

Rapport de Projet - Bagarena - O.R.E.P.A



Mathis BRAHIM
Arthus DAIMEZ
Marine ROY
Léo WEBER

21 Juin 2021

Table des matières

1	Introduction	4
1.1	Présentations	4
1.1.1	Présentation de O.R.E.P.A	4
1.1.2	Présentation des membres	4
1.2	Logos	5
1.3	Histoire de Bagarena	6
1.4	Distribution des tâches	7
2	Retour sur le Cahier des Charges	7
2.1	Présentation du Projet	7
2.1.1	Inspirations	7
2.1.2	But et intérêt	8
2.1.3	Nature	9
2.1.4	Règles du jeu	10
2.1.5	Déroulement d'une partie	10
2.1.6	Aspect opérationnel	11
2.1.7	Aspects technologiques et méthodologique	12
2.2	Planification des taches	12
3	Site	13
4	Graphismes	16
4.1	Personnages et environnement	16
4.1.1	Mise en place des modélisations des personnages	17
4.1.2	Création de Saadel	17
4.1.3	Création de Tyo	19
4.1.4	Création de Niaka	20
4.1.5	Création de Lasnya	21
4.1.6	Création du marchand	21
4.1.7	Création des autels	22
4.1.8	Réalisation des Animations	23
4.2	Maps	27
4.2.1	Création des maps en général	27
4.2.2	Map solo	28
4.2.3	Map 3vs3	30
4.2.4	L'outil Smooth Height	32
4.2.5	Allègement de la map	33
4.3	Menus et interfaces	34
4.3.1	Menus au démarrage	35

4.3.2	Menus durant une partie	39
4.3.3	Interface pendant la partie	41
5	Intelligence Artificielle	43
5.1	IA en mode solo	44
5.2	IA en mode multijoueur	45
6	Mécaniques de jeu	46
7	Multijoueur	48
8	Son et bruitages	51
9	Bande Annonce	52
10	Bilan du projet	53
10.1	Bilan général	53
10.1.1	Les points positifs	53
10.1.2	Les points négatifs	53
10.2	Les problèmes rencontrés	54
10.3	Bilans personnels	55
10.3.1	Léo	55
10.3.2	Mathis	55
10.3.3	Arthus	56
10.3.4	Marine	57
11	Conclusion	58

1 Introduction

1.1 Présentations

Depuis le début du projet nous avons travaillé en équipe et nous avons dû faire face de nombreux problèmes ensemble, il nous semblait alors primordial de former un groupe solide avec des bonnes affinités c'est pourquoi nous avons formé le groupe O.R.E.P.A.

1.1.1 Présentation de O.R.E.P.A

O.R.E.P.A est une société à but non lucratif née le 1^{er} Décembre. Le groupe est composé de 4 étudiants à EPITA Paris, en première année, dans la même classe. Nous sommes de genres et de personnalités différentes. Nos personnalités étant différentes, nous nous retrouvons dans notre vision du travail et notre organisation, avec ces points en commun et une bonne entente de groupe nous pensons que nous formons une belle équipe pour le projet. Après plusieurs discussions on avait fini par trouver un projet qui convenait à l'ensemble du groupe, c'était important pour nous que chacun des membres valide le projet et se sentent à l'aise et soit motivé par le projet. Suite à cela, nous avons décidé de nommer notre groupe O.R.E.P.A qui n'est autre que APERO à l'envers. En effet, après de nombreux mois où nous étions privés d'apéro nous sommes d'autant plus heureux qu'auparavant de nous rassembler autour d'un apéro. En plus du moment convivial et chaleureux qu'est l'apéro il nous rappelle aussi l'été et le retour à la vie normale.

1.1.2 Présentation des membres

Dans cette fabuleuse société, nous pouvons retrouver 4 membres, étudiants de la classe B2, nécessaires au bon déroulement et à l'achèvement de ce merveilleux projet :

- **Léo WEBER**, le chef de groupe. Il a l'oeil rapide et saura repérer les erreurs rapidement. De plus, possédant un bon sens de l'organisation et un travail efficace il saura veiller sur l'avancement et le bon déroulement du projet.
- **Mathis BRAHIM**, aimant la programmation et ayant quelques facilités dans ce domaine, il apportera une grande aide à l'avancée de la partie programmation du jeu. Aimant apprendre et évoluer, ce projet le motive particulièrement puisqu'il lui permettra d'apprendre de nouvelles notions. De plus, le travail en équipe lui plaît, et saura le

motiver d'autant plus.

- **Marine ROY**, le seul esprit féminin de l'équipe, en plus de sa motivation portée sur le projet, son bon sens de la communication permettra une bonne cohésion au sein du groupe, de plus elle possède déjà quelques compétences en modélisation 3D qui seront utiles au projet.
- **Arthus DAIMEZ**, le couteau suisse de l'équipe, il s'est occupé du logo de la société ainsi que celui du projet. De par son expérience, composée de deux projets déjà programmés, et de par des facilités en informatique, il pourra contribuer à l'avancement du jeu. Étant une personne polyvalente, Arthus sait être utile à tous types de tâche, dont sur la création de modèles 3D. Sortant d'une école militaire, la rigueur, l'organisation, la détermination et le travail bien fait sont ses maîtres-mots.

1.2 Logos

Pour rappel, OREPA signifie Apero à l'envers. C'est pourquoi nous avons voulu un logo représentant cette signification. Suite à de multiples recherches, nous nous étions mis d'accord sur la base du logo, à savoir un cocktail. Pour rendre le logo plus personnel nous souhaitions que le nom du groupe fasse apparition sur le verre. Ensuite, Arthus a dessiné plusieurs modèles, que vous pouvez voir ci-dessous. Et enfin, nous avons choisi de prendre le logo en bas à droite parmi les ébauches ci-dessous, en effet nous voulions un logo plutôt simple, les premiers dessins avaient trop de détails et nous nous étions mis d'accord sur la couleur du cocktail.



FIGURE 1 – Ébauches du logo de O.R.E.P.A

Nous avons fait de même pour le logo du projet, nous voulions un logo qui soit raccord avec notre projet, nous nous étions dirigés vers une main vu que notre jeu est de base un jeu de combat, et que la couleur soit rouge, qui représente la couleur du sang, nous avons aussi décidé que le nom du jeu soit présent. Notre talentueux dessinateur Arthus nous a concocté des superbes logos nous avons plus qu'à choisir notre préféré, pour que le logo du groupe soit du même type que celui du projet, nous avons donc choisi un logo simple sans trop de détails.



FIGURE 2 – Ébauches du logo du projet

1.3 Histoire de Bagarena

Sur le continent d'Evros, les combats en équipe ou les duels sont de culture. En effet, il existe une multitude d'arènes dans lesquels des combattants se battent pour la gloire et la renommée.

Certains d'entre eux se battent pour le plus haut grade de combattant, et ont un niveau nettement supérieur aux autres prétendants à ce rôle. Je parle bien évidemment des combattants Saadel équipé de son trident et de son bouclier, il possède une mobilité certaine et une force de frappe non négligeable, Lasya tireuse d'élite, elle possède l'un des meilleurs arcs du continent, en plus d'avoir gagné des compétitions de précision au tir à l'arc, Tyo il ne se laisse jamais écrasé, sa masse et son grand bouclier lui permettent de garder un minimum de distance avec son opposant en plus d'amortir grandement les coups reçus, et enfin Niaka férue de médecine, elle sait soigner et endommager le corps d'une personne avec ses artifices. Chacun d'eux ont leur chance d'accéder au grade ultime, mais certains nouveaux combattants tentent de s'opposer, avec une chance de succès existante ...

1.4 Distribution des tâches

Pour que ce projet puisse voir le jour et qu'il soit bien réalisé, nous nous sommes séparés les tâches.

