

Tales Of The Apocalypse

Rapport de Soutenance 2 - Projet S2



Yanis CHaabane

Jordan FAILLOUX

Tanguy DESGOUTTES

Marc-Emmanuel RAIFFE

Table des matières

1	Introduction	4
1.1	Présentation du projet	4
1.2	Présentation de FIG	4
2	Conception	5
2.1	Génération de villes	5
2.2	Génération du monde	5
2.3	IA	6
2.4	Multijoueur	7
2.5	Menus & Interfaces utilisateurs	9
2.6	Textures, Modèles & Animations	12
2.7	Autres fonctionnalités jouables	13
2.8	Son & Musique	14
2.9	Site Web	14
3	Réalisation	15
3.1	Fonctionnalités jouables actuelles	15
3.2	Problèmes & Solutions	15
4	Synthèse des expériences individuelles	16
4.1	Tanguy	16
4.2	Yanis	16
4.3	Jordan	16
4.4	Marc-Emmanuel	16
5	Prévision	17
5.1	Accord avec le planning	17
5.2	Génération de villes	17
5.3	Génération du monde	17
5.4	IA	18
5.5	Multijoueur	18
5.6	Menus & Interfaces utilisateurs	18
5.7	Textures, Modèles & Animations	18
5.8	Autres fonctionnalités jouables	19
5.9	Son & Musique	19
5.10	Site Web	19
6	Conclusion	20
7	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.

2 Conception

2.1 Génération de villes

1. Travail de Tanguy (responsable) :

Bien que la génération de ville fut complétée pour la soutenance précédente, le système a tout de même été amélioré. Tout d'abord, le système de création d'une route d'un point à un autre a été retravaillé, les routes sont désormais moins linéaires et moins monotones, ce tracé moins prévisible améliore grandement la qualité de la ville. De plus, des petites ruelles se génèrent entre les bâtiments de manière dynamique, précédemment ces ruelles étaient attachées directement aux routes. La structure générale des villes a aussi été revue, une fois les premières connexions de routes établies, des routes se génèrent par extension au niveau de routes déjà générées, cela permet de meilleures liaisons au sein de la ville ainsi qu'une génération de ville plus unique et plus réaliste. Enfin, comme prévu, le système de génération permet de générer des bâtiments de taille 2 x 2.

Une nouvelle notion importante faisant partie de la génération de la ville est la génération des "Prop" (accessoire en anglais), ce nom réfère à l'ensemble des meubles, décos et autres objets. Pour générer ces Props, j'ai implémenté un système de "PropSpot" (emplacement de Prop), chaque PropSpot étant défini par une liste de Prop chacun lié à une probabilité de se générer. Pour inclure ce système dans la génération, il suffit alors de placer ce PropSpot dans un bâtiment du jeu. Ce système de PropSpot est très flexible, on peut créer et placer des PropSpots sans jamais avoir à toucher à du code.

2.2 Génération du monde

1. Travail de Tanguy (responsable) :

A la dernière soutenance, nous générions une ville, désormais nous sommes capable de générer un monde entier. Ce monde a plusieurs "Biome" ou, (pour l'instant "Plain", "Forest et "Arid"), la génération de ces biomes n'est pas complètement aléatoire, ils forment des espaces plus ou moins grand. A l'origine, le monde n'a pas de biome, on remplit certaines cases qui composent ce monde de biomes, ceux-ci sont appelés origine du biome, on fait alors propager chaque case possédant un biome sur ces cases adjacentes (seulement si la case ne possède pas de biome), ce processus s'arrête une fois que toutes les cases du monde ont un biome. Il existe pour l'instant deux types de villes, les villes et les villages. Les villes sont les mêmes que celles présentées dans la partie "Génération de villes", les villages sont des villes plus petites avec moins de connexions de routes, pas de ruelles entre les bâtiments et possiblement des bâtiments différents de ceux dans les villes. Après avoir généré les biomes, on place aléatoirement les villes et les villages. Ensuite on relie les villes aux villes et aux villages proches par un processus similaire à la création de route entre deux points. Pour pouvoir générer ce monde, il a fallu revoir l'intégralité du système de génération de ville.

Désormais, une forêt se génère. La densité et les types d'arbres variant en fonction des biomes. Le système de génération utilise le système de Prop décrit dans la partie "Génération de villes". Cette génération de forêt étant très coûteuse en performance, elle occupe une grande partie du temps de génération du monde.

2.3 IA

1. Travail de Jordan :

Pour la deuxième soutenance nous nous sommes donnés comme objectif de réaliser des intelligences artificielles, notamment un comportement de zombie et d'autres personnages non jouables. Pour l'instant, comme personnage non jouable nous avons un rat qui pourra ensuite être adapté à d'autres animaux. Le déplacement aléatoire et la détection de joueur étant les principales caractéristiques de ces intelligences artificielles, toutes mes recherches se sont centrées sur celles-ci. J'ai commencé par coder le script du zombie car c'est un élément principal du jeu.

Le comportement du zombie est le suivant : il erre aléatoirement en s'arrêtant de temps en temps, dès qu'il détecte un joueur il le poursuit et l'attaque s'il est à bonne portée. Tout d'abord pour le déplacement aléatoire, je prends un point aléatoire A dans un rayon de déplacement prédéfini puis j'utilise une fonction d'Unity pour avoir le point le plus proche de A sur la surface praticable du monde. Le zombie peut ainsi se déplacer sur cette surface. Une des particularités du zombie est qu'il s'arrête parfois de bouger, pour cela j'ai utilisé une coroutine. Une coroutine permet de mettre un temps de pause dans une boucle. Par conséquent, dès que le zombie arrive à destination, il s'arrête quelques secondes puis reprend son déplacement. Ensuite, pour la détection nous devons gérer plusieurs joueurs possibles car un joueur peut contrôler plusieurs personnages. Le zombie doit donc pouvoir gérer tous ces personnages. Pour cela, nous stockons tous les joueurs à portée du rayon de détection dans une liste et le zombie poursuivra le joueur le plus proche de lui parmi ceux dans son angle de vue.

