

# Tales Of The Apocalypse

Rapport de Soutenance 1 - Projet S2



Yanis CHaabane

Jordan FAILLOUX

Tanguy DESGOUTTES

Marc-Emmanuel RAIFFE



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Présentation du projet . . . . .	4
1.2	Présentation de FIG . . . . .	4
1.3	L'équipe de développeurs . . . . .	5
1.3.1	Tanguy Desgouttes (Chef de groupe) . . . . .	5
1.3.2	Yanis Chaabane . . . . .	5
1.3.3	Jordan Failloux . . . . .	5
1.3.4	Marc-Emmanuel Raiffe . . . . .	6
<b>2</b>	<b>Accord avec le Cahier des Charges</b>	<b>7</b>
<b>3</b>	<b>Conception</b>	<b>9</b>
3.1	Génération Procédurale . . . . .	9
3.2	Pathfinding . . . . .	9
3.3	IA . . . . .	10
3.4	Multijoueur . . . . .	10
3.5	Menus & Interface utilisateur . . . . .	11
3.6	Textures, Modèles & Animations . . . . .	12
3.7	Autres fonctionnalités jouables . . . . .	13
3.8	Site Web . . . . .	13
<b>4</b>	<b>Réalisation</b>	<b>15</b>
4.1	Fonctionnalités jouables actuelles . . . . .	15
4.2	Travaux de recherches . . . . .	15
4.3	Problèmes & Solutions . . . . .	17
<b>5</b>	<b>Prévisions</b>	<b>18</b>
5.1	Correction du planning . . . . .	18
5.2	Génération Procédurale . . . . .	18
5.3	Pathfinding . . . . .	18
5.4	IA . . . . .	19
5.5	Multijoueur . . . . .	19
5.6	Menus & Interface utilisateur . . . . .	19
5.7	Fonctionnalités jouables . . . . .	19
5.8	Textures, Modèles & Animations . . . . .	19
5.9	Son et musique . . . . .	19
5.10	Site Web . . . . .	19
<b>6</b>	<b>Conclusion</b>	<b>20</b>
<b>7</b>	<b>Annexes</b>	<b>20</b>
7.1	Sources . . . . .	20
7.2	Annexes . . . . .	20

# 1 Introduction

## 1.1 Présentation du projet

*Tales of the Apocalypse* est un jeu vidéo en 3D visant à recréer une expérience d'un groupe de survivants lors d'une apocalypse. Le joueur aura alors le contrôle total de l'ensemble des membres du groupe et aura pour tâche de les faire survivre. Chaque survivant est unique et possède des traits, des compétences et des caractéristiques propres à lui représentant ses forces et ses faiblesses. TotA n'est ni un jeu de stratégie compétitif ni un simulateur, mais est avant tout un générateur d'histoire, ainsi l'intérêt premier du jeu n'est pas de gagner mais d'assister aux différents événements, qu'ils soient épiques, comiques, tragiques ou dramatiques, vécus par le groupe de survivants.



FIGURE 1 – *Tales of the Apocalypse*

## 1.2 Présentation de FIG

Filiga Inspiration Games (ou FIG) est un groupe formé de quatre étudiants de l'EPITA : Tanguy Desgouttes, Yanis Chaabane, Marc-emmanuel Raiffe et Jordan Failloux. FIG est spécialisé dans la création de jeux vidéos, nous travaillons actuellement exclusivement sur le jeu vidéo *Tales of the Apocalypse* (abrégé TotA). Nous espérons combler notre manque d'expérience dans le domaine par notre détermination au travail et nos connaissances dans le domaine du jeu vidéo. La création de notre groupe se repose sur un mélange harmonieux entre une bonne entente déjà existante, un mélange de différentes compétences et expériences dans le domaine du jeu vidéo et un intérêt partagé pour le concept de TotA.

Le nom de notre groupe, Filiga Inspiration Games, provient d'une réflexion sur nos forces potentielles : la diligence au travail ("Filiga" signifie "diligent" en Samoan) et nos connaissances de différents jeux dont nous allons nous inspirer ("Inspiration"). Enfin, "Games" représente simplement le domaine dans lequel notre groupe a décidé de se spécialiser.



FIGURE 2 – FIG

Le logo de notre groupe est la représentation abstraite d'une figue, provenant de l'abréviation FIG du nom de notre groupe. La figue apparaît comme étant pixelisé reflétant la direction artistique de TotA tournée alors vers un style cubique.

## 1.3 L'équipe de développeurs

### 1.3.1 Tanguy Desgouttes (Chef de groupe)

Étant chef de projet et concepteur du projet, celui-ci me tient particulièrement à cœur. L'idée originale de *Tales of the Apocalypse* peut remonter jusqu'à 3 ans, le premier prototype s'appelait alors *Projet AZ*. Si le jeu a beaucoup évolué au cours des années, l'objectif principal est resté inchangé : satisfaire la question que j'ai pu souvent entendre "Que ferais-je lors d'une Apocalypse ?". Cette idée de jeu m'a amené à découvrir l'informatique et à me familiariser avec de nombreux langages (de GML à C++). Au fil des années, je me suis découvert un intérêt tout particulier dans le développement de jeux vidéos et plus généralement des nouvelles technologies, la prise de décision lors du développement d'un projet me fascine. Ce projet est donc pour moi une des nombreuses raisons pourquoi j'ai rejoint l'EPITA. J'ai pu rencontrer et recruter des personnes compétentes et sérieuses qui, de par leurs différentes expériences, apportent un regard nouveau sur le projet. J'ai une confiance absolue en notre groupe et en la réussite du projet.

### 1.3.2 Yanis Chaabane

Ayant une passion pour l'informatique depuis 2 ans, je m'intéresse à tout ce qui touche aux nouvelles technologies, aux langages de programmation. De par cette passion j'ai d'abord fait le choix d'une Terminale S avec option ISN où j'ai fait mes premiers pas dans la programmation et où j'ai pour la première fois codé en Python et dans les domaines du web. Ce fut d'une grande aide pour moi lors du choix de mon orientation.

Le projet d'ISN fut un projet important qui m'a permis de développer un jeu sur de longs mois et m'a permis évidemment d'améliorer mes compétences dans différents langages de programmation mais aussi de prendre part à un travail de groupe avec la répartition des tâches, ce qui va pouvoir me servir dans ce nouveau projet.

De par cette expérience que j'ai pu avoir l'année dernière je me sens prêt à relever de nouveaux défis lors de ce projet beaucoup plus important. J'ai hâte de pouvoir voir l'avancé de notre jeu et de nos compétences. Étant organisé et perfectionniste, je me donnerai à fond dans ce projet afin d'en pouvoir tirer un jeu qui sera capable de sortir sur les différents marchés de jeux vidéos mais aussi d'en tirer un réel avantage personnel pour mes années à venir à l'EPITA.