Tâches	Léo	Mathis	Arthus	Marine
Edition 3D			S	R
Animation			R	S
Conception de l'environnement			R	S
Physique du jeu		S		R
I.A	S	R		
Musiques et bruitages	R		S	
Création du site web		R	S	
Bande annonce	R	S		
Mise en place multijoueur	S	R		
Responsables Communication	S			R

TABLE 1 – Tableau de la répartition des tâches

R : Responsable de la tâche

S : Suppléant

2 Retour sur le Cahier des Charges

2.1 Présentation du Projet

2.1.1 Inspirations

Lorsque nous avons formé le groupe, on a discuté sur les jeux et les types de jeux qui nous plaisaient. En effet, étant tous les 4 amateurs de jeux vidéos, nous nous sommes rendus compte que nous avions des jeux en commun qui nous avaient marqués. On apprécie les jeux de combat, notamment les jeux de combats tactiques tels que SMITE (développé par Hi-Rez Studios) ou bien League Of Legends (développé par Riot Games). Nous nous sommes donc vite tournés vers ce type de jeu. Suite à cela nous avons réfléchi à un jeu plus simple, un jeu PVP plus classique. C'est donc à ce moment que nous avons pensé au jeu Battlerite (sorti en 2017, par Stunlock Studios). Nous

avons donc choisi un jeu de type combat. Bagarena est né.
Pour la création Bagarena nous nous basons sur plusieurs jeux.

- **SMITE**, en effet, nous nous en sommes inspirés pour son mode Arène dans un de notre mode de jeu FFA (décrit par la suite), ainsi que pour son mode Conflit pour la conception de notre mode 3 vs 3, cependant, certaines différences majeurs restent évidentes, tout d’abord, dans ce jeu les héros sont des dieux de différentes croyances (Égyptienne, Viking, Maya, ...), mais également la vue en TPS est différente de la vue que nous souhaitons avoir. Et enfin ni l’achat d’équipements (pour améliorer nos compétences), ni l’augmentation du niveau ne seront disponible dans Bagarena.
- **League of Legends**, où nous avons comme idée de reprendre l’angle de caméra et le système de sorts. Cependant notre projet ne compte aucun mode de jeu similaire à ce jeu, et nous voulons avoir des graphismes un peu plus cartoon. De plus, la où nous pouvons acheter des items et monter de niveau dans League of Legends, nous ne pourrons pas le faire dans notre jeu.
- **Battlerite**, jeu avec lequel la ressemblance est la plus frappante avec le notre : en effet, nous comptons avoir des graphismes similaire, en plus d’avoir un style de jeu identique. L’angle de la caméra sera le même, le système de déplacements aussi, et enfin le principe des sorts. Néanmoins, ce qui change entre Battlerite et le projet que nous allons mener est que les modes de jeux proposer sont différent : la où le mode arène de Battlerite se finit par l’exécution de tous les membres d’une équipe, notre jeu proposera une fin par destruction d’un autel, ou lorsqu’un seul survivant reste sur le terrain.

2.1.2 But et intérêt

Le but principal de notre jeu est de pouvoir distraire nos joueurs via des sessions de jeux plutôt courtes. L’esprit de Bagarena est celle d’un jeu dynamique avec un rythme rapide mais qui offre cependant une approche stratégique complexe en introduisant des personnages disposant de capacités spécifiques (par exemple : certains personnages seront plus offensifs que d’autres, certains seront plutôt des combattants rapprochés, certains seront plus rapides, d’autres auront des points de dégâts plus élevés,).

L’intérêt est ici de ne pas laisser le temps au joueur de réfléchir mais le forcer à fournir une stratégie adéquate face aux différentes situations qu’il doit gé-

rer. Il va devoir utiliser plusieurs stratégies afin de combattre ses adversaires. Chaque combat étant différent, il devra aussi s'adapter pour combattre ses adversaires.

2.1.3 Nature

Bagarena est un jeu dans lequel les joueurs incarnent un combattant de leur choix, il a à disposition 3 modes de jeux :

- Le premier est un mode 3 vs 3 multijoueur, dont le but est de défendre son Autel des attaques des autres joueurs et de détruire celui du camp adverse.
- Le deuxième mode à disposition est un FFA (Free For All) en multijoueur dont l'objectif est de finir deux fois le dernier survivant d'un groupe de combattants.
- Enfin, le dernier mode est un mode solo qui consiste à défendre son Autel des IAs qui arrivent en vagues de plus en plus nombreuses et difficiles.

Nous avons opté pour des graphismes en trois dimensions, modélisés sur le logiciel Blender, ainsi qu'à l'utilisation du framework Unity. La caméra du joueur sera en vue 3D isométrique. Nos graphismes seront plutôt comme ceux du jeu Fortnite.



FIGURE 3 – Vue isométrique



FIGURE 4 – Graphismes de Fortnite

2.1.4 Règles du jeu

Pour le mode solo :

Le but est donc de défendre un autel face à une horde d'adversaire dont le principal objectif sera de le détruire. Les ennemis vont apparaître par vagues. le but est que l'Autel survive à un maximum de vagues. Ces vagues seront de plus en plus compliquées au fur et à mesure, que ce soit une augmentation du nombre et/ou de la force des ennemis. Entre chaque vagues, l'autel récupérera un peu de ses points de vie.

Concernant le mode multijoueur :

nous proposons 2 modes de jeux différents, un mode FFA et un mode 3 vs 3. Pour le FFA, Il sera organisé en plusieurs rounds dans lesquels seul le dernier survivant gagnera un point. Ici, les règles sont simple : être le dernier en vie. La victoire se fera lorsqu'un joueur auront atteint 3 point. Si au bout de 5 minutes il n'y a pas de vainqueur, celui à qui il restera le plus de point vie sera déterminé comme tel. Pour le 3 contre 3, chaque équipe aura un autel à défendre. Pendant 8 minutes, les deux équipes devront alterner entre défense et attaque, lorsqu'un joueur est tué par un adversaire celui-ci a un certain temps à attendre avant de revenir dans la partie quand le joueur va revivre il sera invincible et ne perdra pas de point de vies pendant quelques secondes. Une partie se termine quand un des deux Autel n'a plus de vie, l'équipe qui aura détruit l'autel ennemie est l'équipe vainqueur.

2.1.5 Déroulement d'une partie

Singleplayer - Lorsque le joueur sélectionne le mode "*Solo*", il arrive immédiatement sur le menu de choix du combattant (en effet, le mode Solo

comporte uniquement "*Défense de l'Autel*") où il devra choisir son combattant pour la partie, puis il devra cliquer sur le bouton "prêt" pour lancer la partie. Il apparaîtra ensuite dans sa zone d'apparition. Une vague ennemie sera lancée 10 secondes après son apparition. Et ensuite il devra tenter de faire survivre son Autel durant le plus de vagues possibles. Il aura une visibilité sur le compteur de vagues durant la partie. Durant sa partie il aura la possibilité d'acheter des bonus chez le marchand. Et enfin, lorsque sa partie sera terminée, le nombre de vagues et la durée auquel l'Autel a survécu seront affichés.

Multiplayer - Lorsque le joueur sélectionne le mode "*Multiplayer*", il sera dirigé vers un écran de choix du mode où il pourra choisir entre le mode "*FFA*" et "*Défense d'autels*". Si il choisit "*FFA*", il arrivera sur la page de choix de son combattant. La partie se lancera lorsque tous les joueurs auront cliqués sur "Prêt". Tous les joueurs vont apparaître sur la map et il y a un décompte de 5 secondes avant de pouvoir attaquer. A la fin de la partie, le nom du vainqueur est affiché. Si il choisit "*Défense d'autels*". Il arrivera sur le menu de choix de l'équipe (il y en a deux). Si le salon de la partie n'est pas complet, les joueurs manquants seront remplacés par des I.A. Les joueurs seront redirigés ensuite sur le menu de choix du combattant. Si un joueur verrouille un choix de combattants, les joueurs de son équipe ne pourront pas choisir le même. Puis, dès que les deux équipes sont prêtes, les joueurs apparaissent dans leur zone d'apparition respectives. A la fin de la partie, le nom de l'équipe gagnante est affichée.

