

EPITA - PROJET S2
PROMO 2025
2020 - 2021
23 juin 2021

Rapport de projet

POPAT RIZWAAN - TORA BENJAMIN - ROMANACCE LUCAS -
RAULINE MATTHIAS

Chef de Projet : TORA Benjamin

Projet The Hunter
Groupe RedHornet

Table des matières :

1	Introduction	4
2	Le projet The Hunter	5
2.1	Synopsis	5
2.2	Objectif	5
2.3	Répartition des tâches	6
2.4	Technologie utilisées	7
3	Réalisation technique	8
3.1	Présentation générale du jeu	8
3.2	Moteur graphique	9
3.2.1	Menus	9
3.2.2	Décors du jeu	15
3.2.3	Animation	17
3.2.4	Caméras	18
3.3	Moteur Physique	19
3.3.1	Collisions	19
3.3.2	Les déplacements	19
3.3.3	Les tirs	20
3.3.4	Les monstres	21
3.4	Mode de jeu	23
3.4.1	Solo	23
3.4.2	Multijoueur	24
3.5	Intelligence artificielle	25
3.6	Moteur audio	27
3.7	Gameplay	27
3.7.1	Les attaques	28

3.7.2	Les armes	29
3.7.3	Les ennemies	33
3.7.4	Les scores	33
3.7.5	Les cinématiques	34
3.7.6	Les objets	37
3.8	Site web	37
3.9	Discord	38
3.9.1	Serveur Discord	38
3.9.2	Bot Discord	40
3.9.3	Reconnaissance du jeu dans Discord . .	42
4	Installation	42
5	Avancement et état du projet	43
5.1	Avancement	43
5.2	État du jeu	44
6	Ressenti individuel et collectif	45
6.1	Popat Rizwaan	45
6.2	Tora Benjamin	46
6.3	Romanacce Lucas	47
6.4	Rauline Matthias	47
7	Conclusion	48

1 Introduction

Dans le cadre de notre projet informatique en InfoSup, nous avons choisi de réaliser un jeu vidéo.

Notre jeu, appelé "The Hunter" est un jeu de tir à la première personne mixé avec un mode de jeu multijoueur avec des points de vue à la troisième personne. Le concept du jeu s'appuie sur un personnage qui lutte contre une horde de monstre pour sa survie, mais également sur un axe plus amusant avec le multijoueur où des joueurs doivent se cacher parmi les éléments du décor pour échapper au chasseur.

Nous vous proposons, par le biais de ce rapport de projet, de découvrir comment le projet "The Hunter" a été réalisé par le groupe "RedHornet" !



FIGURE 1 – Logo de l'entreprise

2 Le projet The Hunter

The Hunter, Le Chasseur en français

2.1 Synopsis

L'action en multijoueur se déroule dans notre monde au japons. Dans un monde post-apocalyptique, notre héros se retrouve coincé dans un aéroport infecté et doit éliminer le plus de monstre possible. En multijoueur nos personnages se retrouve ensemble dans une ville japonaise dans laquelle le joueur désigné comme chasseur doit éliminer tous les joueurs cachés dans le décor.

Vitesse et précision sont là pour faire briller les meilleurs d'entre vous !

2.2 Objectif

Le finalité de notre jeu est d'éliminer le plus d'ennemies possible en solo avec de l'équipement limité.

Avec trois armes à disposition, notre héros devra jouer avec le temps pour pouvoir les récupérer ou récupérer des munitions tout en affrontant des monstres. Avec une difficulté croissante ces derniers seront de plus en plus puissant et résistant !

Le multijoueur fait s'affronter différent joueurs dans un mode de jeu inspiré de PropHunt :

Le chasseur doit éliminer toutes les proies avant qu'il ne meurt ou qu'il tombe à court de munition. Le chasseur perds des points de vie avec le temps. Les proies peuvent se transformer en éléments du décor.

2.3 Répartition des tâches

Nous avons réparti les tâches de manière à ce que tous les membres du groupe travaillent sur des parties du projet qui les intéressaient, mais également pour que tout le monde travaille sur des parties importantes. Il en résulte la répartition des tâches suivante :

Points	Benjamain	Rizwaan	Matthias	Lucas
Système de tir		XX		X
Physique des objets	XX	X		
Animations			X	XX
Déplacement / Caméra des joueurs		XX		X
Éditeur de map / Menu			XX	X
Intelligence artificielle	XX	X		
Site Web	X		XX	
3D : Blender		X	XX	
Audio	X			XX
Réseau		X	XX	

TABLE 1 – Répartition des tâches

Légende :

XX : Responsable
X : Suppléant

2.4 Technologie utilisées

The Hunter a été entièrement codé en C# avec l'utilisation de Photun Pun.

Les terrains de jeu ont été essentiellement modélisés sur Blender 3D avec l'utilisation de très peu d'assets.

Le site web a été réalisé en HTML/CSS. Les pages sont générées grâce au PHP et accompagné d'une base de donnée MariaDB.

Tous les rapports et pièces écrites du projet ont été réalisés à l'aide de LaTeX.



FIGURE 2 – C'est parti!!

3 Réalisation technique

3.1 Présentation générale du jeu

Le jeu dans sa version finale se découpe en 2 mode de jeu : un mode survie en solo dans lequel notre joueur devra éliminer le plus de monstre avant de mourir. Et un mode multijoueur dans lequel deux équipes s'affrontent, le chasseur contre les proies. Le chasseur, équipé d'une arme à feu, a pour mission d'éliminer les proies. Pour gagner, les proies doivent survivre assez longtemps pour que le chasseur meurt ou qu'il tombe à court de munition.



FIGURE 3 – Version française de la jaquette officiel du jeu

3.2 Moteur graphique

3.2.1 Menus

Les menus ont été réalisés par Matthias.

Menu principal

Le menu principal est composé de trois boutons. Le premier bouton "Jouer" renvoie vers un menu de sélection entre le solo et le multijoueur. Le second bouton "Option" renvoie vers le menu des option qui est expliqué plus bas. Enfin il y a le bouton "Quitter" qui comme son nom l'indique permet de quitter le jeu.



FIGURE 4 – Menu principal

Menu principal - Choix

Dans ce menu l'utilisateur peut choisir dans quel mode de jeu aller. Contrairement à la soutenance précédente le mode solo n'est pas juste un salon Photon avec un seul joueur. Il n'y a donc plus de problèmes lorsque plusieurs personnes vont en solo, ainsi que quand elles n'ont pas internet. Le bouton multijoueur quant à lui emmène dans le menu qui est expliqué ci-dessous.



FIGURE 5 – Menu Choix du mode de jeu

Menu Multijoueurs - Choix

Dans ce menu deux choix sont possible pour continuer. Le premier est de créer son propre salon, dans ce cas le joueur est emmené dans un nouveau menu qui lui demande le nom du salon. Concernant le nom du salon quelques restrictions s'appliquent, le nom ne peut pas être vide et doit faire moins de 17 caractères. Le second choix de l'utilisateur est de trouver un salon, pour ce faire dans le nouveau menu ou il est envoyé, une liste de salon lui est montrer et sont joignable d'un seul clique.