### 1.3.3 Jordan Failloux

Je m'appelle Jordan FAILLOUX et je viens de Tahiti. J'ai toujours été intéressé par l'informatique, c'est pour cela que j'ai choisi de faire une Terminale S option ISN. Pour mon avenir professionnel, j'hésitais alors entre la médecine et l'informatique. Étant présent sur le territoire Polynésien, j'ai d'abord décidé de faire PACES. Au cours de l'année, je me suis aperçu que ça ne me plaisait pas. J'ai donc cherché des écoles d'ingénieurs informatique en France et je suis arrivé ici, à l'EPITA.

En ISN, j'ai eu l'occasion de faire un site internet et un jeu vidéo en Python en tant que projet en équipe pour le bac. Ce dernier m'a permis de prendre goût à la réalisation de programmes en groupe. Pour ce projet, je compte utiliser mes expériences acquises précédemment et ne pas refaire les mêmes erreurs pour avoir un meilleur résultat final.

#### **1.3.4 Marc-Emmanuel Raiffe**

J'ai toujours été intéressé par les sciences et à l'aise dans ce domaine. Ces deux raisons m'ont assez naturellement orienté vers un bac S-SI et l'affection que j'avais pour la programmation m'a amené à l'EPITA. Même si je n'avais jamais vraiment appris de langages informatiques avant mes études supérieures je reste confiant pour le projet car les travaux de groupe sont les activités où je suis le plus productif et j'ai toujours beaucoup apprécié les jeux vidéos ce qui m'a permis d'acquérir beaucoup de connaissances maintenant utiles.

En terme de compétences plus strictes je me suis découvert un niveau suffisant en programmation malgré le manque d'expérience. Cela dit je maîtrise certains logiciels comme magicavoxel qui permettront de créer les modèles 3D et illustrations utiles au projet.

## 2 Accord avec le Cahier des Charges

Nous voulons d'abord apporter une légère précision par rapport à notre cahier des charges sur le fait que pour l'instant le jeu n'est disponible qu'en anglais, la langue française pouvant être ajoutée éventuellement comme un bonus dans les options lors de la troisième soutenance.

De plus, en commençant le projet nous nous sommes rendus compte que notre répartition des tâches allait devoir être légèrement modifiée. En effet malgré le fait que Yanis ait commencé à se renseigner et à faire les premiers pas avec le multijoueur et Photon, il a été clair que Tanguy, suppléant de cette tâche était plus à l'aise avec celle-ci. Nous avons donc permuté les rôles de responsable et de suppléant entre le multijoueur et les menus et interfaces graphiques. Yanis s'étant déjà renseigné sur cette tâche a donc pu très vite s'adapter et constituer les premières ébauches du menu principal. Ensuite nous avons décidé de préciser un peu la tâche génération procédurale et de la diviser en deux parties : la partie génération de villes et la partie génération du monde. Comme pour la tâche précédente ces deux tâches seront sous la responsabilité de Tanguy avec comme suppléant Marc-Emmanuel. Les prévisions pour ces nouvelles tâches sont données dans le tableau ci-après. Les différents changements sont résumés dans le tableau de répartition suivant :

	Responsable	Suppléant
<b>Programmation</b>		
Génération de villes	Tanguy	Marc-Emmanuel
Génération du monde	Tanguy	Marc-Emmanuel
Pathfinding	Tanguy	Marc-Emmanuel
IA	Marc-Emmanuel	Jordan
Multijoueur	Tanguy	Yanis
Menus & Interface utilisateur	Yanis	Tanguy
<b>Art</b>		
Textures	Marc-Emmanuel	Jordan
Modèles & Animations	Marc-Emmanuel	Jordan
Son & Musique	Jordan	Tanguy
<b>Site Web</b>		
HTML	Yanis	Jordan
CSS	Jordan	Yanis
<b>Rapport de soutenance</b>		
Rapports de soutenances	Yanis	Tanguy

 Changement de tâche    Modifications de tâches

Tâche \ Soutenance	Soutenance 1	Soutenance 2	Soutenance 3
<b>Programmation</b>			
Génération de villes	85%	100%	100%
Génération du monde	20%	80%	100%
Pathfinding	100%	100%	100%
IA	5%	50%	100%
Multijoueur	60%	80%	100%
Menus & Interface utilisateur	20%	40%	100%
<b>Art</b>			
Textures	5%	30%	100%
Modèles & Animations	20%	40%	100%
Son & Musique	5%	20%	100%
<b>Site Web</b>			
HTML	65%	90%	100%
CSS	65%	90%	100%

 Modifications

## 3 Conception

### 3.1 Génération Procédurale

#### 1. Travail de Tanguy (responsable) :

La génération de la ville a occupé sans aucun doute la plus grande partie de mon travail. Je souhaitais pouvoir créer une ville procéduralement que le joueur puisse explorer complètement. J'ai d'abord commencé par la génération de route, nos routes sont principalement caractérisées par leurs "ouvertures" dans les directions cardinales, il suffit alors de relier des points d'intérêts entre eux en "ouvrant" au fur et à mesure la route. Pour l'instant, toutes les villes ont 9 points d'intérêts plus ou moins reliés entre eux, les ouvertures Nord/Sud/Est/Ouest de la ville, un centre approximatif de la ville et 4 points répartis dans chaque cadran de la ville (Voir Figure 8 à 9).

Une fois ce système mis en place, j'ai commencé à réfléchir à un moyen de mettre des bâtiments dans la ville. J'ai décidé de générer des bâtiments seulement à côté des routes, tout d'abord des bâtiments de petites tailles (carré de 1x1) simple à positionner face à la route puis des bâtiments plus grands (rectangle de 2\*1) (Voir Figure 10 à 11).

Placer ces bâtiments rectangles a été pour l'instant la plus grande difficulté rencontrée, en effet les bâtiments devaient avoir une entrée du côté de la route tout en se générant sur deux emplacements. Au cours du développement de nombreux problèmes ont pu être rencontrés et résolus, par exemple la partie supplémentaire des rectangles n'empêchait pas d'autres bâtiments de se générer. Au final, ce système génère une ville satisfaisante tout en étant extrêmement flexible, l'ajout et la suppression de bâtiment ne demandant même pas de regarder le code.

### 3.2 Pathfinding

#### 1. Travail de Tanguy (responsable) :

J'ai travaillé sur l'implémentation d'un système de déplacement "Clic to move", pour déplacer les personnages, le joueur sélectionne le ou les personnages qu'il souhaite contrôler et a juste à cliquer pour les déplacer. Pour implémenter ce système j'ai utilisé la fonction "NavMesh" ("Navigation Mesh" ou "réseau de navigation") qui permet de définir une surface où un chemin d'un point A à un point B peut être trouvé.