2.1.6 Aspect opérationnel

Rappelons que notre groupe est à but non lucratif. De ce fait nous ne percevons aucun bénéfice de quelques sortes. Notre jeu n'a pour l'instant pas vocation à être rentable. C'est pourquoi les entrées d'argent se résumeront aux dons pour l'instant. Dans un futur plus ou moins proche, les entrées d'argent pourront être la monétisation du jeu, la vente d'objets (dans le jeu), la publicité, et la vente de produits dérivés. Nous n'aurons pas besoin d'investir dans des ordinateurs ou autres matériels informatiques, en effet, nous possédons tous le matériel nécessaire et suffisant à la réalisation du projet. Les dépenses effectués pour le projet seront essentiellement de l'ordre de la nourriture, des livres d'apprentissage ainsi que d'autres achats nécessaires. De plus, l'hébergement de notre site internet fera peut être l'objet d'un léger financement. Nous n'aurons également pas besoin de payer des licences de logiciels. En effet, grâce à l'école nous avons déjà Rider. Et pour la modélisation 3D nous utiliserons Blender qui est gratuit. En terme de communications, le

seul coût éventuel que nous devrions avoir concerne la production papier (cahier des charges, rapports de soutenances). De plus, nous souhaitons faire un compte Twitter, pour obtenir de la visibilité et informer nos joueurs de l'avancement du projet.

2.1.7 Aspects technologiques et méthodologique

Pour l'aspect technologique, nous avons accès à une vaste variété de matériaux et de logiciels pouvant nous aider à développer et concrétiser au mieux notre projet. Nous sommes tous les quatre équipés d'un ordinateur personnel, de plus, si besoin nous avons à disposition les locaux de l'EPITA pour pouvoir utiliser tous les logiciels dont nous aurons besoin. Nous nous servirons de moteur de recherche tel que Google pour nos recherches, Blender pour la modélisation du jeu et bien évidemment Unity. Nous allons utiliser Rider, et Visual Studio pour coder en C#. L'interface graphique et les modèles 3D vont être réalisées sous Blender. Concernant la partie multijoueur, nous allons utiliser le moteur réseau *Photon Unity Network*. Pour le logo, nous l'avons fait sur Krita et Gimp à l'aide d'une tablette graphique. Pour finir, tout support papier sera rédigé en L^AT_EX.

Concernant l'aspect méthodologique, nous avons à disposition une bibliothèque contenant les informations nécessaires nous expliquant comment utiliser des logiciels ou développer dans certains langages, et nos cours de l'école. Nous aurons également la possibilité de se faire aider par les professeurs de l'EPITA, les ACDC, des connaissances des membres du groupe, et les étudiants des années supérieures qui ont plus d'expérience que nous. Aussi, nous disposons des ressources internet telles que les différents tutoriels, manuels numériques sur les logiciels et forums d'entraide.

2.2 Planification des tâches

La réalisation de ce projet c'est déroulé sur environ 5mois, pour bien articuler le projet nous avons au total 3 soutenances durant lesquels nous avons du montrer notre avancement du projet. Afin de pouvoir terminer le projet à temps et de le terminer correctement, nous avons dû bien nous organiser et quoi de mieux que un tableau des objectifs d'avancement.

	1ère Soutenance	2ème Soutenance	3ème Soutenance
Edition 3D	30%	60%	100%
Animation	20%	50%	100%
Conception de l'environnement	25%	50%	100%
Physique du jeu	0%	50%	100%
I.A	20%	60%	100%
Musiques et bruitages	30%	60%	100%
Création du site web	33%	66%	100%
Bande annonce	10%	25%	100%
Mise en place multijoueur	50%	75%	100%
Responsables Communication	33%	66%	100%

TABLE 2 – Tableau des objectifs d'avancement

3 Site

(Mathis)

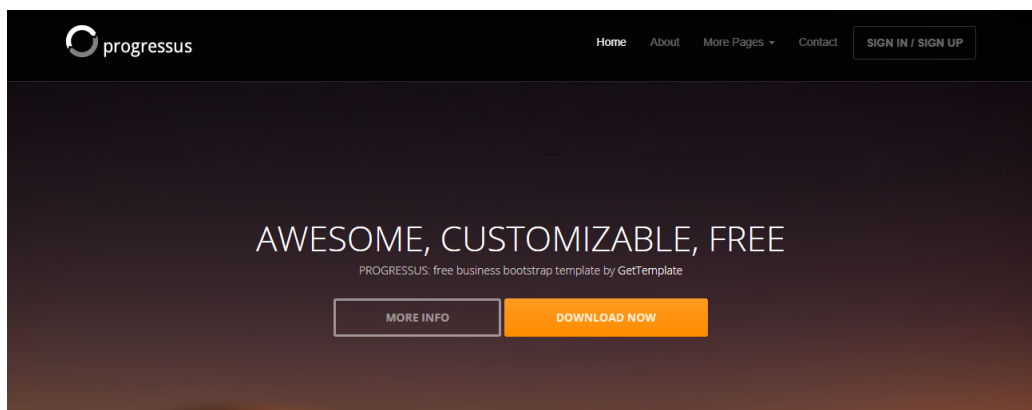
Le site web était terminé globalement dès la première soutenance, j'ai uniquement continué à le mettre à jour avec les différents tweets, rapports de soutenance et les différentes informations et liens de téléchargements concernant la version "finale" et son installation. J'avais souhaité réaliser le site web rapidement pour que l'on puisse se concentrer sur d'autres problématiques et tâches du projet. Pour moi, le site n'était pas l'élément sur lequel nous devons passer le plus de temps, d'où ce choix de l'avoir commencé rapidement et d'y avoir consacré du temps uniquement sur une seule période. Je précise également que je n'avais aucune connaissance web, je me suis donc documenté essentiellement sur Youtube.

Je souhaitais le réaliser essentiellement en langage HTML et CSS. Après de nombreuses heures de documentation je me suis lancé, cependant je me suis rendu compte au bout d'une dizaine d'heures de travail que la partie CSS (graphique) du site allait être difficile à réaliser. En effet, mon manque d'expérience me pénalisait, le site avait un design beaucoup trop ancien. Ci-dessous le design de ma première version.



FIGURE 5 – Première version du site

J'ai donc cherché une solution plus simple. J'ai essayé d'apprendre l'utilisation du framework Hugo et du langage Markdown, cependant j'ai eu du mal avec ceux-ci. Je souhaitais donc rester sur la conception d'un site web en HTML/CSS, après quelques recherches je me suis rabattu sur l'utilisation d'un template Bootstrap (collection d'outils utiles à la création du design de sites et d'applications web) basique pour pouvoir tout recréer de moi-même tout en ayant quelques objets à réutiliser, notamment le menu et quelques composants CSS. De plus, Bootstrap permettant d'avoir un site responsive, ceci me semblait être un atout. Le site a donc totalement été recréé et codé de moi-même, j'ai uniquement gardé le menu, la forme du footer (pied de page), et réutilisé les classes CSS utilisés dans la page de démo du template pour mettre en forme la partie HTML.



The best place to tell people why they are here

The difference between involvement and commitment is like an eggs-and-ham breakfast:
the chicken was involved, the pig was committed.

FIGURE 6 – Progressus - Template Bootstrap utilisé

Le site est organisé de la manière suivante :

- Accueil (présentation rapide du jeu (histoire, modes de jeu), bande annonce) .
- A propos
 - Origine du groupe.
 - Présentation des membres du groupe.
 - Buts et intérêts.
 - Principaux outils et logiciels utilisés pour le projet.
 - Règles du jeu.
 - En bas de la page, téléchargements des rapports et cahier des charges.
- Téléchargements
 - Téléchargements du jeu (version en .exe).
 - Téléchargement du manuel d'installation, mais également présence d'une version simplifiée affichée directement sur le site.
- Actualités
 - Dernier Tweet du projet.
 - Notes de mises à jour et de publications de version du jeu.