Le rat a pratiquement les mêmes caractéristiques : il se déplace aléatoirement sans s'arrêter et lorsqu'il rencontre un joueur il fuit au lieu de le poursuivre. De plus, le rat possède un angle de vue de 360 degré donc dès qu'un joueur entre dans son rayon de détection il fuit avec une vitesse augmentée.

Nous avons également implémenté une génération aléatoire de noms et prénoms pour les personnages du joueur. Pour cela, nous avons pris une base de données de noms et de prénoms.

2.4 Multijoueur

- Travail de Tanguy (responsable) :

Pour cette soutenance, mon rôle dans le multijoueur était d'adapter les systèmes mis en place décrits dans les parties "Génération de villes" et "Génération du monde". Il existe une limite au nombre d'éléments pouvant être suivis par Photon sur le réseau, cette limite étant fixé à 999. Dépassant cette limite très rapidement (et ce même pour un petit monde), j'ai du, tout au long de l'implémentation de ces systèmes, créer des alternatives n'utilisant pas d'observateur Photon.

Contrairement à la soutenance précédente, les personnages n'ont plus d'observateur Photon. Il existe une instance qui a un observateur Photon qui reçoit et envoie sur le réseau les différentes entrées des joueurs par un système de commandes (très similaire à un système de commandes d'un IRC). Par exemple, lorsqu'un joueur souhaite déplacer un personnage, la commande correspondante ("SetDestination" dans ce cas là) est envoyée localement à l'instance décrite précédemment, cette instance du joueur envoie la commande sur le réseau, cette même instance chez les autres joueurs reçoit cette commande et l'envoie au personnage indiqué par un des paramètres de la commande et exécute la fonction correspondante indiquant au personnage la destination indiquée par les paramètres de la commande.

Un système de commande similaire à celui décrit ici a aussi été implémenté pour les "Props", ceux-ci reçoivent et envoient aussi des messages à une instance s'occupant de relayer ces messages sur le réseau. Ces systèmes de commandes permettent de se libérer des limites imposées par Photon.

- Travail de Yanis (suppléant) :

Après la première soutenance nous avons décidé de reprendre tout notre système de connexion au multijoueur en partant d'une idée de Tanguy :

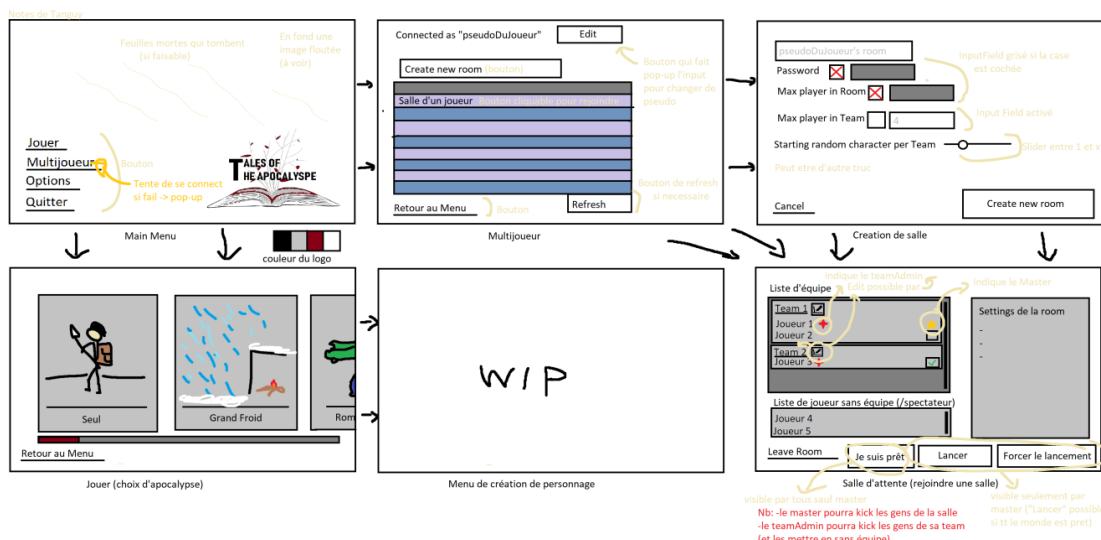


FIGURE 3 – UI

Nous avons donc fait une amélioration avec l'accès plus rapide et efficace à des salles mais aussi de gérer les joueurs et leurs équipes, ainsi que le lancement du jeu par le "MasterClient" lorsque tous les joueurs ont dit qu'ils étaient prêts.

Pour être plus précis le travail s'est divisé en différentes sous-parties :

- La recherche de salle : On utilise désormais les accueils implémentés par Photon et qui nous permettent d'accéder aux listes des salles de chaque accueil. Un tri est effectué sur la liste qu'on reçoit pour que l'on affiche que les salles qui soient accessibles, c'est-à-dire que le nombre de joueurs maximum de la salle n'est pas encore atteint et qu'elles soient considérées comme visibles et ouvertes.
- La création d'une salle : J'ai implémenté la création personnalisée par l'utilisateur de différents paramètres pour la salle à créer. Ces paramètres sont ajoutés à la liste des paramètres de la salle. Ils sont composés de la valeur maximale de joueurs autorisés dans la salle, de la valeur maximale de joueurs dans chaque équipe, du nombre de personnages initialisés dès le début de la partie dans chaque équipe, de la taille de la génération du monde.
- L'attente dans une salle : Pour l'attente dans la salle, soit créée, soit rejointe, j'ai implémenté avec Tanguy tout un système de joueurs et d'équipes avec deux classes "Team" et "Player" qui s'occupent de gérer la création d'équipes et de nouveaux joueurs dans ces différentes équipes ou de les retirer de celles-ci.
- Dans le jeu : Les différents joueurs et équipes sont sauvegardés grâce à la classe "PermissionsManager" qui utilise la fonction "DontDestroyOnLoad" fourni par Unity pour ne pas se faire détruire lors d'un changement de scène et donc garder les différentes informations sur les joueurs et les équipes créées dans la scène précédente. Ainsi dans le jeu, le système de contrôle des personnages va dépendre de l'équipe à laquelle appartient le joueur. En effet chaque joueur peut contrôler les personnages qu'il a créés et ceux créés par les membres de son équipe.