Le déplacement d'un objet sur cette surface est aussi géré par Unity, l'objet est alors appelé un "agent" (ou "NavMeshAgent"). Il reste alors la sélection de l'objet qu'on souhaite déplacer et la sélection de sa destination, pour se faire, on utilise le système de "RayCast" d'Unity qui permet de détecter des GameObjects sur une ligne (ou un segment) donnée.

Ainsi, le joueur peut sélectionner un ou plusieurs personnages avec le clic gauche de la souris puis marquer une destination avec le clic droit, tous les personnages sélectionnés vont se déplacer à cette destination. Ce système de déplacement bien que traditionnel convient parfaitement à notre jeu de par sa simplicité.

#### 2. Travail de Marc-Emmanuel (suppléant) :

L'implémentation du "pathfinding" ayant déjà été faite (jusqu'alors les personnages étaient des cubes se mouvant) il a fallu utiliser un animator, changeant l'animation d'un modèle en fonction des données d'entrée. Les données d'entrée sont issues du NavMeshAgent qui nous permet de faire le "pathfinding" et sont retranscrites en une certaine quantité de vitesse et de degré de rotation par une portion de code. Une fois tout cela en place et les différents problèmes résolus on a pu avoir une animation fluide et qui suivait le déplacement du personnage.

### 3.3 IA

Pour le moment seul un élément a été implémenté : lorsqu'un personnage ne fait pas partie d'une équipe, alors il se déplace aléatoirement dans une zone définie.

### 3.4 Multijoueur

1. Travail de Tanguy (responsable) :

Après m'être renseigné sur Photon, J'ai commencé par implémenter un multijoueur fonctionnel, le joueur pouvait alors se connecter au serveur de Photon, créer une salle si elle n'existe pas déjà et rejoindre la salle si elle existait. Une fois la salle rejointe il pouvait déplacer un cube et voir se déplacer les cubes des autres joueurs. Il pouvait ensuite quitter la salle et retourner au menu. Le but de cette implémentation était d'avoir un système fonctionnel sans se soucier de l'esthétique. Par la suite, le système a été complété et amélioré par Yanis (notamment avec l'ajout d'une liste des joueurs dans les "chambres d'attente").

J'ai aussi mis en place un système de permissions qui permet au joueur de jouer en équipe contre d'autres équipes, un joueur ne peut contrôler un personnage que si ce personnage est lié à son équipe. A noter aussi que chaque joueur et chaque équipe a une couleur attribuée qui s'affiche en surbrillance autour d'un personnage sélectionné. La mise en place m'a forcée à me renseigner sur les fonctions "RPC" de Photon, c'est-à-dire la transmission directe de données entre joueurs.

A ce stade du développement du jeu, le joueur peut donc se connecter dans une salle et contrôler des personnages qui appartiennent à son équipe, il peut aussi voir d'autres personnages être déplacés mais ne peut pas les déplacer, l'accès lui étant refusé par le système de permissions.

2. Travail de Yanis (suppléant) :

Mon travail sur le multijoueur a commencé avec le menu du multijoueur. Lorsque nous appuyons sur le bouton "Multijoueur" à partir du menu principal alors nous arrivons sur une nouvelle scène où nous pouvons nous connecter aux serveurs fournis par Photon (ceux-ci permettant d'avoir jusqu'à 20 joueurs connectés en simultanés).

A partir de ce moment si le joueur se connecte, alors le jeu va retenir son nom en l'ajoutant en plus dans les préférences du joueur, c'est-à-dire que lorsque le joueur relancera le jeu, un pseudo lui sera proposé automatiquement, le dernier qu'il a entré lors de ses parties précédentes. Il aura alors le choix entre soit créer une nouvelle "room" (nouvelle partie) ou en chercher une déjà existante avec comme nom celui entré par le joueur. Quel que soit son choix (hors celui du retour au menu) il arrive dans une "chambre d'attente" où il peut voir les joueurs déjà présents dans la partie ou dans la même "chambre d'attente" que lui. Cette liste de joueurs se met à jour à chaque fois qu'un joueur arrive dans cette "chambre" ou la quitte.

Pour faire cela on utilise les différents événements qui sont envoyés par Photon tels que : "OnJoinRoom", "OnPhotonPlayerConnected" et "OnPhotonPlayerDisconnected". Quand ces événements sont appelés on met à jour la liste de joueurs.

Le joueur a alors de nouveaux trois choix : celui de rejoindre la partie, celui de retourner au choix des différentes "rooms" ou bien celui de retourner au menu principal (il se déconnecte alors du serveur Photon).

### 3.5 Menus & Interface utilisateur

1. Travail de Yanis (responsable) :

(a) Le menu principal :

J'ai implémenté quatre boutons : un bouton "Solo" qui permet d'accéder au mode hors ligne, individuel du jeu (pas encore implémenté dans notre jeu). Suit le bouton "Multijoueur" qui permet comme son nom l'indique d'accéder au mode en ligne pour pouvoir jouer des parties avec des joueurs en ligne. Nous avons ensuite le bouton "Options" qui nous permet de régler les différents paramètres du jeu tels que le volume sonore, la qualité graphique du jeu ou encore la résolution du jeu c'est-à-dire la taille de l'écran utilisé pour jouer au jeu. Enfin un bouton "Quit" permet de quitter le jeu, de l'arrêter.

Pour faire cela on attache un script à chaque bouton pour que lorsque celui-ci est cliqué cela fasse un appel à une fonction qui peut activer, désactiver certains éléments ou bien charger d'autres scènes.

(b) Connexion au Multijoueur :

De nombreux éléments "UI" ont été ajoutés tels que : un bouton pour se connecter au serveur de Photon pour le multijoueur mais il ne marche que lorsque le joueur a rentré un pseudo dans un "input field" situé juste en dessous. Après cette étape les deux éléments cités ci-dessus se désactivent et apparaissent 3 nouveaux éléments : un autre "input field" dans lequel le joueur entre le nom de la "room" qu'il souhaite rejoindre si elle existe ou créer sinon. Il peut donc cliquer sur deux boutons : "Create" qui va créer une "room" avec le nom donné ou "Search" qui va faire rejoindre la "room" avec le nom donné si elle existe.

Le joueur peut donc arriver dans une "chambre d'attente" où il y a trois nouveaux éléments : un bouton "Leave Game" pour retourner au choix des "rooms", un bouton "Join Game" pour rentrer enfin dans la partie et un "scroll view" qui permet d'afficher les noms des joueurs présents que l'on peut faire défiler de haut en bas.