Le footer contient notre adresse de contact, les différents réseaux de notre groupe pour nous suivre, et le copyright.



Histoire

Sur le continent d'Evros, les combats en équipe et les duels sont de culture. En effet, il existe une multitude d'arènes dans lesquels des combattants se battent pour la gloire et la renommée. Certains d'entre eux se battent pour le plus haut grade de combattant, et ont un niveau nettement supérieur aux autres prétendants à ce rôle. Je parle bien évidemment des combattants Saadel (équipé de son trident et de son bouclier, il possède une mobilité certaine et une force de frappe non négligeable), Lasnya (tireuse d'élite, elle possède l'un des meilleurs arcs du continent, en plus d'avoir gagné des compétitions de précision au tir à l'arc), Tyo (il ne se laisse jamais écraser, sa masse et son grand bouclier lui permettent de garder un minimum de distance avec son opposant en plus d'amortir grandement les coups reçus), et enfin Nyaka (fée de médecine, elle sait soigner et endommager le corps d'une personne avec ses artifices). Chacun d'eux ont leur chance d'accéder au grade ultime, mais certains nouveaux combattants tentent de s'opposer, avec une chance de succès existante...

FIGURE 7 – Page d'accueil de Bagarena

Pour héberger le site, nous utilisons GitHub Pages qui permet de transformer un projet en hébergement pour site web gratuitement.
Le lien du site web : <https://skraam308.github.io/BAGARENA-website>
J'ai pris énormément de plaisir à faire ce site, et j'y ai consacré vraiment beaucoup de temps afin d'avoir un beau site. J'avais toujours souhaité apprendre à faire un site web et ce fut l'occasion pour moi d'apprendre. Désormais, il faudrait juste que j'apprenne à maîtriser le CSS pour faire des designs modernes, ainsi que le Javascript pour intégrer des mécanismes plus complexes à un site web.

4 Graphismes

4.1 Personnages et environnement

(Marine)

4.1.1 Mise en place des modélisations des personnages

Pour l'ensemble du groupe, il nous était essentiel que les modélisations 3Ds soient faites main et par l'un d'entre nous, cela permettait déjà d'avoir un jeu bien plus personnel et authentique, mais surtout, pour qu'on puisse avoir un réel contrôle sur tout et qu'on puisse réaliser des personnages qui plaisent réellement à l'ensemble du groupe. Comme Marine était la seule personne du groupe qui avait déjà utilisé des logiciels de modélisation 3D, en plus de sa motivation, nous l'avons donc désigné pour qu'elle réalise ces fameuses modélisations.

Comme précisé dans le cahier des charges avant le lancement de la partie le joueur devra choisir le combattant qui devra jouer durant toute la durée de la partie, pour laisser un maximum de choix aux joueurs, nous avons opté pour 4 combattants uniques de genre et de types différents, pour une équité nous avons mis 2 hommes et 2 femmes. Chaque combattant a des capacités et des caractéristiques différentes qui offrent au joueur de multiples gameplay, ainsi le joueur aura le choix entre **Saadel** un combattant corps-à-corps avec un trident, **Lasnya** combattant à distance avec un arc **Tyo** qui sera tout simplement un tank, et pour finir **Niaka** qui sera un soutien.

Comme les modélisations sont des pièces essentielles au projet je ne pouvais pas les négliger, d'autant plus que Arthus en avait besoin plus ou moins rapidement pour pouvoir réaliser les animations. J'ai donc rapidement démarré la modélisation de tous les combattants, comme il me semblait impensable de fournir à mon groupe des modélisations bâclées et très peu esthétiques j'ai fait en sorte de donner le meilleur de moi-même et de travailler régulièrement afin de ne pas pénaliser mon équipe. Pour réaliser les modélisations, j'ai utilisé le logiciel gratuit *Blender*. Après m'être accoutumée au logiciel j'ai enfin pu commencer à créer les combattants.

4.1.2 Création de Saadel

Comme Saadel est un combattant assez classique, j'ai décidé de commencer par celui-ci. Pour créer un personnage sur blender, on commence par des formes simples, comme un carré ou un triangle, et puis à l'aide des outils à notre disposition on les modifie pour en faire des nouvelles formes plus complexes. Ainsi de suite, on fini par obtenir des bouts de notre modélisation (un bras, une jambe etc.) et une fois que toutes les parties nous plaisent, on a plus qu'à les rassembler ensemble et former notre personnage. C'est de cette manière que j'ai fait mes premières modélisations. Une fois le corps terminé, il ne me restait plus que les armes à fabriquer, Saadel est un combattant équipé d'un trident et d'un bouclier, après les avoir modélisés j'ai

dû rajouter des os (armature) sur les combattants, afin que par la suite il puisse être correctement animé par Arthus. Ensuite il ne me restait plus qu'à appliquer les textures sur Saadel pour qu'il devienne encore plus beau. Mais avant ça, j'ai préféré commencer le reste des autres modélisations. Après avoir fini les autres modélisations des combattants j'ai décidé de les exporter en *.FBX* pour pouvoir les importer sur Unity, malheureusement c'est là que les problèmes ont commencé, lorsque mes modélisations étaient importées dans Unity nombreuses apparaissaient avec des trous, voire des membres en moins, c'est alors pourquoi j'ai dû retravailler quasiment entièrement toutes mes modélisations.

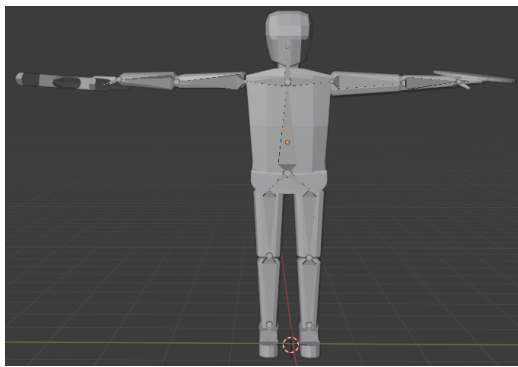


FIGURE 8 – Première version de Saadel

J'ai pu en profiter pour les rendre plus esthétiques, après avoir passé plusieurs heures sur Saadel celui-ci était pour une nouvelle fois fini, et cette fois-ci je l'ai directement importé sur Unity pour vérifier s'il possédait des problèmes. Cette fois-ci c'était la bonne, il apparaissait entièrement dans Unity. Ainsi, j'ai pu me consacrer aux textures des combattants, pour ne pas embrouiller les joueurs avec les couleurs des combattants et les couleurs d'équipe j'ai utilisé des couleurs assez sobres, pour Saadel j'ai utilisé un gris assez clair pour ses habits, pour les armes/boucliers j'ai utilisé un gris bien plus foncé, pour rappeler le côté métallique des armes, et pour ses cheveux j'ai utilisé un rose clair. Pour éviter tout travail à la dernière minute, je me suis empressée de l'exporter dans Unity pour voir le résultat final et vérifier s'il y a des problèmes, c'est à ce moment que j'ai vu que j'avais aussi un problème de texture, en effet, elles n'étaient justes pas présente sur Unity, après de longues recherches sur internet pour trouver la cause de mon problème, j'ai fini par trouver la solution, les textures que j'avais utilisées n'étaient pas compatible avec Unity. Une fois le problème réglé, j'ai pu importer ma modélisation 3D de Saadel dans Unity et tout fonctionnait parfaitement.