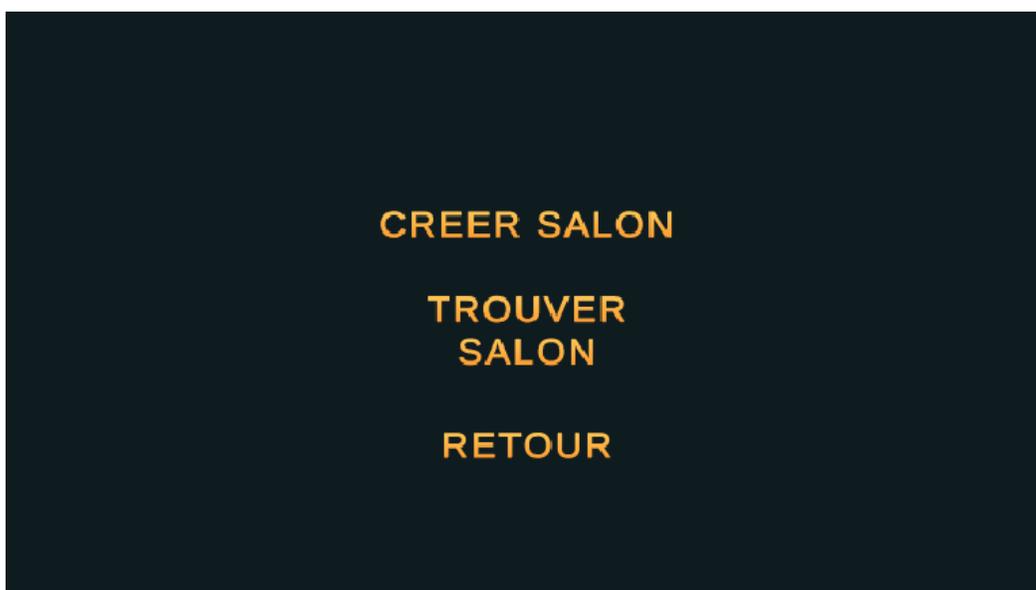


FIGURE 6 – Menu Multijoueurs / Choix

Menu Multijoueurs - Salon

Dans un salon apparaissent plusieurs options, cependant seul le créateur du salon peut les voir. Dans l'éventualité où le propriétaire du salon décide de partir, un nouveau propriétaire est choisi si il reste des joueurs. Sinon le salon est détruit. Parmi les options disponibles, il y a la possibilité de rejoindre le salon en pleine partie, qui est désactivée par défaut. Ainsi que la visibilité du salon pour des utilisateurs qui recherchent un salon pendant que la partie est en cours, cette option est aussi désactivée par défaut.



FIGURE 7 – Menu Multijoueurs / Salon d'avant partie

Menu option

Dans le menu option plusieurs paramètres sont modifiables. Premièrement il y a des paramètres sonore avec le volume qui modifie le son entier du jeu, musique et bruit additionnel tel que le tir d'une arme. Il y a aussi le paramètre pour la musique qui ne modifie que la musique du jeu et permet de toujours savoir ce qu'il se passe en partie.

Ensuite il y a les paramètres graphiques, l'utilité de ces paramètres est plutôt évidente. A l'exception de la synchronisation verticale qui permet de limiter à 60 images par seconde les images envoyées à l'écran dans l'éventualité où la fréquence d'image choisie soit plus élevée que le taux de rafraîchissement de l'écran, ce qui provoquerait un déchirement des images affichées.

Enfin il y a le bouton pour modifier le pseudo choisi lors du premier lancement du jeu et qui est sauvegardé localement sur un fichier nommé "pseudo.tpas".

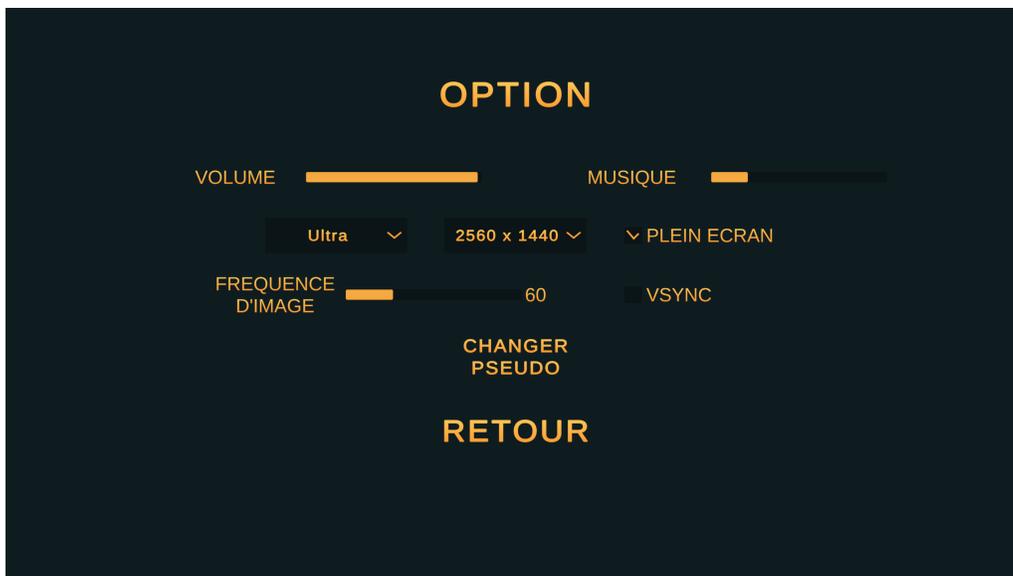


FIGURE 8 – Menu option

Menu pause

Premièrement en multijoueur le menu de pause n'interrompt pas le déroulement de la partie en mettant pause au jeu. Contrairement au menu de pause du solo qui lui arrête le temps du jeu pour permettre au joueur de changer des réglages ou de mettre pause sans se faire attaquer par des zombies.

Le menu de pause possède quatre boutons. Le premier comme son nom l'indique permet de continuer la partie. Le second permet de régler des options comme hors partie à l'exception de quelques unes qui ne sont pas disponibles. Cependant des réglages comme la sensibilité ne sont réglables qu'en partie pour pouvoir mieux tester.

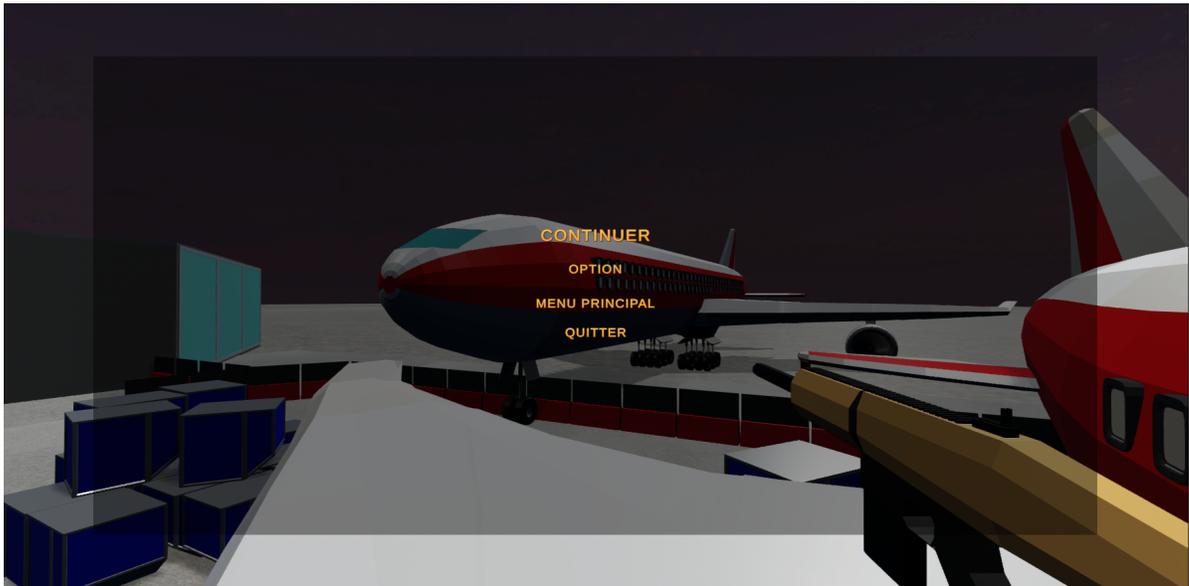


FIGURE 9 – Menu option

3.2.2 Décors du jeu

Solo

Matthias

La carte a été entièrement réalisé dans Blender. Concernant les objets qui sont place dans le bâtiment de l'aéroport, à l'exception des chariot à valises, des valises et de la voiture, le reste des objets a été réalisé pour le jeu dans Blender.