Pour cela on va regarder chaque joueur présent dans la liste donnée par Photon qui contient les joueurs de la "room" et on va tous les ajouter dans la liste affichée. Sinon il s'agit d'une mise à jour de la liste affichée en ajoutant le joueur en tant que dernier ou en regardant l'index du joueur partant dans la liste pour le retirer de la liste affichée. Dans tous les cas on va gérer l'affichage de la liste en ajoutant un "GameObject" représentant le joueur avec son nom et en le déclarant fils d'un autre "GameObject" qui va représenter la liste des joueurs. Ces fils vont donc se mettre les uns à la suite des autres de façon verticale.

Enfin tout le long de ce processus il y a un bouton "Menu" situé en bas qui permet de revenir au menu principal. Finalement lorsque le joueur rentre dans la partie il y a une barre qui montre le chargement de la partie (il s'agit d'un "slider" qui évolue selon le chargement d'une nouvelle scène de façon asynchrone).

(c) L'inventaire :

— Explication visuelle : J'ai implémenté un inventaire pour chaque joueur. Pour cela il a fallu mettre un canvas en mode "World space" pour que ce dernier puisse suivre la position du joueur auquel il est attaché. Il y a donc encore une fois un "scroll view" afin de pouvoir faire défiler les différents éléments de l'inventaire. Ces éléments sont constitués d'un élément comprenant une image et un bouton servant d'icône pour l'item, d'un bouton pour enlever l'item de l'inventaire et d'un texte qui permet de voir le nombre d'items dans un emplacement de l'inventaire. Si le nombre d'items par emplacement dépasse celui autorisé, alors l'item sera ajouté dans un nouvel emplacement et ainsi de suite

jusqu'à arriver au maximum de capacité de l'inventaire. L'inventaire de chaque joueur sélectionné est affiché lorsque le joueur appuie sur la touche E et disparaît s'il appuie de nouveau sur cette touche .

Ensuite pour remplir son inventaire le joueur peut interagir avec des objets au sol : s'il arrive suffisamment près de l'objet en question alors il va être ajouté pour l'instant dans son inventaire s'il n'est pas déjà plein et cela va avoir pour effet de détruire l'objet du monde. Ces objets, stockés dans l'inventaire sont utilisables selon leurs fonctionnalités. Par exemple une pomme pourra être mangée. Donc lorsque le joueur cliquera sur l'item de la pomme dans l'inventaire il sera enlevé de l'inventaire s'il n'en contenait qu'un seul ou décrémenté sinon et la faim du joueur diminuera, les informations du joueur étant mis à jour.

- Explication fonctionnelle : Pour effectuer l'ajout dans l'inventaire d'un item, pour cette première soutenance nous avons deux manières de le faire. Soit nous pouvons interagir directement avec l'objet selon une certaine de zone de rencontre définie par un rayon autour de cet objet. L'objet possède un script dérivant d'une classe "Interactable" pour que ce dernier puisse avoir un item différent et appelle alors les différentes fonctions pour ajouter l'item dans l'inventaire. L'autre manière de faire est pour l'instant une manière pour tester l'inventaire. Le joueur peut appuyer sur la touche I et un item au hasard est choisi et ajouté dans l'inventaire parmi une liste d'items mise dans un "Scriptable Object".

Dans les deux cas, l'appel de l'ajout à l'inventaire va aller vérifier dans un dictionnaire si l'item peut être ajouté, c'est-à-dire si l'inventaire n'est pas déjà plein, sinon l'item va soit être ajouté dans le dictionnaire s'il n'en faisait pas partie, soit il va incrémenté sa valeur associée. Si l'item a pu être ajouté correctement au dictionnaire, l'inventaire va être mis à jour par des fonctions qui vont gérer les différents éléments de chaque emplacement. On va afficher chaque emplacement et permettre l'interaction avec celui-ci en gérant son incrémentation ou sa décrémentation.

## 2. Travail de Tanguy (suppléant) :

Afin de commencer à différencier différents personnages, j'ai implémenté un système d'informations du personnage. En haut à gauche de l'écran on va avoir un cadran d'informations avec le nom du personnage et ses différents attributs que sont : la Force, l'Intelligence, la Perception, le Mental, le Social qui vont de 0 à 100, une barre de faim temporaire a aussi été mise pour tester les interactions avec l'inventaire. Cette boîte d'informations apparaît lorsqu'un personnage est sélectionné. Si plusieurs personnages sont sélectionnés, c'est la boîte d'informations du dernier personnage sélectionné qui apparaîtra.

De nombreux facteurs vont faire évoluer un personnage. Le seul mis en place pour le moment est l'inventaire. Certains objets sont utilisable dans l'inventaire, par exemple le joueur peut cliquer sur emplacement représentant de la nourriture pour modifier la faim d'un personnage, le cadran d'informations sera alors mis à jour.

## 3.6 Textures, Modèles & Animations

### 1. Travail de Marc-Emmanuel (responsable) :

#### (a) Modèles :

Comme prévu tous les modèles faits pour le jeu l'ont été en voxel. Le travail de modélisation a commencé avec des essais d'importation sous Unity avec le premier modèle fait pour le jeu, un réfrigérateur(figure 14). Ce modèle a été suivi par beaucoup d'autres qui avaient pour but

de supporter la génération procédurale et le "pathfinding". Ainsi ont suivi des modèles de routes et de structures de bâtiments (figure 12, figure 13). Au vue de l'avancement prévu, la priorité était de trouver une génération viable avant de se lancer dans la conception globale. Une fois la génération finie j'ai pu me lancer dans la modélisation de bâtiments de différentes tailles et formes ainsi que d'objets qui seront interactifs (figure 16).

(b) Animation :

J'ai débuté le travail d'animation entre la conception des premiers modèles de test et le début de conception des modèles finaux. Il a fallu dans un premier temps créer nos personnages, premières cibles de notre animation. Initialement les modèles pensés pour les personnages devaient se faire sans membres animés mais ont vite évolués pour profiter des options que propose Unity. Ainsi le premier modèle en "T-POSE", posture standard pour l'animation où le personnage est debout, les bras levés sur le coté (comme un T), a vu le jour. Il a fallu le munir d'un squelette puis j'ai pu utiliser la banque d'animation humanoïde de Mixamo pour tester les animations (figure 15). Après quelques essais infructueux avec différents modèles arrivés à ce stade j'ai enfin pu trouver le style de modèles qui ont le meilleur rendu en animation. Il a ensuite simplement fallu me constituer une banque d'animation pour commencer à travailler sous Unity.

