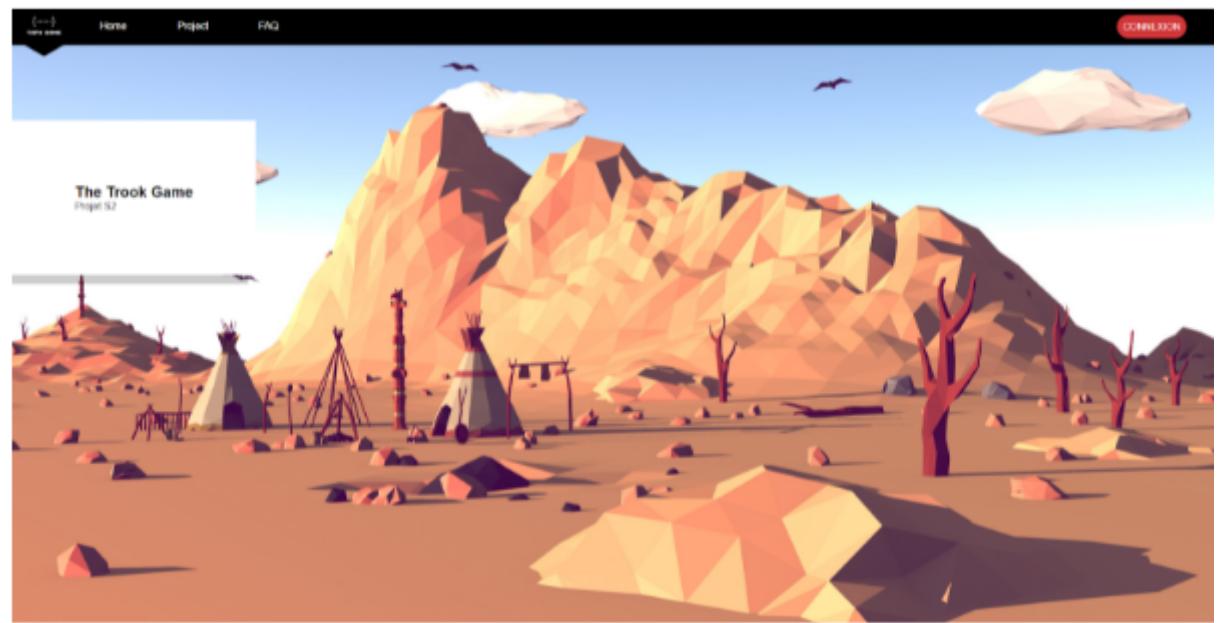


## 6.2 Site web

Au niveau du site web, nous avons commencé à coder un design de base, et une page d'accueil. Nous avons choisi un design épuré qui fait plus professionnel. Nous avons donc développé une simple Header et une simple Footer.



Nous avons également décidé de mettre en place une belle bannière d'accueil pour donner envie au public de jouer à notre jeu. Vous pouvez la voir ci-dessous.



## 6.3 Prochainement

Pour la prochaine soutenance le multijoueur sera implémenté d'une fonctionnalité de chat entre joueur ainsi qu'un échange d'item dans leurs inventaire afin que le système de multi soit au point. De plus, il faut absolument que le système de réseau soit opérationnel (HTTP compris), et qu'il soit fonctionnel. Une fois tout correct, il ne nous restera plus qu'à le greffer au jeu. Je tiens à préciser que le jeu a été développé dans la logique du multijoueur.

Au niveau du site internet, j'aimerais faire un système de compte (connexion et inscription) directement relié au jeu, et terminer les pages "À propos", "Règles", et "Statistiques" qui seront créés à l'avenir..

De plus, un système de téléchargement du Trook Game sera directement accessible via le site internet qui sera à disposition du grand public. A partir de ce téléchargement le jeu sera potentiellement disponible pour la dernière soutenance a partir d'un launcher ou une connexion sera requise. Cette connexion se fera à partir du compte créé sur le site web.

## 6.4 Difficultés rencontrés

J'ai rencontré pas mal de problèmes autant en C qu'en Web.  
Au niveau du HTTP, la documentation est moins abondante ce qui rend la tâche moins évidente. Encore une fois, la quantité de code est importante et il ne faut pas se perdre et être assez méthodique.

Pour ce qui est du site, j'ai eu de la chance d'avoir déjà fait un peu de CSS auparavant ce qui m'a grandement facilité la tâche. J'ai cependant eu quelques soucis en PHP lors de la mise en place de ma classe "Page". Certaines erreurs apparaissent mais je m'en suis sorti encore une fois grâce aux forums

Au niveau de codage du multijoueur c'était plutôt compliqué au premier abord car n'ayant jamais touché à Unity c'était un défi à relever. Le premier obstacle était évidemment de se renseigner et comprendre le fonctionnement de Unity ainsi que ses outils les plus basiques. Le premier défi était de faire en sorte que le multijoueur se puisse faire en local.

# Chapitre 7

## Conclusion

Pour conclure nous avons déjà une base solide pour pouvoir avancer dans notre projet. Pour la prochaine soutenance nous aurons réussie à implémenter le mode multijoueur, ainsi que de nouvelle fonctionnalités pour commencer à rendre le gameplay intéressant. Le but étant que pour la dernière soutenance nous ayons plus qu'à nous appliquée à rendre le jeu le plus équilibré possible avec un gameplay plaisant à jouer.