2.5 Menus & Interfaces utilisateurs

1. Travail de Yanis (responsable) :

- Dans le menu : Si l'utilisateur clique sur "Solo" alors il n'arrive plus directement sur la prochaine scène mais il passe d'abord par un choix. J'ai implémenté différents "panels" qui contiennent les images de chaque mode de jeu disponible, les modes de jeu correspondants aux événements qui se dérouleront durant la partie (ex : un mode seul, un mode grand froid, etc). Chaque "panel" est donc composé d'un bouton afin de choisir le mode de jeu. Après avoir appuyé sur le bouton d'un des "panels", on charge la prochaine scène en solo et on garde l'information du mode de jeu choisi grâce à la classe "Mode" sur laquelle on utilise aussi la fonction "DontDestroyOnLoad".
- Dans le multijoueur : Tout d'abord j'ai ajouté un composant de texte en haut lors de la recherche de salle qui affiche le pseudo avec lequel le joueur s'est connecté au serveur Photon. A coté j'ai implémenté la possibilité de modifier ce pseudo. Il y a donc un bouton "Edit" qui permet de faire apparaître un champ dans lequel le joueur peut taper un nouveau pseudo et appuyer sur le bouton "Valid". Ce nouveau pseudo remplace alors l'ancien, est affiché et est sauvegardé pour le reste de la partie et sera gardé en mémoire lors de prochains lancements du jeu.

Il y a ensuite le bouton pour créer une nouvelle salle et la liste des salles qui se trouve dans un composant "ScrollView" et qui donc permet de faire défiler les différentes salles si elles existent. Les salles sont ajoutées les unes en-dessous des autres dans ce "ScrollView" qui est mis à jour à chaque fois qu'une salle est créée, fermée ou détruite. Chaque salle affichée possède un composant bouton qui permet de la rejoindre. Si la salle est déclarée comme privée, il faut alors rentrer un mot de passe. Un "panel" s'affiche alors dans lequel se trouve un champ où l'utilisateur doit rentrer le mot de passe de la salle. Si le texte rentré dans le champ correspond bien au mot de passe, le joueur peut alors rejoindre la salle.

Si on décide de créer une salle, on arrive dans la partie de création des paramètres de la salle. Il y a alors un champ où renseigner le nom que l'on souhaite assigner à la salle. Viennent ensuite les différents paramètres optionnels tels que la possibilité d'un mot de passe, la valeur maximale de joueurs dans la salle, la valeur maximale de joueurs dans chaque équipe et la taille souhaitée pour la génération du monde. Chaque paramètre possède plusieurs composants : un interrupteur (un "toggle") qui permet d'activer ou de désactiver le paramètre et un champ où l'utilisateur associe la valeur souhaitée au paramètre si celui-ci est activé par le "toogle". Un dernier paramètre est présent, il s'agit du choix du nombre de personnages qui vont apparaître dès que le chef d'une équipe va rentrer dans la partie. Ce paramètre est associé à un curseur qui permet donc de choisir entre 0 et 5 le nombre de personnages à faire apparaître.

Lorsque l'utilisateur appuiera sur le bouton "Create Room" il va alors créer une salle en lui ajoutant donc les différents paramètres personnalisés. Si un paramètre est désactivé ou n'a pas de valeur associée dans le champ, une valeur par défaut lui sera attribuée lors de la création de la salle. Il peut sinon appuyer sur le bouton "Cancel" qui le ramènera à la connexion au serveur Photon.

Si le joueur a créé une salle ou en rejoint une, il arrive alors dans la salle d'attente où il peut choisir de rester sans équipe, d'en créer une nouvelle ou d'en rejoindre une déjà existante. Pour ce faire j'ai implémenté deux "ScrollView" : un qui est la liste des joueurs sans équipes, l'autre correspondant à la liste des équipes et des joueurs présents dans ces équipes.

Pour créer une équipe il y a un champ dans lequel on peut mettre le nom de l'équipe à créer et un bouton pour procéder à sa création. Une équipe peut être créée s'il n'existe aucune autre équipe avec le même nom. La liste des équipes est alors mise à jour pour tous les joueurs avec celle nouvellement créée. Pour rejoindre une équipe il suffit simplement de cliquer le nom de l'équipe dans la liste qui possède un composant bouton. Si un joueur sans équipe en rejoint une, la liste des joueurs sera alors mis à jour sans ce dernier. S'il était déjà dans une équipe le joueur peut en créer une nouvelle ou en rejoindre une. L'affichage sera alors mis à jour en mettant le joueur dans sa nouvelle équipe. De plus si l'équipe qu'il vient de quitter ne contenait pas d'autres joueurs elle sera effacée de la liste des équipes dans l'affichage.

Chaque joueur a la possibilité de montrer qu'il est prêt à lancer la partie. Un bouton "Ready" est en effet présent en bas et a pour action d'afficher à côté du nom du joueur une image montrant qu'il est prêt et cette image sera visible par tous les joueurs. On peut aussi voir un "panel" dans lequel sont affichées les différentes informations par rapport aux paramètres de la salle.