Multijoueur

Lucas

Pour la map multijoueur, elle à été réalisé à l'aide de modèle 3D trouvé sur unity ainsi que de modele réalisé par notre groupe. La Map Asian City est inspirée d'un quartier japonais avec des maisons inspiré de l'architecture des banlieue du Japon. La map est assez étroite et très dense, ce qui est un point fort pour notre mode de jeux. Les intérieurs des maisons on été entièrement modélisé à la main, car nous voulions que les joueurs aient la possibilité de se cacher à la fois à l'extérieur ainsi qu'à l'intérieur.



FIGURE 10 – AsianCity

Les intérieurs de map sont agencé de manière à perdre le Hunter. Plusieurs maisons se ressemblent et de ce fait, il est plus difficile pour le Hunter de trouver ces cibles. Certains éléments dans les bâtiments sont exclusifs à leur bâtiment et d'autre sont utilisé plusieurs fois. Cette répétition des objets est volontaire, car ce sont des lieux de cachette pour les Hiders.



FIGURE 11 – Une maison de AsianCity

3.2.3 Animation

Lucas

Notre joueur le Hunter est un cyborg, il possède des animations pour sauter, marcher, courir. Le Hiders quant à lui ne possède pas d'animation, car le but du jeu est qu'il se transforme en objet et non qu'il reste avec son modèle 3D d'origine. Du côté vu du joueur, le hunter possède une animation de tir avec la vue à la 1re personne. De plus, Rizwaan et Matthias ont ajouté des animations pour le zombie ainsi que pour les armes ramassable du mode solo.

3.2.4 Caméras

Concernant les caméras, Matthias et Rizwaan se sont occupé d'intégrer dans les prefabs des joueurs des caméras de sorte que le chasseur puisse jouer à la première personne et les proies à la troisième. Les scripts C# associés aux caméras permettent aux joueurs de regarder leur environnement et de viser. Le plus compliqué était d'implémenter une caméra à la troisième personne car les proies changent d'apparence. Pour faciliter le jeu des proies, elles peuvent grâce à la touche "H" rendre invisible localement leur apparence pour mieux viser l'objet pour se transformer. Pour finir nous avons ajouté une fonction qui permet après la mort d'un joueur de suivre la partie depuis un mode spectateur.

Avec une caméra libre pour pouvoir suivre les déplacements des joueurs et être au cœur de l'action.

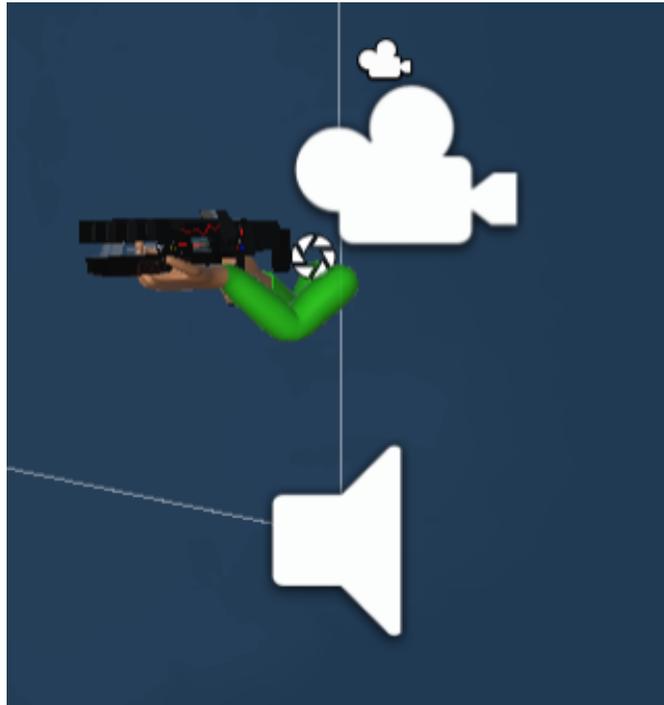


FIGURE 12 – Position de la caméra par rapport à l'arme

3.3 Moteur Physique

3.3.1 Collisions

Rizwaan et Benjamin Les collisions sont en effet une partie très importante du moteur physique. Elles interviennent en tout temps et pour tous les personnages. C'est la dynamique même de tout jeu de tir.

Nous avons attribué à chaque éléments présent sur la scène une paroi de collision pour pouvoir le prendre en compte dans les scripts (Exemple : tir, attaque des zombie, transformation objet). Nous avons également préparé le terrain du mode solo pour pouvoir accueillir l'intelligence artificielle, et pour le multijoueur nous avons ajouté des collisions pour permettre aux joueur d'entrer dans les maisons.

3.3.2 Les déplacements

Matthias + Rizwaan + Lucas

Les collisions lors des déplacements des entités sont très importantes car cela permet au joueur de se déplacer dans un environnement plus conforme à la réalité et cela permet aussi à notre intelligence artificielle de pouvoir suivre le joueur.

Que ça soit la gravité ou encore monter et descendre des escaliers, les parois de collision des objets sont définis pour permettre aux joueurs de jouer sans se heurter contre une parois qu'il ne voit pas en plein milieu de terrain de jeu. Nous aussi du adapter les mécanique de déplacement pour les différents mode de jeux que ça soit en multijoueurs avec le Hunter et le Hiders mais aussi en solo.

3.3.3 Les tirs

Rizwaan et Benjamin

Le système de tir a été créé avec le système de Raycast.

Plus concrètement, un rayon partira du centre de la caméra du joueur et aura comme trajectoire une ligne droite jusqu'à rencontrer un obstacle. Nous avons intentionnellement choisis les objets pouvant être ignorés comme les bordure de terrain ou encore les armes pouvant être ramassé au sol.

Nous avons donc pue choisir les cibles valides et appliquer les dégâts et effets suivant le joueur : Le chasseur utilise des armes à feu pour infliger des dégâts aux monstres et aux proies alors que les proies elles utilise ce système pour sélectionner un objet du décor pour se métamorphoser.

Les tirs sont synchronisés en ligne grâce aux méthode RPC de Photon Pun, celles-ci permettent au joueur qui lance la fonction d'envoyer les information au client qui va renvoyer la fonction sur les autres instances. Donc si un joueur tir, une RPC va être envoyée au client puis aux instance qui vont exécuter ce même tir. Cependant avec Photon, nous pouvons ajouter une condition qui change tout ! Lors du tir, nous allons regarder si c'est bien le joueur qui lance la RPC qui exécute la fonction. Résultat : Un joueur tir (c'est lui qui lance la RPC), les autres instances reçoivent l'information du tir mais ils n'ont pas lancé la RPC, ils vont donc voir le joueur tirer sans eux même tirer.