FIGURE 9 – Version finale de Saadel dans Unity

4.1.3 Création de Tyo

Après avoir fini ma première version de Saadel, je me suis attaqué à la modélisation de Tyo, du fait que Tyo soit un tank, cela rendait la modélisation plus technique, en effet le gabarit d'un tank est bien plus volumineux que celui d'un simple combattant, pour qu'il soit crédible je l'ai fait plus grand et imposant. J'ai aussi modélisé ses armes, qui sont un grand bouclier avec des pics et un grand marteau. Comme pour toutes mes modélisations je lui ai fait aussi une armature pour l'animer, comme expliqué plus tôt la modélisation de Tyo avait aussi un problème lors de l'exportation vers Unity. J'ai donc dû retravailler ma modélisation.

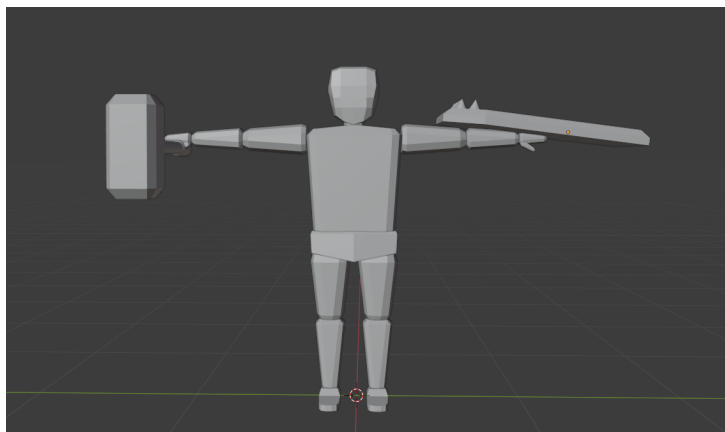


FIGURE 10 – Première version de Tyo

Après avoir réussi à faire une modélisation qui apparaissait entièrement

dans Unity, il ne me restait plus que les textures pour terminer Tyo. Comme pour Saadel j'ai utilisé un gris qui est un peu plus sombre que celui de Saadel, pour les armes j'ai utilisé le même gris, et pour terminer je lui ai fait des cheveux noirs. Et une fois sur Unity, tout apparaissait correctement

4.1.4 Création de Niaka

Ensuite, après Tyo, j'ai commencé à travailler sur Niaka qui était elle aussi un nouveau challenge, celui-ci était de devoir modéliser une femme, cela rendait la tâche un peu plus compliquée car il fallait qu'elle ressemble le plus possible à une femme, et donc il fallait lui donner des formes. J'ai donc fait de mon mieux pour qu'elle ressemble un minimum à une femme, vu que Niaka est le soutien, elle ne possède pas d'armes, j'en avais donc aucune à modéliser pour elle, comme le reste je lui mis une armature, et comme les autres j'ai aussi eu des soucis avec cette modélisation dans Unity avant de faire les textures.



FIGURE 11 – Première version de Niaka

Dès que le problème a été réglé, j'ai pu lui appliquer des textures pour changer des couleurs de Saadel et Tyo, et toujours pour éviter les confusions entre les couleurs des équipes et celles des combattants, j'ai pris un marron pour les vêtements de Niaka et je lui ai mis des cheveux blonds. Une fois sur Unity, tout fonctionnait parfaitement.



FIGURE 12 – Les versions finales de Tyo(droite) et Niaka(gauche)

4.1.5 Création de Lasnya

Il me restait plus que ce dernier combattant à modéliser, je m’y suis mise après avoir fini la première version de Niaka, comme j’avais déjà fait Niaka, je me sentais déjà un peu plus à l’aise avec l’idée de modéliser une femme. Après avoir modélisé tout son corps, j’ai dû m’occuper des armes, Lasnya est une combattante qui attaque à distance et pour cela elle possède un arc, j’ai donc modélisé son arc, ce qui fut assez compliqué, en effet, faire des arcs de cercle sur Blender ce n’est pas toujours très facile. J’ai aussi dû modéliser une flèche de l’arc pour animer l’attaque de Lasnya. Puis, pour finir je lui ai fait son armature. À la différence de mes autres modélisations, comme j’ai commencé à la modéliser après les problèmes d’importations sur blender, je n’ai pas eu besoin de retravailler Lasnya, j’ai pu lui mettre directement ses textures, pour ses vêtements je lui ai mis un violet foncé, des cheveux marrons, et son arc est lui aussi marron pour faire un arc en bois, et tout fonctionnait également très bien dans Unity.

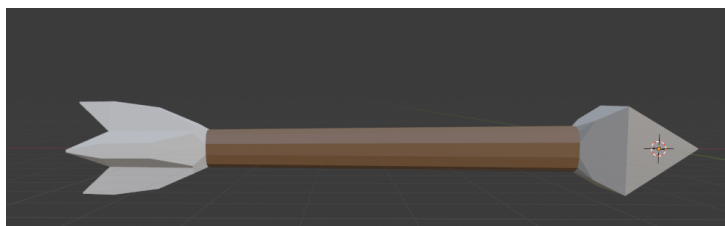


FIGURE 13 – Flèche de l’arc de Lasnya

4.1.6 Création du marchand

Au fur et à mesure de l’avancement du projet, nous avons décidé d’y rajouter un marchand dans le mode Solo, qui se trouvera à côté de l’autel à défendre, celui-ci permettra au joueur d’acheter des bonus, tel que de la vie

en plus et augmenter les dégâts de ses attaques durant la partie. J'ai donc dû modéliser ce fameux marchand. Comme les autres modélisations, je lui ai aussi mis une armature car il sera animé, pour qu'il ne soit pas complètement statique durant la partie, ensuite je me suis occupé des textures, vu que le marchand sera uniquement disponible en mode solo, je me suis permis de lui mettre un haut vert, ainsi qu'un pantalon et des cheveux noirs. Et voilà, c'était le dernier personnage que j'ai dû modéliser pour le projet.



FIGURE 14 – Les versions finales du marchand et de Lasnya

4.1.7 Création des autels

En parallèle des personnages que je devais modéliser, il me restait les Autels à défendre à modéliser, pour savoir comment les modéliser je me suis inspirée des autels que l'on peut retrouver dans certains autres jeux, avec ça j'ai pu faire la modélisation des autels. Les autels sont principalement composés d'une grosse pierre en lévitation, retenue par des chaînes, et d'un socle. Sur la première version, le socle était de base carrée, cependant lorsque nous l'avons importé sur la map du solo, on s'est dit qu'un socle arrondi rendrais mieux dans la map. Après avoir changé ce léger détail, je lui ai rajouté, comme les personnages, une armature, elle permet à l'autel d'être animé durant la partie. Les autels seront disponibles dans le mode de jeu solo, et dans le mode défense d'autel en multijoueur. Pour pouvoir différencier l'autel de l'équipe adverse à celui de son équipe, la pierre de l'autel ne sera pas la même. Nous avons donc fixé les couleurs suivantes, pour le mode Solo l'autel sera vert, et pour le multijoueur l'un sera bleu et l'autre rouge. Pour les textures, je les ai d'abord incorporées dans Blender, le socle est gris clair et les chaînes qui retiennent la pierre sont gris foncé, et la couleur de pierre est du coup soit verte, bleue ou rouge, cependant lors de l'importation vers Unity, la couleur de la pierre nous plaisait pas, elle semblait lisse et cela s'éloignait trop d'une texture de pierre. Nous avons donc décidé de faire directement les textures

sur Unity, ceci nous a permis de mettre plus de volume dans les textures et de rendre la pierre bien plus réaliste. Après l'application des textures sur Unity, nous avons réussi à avoir notre pierre qui brille avec ses chaînes grises et un socle avec une texture de pierre.