Il existe aussi des fonctionnalités disponibles uniquement pour certains joueurs. Le joueur ayant créé une équipe possède le droit de détruire son équipe et/ou de retirer des joueurs de l'équipe via un bouton rouge affiché à droite du nom de l'équipe et/ou du joueur. Ce bouton n'est donc visible que pour le joueur en ayant les permissions. Le joueur ayant créé la salle a quant à lui les mêmes permissions que celles citées précédemment mais il peut le faire de manière générale, c'est-à-dire qu'il peut détruire n'importe quelle équipe, retirer n'importe quel joueur d'une équipe et même retirer un joueur de la salle s'il n'avait pas d'équipe. Une mise à jour de l'affichage des équipes et des joueurs sans équipes est à chaque action de ces utilisateurs effectuée. De plus, ce joueur possède deux boutons de plus que les autres : un bouton "Launch" qui permet de lancer la partie si tous les joueurs sont prêts (et ont donc leur image associée affichée) et un bouton "Force Launch" qui lance la partie même si certains ne sont toujours pas prêts.

- Dans le jeu : J'ai implémenté de nouvelles interfaces utilisateurs telles qu'un menu de pause (le jeu continue quand même en arrière-plan) qui nous permet soit de retourner dans le jeu, soit de gérer les options du jeu, soit de retourner au menu ou bien de quitter complètement le jeu. Ensuite j'ai repris la boîte d'informations qui avait été faite par Tanguy et j'en ai modifié l'affichage. Cette boîte d'informations possède aussi un bouton qui permet d'afficher plus d'informations sur les différentes caractéristiques et capacités du personnages.

J'ai aussi ajouté un "panel" en haut de l'écran qui affiche les références de chaque personnage que nous pouvons contrôler. Pour le moment il s'agit d'images avec le nom du personnage et possèdent un composant "Button" qui permet de sélectionner et de contrôler un joueur et de ramener la caméra au-dessus de celui-ci. Ensuite le système d'affichage d'inventaires a été complètement refait : j'ai implémenté à droite un "Scroll View" dans lequel sont ajoutés les différents inventaires et interfaces des personnages dès lors qu'ils sont sélectionnés par l'utilisateur. Pour afficher cette liste d'interfaces il suffit d'appuyer la touche E si au moins un personnage est sélectionné.

Il existe deux types d'interfaces : tout d'abord l'interface du joueur composée de différents onglets cliquables et donc répartie en différentes parties : il y a en premier lieu l'inventaire du personnage sélectionné dans lequel on peut voir les différents objets qu'il possède et avec lesquels on peut interagir comme à la soutenance précédente avec l'ajout en plus d'un bouton qui affiche la description de l'objet souhaité.

Le deuxième onglet représente les équipements du joueur. Lorsque le joueur possède des objets dans son inventaire qui sont de types "Equipable" ou "Wearable" et qu'il clique dessus ils seront enlevés de l'inventaire et mis dans cette onglet selon les parties du corps auxquels ils doivent être mis, dans les emplacements adéquats).

Le troisième onglet est le système d'artisanat où le joueur peut choisir de fusionner deux objets pour en former un nouveau. Pour cela il doit posséder le bon nombre d'objets pour faire le nouvel objet. Si c'est le cas, la case correspondant au nouvel objet possédera un carré vert. Il y a des conditions pour pouvoir procéder à la création de nouveaux d'objets. Si le joueur ne remplis pas les conditions pour certains, l'objet récupérable sera grisé et le joueur ne pourra pas interagir avec.

Enfin le dernier onglet est l'onglet santé dans lequel sont affichées les différentes statistiques de santé du joueur ainsi que ses blessures.

Le deuxième type d'interface que l'on peut retrouver dans la liste des interfaces sont les inventaires des meubles par exemple. Une interaction est alors possible entre l'inventaire d'un meuble et les interfaces des personnages sélectionnés. Lorsque le joueur va cliquer sur l'objet présent dans l'inventaire du meuble cela va alors le supprimer de celui-ci et l'ajouter dans l'inventaire du premier personnage sélectionné.

Finalement, j'ai implémenté une barre de construction dans laquelle se trouve différents boutons qui permettent de lancer le mode construction implémenté par Tanguy avec le bon élément à construire.

2. Travail de Tanguy (supplément) :

Comme indiqué précédemment, mon travail fut de conceptualiser les interfaces et des fonctionnalités du menu, de la création de salle, de la salle d'attente (cf figure 3) ainsi que les interfaces des personnages (inventaire, équipement, artisanat et santé). Ces interfaces et fonctionnalités ont été implémentés et décrits par Yanis dans ses parties : dans "Multijoueur" et "Interfaces utilisateurs".

2.6 Textures, Modèles & Animations

1. Travail de Marc (responsable)

(a) Textures :

- Le premier travail sur les textures du jeu entre la première et la seconde soutenance a été de faire les images présentes dans les menus. L'idée était d'avoir un environnement rappelant le contexte apocalyptique du jeu sans pour autant que l'endroit soit reconnaissable. En conséquence ces images sont floutées et c'est la raison pour laquelle des images réalistes et non du pixel art ou des scènes voxel ont été choisies, les deux dernières options essayées collant plus avec le style général voulu mais rendant beaucoup moins bien.
- La seconde tâche, plus majeure, a été de commencer à faire toutes les icônes dont nous nous servons dans le jeu. Déjà les icônes d'interface pour le menu par exemple les boutons de validation et d'exclusion du menu multijoueur mais aussi toutes les icônes d'objet visibles dans les inventaires du jeu, on trouve en vrac les armes, les meubles et les composants de construction qui ont chacun besoin de leur texture facilement identifiable.

(b) Modèles :

- Le travail sur les modèles du jeu s'est fait dans la continuité de ce qui avait été fait pour la première soutenance mais à un rythme bien plus soutenu.