Pour finir, ce système de RPC nous a permit de synchroniser les dégâts, les particules de tirs, la santé, les mouvements et les scripts de fin de partie.

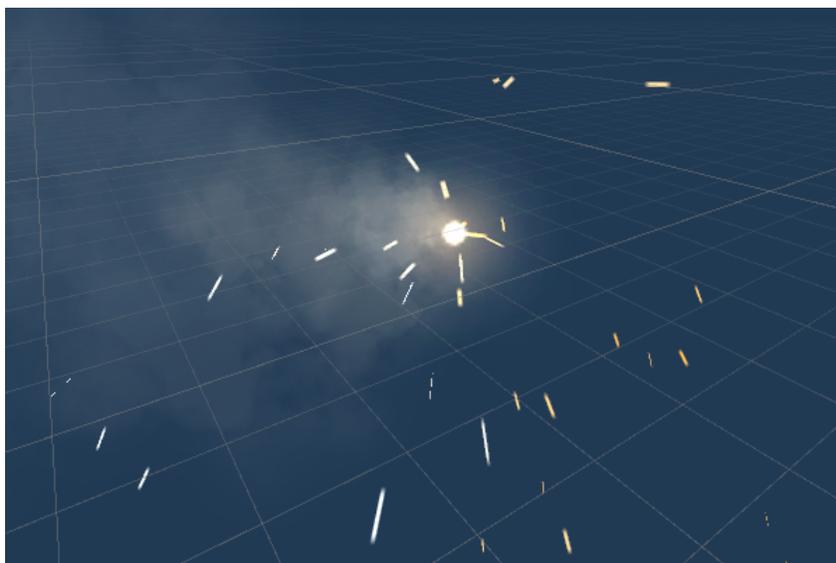


FIGURE 13 – particule d’impact sur surface

3.3.4 Les monstres

Benjamin et Rizwaan

Dans notre jeu les monstres sont contrôlés par notre intelligence artificielle, ces derniers ont pour but d’éliminer notre héros. Ils suivront ces déplacement afin d’être assez proche pour lui infliger des dégâts. Ce qui veut dire que les monstres doivent être en capacité de suivre les joueur, pour cela les monstres sont équipés du même système de collision que le joueur. Si le joueur monte des escaliers, les monstres pourront aussi, et tout ça grâce à un système de baking qui permet à l’IA d’identifier absolument toute la map et de s’y créer des chemins. En effet, il a fallu adapter les maps de manière à ce qu’elles puissent être identifiées comme jouable de la part de l’IA, une fois ceci fait, en début de partie l’intelligence artificielle n’a plus qu’à analyser toute la map, elle sera dans la capacité de trouver par elle-même tous les chemins qu’un joueur pourrait suivre uniquement en marchant en courant. Le seul et unique défaut de ce système est que si le joueur est obligé de sauter au cours de son chemin, l’intelligence artificielle ne pourra pas analyser le chemin car elle est elle-même dans l’incapacité de sauter.

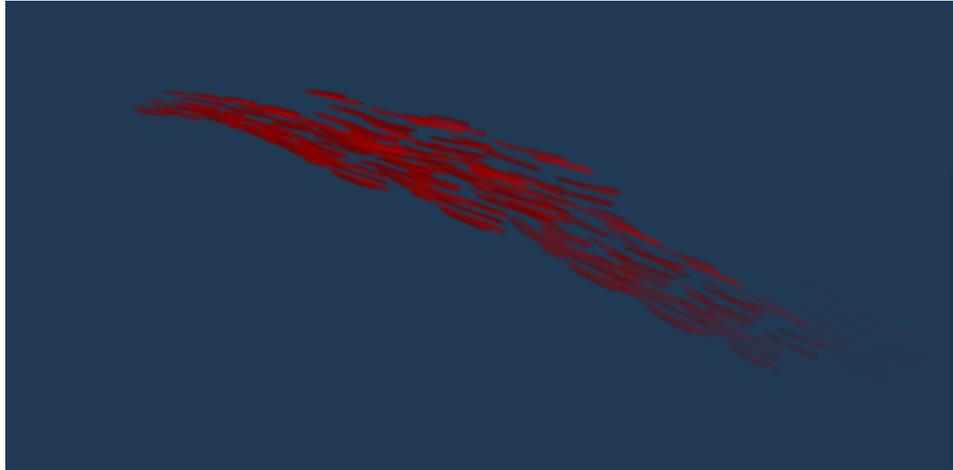


FIGURE 14 – particule d’impact sur ennemies

Nous avons ensuite ajouté plusieurs script C# pour donner à ces derniers plusieurs caractéristiques :

Les dégâts qu’ils infligent au joueur, les points de vie qu’ils possèdent, le temps entre chaque vague, le score, la vague actuelle, les sons associé, les animations jouées et la gestion des scripts associé au déplacement et à la prise de dégâts / mort de ces derniers). De plus les dégâts et les points de vie de ces derniers augmente au fur et à mesure de la partie (à chaque nouvelle vague) et les zombie éliminés permettent d’incrémenter le score. Nous gérons aussi les collision qui disparaissent dès le lancement de l’animation de mort ,pour pouvoir tuer plusieurs zombies alignés, et les différents points d’apparition dans ces scripts.

3.4 Mode de jeu

3.4.1 Solo

Toute l'équipe Pour le solo, le mode de jeu est une survie face a des zombies. Pour gérer la partie un nous avons créer un objet "Game Controller" qui possède le script de la partie. Le fonctionnement du script est assez simple, il y a une liste qui représente l'ensemble des zombies présent dans la scène. De plus un entier représente le nombre de zombie qui doivent apparaître, cela permet de ne pas avoir un nombre trop important de zombie en même temps dans la scène.

Pendant la partie une vérification du nombre de zombie dans la liste est effectuée et trois cas sont possibles. Le premier est que la liste a le nombre maximum de zombie et rien ne se passe. Le second cas est que le nombre maximum de zombie n'est pas atteint dans la liste mais qu'il reste des zombies a apparaître, dans ce cas la les zombies apparaissent. Pour le dernier cas, la liste est vide et il ne reste pas de zombies a faire apparaître, dans ce cas la manche passe et les joueurs ont 5 secondes avant l'arrivée des zombies de la manche.

3.4.2 Multijoueur

Toute l'équipe

Pour le multijoueur Photon a été utilisé. Pour cette dernière soutenance il restait beaucoup de fonctionnalités à synchroniser.

Par exemple la transformation en objet qui était précédemment uniquement locale est maintenant propagée sur l'ensemble des joueurs. Pour faire cela j'ai utilisé une fonction très pratique de Photon qui est appelée Propriétés Customisés. Cela m'a permis d'envoyer un hash qui indique aux autres joueurs qu'une transformation a été effectuée et qu'ils doivent l'effectuer aussi localement. Cela permet de facilement avoir toutes les transformations synchronisées sans perte d'information.

Une autre synchronisation a été la synchronisation des dégâts grâce à une RPC qui permet de dire au joueur touché qu'il doit perdre de la vie localement. C'est une fonction qui nous a pris beaucoup de temps à implémenter pour que seul la personne concernée subisse les dégâts.

De plus, comme nous avons maintenant des animations pour le Chasseur il nous a fallu les synchroniser. Heureusement Photon possède un "Photon Animator View" qui permet de synchroniser facilement les variables utilisées pour passer entre les différentes animations.