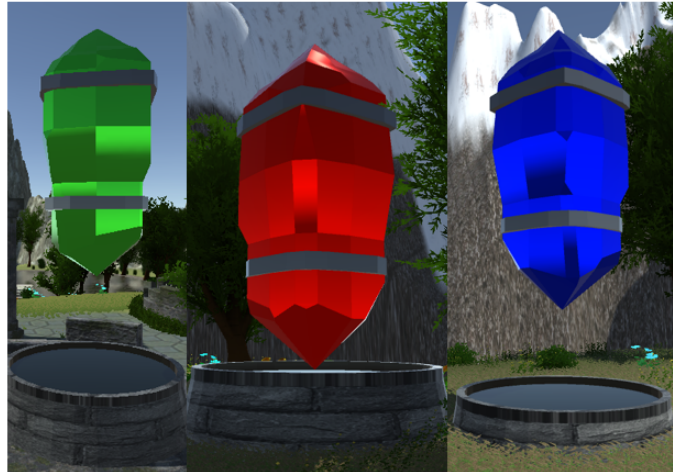


FIGURE 15 – Dernière version des trois autels

4.1.8 Réalisation des Animations

(Arthus)

Au début du projet, j'avais téléchargé quelques animations sur "<https://assetstore.unity.com>". Néanmoins, après avoir reçu les modèles 3Ds de Marine, j'ai réfléchi et j'ai découvert que ce n'était pas possible de les combiner. Les modèles en ligne dont provenaient les animations n'avaient pas le même nombre d'os. Ainsi nos personnages ne pouvaient pas s'articuler correctement. C'est à ce moment-là que j'ai commencé à faire les animations moi-même, en commençant par l'attaque lourde et l'attaque légère de Saadel, le combattant au trident.

Ensuite, comme Marine l'a précisé auparavant, nous avons dû refaire nos modèles 3Ds ainsi les animations que j'avais faite avaient disparu.

Avant cet accident, Marine m'avait fait les modèles 3Ds des autels, que j'ai pu par la suite les animer. Je leur ai donné une animation de vie, qui les

fait s'élever et tourner autour d'eux-même, dans le but de donner du dynamisme à notre jeu, et une animation de mort pour quand l'autel est détruit. Pour les attributions des animations, j'ai utilisé un "Animator Controler" et un booléen "IsDead" qui représente la mort.

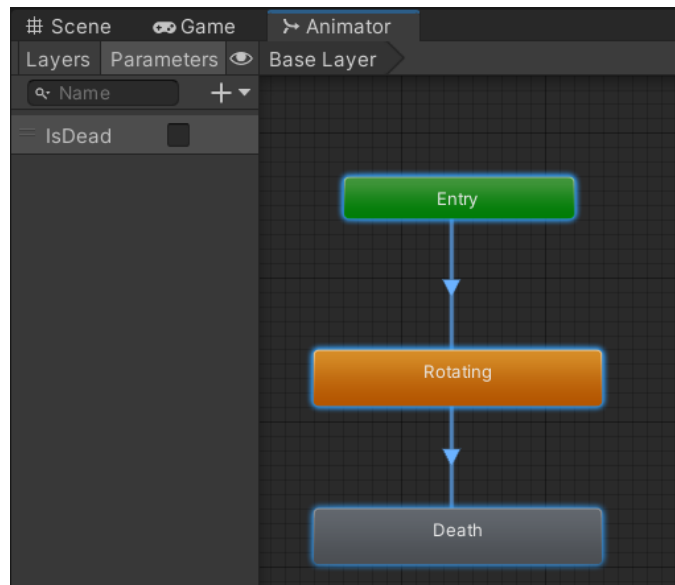


FIGURE 16 – Animator Controler des Autels

Cependant, j'ai consacré beaucoup de temps à faire les animations de tous les personnages sur le logiciel gratuit Blender. Ce fût long et périlleux car ce ne fût pas une chose facile, comme problèmes rencontrés nous pouvons citer le fait que je ne savais pas comment enregistrer correctement une position, le fait que les animations ne se sauvegarder parfois pas, etc... Comme pour Unity j'ai dû créer plusieurs tests et j'ai dû beaucoup me renseigner sur ce logiciel et les fonctionnalités qu'il offrait.

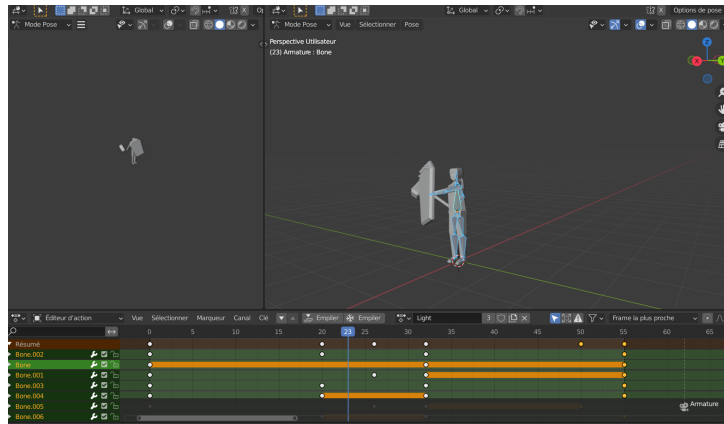


FIGURE 17 – Animation d'attaque lourde de Tyo sur Blender

J'ai commencé par faire l'unique animation du marchand, l'animation de vie, puis je l'ai incorporé sur la map du solo et je lui ai associé un "Animator Controler" qui fait tourner en boucle cette animation.

Tous les personnages jouables quant à eux possèdent 5 animations différentes, et ce pour chacun des personnages (il y en a 4 je rappelle) dont l'attaque Lourde (Heavy), l'attaque légère (Light), la marche avant (Forward), la marche arrière (Backward), et enfin une animation de respiration (Idle). Ce qui a fait au total 20 animations.

Lorsque les animations furent finies, j'ai pu les incorporer dans le projet Unity. Puis sur chacun des personnages, j'ai créé des "Animator Controler" qui est une sorte "d'arbre d'animations", ceux-ci seront plus complexes que les précédents car beaucoup plus d'animations et de conditions. Dans celui-ci, j'ai créé 4 booléens qui représentent chaque animation (Forward pour la marche avant, Backward pour la marche arrière, Heavy pour l'attaque lourde, et Light pour l'attaque légère). Chaque personnage possède un Animator Controler différent ce qui a pour effet que j'ai dû réitérer l'opération 4 fois en y insérant les animations respectives au personnage. Néanmoins, j'ai donné les mêmes noms de variables pour faciliter la tâche à Mathis quant à l'activation des animations par ses booléens.

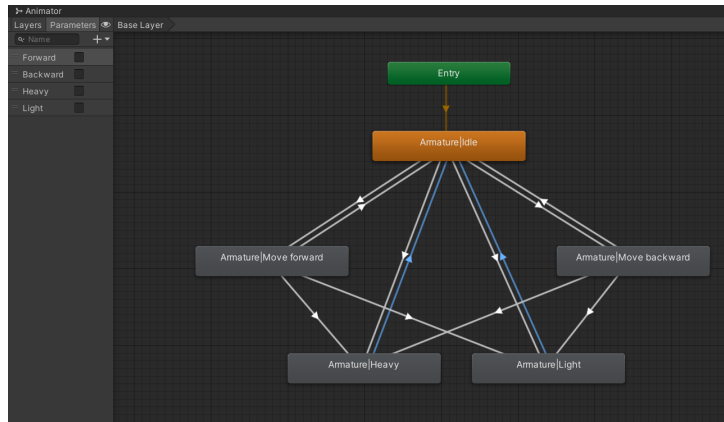


FIGURE 18 – Animator Controller de Unity

Un petit résumé des animations :