Au niveau modèles nécessaires à la conception des villes j'ai jusqu'à présent pu terminer onze bâtiments différents. Pour chacun d'entre eux j'ai d'abord fait un modèle générique avec le découpage des pièces, des escaliers et de quelques éléments dont l'interaction n'est pas possible dans le jeu final comme des sonnettes à l'entrée des immeubles qui seraient hors service dans le cadre d'une apocalypse. J'ai ensuite fait pour chaque plusieurs versions des différents étages des bâtiments, ajoutant des détails comme de la salissure ou encore des effondrements de façade. Cette partie de la conception a pour but de ne pas avoir un environnement trop redondant. La plupart de ces modèles sont déjà ajoutés au jeu mais une partie est simplement encore en conception en attente de formes détaillées alternatives.

Pour ce qui est des modèles de meubles et autres éléments qui constituent la génération des pièces mais aussi des rues avec des poubelles ou encore des 'espaces de nature' avec une végétation variée le travail a là aussi suivi son chemin. J'ai pu finaliser beaucoup de modèles (lit, établis, poubelles, meubles de stockage en tout genre, éléments plus spécifiques à certains bâtiments comme des rayons pour les espaces de vente) qui ne sont pas encore présents dans le jeu car il faudra d'abord définir toutes leurs positions possibles, action indispensable pour la génération de ville. La raison pour laquelle cela n'est pas encore fait a été une question de temps mais c'est surtout qu'il sera préférable de le faire une fois tous les modèles en main pour éviter de revenir sur la position de la génération pour ajouter de nouvelles choses et tout faire efficacement en une fois.

(c) Animations :

- L'animation n'a pas beaucoup évoluée entre cette première et seconde soutenance, nos objectifs initiaux ayant déjà été remplis dès la première. Il y a néanmoins eu un travail de recherche sur l'animation de plusieurs modèles assemblés entre eux et aussi le début d'une constitution d'une banque d'animation personnalisée pour les zombies qui doivent avoir leurs propres manières de se déplacer.

(d) Résumé :

- Les textures et les modèles sont en bon avancement, les avancements étant surtout théorique et le travail de la première soutenance étant voué à être réutilisé.

2.7 Autres fonctionnalités jouables

1. Travail de Tanguy :

(a) Système de traits :

Comme annoncé dans le cahier des charges, nous souhaitons donner une identité au personnages. Cet identité se fait principalement par le système de traits. Lors de la création d'un personnage, des traits lui sont attribués, ces traits modifient les caractéristiques du personnage, par exemple le trait "Strong" (fort) donne +10 à la force du personnage. Il existe plusieurs catégories de traits : physique, mental, professionnel et expérience apocalyptique. Les traits physiques regroupent l'ensemble des traits caractérisants le corps du personnage, les traits mentaux regroupent l'ensemble des traits concernant la condition mentale, les compétences sociales et les connaissances d'un personnage, les traits professionnels regroupent l'ensemble des traits concernant l'expérience professionnel acquise avant le début de l'apocalypse, enfin, les traits d'expérience apocalyptique regroupent l'ensemble des traits concernant l'expérience pendant l'apocalypse. Ce système de trait permettra au joueur de s'attacher plus facilement à ses personnages, ainsi que rendre ces personnages plus mémorables.

(b) Système de santé et de combat :

Pour une expérience de jeu plus intéressante, nous avons implémenté un système de santé relativement complexe. Chaque personnage à plusieurs parties du corps avec des spécificités différentes(tête, jambes, torses, bras, mains etc). Chaque partie du corps peut avoir des blessures. Chaque blessure pouvant avoir des effets différents (saignement, infection, fracture, etc).

Le système de combat existe en parallèle du système de santé. Chaque personnage peut s'équiper d'armes et d'armures. Les armes ont des caractéristiques nombreuses : à distance ou non, tranchante, dégât de l'arme, vitesse d'attaque, etc. Les armures en ont aussi : résistance au coupure, résistance à la température, etc.

Pour illustrer ces deux systèmes prenons un exemple, un personnage en attaque un autre avec un sabre, s'il réussit le personnage est coupé à la jambe, la blessure va saigner, il risque de mourir si son stock de sang se vide. Aussi, comme c'est la jambe qui a été touchée, sa vitesse de déplacement change grandement. De plus, si la coupure est trop importante la jambe est entièrement coupée, le personnage perd alors sa jambe et son pied. Enfin, la douleur provoquée par la blessure affecte les capacités du personnages et peut le faire s'évanouir.

(c) Système d'artisanat et de construction :

Les personnages peuvent créer des nouveaux objets à partir de leurs objets via l'onglet artisanat de leur inventaire. Il existe des recettes permettant d'établir quelles combinaisons d'objets donnent quels objets. De plus, les joueurs peuvent placer des plans de construction, les personnages pourront finir la construction de cette construction en utilisant des objets.

2.8 Son & Musique

1. Travail de Jordan :

Pour cette deuxième soutenance nous voulions implémenter les sons et les musiques que nous avions trouvé pendant la première soutenance pour donner un peu d'ambiance, ce qui est important dans notre jeu. Je me suis donc renseigné sur le fonctionnement de l'audio sur Unity qui est plutôt complet : les sons dans un environnement 3D, le mixage des sons et musiques, les sources de sons et le receveur. Pour gérer tout ça nous avons un "AudioManager" qui, comme son nom l'indique, gère toutes les sources audio. On y stock tous les sons et toutes les musiques et on peut y régler leur nom, leur volume, choisir si elles se répètent, etc. On peut ainsi faire jouer n'importe quel son ou musique n'importe quand et n'importe où dans un script en une ligne de commande. Pour l'instant, nous avons implémenté un cycle de musique en lien avec le cycle jour/nuit. La musique change lorsque la nuit tombe et revient au levée du soleil. Nous entendons aussi un son d'alerte lorsqu'un zombie nous repère. Pour éviter que le son se répète tant que nous sommes dans la vision du zombie, nous créons un nouvel objet quand le son a été joué pour montrer qu'il a déjà été joué une fois puis on le détruit si le joueur sort de la vision du zombie.