3.5 Intelligence artificielle

Benjamin et Rizwaan

L'intelligence artificielle des ennemis a subi de grandes améliorations au fil de l'année. Désormais, les ennemis ont la faculté d'analyse du terrain dans lequel ils évoluent. Ils savent donc où sont les joueurs et les obstacles proches d'eux. Ceci leur permet d'avoir une réponse adaptée à un problème donné.

Pour pouvoir permettre à l'ennemi de prendre en compte son espace afin d'agir en conséquence, nous nous sommes servi d'un périmètre établi autour de lui pour pouvoir détecter les éléments environnant.

L'intelligence artificielle identifie tout l'environnement pour s'y créer des chemins et si le joueur se trouve dans le périmètre de détection, alors il se dirigera vers lui en contournant les obstacles. Mais pour éviter tout problèmes liés à un monstre coincé dans un obstacle ou juste perdu, nous avons établi un chemin qu'il devra suivre si le joueur n'est pas dans son périmètre de détection.

Ainsi, le monstre trouvera toujours le joueur (et pas l'inverse).

Nous avons également ajouté les caractéristiques classique d'un zombie tel que la vie, les dégâts ou encore la vitesse de déplacement. Notre intelligence artificielle utilise un NavMesh Agent pour analyser le terrain et des scripte pour se déplacer vers le joueur ou vers les chemins pré-établis. De plus, avec un système d'apparition par manche, notre intelligence artificielle apparaîtra sur le terrain jusqu'à la mort du joueur en copiant le modèle du zombie avec ses attributs.

Pour finir nous tenons à préciser que notre intelligence artificielle est terminée et qu'elle fonctionne à merveille!

Parlons maintenant des dégâts et de la santé de ces monstres : En début de partie les monstres infligeront peu de dégâts, il pourront éliminer notre héros 10 coups et être éliminé en 2 coups de pistolet (arme de début).

Cependant, à chaque vague de monstre ces derniers gagneront 5 points de santé et 2 de puissance. Tandis que notre héros se soignera de 30 points de santé à chaque vague.

3.6 Moteur audio

Lucas, Matthias Au niveau de l'implémentation des sons, nous avons importé différents sons qui donnent à notre jeu son identité, comme un son de victoire et de défaite, les musiques de menu et d'ambiance, mais aussi les sons de déplacement de tir et de rechargement. De plus pour le jeu en solo, chaque zombie possède son propre son qui est spatialisé en 3D.

3.7 Gameplay

Les idées pour le gameplay de The Hunter ont été réfléchies par toute l'équipe. Matthias et Lucas ont réalisés les objets et les terrains de jeu. Rizwaan et Benajmin se sont occupé des armes et des effets de tir, ils ont aussi réalisé l'intelligence artificielle ainsi que les cinématiques. Matthias s'est occupé des scripts qui s'occupe du bon déroulement des parties.



FIGURE 15 – Le chasseur

3.7.1 Les attaques

Solo

Rizwaan et Benjamin se sont occupé des animations d'attaque des monstres ainsi que les effets d'impact de balle et de fumer qui sort du canon des armes.

Les personnages équipés d'armes à feu pourront utiliser le cliqué gauche de la souris pour tirer et la touche "R" pour recharger l'arme équipé.

Multijoueurs

L'équipe entière a travaillé sur le multijoueur, seul le chasseur est équipé d'une arme. Il faut 5 balles pour éliminer une proies. Avec 60 balles disponible pour un maximum de quatre proies et une durée de vie limité, le chasseur devra être attentif à tout les mouvements et sons de son environnement car comme dit auparavant, les proies émettent des sons à intervalle régulier de 30 secondes. Les proies pourront utiliser la touche "E" pour sélectionner un objet et se transformer, et utiliser la touche "H" pour avoir une vue plus dégagée, une vue sans leur apparence pour pouvoir sélectionner de plus petits objets. Il faut préciser que le chasseur lui, voit toujours l'objet de la proie.

3.7.2 Les armes

Les armes ont été modélisées sur Blender 3D par Matthias avec l'aide de toute l'équipe. Notre jeu comporte trois armes différentes : Un pistolet nommé "Desert Eagle", un fusil d'assaut "Fn Scar" et un canon laser automatique "Laser Gun". Ces trois armes sont jouable en solo et seul le Laser Gun est disponible en multijoueur. Les armes possèdent des particules d'impacts et de tir ainsi que des effets sonore avec leurs propre caractéristique (dégâts, cadence, porté, temps d'apparition en solo, lieu d'apparition, script pour ramasser l'arme) fait par Rizwaan.



FIGURE 16 – Un zombie

Desert-Eagle

Dans un premier temps, nous avons donc le desert-eagle. C'est une arme que l'on peut retrouver dans tout type de jeu, c'est aussi l'arme la moins puissante de toutes celles qui sera possible de trouver. Elle inflige 25 point de dégâts pour une arme coups par coups avec une porté d'attaque de 200 unité de distance.

En lançant le mode solo, le joueur commencera donc avec cette dernière et aura la possibilité en explorant la map de se procurer d'autres armes à propos des quels nous parlerons par la suite.



FIGURE 17 – Desert Eagle

FN-Scar

Pour continuer, nous avons donc la "FN-Scar", arme qui ne sera utilisable, tout comme le desert eagle, uniquement dans le mode solo. Cette fois, ce sera au joueur de la trouver sur la map et de se l'approprier pour combattre les intelligences artificielles.

A la différence du desert eagle, cette arme est automatique, de plus, elle fait plus de dégâts par secondes. Avec 20 points de dégâts par tir, une cadence de tire de 10 unité et une portée de 200 unité, elle est donc bien plus efficace et permettra au joueur de sortir de situation délicates.



FIGURE 18 – Fn Scar

Laser-Gun

Enfin, la dernière arme qui sera présente sur le jeu et modélisée entièrement par l'équipe est la suivante : Le "Laser-Gun".

Le Laser Gun est une arme qui à la différence des autres, sera disponible à la fois dans le mode solo et le mode multijoueur. C'est l'arme la plus adaptée à toutes les situations par ses caractéristiques mais aussi l'arme la plus poussée en terme de modélisation.

Elle sera par conséquent, pour le mode solo, plus compliquée à trouver que les autres mais permettra de prendre l'avantage dans bien plus de situations que les autres armes. Elle inflige 100 points de dégâts par balle, avec un cadence plus réduite de 5 unité et toujours une portée de 200 pour le mode solo. Cependant en multijoueurs, elle inflige 20 points de dégâts avec la cadence et la même portée.



FIGURE 19 – Laser Gun

3.7.3 Les ennemies

Rizwaan, Matthias et Benjamin

Les ennemies de mode solo sont représentés par des zombies avec les animations qui vont avec (course, attaque et mort). En multijoueur, les proies apparaissent en tant que capsule pour obliger l'utilisateur à se transformer en élément du décor sans quoi le chasseur le trouvera rapidement et se fera éliminer, et le chasseur est un androïde (cyborg) équipé du "Laser Gun".

3.7.4 Les scores

Matthias et Rizwaan

Le multijoueur ne possède pas vraiment de score mais plutôt d'une victoire pour l'un des deux camps. Cependant pour le mode solo, le score est représenté par le nombre de zombies tués, il est disponible visuellement pendant toute la partie dans la zone supérieure droite de l'écran. Il est aussi accompagné par le nombre de vagues passées dans le coin supérieur gauche de l'écran. De plus un écran de fin de partie résume le score obtenu avant de renvoyer le joueur au menu principal.