### 3.7 Autres fonctionnalités jouables

1. Travail de Tanguy :

Une fois la génération de ville terminée, j'ai implémenté le déplacement de la caméra du joueur dans le monde. Dans Tota le joueur voit le monde en vue de dessus, le déplacement horizontal de la caméra se fait avec les touches directionnelles ou les touches WASD. Le déplacement vertical de la caméra dans le monde a été plus difficile à mettre en place, en effet le joueur doit pouvoir voir l'intérieur des bâtiments à travers le sol et le toit, pour ce faire, le joueur utilise la molette de la souris pour définir l'étage qu'il veut voir, à chaque fois qu'un nouvel étage est défini tous les étages supérieurs à celui choisi ne sont pas montrés à l'écran (évidemment cette disparition des bâtiments se fait localement et n'est pas transmise aux autres joueurs).

### 3.8 Site Web

Nous avons décidé de nous occuper dès le début du site Internet car nous avons choisi de faire un site statique pas très complexe avec les différentes informations inhérentes au jeu et à Filiga. Donc nous nous sommes mis d'accord sur le fait qu'étant pas très long à faire il serait plus rentable de le faire tôt pour garder le plus de temps possible à la réalisation du jeu en lui-même. Vous pouvez retrouver notre site sur : filiga.me

1. Travail de Yanis (responsable HTML) :

Je me suis occupé de générer les différentes pages HTML :

- Une page d'accueil avec un menu pour accéder aux autres pages, une brève introduction au projet et probablement un aperçu du jeu *Tales Of The Apocalypse*.
- Une page de présentation du projet, de son origine et de son but.
- Une page pour présenter chaque membre du groupe.
- Une page concernant l'avancement du projet et la répartition des tâches.
- Une page relatant les différents problèmes rencontrés et les solutions envisagées lors de ce projet.

- Une page de téléchargements où il sera possible de télécharger le jeu mais aussi une version du rapport de soutenance finale et une version lite du projet.
- Une page d'annexes où seront données les références aux logiciels, images, sons, bibliothèques, applets et autres éléments que nous aurions pu utiliser.

Je me suis de plus occupé de faire l'hébergement du site grâce GitHub Pages avec le nom de domaine pris grâce à namecheap.com, pour que l'on puisse accéder au site par un lien ayant un nom nous représentant, nous Filiga.

## 2. Travail de Jordan (responsable CSS) :

Mon travail a alors été de faire la mise en page de ces différentes pages et autres fonctionnalités du site :

- Menu
- Banderole
- Bouton d'accueil et bouton pour revenir en haut
- Animation des boutons
- Uniformisation du site
- Mise en place des liens
- Interface d'affichage des profils en JavaScript

Pour cela, pratiquement l'intégralité du site a été réalisée en CSS sauf l'interface d'affichage des profils qui a été faite en JavaScript. Ne voyant pas le moyen de faire une telle interface en CSS et n'ayant aucune connaissance en JavaScript, je me suis aidé d'un template que j'ai modifié pour obtenir ce que je recherchais. Il faut d'abord commencer par découper chaque page en différentes parties selon ce qu'on souhaite afficher. Pour cette division on utilise des blocs "div" auxquels on ajoute soit un "id" soit une "class" pour y faire appel dans le CSS de la page. On peut alors effectuer de nombreux éléments de mise en page tels que l'alignement du texte, la position de certains éléments par rapport à d'autres, la marge intérieur et extérieur des différents blocs, gérer la couleur de fond, mettre des images en arrière-plan etc.

## 4 Réalisation

### 4.1 Fonctionnalités jouables actuelles

Le joueur peut créer une salle d'attente, d'autre joueurs peuvent le rejoindre s'ils connaissent le nom de la salle. Une fois que tous les joueurs sont là, le joueur qui a créé la salle lance la génération de la ville. Une fois la ville générée, les joueurs peuvent rejoindre la partie. Ils ont chacun leurs équipes et peuvent faire apparaître un personnage qui rejoint leur équipe en appuyant sur la touche espace. Ils peuvent déplacer leur personnage (et n'ont pas le contrôle des autres personnages). Pour faire survivre leurs personnages, ils doivent leur faire ramasser de la nourriture trouvable dans certains bâtiments. Une fois ramassé, l'objet est ajouté dans l'inventaire et est utilisable pour nourrir le personnage.

### 4.2 Travaux de recherches

#### 1. Yanis :

Ne connaissant rien d'Unity au début du projet j'ai d'abord suivi les nombreux tutoriels mis à disposition sur le site officiel d'Unity pour prendre en main le moteur de jeu. J'ai donc pu apprendre les bases des interfaces d'Unity et l'essentiel du logiciel, les bases des scripts en C# sous Unity et comment cela fonctionnait, les éléments de physique, d'audio, de graphiques, d'interfaces utilisateurs d'Unity, mais aussi d'apprendre dès le début les bonnes manières et pratiques à prendre avec ce logiciel.

Ensuite j'ai passé de nombreuses heures à me documenter sur les moyens d'accomplir les différentes tâches dans lesquelles j'étais impliquées. Ainsi j'ai fait des recherches sur les différents moyens d'établir un jeu en multijoueur celles-ci m'ont conduit à faire approuver au groupe l'utilisation de photon pour notre jeu qui nous permet d'avoir un serveur sur lequel nous pouvons faire jouer des joueurs en ligne. J'ai donc ensuite recherché quels étaient les moyens fournis par Photon pour adapter le multijoueur à notre jeu et gérer les envois d'informations aux différents joueurs. Mes recherches ont aussi été orientées vers ma tâche principale, celle des menus et interfaces utilisateurs. J'ai donc appris à utiliser les éléments "UI" proposés par Unity et à les adapter à notre jeu à travers des vidéos explicatives et des tests faits par moi-même pour être à l'aise avec ces nouveaux éléments.

De nombreuses recherches ont été menées pour voir comment on pourrait gérer l'inventaire selon nos besoins, avec une question celle de savoir si l'on pouvait mettre en oeuvre la méthode du "Drag and Drop". Par rapport à ce que nous avions mis en place comme système pour l'inventaire cela ne paraissait pas possible. Nous avons donc opté pour une simple interaction, celle d'appuyer sur une touche pour récupérer les item, de cliquer dessus ou encore de faire une interaction par proximité. Il a aussi été utile de faire des recherches pour comprendre l'utilisation des "Scriptable Objects" qui sont très intéressants pour contenir de nombreuses informations sur certains éléments et n'ont pas besoin d'être attachés à un "GameObject", ce qui en font des outils très puissants. Cependant mes recherches ne sont pas restées seulement sur Unity. En effet, étant en partie responsable du site Web et malgré mes quelques connaissances dans ce domaine, j'ai dû aussi rechercher des moyens de faire héberger notre site afin qu'il soit disponible sur Internet.