2.9 Site Web

1. Travail de Yanis :

Je me suis occupé de la maintenance du site, c'est-à-dire que je me suis occupé de continuer à ajouter du contenu au site par rapport à ce que nous avons fait et ajouté à notre jeu avec l'ajout de nouvelles images et gifs par exemple. J'ai aussi commencé à travailler sur une version du site utilisant un serveur pour l'implémentation d'un site dynamique avec une base de données afin d'avoir des utilisateurs qui peuvent se connecter ou créer des comptes. Ils peuvent aussi envoyer des messages d'avis du jeu sur la page d'accueil ou encore consulter leur page de profil avec les différentes informations concernant leur compte.

2. Travail de Jordan :

Nous avions de l'avance sur le site pendant la première soutenance, la majorité de la mise en page étant déjà faite, il ne me restait plus qu'à faire que le site soit adaptatif au petits périphériques. Je n'avais jamais fais de site dont la mise en page était adaptée à l'appareil utilisé auparavant, donc j'ai du me renseigner en amont. Globalement, si la vue du site devient trop petite le site s'adapte :

- le menu horizontal s'affiche en verticale,
- le bouton animé de l'accueil disparaît,
- la taille des polices du site diminuent,
- le contenu occupe 90% de la largeur de l'écran au lieu d'avoir une taille fixe,
- les tableaux et les gifs sont affichés les uns en dessous des autres.

Nous avons aussi implémenté une base de donnée dans le site grâce au PHP comme prévu. Les pages en PHP ne sont pas encore mises en page et la mise en page globale n'est pas encore terminée.

3 Réalisation

3.1 Fonctionnalités jouables actuelles

Pour cette deuxième soutenance, les joueurs peuvent créer des salles avec des réglages spéciaux, d'autres joueurs peuvent les rejoindre, ils peuvent se mettre ensemble ou non. La taille de la carte où ils jouent étant décidée par celui qui a créé la salle. Le monde dans lequel les joueurs évoluent est un monde ouvert avec un réseau de villes et de villages inter-connectés avec des biomes différents impactant sur le type de végétation générée. Les bâtiments générés dans les villes ont des tailles et des formes diverses. Dans ces bâtiments se génèrent des meubles et des décorations possiblement propres aux bâtiments. Les personnages peuvent fouiller ces bâtiments pour trouver des objets à fonctions variées comme de la nourriture, des armes, des vêtements et des matériaux. Pour trouver ces objets, les personnages fouillent dans des conteneurs qui peuvent stocker les objets. Ces objets peuvent aussi être utilisés pour créer d'autre objets via de l'artisanat, les joueurs peuvent aussi placer des plans de construction que les personnages pourront construire. Chaque personnage a des traits qui le caractérisent, son état de santé est défini par l'état de ses différentes parties du corps. Les personnages peuvent se battre entre eux, ils peuvent mourir de saignements et d'infections. Leur état de santé impact leurs mouvements et leurs caractéristiques. Les joueurs peuvent aussi être mordus par des zombies. Ils peuvent aussi attraper des rats qui fuient à leur approche.

3.2 Problèmes & Solutions

De par notre expérience et nos méthodes acquises pour la première soutenance, nous n'avons pas eu de problèmes trop importants. Le seul problème rencontré était l'envoi de donnée en masse lors de la génération d'un très grand monde provoquant la déconnexion des joueurs du serveur de Photon. La solution étant d'optimiser la génération pour ne pas envoyer toutes les données en même temps.

4 Synthèse des expériences individuelles

4.1 Tanguy

L'expérience du projet jusqu'à présent a pour moi été utile et révélatrice sur plusieurs plans. Tout d'abord sur le plan de la programmation j'ai pu grandement m'améliorer et me familiariser avec les codes à respecter pour travailler en équipe. Le fait d'apprendre en autodidacte à utiliser Unity m'a aussi permis de changer mes méthodes d'apprentissage, une rigueur plus importante étant nécessaire. Sur le plan du travail d'équipe, ayant été désigné chef de projet, j'ai pu expérimenter les meilleurs moyens de répartir le travail au sein du groupe et de garder ce dernier soudé et motivé, expérience et pratiques qui me seront à coup sûr utiles dans le futur.

4.2 Yanis

Après 4 mois et demi à avoir travaillé sur le projet, je peux dire que celui-ci apporte de nombreuses choses. En effet il m'a permis de travailler sur le côté programmation et de m'améliorer dans ce domaine mais aussi de me familiariser avec de nouveaux éléments tel qu'Unity ou encore du côté du site Web. Ce projet m'a aussi apporté beaucoup pour mon futur métier d'ingénieur avec la réalisation du projet en groupe, de la division du travail et de l'échange d'informations entre membres afin de toujours être au courant des avancées des autres. Ce projet m'a permis aussi d'apprendre à toujours chercher des solutions aux problèmes que nous rencontrons et de persévérer jusqu'à la réussite et que même si des fois, psychologiquement c'est dur de rester bloquer pendant des heures sur un même problème, j'ai appris à ne pas lâcher car c'est un travail de groupe et je ne peux pas juste abandonner et laisser les autres se débrouiller avec les problèmes que je leur ai laissé.

4.3 Jordan

Travailler sur ce projet ces quatre derniers mois, j'ai pu apprendre beaucoup de nouvelles choses. Tout d'abord de nouvelles notions de programmation en C# mais aussi de développement web, notamment en CSS avec la réalisation de site adaptatif. Travailler en groupe de quatre m'a également beaucoup appris. En effet, c'est très différent que de travailler en groupe de deux comme j'ai pu le faire en ISN. Grâce à cela, j'ai pu me rendre compte de l'importance de la répartition des tâches et de la communication dans un groupe.