FIGURE 20 – Fin de partie en solo

3.7.5 Les cinématiques

Benjamin

Abordons désormais le sujet des cinématiques. Entre la deuxième et dernière soutenance, nous avons donc travaillé sur les cinématiques. En effet ce fut un travail relativement compliqué qui présenta pas mal d'obstacles que nous allons vous expliquer par la suite.

Pour se faire, nous sommes donc parti sur l'idée d'utiliser une caméra qui serait mobile d'elle même sans que personne n'est rien à faire. Pour se faire, nous avons donc utiliser une fonctionnalité directement intégré a Unity qui se nomme animations.

Le système d'animations nous a permis d'utiliser une caméra et d'enregistrer ses déplacements afin d'en faire une sorte de "vidéo", une cinématique.

Une fois tous les mouvement de caméras fait, il a fallu les placer dans un ordre précis et faire en sorte que le chemin qu'elle suive soit cohérent. Une fois cela fait, la caméra parcourait la map afin de donner une vision global de cette dernière au joueur.

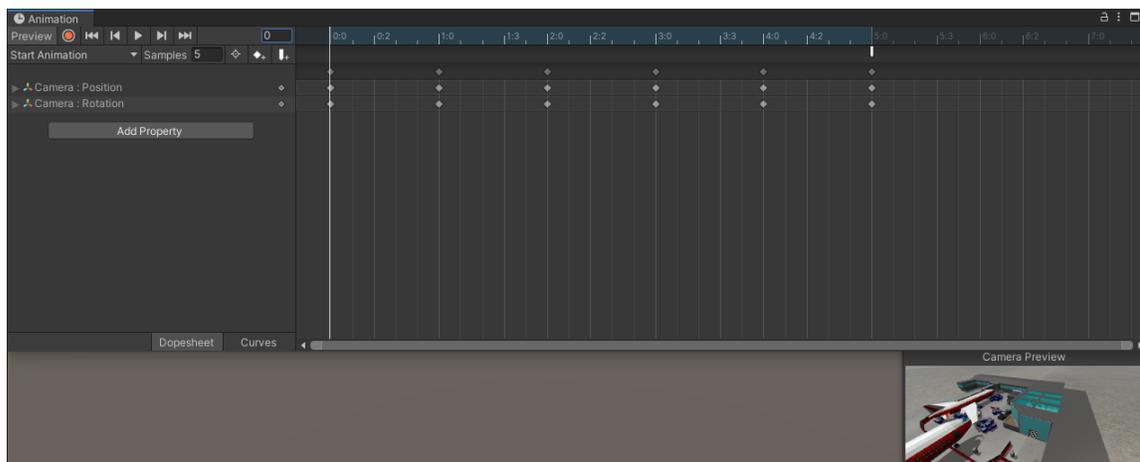


FIGURE 21 – Caméra position 1

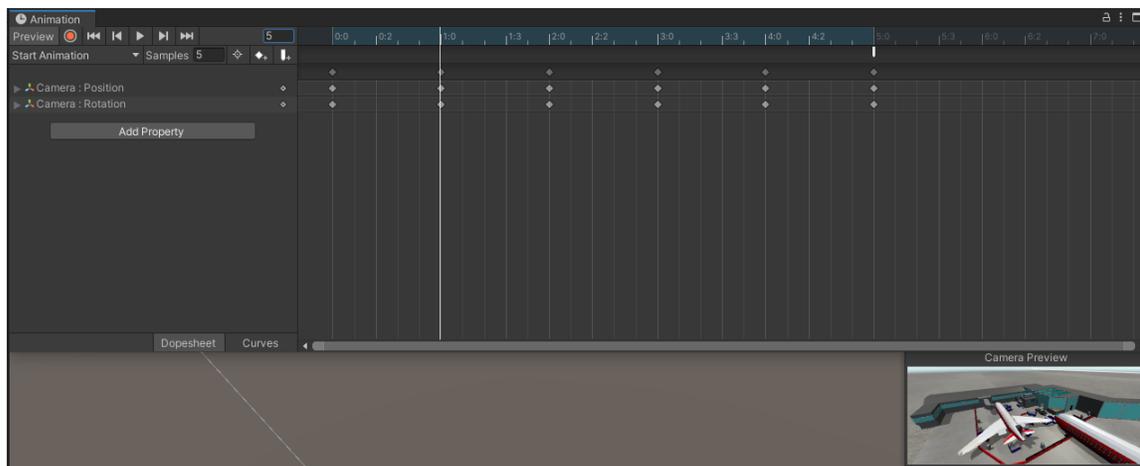


FIGURE 22 – Caméra position 2

Dès lors, des problèmes se sont présentés, en effet la vitesse était bien trop de parcours de la caméra était bien trop élevée, et les mouvements n'étaient pas fluide et pas naturels, ce qui apportait un mauvais ressenti au joueur. Nous avons donc trouver un moyen de ralentir la vitesse de parcours de la caméra ainsi que ses déplacements afin de donner une sensation agréable à celui qui regarde la cinématique. Une fois tout cela fait, il ne manquait plus qu'un dernier détail, la cinématique devait se déclencher au lancement du jeu puis une fois terminée, le joueur devait directement entrer dans la vue du personnage. C'est après pas mal de recherches que nous avons trouver la solution. En effet, nous nous somme servis d'un système de priorité directement intégré à Unity. Pour se faire, nous avons donner une certaine priorité à la caméra présentant la cinématique, plus élevée que celle du joueur. Au fur et a mesure que la cinématique se déroule, l'entier représentant la priorité de la caméra cinématique diminue, pour finalement arrivé a l'entier correspondant à celle du joueur et donc entrer directement dans le point de vue du personnage.

3.7.6 Les objets

Matthias

Comme je l'ai déjà dit dans la partie sur les décors, les objets ont été réalisés dans Blender manuellement. Pour les armes, leur conception a déjà été abordé. Le seul soucis a été concernant les matériaux qui ne s'exportent pas bien a partir de Blender mais ça n'a été qu'un problème mineur. Concernant les objets de transformation, j'ai cherché les objets ayant le plus de chance de faire sens dans une utilisation sur la carte. Les objets comme le banc ou la canette n'ont pas été très long a réaliser. Cependant la réalisation du distributeur a été un peu plus complexe pour réaliser les vis sans fin qui poussent les produit. Enfin quelques objets comme le vélo ou la lampe asiatique viennent d'objet des assets des maisons qui ont été utilise pour la carte du multijoueur.

3.8 Site web

Matthias + Benjamin

Le site web a quelque peu changé depuis la dernière soutenance. Le plus gros changement se trouve dans l'apparence générale avec en plus toutes les informations qui sont maintenant disponible. De plus les pages ont été simplifiées grâce à un peu de code JavaScript. Cela a permis de rendre les pages plus accueillantes et de réduire le temps passé dans les chargements. De plus, la base de donnée est toujours présente en lien avec le code JavaScript pour montrer l'historique du déroulement du projet. Un lien avec notre serveur Discord est aussi présent et permet de rapidement prendre contact avec nous à la moindre question. Les images rectangulaire présentes un peu partout sur le site ont été réalisés sur Photoshop grâce a des images vectoriels libre de droit pour les pictogrammes au centre.

3.9 Discord

3.9.1 Serveur Discord

Benjamin

Comme vous avez pu le voir avec le site web, nous avons laissé un moyen de nous contacter, et pour cela, nous sommes passés par la plate-forme discord. C'est la raison pour laquelle nous allons vous en parler par la suite.