## 2. Tanguy :

Au début du projet, j'ai décidé d'écouter les élèves des années précédentes qui conseillaient de commencer par la partie multijoueur du projet. Après m'être familiarisé avec Unity, je me suis donc renseigné sur les différents moyens d'implémenter un mode multijoueur. Après diverses recommandations, nous avons choisi d'utiliser l'API Photon. Bien que Photon permette une gestion simple du multijoueur, la compréhension et la mise en place d'un tel système fut une première pour moi.

Pour la génération, j'ai commencé par chercher à savoir si des méthodes de génération de ville existaient déjà, malheureusement les méthodes les plus proches de ce que je désirais ne me satisfaisaient pas, par exemple, un tutoriel de "Holistic3d" disponible sur Youtube permettait de générer un quadrillage de route et de remplir les espaces vides par des bâtiments de tailles similaires. J'ai donc élargi cette recherche à l'ensemble des méthodes de génération procédurale, trouvant de nombreuses méthodes de génération pour "Rogue-like" (genre de jeu basé ce système). Bien que ces méthodes n'étaient pas directement utiles au projet, elles m'ont permis de repenser la notion de génération procédurale.

## 3. Marc-Emmanuel :

Comme début logique du travail en groupe j'ai du me renseigner sur le fonctionnement de git ainsi que sur celui d'Unity, cela étant la première fois que j'ai eu à utiliser ce logiciel. J'ai donc suivi des tutoriels vidéos pour apprendre les bases ainsi que les choses à ne pas faire ou à éviter. Par rapport à mon travail personnel j'ai dû me renseigner sur le logiciel MagicaVoxel, bien que je l'utilisais déjà auparavant je ne savais pas comment exporter des modèles et comment leur attribuer certaines textures ou propriétés. J'ai également dû apprendre à utiliser un Animator sous Unity pour pouvoir mettre en place l'animation ainsi que me renseigner sur le fonctionnement des agents NavMesh pour les utiliser. Enfin j'ai fait un travail de recherches sur des choses que je n'ai pas encore eu l'occasion d'implémenter, notamment tout ce qui touche à une modification procédurale des modèles 3D qui nous permettrait d'avoir une diversité dans le jeu sans avoir à créer un nombre important de modèles différents.

## 4. Jordan :

Yanis et moi étions les seuls à avoir de l'expérience dans la création de site web et de nous deux, j'avais le plus de connaissances en CSS. Par conséquent, j'étais chargé de la conception visuelle du site. De plus, grâce à Tanguy et Yanis qui avaient bien la programmation du jeu, j'ai pu me consacrer à la création du site internet. Lorsqu'un membre du groupe veut rajouter du contenu au site, il l'ajoute directement sur le fichier en Html et met à jour via GitHub. Ensuite je récupère le fichier pour faire la mise en page. J'ai du donc faire quelques recherches pour comprendre le système de branche et de "merge" de git.

Après avoir fini le site web, j'ai commencé à travailler sur le jeu. J'ai commencé par apprendre les bases de Unity en suivant des tutoriels. Ensuite Tanguy m'a expliqué leur avancé dans la programmation et le fonctionnement de plusieurs scripts. Je me suis aussi documenté sur les "delegates" et les "events" pour enfin réaliser un cycle jour-nuit avec un système d'heures en fonction de la rotation du soleil. Je viens tout juste de commencer la programmation sous Unity et ne maîtrise pas encore totalement son fonctionnement. Je suis encore en train d'approfondir mes connaissances sur Unity pour les prochaines soutenances.

### 4.3 Problèmes & Solutions

Jusqu'alors quelques problèmes ont été rencontrés. Ces problèmes n'ont pas été si importants que cela aurait pu être. Voici une liste de nos quelques problèmes :

1. L'utilisation de git en équipe : Nous avions tous été formés à git à travers nos nombreux TPs de programmation mais l'utilisation de git en équipe de plusieurs personnes est évidemment différente. Il a donc fallu faire des branches pour que chacun puisse travailler sur les différentes parties qui lui incombaient. Cependant il fallait très souvent gérer les nombreux conflits lorsqu'on voulait merge (faire une mise en commun dans la branche principale). Ces nombreux conflits étant relatifs à de petites différences lorsqu'on travaillait sur les mêmes scènes Unity étaient très dures à gérer car ces fichiers ne sont pas facilement manipulables comme peuvent l'être les scripts en C# dans lesquels même si des conflits sont présents, on comprend assez aisément comment les résoudre. Nous avons donc décidé de mieux gérer ces conflits en gardant l'idée que chacun travaille sur une branche et en même temps nous faisons des copies des éléments à potentiels conflits. Par exemple pour ne pas avoir de conflits sur un prefab, Tanguy et Yanis vont faire une copie du prefab et travailler chacun sur leur copie. Au moment de la mise en commun via le merge de git, il n'y a donc pas de conflits sur ce prefab, et il faut faire les quelques changements apportés par chacun sur le prefab principal.
2. Un autre problème a été avec la génération procédurale. A un moment du développement, le système s'est heurté à un mur, Photon ne pouvait pas supporter la transmission d'autant d'éléments en multijoueur, en effet jusqu'alors il transmettait l'entièreté des routes et des bâtiments aux autres joueurs. Il fallait donc trouver un moyen de n'envoyer que les informations essentielles aux autres joueurs. Les fonctions RPC ("Remote Procedure Call") permettent d'envoyer des informations de certains types aux autres joueurs (comme des nombres ou des chaînes de caractères) à travers l'appel de fonctions spéciales. Il a donc fallu traduire les informations de génération en une suite de nombre (rangée dans une liste), les envoyer aux autres joueurs puis les retraduire en informations de génération. Ainsi dans la version actuelle du projet, la génération est décidée par un seul joueur, dit "Master" (celui qui a créée la "room") puis les informations de génération sont compactées dans une liste de nombres puis décompactées par les autres joueurs, la génération des bâtiments se fait donc localement.
3. Problème lors de l'adaptation au modèle 3D du personnage et son animation. Avant d'importer ces derniers éléments on utilisait des cubes pour imiter des joueurs, on pouvait alors les faire se déplacer. Mais à partir du moment où l'on a remplacé ces cubes par les modèles 3D des personnages et leurs animations cela ne marchait plus correctement. Le problème ne venait pas des animations en elles-mêmes ni de l'Animator mais des informations envoyées par le navmesh. L'agent ne détectait simplement pas bien le sol et le personnage flottait au dessus, ce qui perturbait le programme. Il a simplement fallu changer quelques paramètres pour qu'il détecte bien le sol. Un autre problème était que le modèle 3D n'était pas défini sous Unity en tant qu'"humanoid", il a donc fallu le changer pour que Unity gère bien les articulations et les propriétés du modèle.

## 5 Prévisions

### 5.1 Correction du planning

Pour cette soutenance nous sommes dans les temps que nous avions prévu lors de notre cahier des charges.