4.4 Marc-Emmanuel

M'étant occupé majoritairement de la modélisation, de l'animation et des aspects graphiques pendant ces quatre premiers mois c'est surtout dans ce domaine que je me suis amélioré. Voir ma propre progression au fil du travail a été un facteur de motivation majeur et les tâches que j'effectuais en une heure au début de ces quatre mois ne me prennent souvent plus que vingt à trente minutes maintenant. Au-delà de ça j'ai pu progresser un peu en programmation et ai appris beaucoup de fonctionnalités de Unity en suivant les autres travaux en cours du projet, notamment en lisant les différents scripts du jeu. Enfin la chose la plus importante que j'ai pu expérimenter et pratiquer durant ces quatre mois est le travail en équipe, les sentiments de devoir envers le groupe sont très positifs et m'ont permis d'avancer même quand il m'est arrivé de craquer un peu après de longues sessions de travail. C'est cette dernière partie de l'expérience qu'est le projet qui me sera sans doute la plus bénéfique pour le futur et que j'ai la plus appréciée jusqu'à maintenant.

5 Prévision

5.1 Accord avec le 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
Programmation			
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%
Art			
Textures	5%	30%	100%
Modèle & Animations	20%	40%	100%
Son & Musique	5%	20%	100%
Site Web			
HTML	65%	90%	100%
CSS	65%	90%	100%

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

5.2 Génération de villes

La génération de la ville a été complété à 100%. Nous ne prévoyons pas d'améliorer la génération de la ville en elle-même. Pour la prochaine soutenance il faudra rajouter les "PropSpot" (emplacement de Prop) à chaque nouveau bâtiment, ce processus ne nécessitant pas de toucher au code derrière la génération.

5.3 Génération du monde

La génération du monde est complétée à 100%, nous sommes en avance. Pour la prochaine soutenance, nous pouvons optimiser la génération pour accélérer le temps de chargement.

5.4 IA

Pour la suite de la conception des intelligences artificielles il faudra créer un modèle d'intelligence artificielle gérant les personnages non jouables non hostiles avec lesquels le joueur pourra interagir. Sur le modèle des zombies et du rat déjà programmé il faudra ajouter d'autres animaux au jeu, la liste précise n'étant pas définie à ce jour sans pour autant que les idées et les méthodes manquent.

5.5 Multijoueur

Les fonctionnalités de connexion et de création de salle, les systèmes de commandes entre Prop et personnages sont finies. Pour la prochaine soutenance, il suffira d'adapter toutes les nouvelles fonctionnalités au multijoueur. De plus nous implémenterons un canal de communications entre joueurs. En effet chaque joueur pourra soit envoyer un message à tous les autres joueurs de la partie ou envoyer un message seulement aux joueurs de son équipe.

5.6 Menus & Interfaces utilisateurs

En termes d'interfaces utilisateurs, il faudra encore polir le design pour essayer d'obtenir des interfaces plus belles. Il faudra aussi continuer à ajouter les différentes textures d'objets faites par Marc.

5.7 Textures, Modèles & Animations

1. Textures :

- Pour la suite il va falloir faire celles des menus maintenant qu'ils sont mis en place, la possibilité de changer le design de choses comme les boutons du menu est aussi d'actualité mais la majeure partie des textures à faire sont celles de tous les objets récupérables, faisables et observable en jeu dans les différents inventaires. Le style cubique de ces textures va rester pour continuer à coller avec le style général du jeu.

2. Modèles :

- Pour la fin de ce projet il va falloir finir tous les modèles nécessaires au jeu et les implémenter sur Unity, à savoir encore une dizaine de bâtiments, les modèles de personnalisation pour les personnages qui vont permettre de les différencier (vêtements, expressions faciales) et surtout beaucoup de meubles et autres objets avec lesquels on peut interagir qui donneront de la vie au jeu et qui devront avoir des positions définies pour chaque bâtiment, route et 'espace de nature'.

3. Animations :

- Au niveau des animations il va falloir incorporer celles des personnages non joueurs hostiles tels que les zombies et aussi celles nécessaires pour les actions de construction ou de combat (ex : avoir une animation pour les différents coups des personnages ou encore des animations de mort)

5.8 Autres fonctionnalités jouables

Les fonctionnalités de jeu qui sont nécessaires et ne sont pas encore implémentées sont, tout d'abord, le système de santé qui va consister en l'implémentation de maladies, de la température et soin et le système d'apprentissage de savoir qui va permettre aux personnages d'augmenter leurs compétences et de débloquer des nouvelles recettes.

5.9 Son & Musique

Pour ce qui est des sons, les bruitages nécessaires au jeu ont déjà été recueillis et archivés et il ne reste donc qu'à tous les implémenter et à les faire coller avec les actions auxquelles ils correspondent. Pour les musiques, celles ci ayant été produites depuis la première soutenance et certaines déjà mises en place, l'ajout ou la modification d'éléments à ce niveau là va déprendre des envies pour le jeu plus que d'un besoin.

5.10 Site Web

Pour le site web, il s'agira de continuer à implémenter le contenu et finir la partie relation entre le site et la base de données afin d'avoir un site dynamique dans le sens des fonctionnalités utilisateurs.

6 Conclusion

A l'image de ce qui s'est passé avant la première soutenance cette période de travail s'est réalisé sans vraiment d'accroc et les objectifs fixés ont été respectés malgré la charge de travail.

Chacun continue de bien travailler au sein du groupe et de l'aide est apportée dès que nécessaire si un des membres a des soucis sur un aspect de son travail.

L'objectif principal de cette soutenance à savoir avancer globalement vers la finalisation du jeu grâce aux bases établies dans la première soutenance et proposer un jeu convaincant et fonctionnel est un succès, la maîtrise des outils et logiciels de travail acquise nous ayant fait atteindre le rythme prévu. La charge de travail restante est tout de même très importante si l'on considère le peu de temps restant avant la date d'échéance mais la quasi finalisation de la partie fonctionnelle et programmation (80%) et l'avance prise dans les tâches artistiques et la mise en forme des interfaces va permettre de passer sur un rythme de finalisation et d'implémentation qui ne devrait soulever aucun problème.