- **Idle** : Cette animation est l'animation de vie, le personnage bouge son corps mais pas les jambes, et ce de manière légère. Cette animation tourne en boucle quand nous ne touchons à aucune touche, ou alors lorsque deux touches contraires sont activées en même temps (comme par exemple avancer et reculer en même temps), autrement dit lorsqu'il est censé être immobile.
- **Forward** : Cette animation est l'animation de marche vers l'avant (contrôlée par le booléen 'Forward'), le personnage bouge l'intégralité de son corps de manière prononcé. Sur cette animation, nous avons fait pour que l'animation se coupe quand on appuie plus sur la touche "Z", lorsqu'on attaque ou alors lorsqu'on appuie sur "Z" et "S" en même temps. Pour faire ceci, il suffit de décocher la case 'Has Exit Time' (les liens gris).
- **Backward** : Cette animation est l'animation de marche vers l'arrière (contrôlée par le booléen 'Backward'), comme précédemment, le personnage bouge l'intégralité de son corps de manière prononcé. Sur cette animation aussi, nous avons fait pour que l'animation se coupe quand on appuie plus sur la touche "S", lorsqu'on attaque ou alors lorsqu'on appuie sur "Z" et "S" en même temps. Pour faire ceci, il suffit de décocher la case 'Has Exit Time' sur le lien entre Idle et Backward.
- **Heavy** : Cette animation est celle de l'attaque lourde, c'est une animation lente, pour une attaque lente mais puissante. Tout le haut du corps est sollicité et ce de manière très visible. Cette animation quant à

elle possède un "Exit Time" (les flèches bleues) c'est-à-dire que lorsque le booléen (situés sur la gauche de l'image ci-dessus) passe sur "True" l'animation se fait même si le booléen passe sur "False". Nous nous sommes servis de cela car cela permet de simplifier le code et de bloquer le personnage tant que l'animation n'est pas finie. Sans cela, tout aurait été plus complexe car chaque personnage à un temps d'attaque lourde différent.

- **Light** : Cette animation est celle de l'attaque légère, c'est une animation rapide, pour une attaque rapide mais peu puissante. Tout le haut du corps est sollicité et ce de manière très visible. Cette attaque, comme l'attaque lourde, possède un "Exit Time" pour les mêmes raisons.

4.2 Maps

(Arthus)

4.2.1 Création des maps en général

La création des maps fût la partie la plus conséquente de mon travail. Faire des maps bâclées m'était totalement inconcevable. Pour le design de celles-ci, j'ai utilisé un logiciel que je n'avais jamais utilisé avant la création du groupe : Unity. Pour comprendre son fonctionnement je me suis beaucoup renseigné sur divers sites, dont YouTube, et j'ai créé plusieurs maps de tests pour chacune des maps avant d'arriver à ce qu'elles sont aujourd'hui.

Les difficultés de prises en mains étaient nombreuses :

- Le déplacement de la caméra d'édition, qui est très spéciale et pour laquelle il m'a fallu beaucoup de temps d'adaptation.
- L'extrusion du terrain pour créer des hauteurs dont les pentes peuvent être surmontables, car oui pendant très longtemps je ne savais pas qu'il y avait l'outil "Smooth Height", dont les spécifications seront spécifiées un peu plus tard.
- Les assets nécessitent aussi pas mal de documentations pour le seul fait de les ajouter.

4.2.2 Map solo

Pour cette map, qui fût la première créé, j'ai tout d'abord créer un terrain de 200*200 sur lequel j'y ai fait une colline centrale et des montagnes tout autour pour délimiter la carte de jeu. Par la suite je me suis occupé des couleurs et des éléments naturels comme la végétation et un étang. Pour la végétation j'ai utilisé deux outils appelés "Paint Trees" (pour les arbres, qui leur attribut des collider, ce qui permet de ne pas pouvoir les traverser) et "Paint details" (pour les herbes, les fleurs, les cailloux, etc ...) Celui-ci ne met pas de collider contrairement au précédent) qui permet de placer des objets de manière plus naturel sur la map. Il permet aussi un important gain de temps car il évite de placer chaque objet uns à uns, il place dans une zone cible une certaine proportion d'objets. Enfin un gain sur la taille et les performances du jeu lors du chargement de la map. Enfin j'y ai incorporé des infrastructures dont une auberge et une petite maison.

Par la suite, j'ai mis des murs invisibles tout autour de la map pour éviter que les joueurs ne tombent dans le vide. Des brouillards sont présents sur les 4 entrées de la map, ce qui permet à Mathis d'y mettre les spawners (lieux où apparaissent les IA) de mobs (méchants). Cela a pour but que le joueur ne voit pas les IA apparaître magiquement mais qu'il y ait un peu de cohérence et de réalisme, mais aussi que le joueur ne voit pas le vide derrière la map.

Cette map est constituée d'assets récupérés principalement sur le site "<https://assetstore.unity.com>". Ces assets sont, par exemple, des textures pour les montagnes, des types d'herbes, des fleurs, des préfabriqués (comme des bâtiments ou des objets à placer). Au début, j'ai utilisé un "autel" récupéré sur ce site pour tester la carte, et d'avoir un premier aperçu. Cependant, plus tard, lorsque les modèles 3Ds et les animations des autels ont été finies, on a tous remplacé par nos propres objets.

Enfin sur cette carte j'ai bien vérifié que tous les objets aient un collider (dont les maisons par exemple) et donc qu'on ne puisse pas passer au travers, dans le cas contraire j'en ai créé moi-même. Par la suite, j'ai ajouté un marchand et son décor proche de l'autel une fois que sont modèle 3Ds et ses animations ont été elles aussi finies.



FIGURE 19 – Marchand du solo



FIGURE 20 – Magasin du Marchand

Quelques images de cette carte :

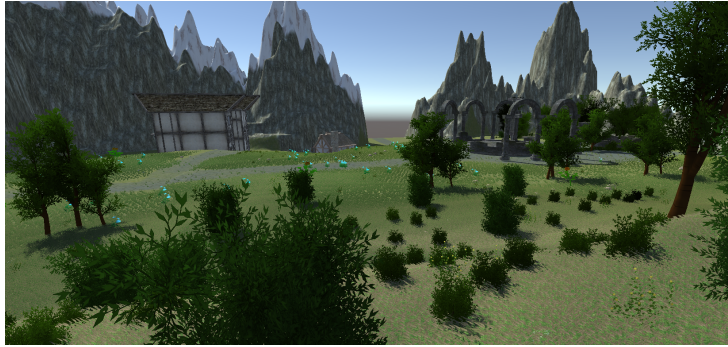


FIGURE 21 – Capture d'écran de la map solo



FIGURE 22 – 2ème Capture d'écran de la map solo

4.2.3 Map 3vs3



FIGURE 23 – Capture d'écran de la map 3vs3

Cette map contrairement à l'autre est une map plus en long pour que les joueurs doivent faire du chemin avant d'attaquer l'autel adverse. Sinon le jeu serait trop facile et rapide!

Pour ce qui est du design de la map, j'ai tout d'abord fait une map relativement plate et en l'entourant de quelques murs, mais cela n'était pas très esthétique, ainsi je l'ai totalement refaite en élevant toute la map et en créant sur les bords de map une crevasse (inaccessible car protégée par des murs invisibles) et sur les bords externes de la carte, des montagnes sont présentes. Ce contraste est très intéressant et donne de la profondeur à la carte. Néanmoins quand cette idée fût fixée, le même procédé que la map solo a été appliqué, autrement dit lorsque la forme de la carte me plaisait, j'y ai mis des textures, puis les éléments naturels, et enfin nos infrastructures.