En effet, un serveur discord avait été créé au début de l'année, afin que nous puissions communiquer entre nous. Par la suite nous avons intégré un moyen de rejoindre notre discord sur notre site web, il fallait donc que ce dernier soit présentable pour les arrivants, une refonte totale a donc été faite.

Dans un premier temps : La refonte esthétique, des salons textuels ainsi que des salons vocaux ont été créés afin que les joueurs puissent communiquer entre eux, que ce soit pour de la recherche de joueur, pour parler de tout et de rien ou débattre entre eux.

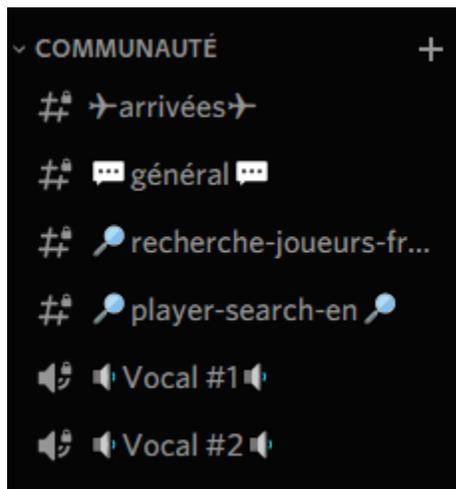


FIGURE 23 – Salons Communauté

De plus, une partie administrative a aussi été créée : En effet, la relation joueurs et développeurs est devenu possible avec la création de salons prévus à cette effet. Les joueurs ont des salons qui leurs sont dédiés et qui sont visibles par les développeur leur permettant de remonter toutes sortes de problèmes liés au jeu, de poser des questions aux développeurs ou encore de suggérer leurs idées pour améliorer le jeu.

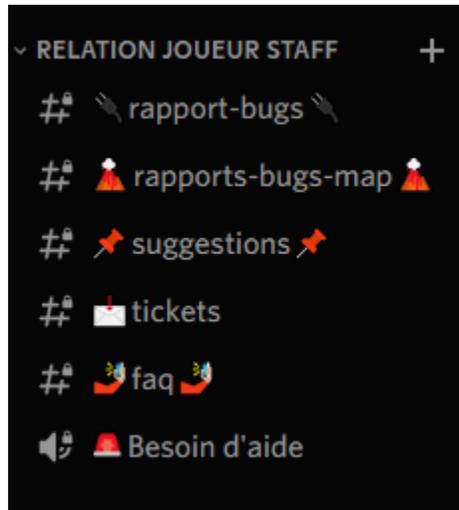


FIGURE 24 – Relation joueurs développeurs

Le but de ce travail était de pouvoir réunir la communauté qui joue au jeu de manière à ce que les joueurs puissent communiquer / débattre entre eux et entrer en communication avec les développeurs pour tout types de problèmes ou de questions.

3.9.2 Bot Discord

Benjamin

Nous avons donc pu parler de la refonte complète qui a été faites avec le serveur discord. Cependant, nous avons beaucoup parlé de l'aspect esthétique et pratique mais nous n'avons aucunement abordé l'aspect technique. En effet, il n'y a pas que l'esthétique qui a été refait.

Dans un premier temps, nous avons fait en sorte que les joueurs rejoignant le discord aient directement et automatiquement le rôle joueur leur permettant ainsi d'avoir accès aux différents salons.

Par la suite, nous avons aussi ajouter un bot qui permet l'ouverture de tickets, aussi appelées demandes. Ce bot n'a pas été créé mais configuré par nos soins afin que tous les joueurs puissent ouvrir des demandes privées, uniquement visible par le staff et permettant d'avoir des réponses rapides et efficaces à leurs demandes si jamais la situation l'oblige. Il est donc possible pour les joueurs d'ouvrir un ticket qui ouvrir un salon textuel que seuls les staffs et le joueur lui même sont dans la capacité de voir, dès lors le joueur peut faire sa demande et une fois que cette dernière a été traitée, lui ou les membre du staffs peuvent fermer le ticket et en garder une copie si besoin est.

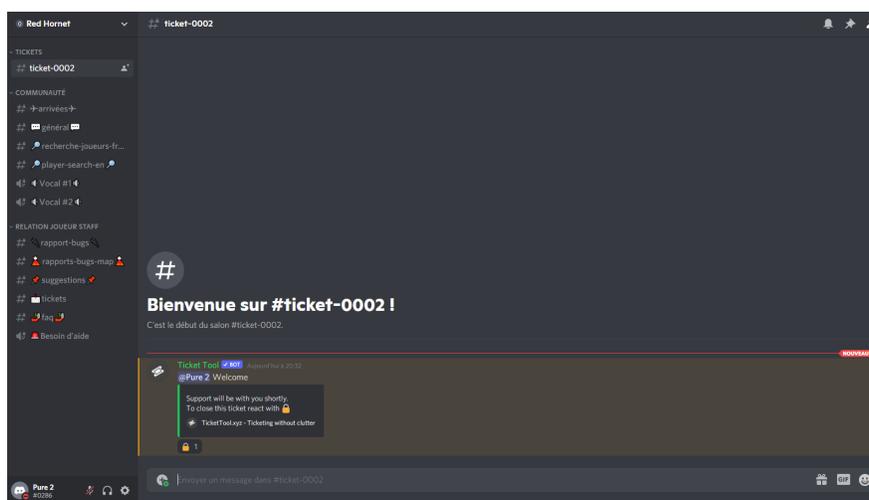


FIGURE 25 – Système de tickets

Enfin, nous avons aussi créé notre propre bot : Le "Red Hornet's Bot". Nous avons pour cela utilisé Node.JS, nous permettant d'avoir accès à toute la bibliothèque mise publiquement par discord afin que les développeurs puissent utiliser toutes les fonctionnalités de la plate-forme pour leurs bots. Nous avons donc un bot privé, qui nous permet de gérer tout l'administration.

En effet ce bot est dans la capacité dans un premier temps de nous montrer tous les logs du serveur au cas où nous en avons besoin pour retrouver des éléments que l'on aurait pu manquer sur le serveur. De plus le bot est aussi dans la capacité d'expulser, de bannir, d'empêcher l'envoi de messages et bien d'autres choses avec une simple commande à entrer.

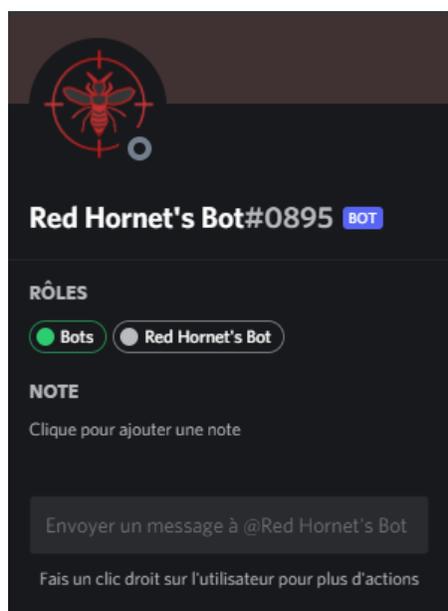


FIGURE 26 – Bot Red Hornet

En résumé, ce bot nous permet d'automatiser tout ce qu'un gérant de serveur discord peut avoir besoin, mais aussi de garder la main mise sur tout ce qui se passe sur ce dernier, même en notre absence. Nous avons jugé que c'était un point plus qu'important étant donné que ce serveur est le seul moyen de nous contacter.