7 Annexes

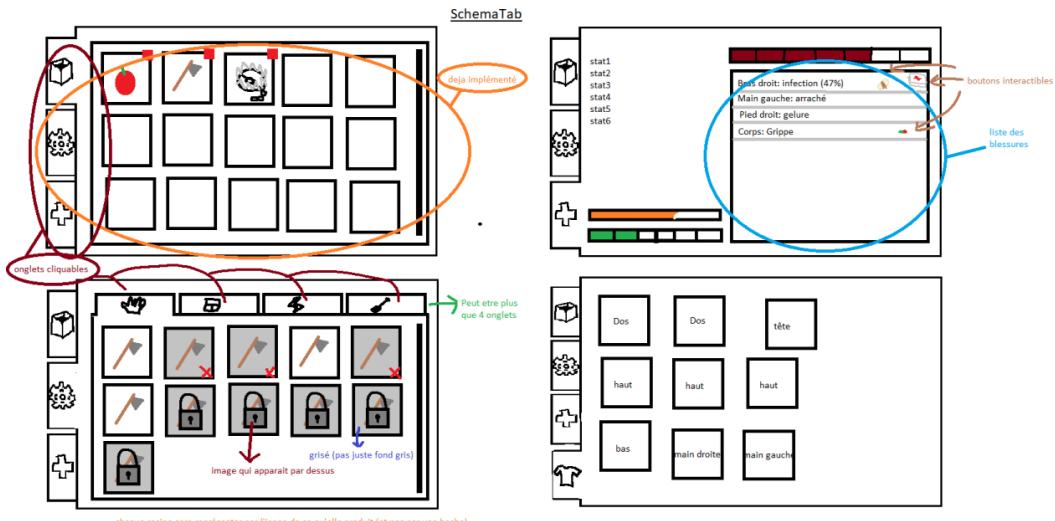


FIGURE 4 – Interface

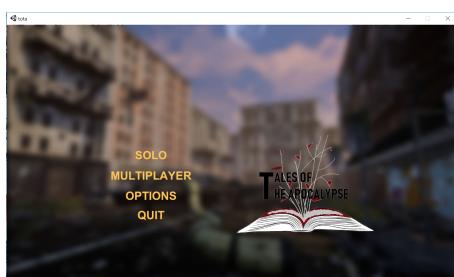


FIGURE 5 – Menu principal



FIGURE 6 – Choix du mode solo

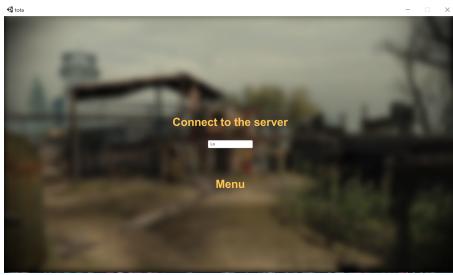


FIGURE 7 – Connexion au serveur

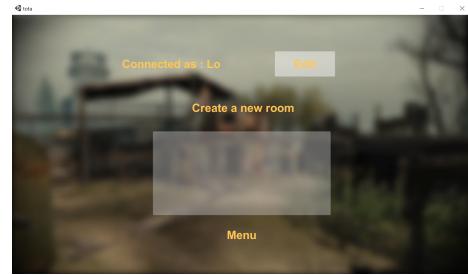


FIGURE 8 – Crédation d'une salle

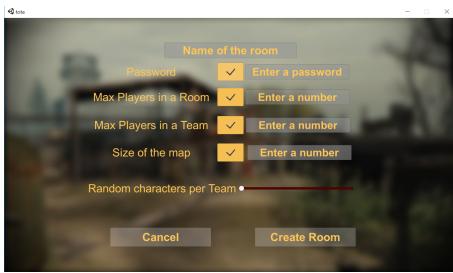


FIGURE 9 – Paramètres de la salle

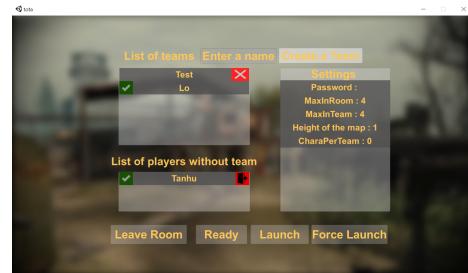


FIGURE 10 – En attente dans la salle



FIGURE 11 – Interfaces utilisateurs



FIGURE 12 – Génération du monde

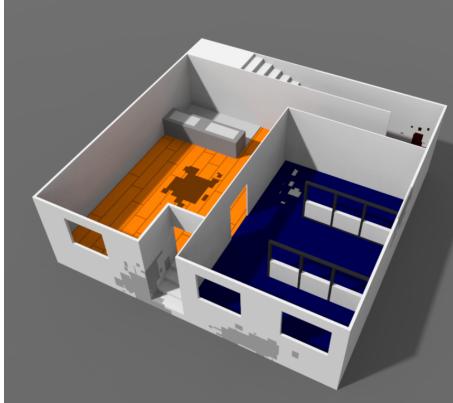


FIGURE 13 – Bâtiment



FIGURE 14 – Bâtiment

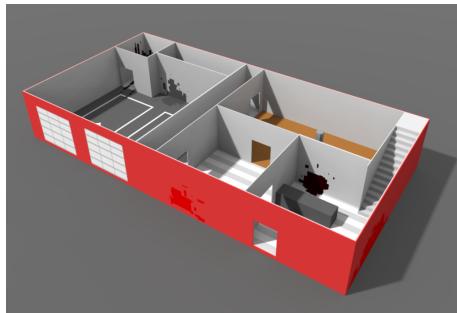


FIGURE 15 – Bâtiment



FIGURE 16 – Lits



FIGURE 17 – Rayons



FIGURE 18 – Établis