Tâche \ Soutenance	Soutenance 1	Soutenance 2	Soutenance 3
<b>Programmation</b>			
Génération de la ville	85%	100%	100%
Génération du monde	20%	80%	100%
Pathfinding	100%	100%	100%
IA	5%	50%	100%
Multijoueur	60%	80%	100%
Menus & Interface utilisateur	20%	40%	100%
<b>Art</b>			
Textures	5%	30%	100%
Modèle & Animations	20%	40%	100%
Son & Musique	5%	20%	100%
<b>Site Web</b>			
HTML	65%	90%	100%
CSS	65%	90%	100%

■ Tâche en avance par rapport au pourcentage prévu ■ Tâche effectuée

### 5.2 Génération Procédurale

Le système de génération procédurale de ville a été complété à 100%, bien qu'il puisse encore être amélioré, on a atteint un niveau plus que satisfaisant et présentable lors de la dernière soutenance. Le système est extrêmement flexible et peut accepter des nouveaux bâtiments sans toucher au code, ainsi nos objectifs ont été remplis. Maintenant que la génération de ville est terminée, on peut continuer à implémenter la génération du monde c'est-à-dire un réseau de villes connectées entre elles.

### 5.3 Pathfinding

Le système de Pathfinding a été complété à 100%, il a été supplémenté par le système de sélection et de permissions qui garantissent le contrôle et le mouvement des personnages. Vu que le système est fonctionnel nous n'aurons plus besoin de le modifier pour les prochaines soutenances. Tout de même notre implémentation du système reste flexible et peut être changer facilement permettant une maintenance facile du projet.

## 5.4 IA

Nous souhaitons pour la prochaine soutenance implémenter de nouveaux comportements des PNJ (personnages non jouables) ainsi que les premiers ennemis (mort-vivants) mais aussi les premiers éléments de "narration" sous la forme d'évènements. On espère pouvoir interagir avec des PNJ et même permettre aux équipes de les recruter.

## 5.5 Multijoueur

Notre système de connexion dans des salles est fonctionnel avec des fonctionnalités en plus, nous sommes donc en avance. Tout de même il sera nécessaire de continuer à envoyer les différentes informations à chaque joueur à chaque nouvel ajout d'éléments dans le jeu.

## 5.6 Menus & Interface utilisateur

Pour les prochaines soutenances il faudra ajouter un menu de pause (en mode multijoueur, la partie continuera quand même en arrière plan) où le joueur pourra décider de gérer les paramètres du jeu, de retourner dans la partie ou bien de la quitter. Il faudra aussi ajouter au début de la partie le choix de son équipe en créant son équipe et en lui donnant un nom ou bien en tapant le nom d'une équipe déjà présente dans la partie (ou en cliquant sur le nom). Yanis devra s'occuper de mettre en place de nouveaux éléments d'interfaces utilisateurs tels que : un affichage en haut de l'écran de la représentation des différents personnages contrôlés, un système d'inventaire pour les meubles mais aussi implémenter des choix d'actions cliquables lorsqu'on sélectionne un objet en faisant un clic droit dessus.

## 5.7 Fonctionnalités jouables

Pour la prochaine soutenance, on souhaite implémenter de nouvelles fonctionnalités :

- Des PNJ et des zombies
- Un système de combat à distance et au corps à corps
- Un système d'artisanat
- Un système de température
- Une génération du monde avec des villes reliées entre elles par des routes

## 5.8 Textures, Modèles & Animations

D'ici les prochaines soutenances nous souhaitons avoir une progression linéaire de la création de modèles et d'éléments du jeu pour ne pas être pris de court, ce sera donc un travail de fond. Pour ce qui est de la forme nous souhaitons introduire des animations pour d'autres objets que nos personnages (ex : porte) mais également pouvoir ajouter des objets visibles sur les modèles 3D depuis le jeu lors de leur utilisation (ex : torche).

## 5.9 Son et musique

Pour l'instant, le jeu n'a pas de son mais on a pu trouver des sons et bruitages libres de droits. De plus, une connaissance nous fourni ses propres musiques pour le jeu (son soundcloud est en annexe).

## 5.10 Site Web

Pour le site web pour les prochaines soutenances nous souhaitons implémenter d'abord du contenu par rapport à nos avancées, améliorer le visuel (et donc le CSS) de certaines pages et commencer à gérer le "responsive design" du site c'est-à-dire gérer la mise en page du site selon les différents appareils sur lesquels il peut s'afficher.

## 6 Conclusion

Le début de ce projet s'est réalisé sans accrocs. Notre dynamisme et notre organisation nous ont permis de respecter nos objectifs fixés par notre cahier des charges.

Finalement, nous sommes un groupe qui arrive à travailler ensemble, ce qui représente un avantage majeur pour la suite. Nous avons pratiquement tout ce que nous avions prévus pour cette première soutenance.

Notre objectif principal pour cette soutenance était de faire toutes les bases du jeu en terme de multijoueur, de génération procédurale et de pathfinding ainsi que de maîtriser l'utilisation d'Unity, de GitHub et de MagicaVoxel.

Il nous reste beaucoup de chose à faire pour la deuxième soutenance comme prévu dans le cahier des charges, c'est pourquoi nous avons essayé de prendre assez d'avance sur celle-ci. En effet lors de la deuxième soutenance le but sera d'avoir un prototype du jeu qui sera déjà assez convaincant avec les éléments les plus importants incorporés.

## 7 Annexes

### 7.1 Sources

- <https://doc.photonengine.com/en-us/pun/v1/demos-and-tutorials/pun-basics-tutorial/intro>
- <https://docs.unity3d.com/Manual/index.html>
- <https://Unity3d.com/fr/learn/tutorials>
- <https://pages.github.com/>
- <https://nc.me/>
- <https://www.w3schools.com/>
- <https://www.udemy.com>
- [https://www.youtube.com/channel/UCYbK\\_tjZ2OrIZFBvU6CCMiA](https://www.youtube.com/channel/UCYbK_tjZ2OrIZFBvU6CCMiA)
- <https://www.youtube.com/channel/UCCGIF1XekJqHYIafvE7l0c2A>
- <https://www.youtube.com/user/Cercopithecan>
- <https://soundcloud.com/thescrapboy/tracks>