3.9.3 Reconnaissance du jeu dans Discord

Matthias

En utilisant le site pour les développeurs de Discord j'ai pu faire rentrer dans la base de donnée le jeu pour que "Discord Rich Presence" le service de partage d'activité de jeu reconnaisse le jeu. Pour cela, quand le jeu est lancé et que le partage d'activité est activé tous les amis Discord peuvent voir quel est le jeu en cours. De plus en mettant à jours au bon moment dans le code, la ligne ci-dessous "Dans le menu" change quand on arrive dans un salon, que l'on lance une partie multijoueur ou que l'on va en partie solo.

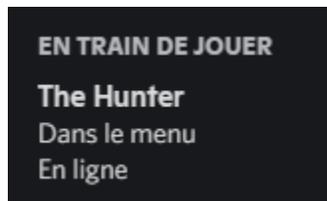


FIGURE 27 – Discord Rich Presence

4 Installation

Matthias

Pour l'installateur j'ai utilise Inno Setup ce qui a permis de facilement faire le script d'installation ainsi que celui de désinstallation qui est créer en même temps.

5 Avancement et état du projet

5.1 Avancement

Points	Avancement (Pourcentage)
Système de tir	100
Physique des objets	100
Animations	100
Déplacement / Caméra des joueurs	100
Réseau	100
Editeur de map / Menu	100
Intelligence artificielle	100
Site Web	100
3D : Blender	100
Audio	100

TABLE 2 – Progression actuelle du projet

5.2 État du jeu

Points	Retards	Terminé.e
Système de tir		X
Physique des objets		X
Animations		X
Déplacement / Caméra des joueurs		X
Réseau		X
Editeur de map / Menu		X
Intelligence artificielle		X
Site Web		X
3D : Blender		X
Audio		X

TABLE 3 – État actuel du projet

6 Ressenti individuel et collectif

6.1 Popat Rizwaan

J'ai beaucoup appris lors du développement de notre jeu. Mes compétences ont été mises à l'épreuve car c'est mon premier grand projet.

Durant cette année, j'ai appris à mieux travailler en groupe et à gérer des tâches communes. J'ai également découvert Unity et plusieurs aspects qui entourent le monde du jeu vidéo comme les animations, la modélisation, le HTML et le CSS ou encore la programmation objet. C'est avec beaucoup de travail et de communication qu'aujourd'hui je suis heureux d'avoir fini ce projet.

J'ai vraiment apprécié travail avec mon groupe car même si j'étais affecté à certaines tâches, j'ai pu apprendre et découvrir leurs travaux. J'ai essentiellement codé en C# pour notre jeu mais j'ai pu apprendre comment créer un environnement de jeu, ajouter des animations ou encore intégrer des sons au sein du jeu.

Je ne compte pas en rester là et pense continuer à créer des projets. Pour finir, que ce soit de la modélisation, du code ou des ajouts de sons, j'ai apprécié chaque moment passé sur ce projet et je n'oublierai jamais cette expérience qui m'a tant appris

6.2 Tora Benjamin

Ce projet a été une totale découverte pour ma part, et elle fut vraiment intéressante.

En effet, avant cela, je n'avais jamais touché à un système du type de Unity, c'est la raison pour laquelle je me suis beaucoup investi et ai beaucoup appris. Pour ma part, je me suis énormément occupé de l'aspect code du jeu avec l'intelligence artificielle, le système de tirs, la barre de vie ou encore même le site internet et bien d'autres choses. C'est une expérience qui m'a appris à être rigoureux, autonome, et efficace lorsque je code.

Cependant mon expérience ne s'est pas arrêté au code puisque sur la fin j'ai aidé dans plusieurs domaines aux quels je n'avais pas touché tout au long. Par exemple j'ai pu m'attarder sur la modélisation 3D en modélisant certains objets de la map solo, j'ai aussi pu m'attarder sur la gestion de communauté avec le serveur discord, découvrir de nouveaux langages avec le JavaScript pour le bot mais aussi toucher aux maps et bien d'autres choses.

En conclusion, cette expérience m'a permis d'apprendre énormément dans énormément de domaines et restera gravée dans ma mémoire!

6.3 Romanacce Lucas

Tout au long de ce projet, je me suis familiarisé avec le moteur de jeux unity.

J'avais déjà commencé le développement d'un jeu avant The Hunter, mais le fait d'être en groupe était bien plus motivant et on pouvait compter sur les compétences de chacun pour différentes tâches. Je me suis plutôt occupé de l'aspect visuel du jeu, avec l'implémentation des animations, la réalisation de modèle 3D ou encore le design de notre jaquette de jeux. Néanmoins, j'ai aussi touché à la partie technique dans les scripts de déplacement par exemple.

Je retiens que du positif de ce projet que ça soit dans la collaboration avec l'équipe, mais aussi dans l'apprentissage du moteur de jeux Unity.

6.4 Rauline Matthias

Grâce à ce projet j'ai pu laisser cour à mon imagination pour la réalisation de fonction face à des problèmes qui ont été rencontrés. De plus j'ai pu apprendre à me servir de Unity et de Blender entièrement, de même que pour les langages HTML, CSS, PHP et un peu de SQL. Mais personnellement le plus gros changement est le fait de travailler en équipe avec des personnes qui ont des attentes, sachant que j'en ai aussi. Mais aussi plein d'idées et de compétences qui auraient été trop longues à toutes apprendre tout seul. De plus le support et la motivation que l'on se donne les un aux autres permet de mieux avancer. C'est aussi pour ça que je suis content que le groupe soit resté entier tout au long de l'année.

7 Conclusion

Vous l'aurez compris, nous avons été particulièrement motivés par ce projet et ce jusqu'à la fin de l'année. En effet, ce dernier nous a permis à tous de découvrir ce qu'était un travail de groupe, en situation, à longue durée, avec ses réussites et ses échecs.

D'autre part, certains casse-têtes ont mis nos nerfs à rude épreuve mais le résultat final prévaut et prouve que nous avons su les résoudre. Enfin, nous sommes tous d'accord pour dire que ce fût une riche expérience sur le plan décisionnel. Il a fallu apprendre à s'écouter, à exposer ses points de vue mais aussi à accepter la critique et en tirer le meilleur possible ; tout le monde y a gagné.

Ainsi s'achèvent ce rapport de projet et par la même notre année de SUP à l'EPITA. Toute l'équipe de RedHornet a été très heureuse de vous faire partager sa passion et son travail tout au long de cette année !



FIGURE 28 – Merci encore pour cette année !

Table des figures

1	Logo de l'entreprise	4
2	C'est parti!!	7
3	Version française de la jaquette officiel du jeu	8
4	Menu principal	9
5	Menu Choix du mode de jeu	10
6	Menu Multijoueurs / Choix	11
7	Menu Multijoueurs / Salon d'avant partie	12
8	Menu option	13
9	Menu option	14
10	AsianCity	16
11	Une maison de AsianCity	17
12	Position de la caméra par rapport à l'arme	18
13	particule d'impact sur surface	21
14	particule d'impact sur ennemies	22
15	Le chasseur	27
16	Un zombie	29
17	Desert Eagle	30
18	Fn Scar	31
19	Laser Gun	32
20	Fin de partie en solo	34
21	Caméra position 1	35
22	Caméra position 2	35
23	Salons Communauté	38
24	Relation joueurs développeurs	39
25	Système de tickets	40
26	Bot Red Hornet	41
27	Discord Rich Presence	42
28	Merci encore pour cette année!	48

Liste des tableaux

1	Répartition des tâches	6
2	Progression actuelle du projet	43
3	État actuel du projet	44