### 7.2 Annexes



FIGURE 3 – menu

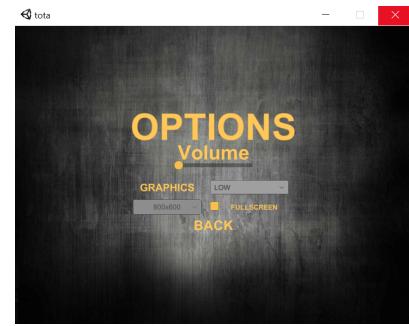


FIGURE 4 – options

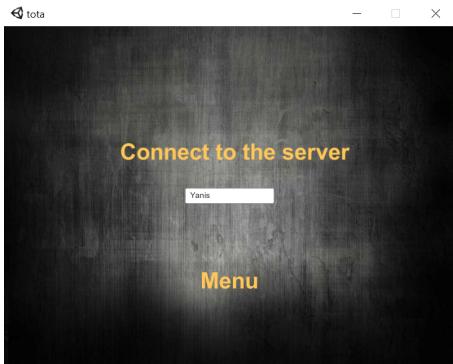


FIGURE 5 – Connexion au serveur

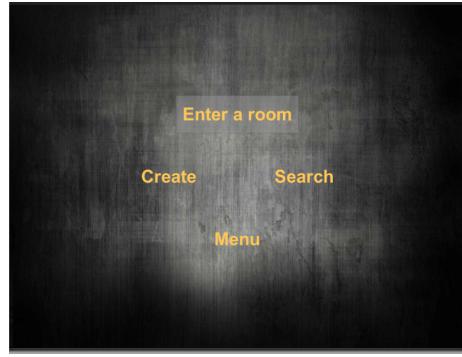


FIGURE 6 – Creation d'une "room"

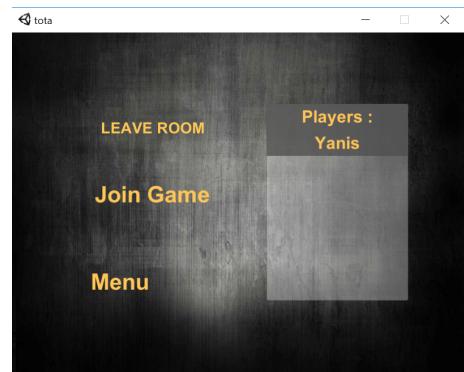


FIGURE 7 – En attente dans une "room"

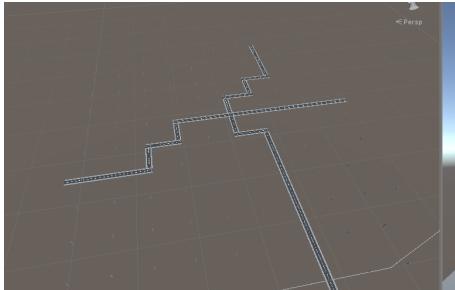


FIGURE 8 – Première ébauche de génération de routes

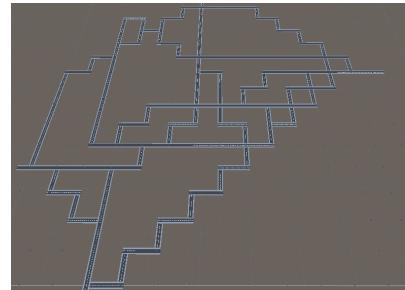


FIGURE 9 – Évolution de la génération procédurale

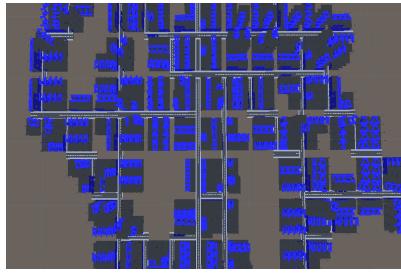


FIGURE 10 – Génération procédurale avec les bâtiments

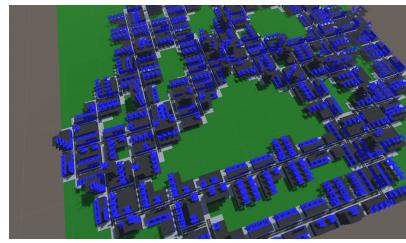


FIGURE 11 – Version actuel de la génération

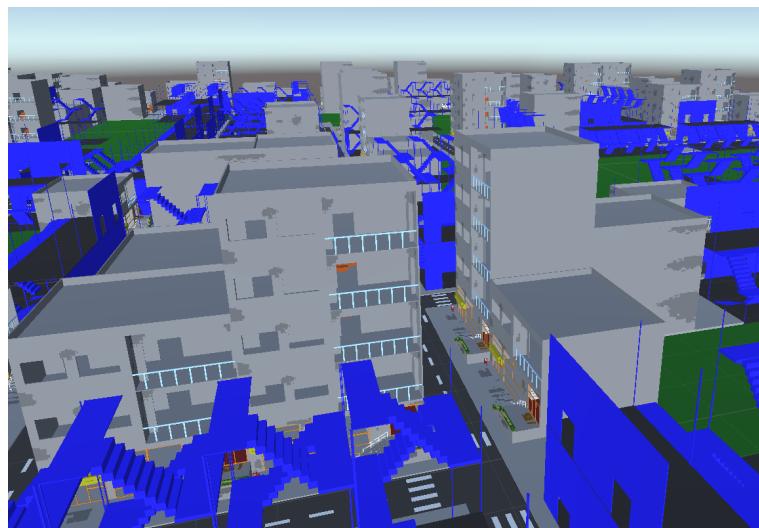


FIGURE 12 – Plan rapproché de la ville

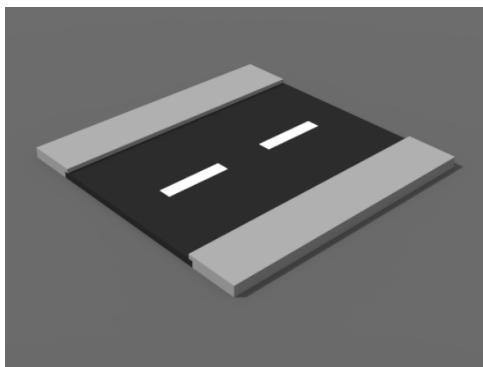


FIGURE 13 – Première modélisation de route

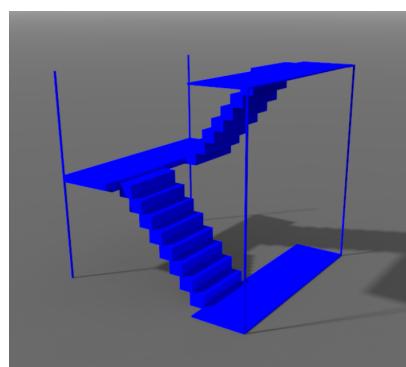


FIGURE 14 – Modélisation d'un escalier



FIGURE 15 – Modélisation d'un frigo



FIGURE 16 – Modélisation d'un personnage



FIGURE 17 – Modélisation d'un bâtiment

Vous pouvez retrouver de nombreux gifs montrant l'avancée de certaines fonctionnalités du jeu sur notre site dans la page de programme. Les liens de nos musiques sont aussi accessibles sur notre site.