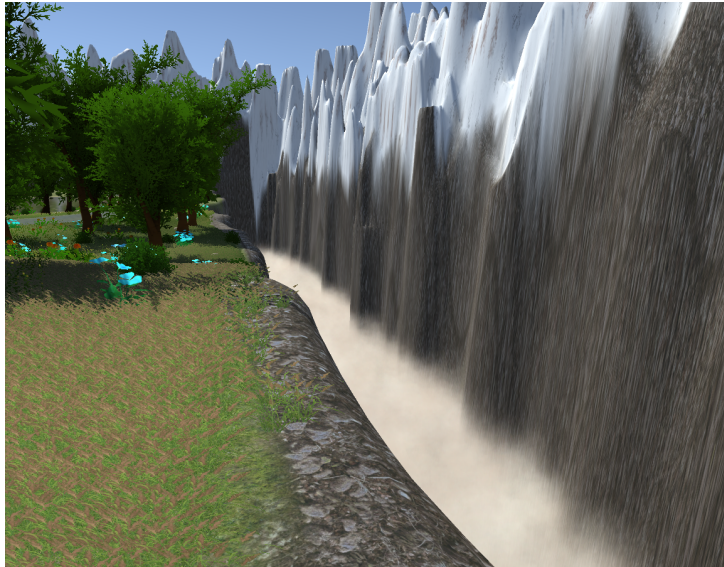


FIGURE 24 – Crevasse de la map 3vs3

Au lancement de la partie, un brouillard sera présent sur le centre de la map et des murs invisibles nous empêcheront de sortir de nos camps. Après un certain délai, via un script de mon camarade Mathis, le brouillard s'arrêtera progressivement et les murs disparaîtront pour laisser la place au jeu!

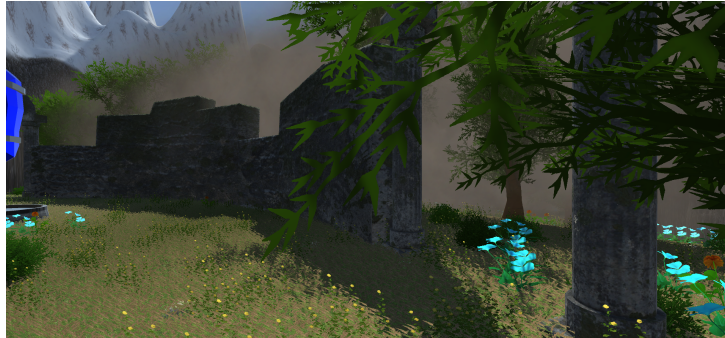


FIGURE 25 – Brouillard de début de la map 3vs3

4.2.4 L'outil Smooth Height

Nous avons quelques soucis car cette map était beaucoup trop lourde, je me suis ainsi penché sur la question de comment régler ce problème, c'est ainsi que j'ai découvert un outil qui s'appelle "Smooth Height".

Ce dernier permet d'arrondir les courbes (qui sont à l'origine des triangles), ce qui permet d'alléger le fichier et de réduire le temps de chargement de la carte de jeu.

Cet outil a aussi un autre avantage : celui de rendre les montés plus belles. Au début pour les faire, je devais le faire manuellement en changeant les hauteurs, sauf que comme dit précédemment, et cela ne rendait pas bien du tout. (Cf. Images ci-dessous)



FIGURE 26 – Pente faites à la main



FIGURE 27 – Pente faites avec l'outils Smooth Height

4.2.5 Allègement de la map

L'outil Smooth Height nous a permis de réduire la taille du jeu mais cela resté tout de même trop lourd. Comme les deux maps sont présentes sur le

même projet, juste dans des scènes différentes, les assets ont besoin d'être présents qu'une seule fois même s'ils sont utilisés dans les différentes maps, ce qui nous permet d'économiser plus de place.

Néanmoins ce n'est pas tout, car même à ce point-là, j'ai décidé "d'épurer" les assets utilisés pour qu'ils prennent encore moins de place. De manière plus précise, j'ai retiré chaque fichier non utilisé car cela prenait beaucoup de place pour rien.

Cependant, ce n'est pas une tâche facile et rapide ! Cela m'a pris plusieurs heures, car j'ai dû m'assurer que rien de ce qui est utilisé n'a besoin de ce qui va être supprimé, au risque d'avoir des erreurs et de rendre le fichier, et donc le jeu, corrompu.

Pour résumer cette partie de création des cartes de jeu, je peux dire que j'ai pris beaucoup de plaisir à designer les maps, même si cela me poussait parfois à bout.

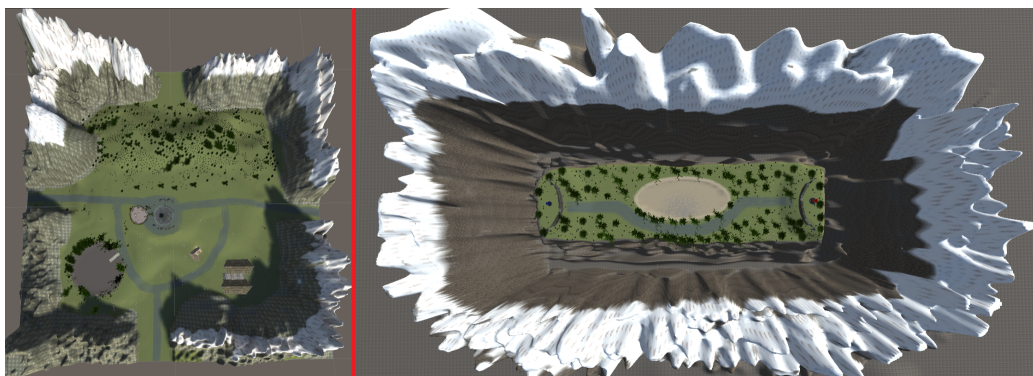


FIGURE 28 – Les maps en résumé

4.3 Menus et interfaces

(Mathis)

Les menus du jeu m'ont pris beaucoup de temps, en effet, il a fallu que j'apprenne à maîtriser les liaisons entre les scripts et les éléments "Unity UI". Les menus ont constamment évolué, à savoir qu'à chaque soutenance de nouveaux menus ont fait leurs apparitions.

4.3.1 Menus au démarrage

Pour les menus, nous avons utilisé les éléments d'interface utilisateur proposés par Unity. Nous avons créé une première scène nommée "MainMenu", qui permet au joueur de choisir entre le mode "Solo" et le mode "Multijoueur", mais également de pouvoir quitter, ou modifier les paramètres via un sous-menu "Paramètres". Nous avons choisi une police originale, restant toujours dans l'univers et le style de notre jeu. De plus, concernant le choix des couleurs, nous avons opté pour des couleurs assez sombres pour coller également à l'ambiance et l'univers de Bagarena.



FIGURE 29 – Menu Principal

Pour la partie Options, j'ai ajouté la possibilité de diminuer le volume du jeu avec un Slider, de changer la résolution (dans le Dropdown il y a toutes les résolutions prises en charges par l'ordinateur de l'utilisateur), et enfin la possibilité d'être en plein écran ou non, avec un Toggle à cocher ou décocher. Voir la capture d'écran ci-dessous.

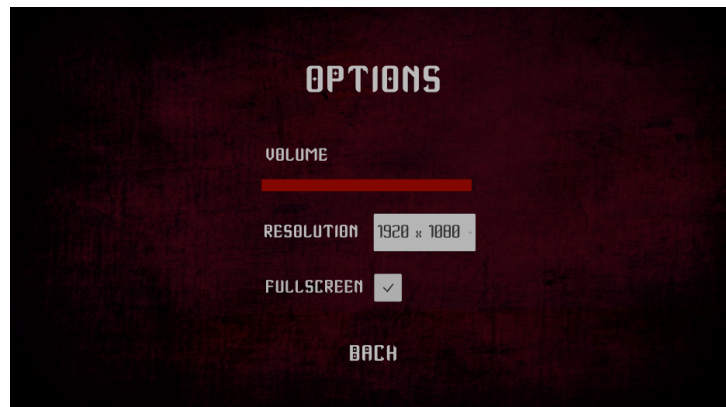


FIGURE 30 – Menu des options

Lorsque l'utilisateur sélectionne le mode "Solo", il va être redirigé vers la scène "Solo" où il va pouvoir choisir son combattant.



FIGURE 31 – Menu de choix de personnage en Solo

S'il sélectionne le mode "Multiplayer", il va être redirigé vers la scène "Multiplayer", celle-ci contient les menus propres à la partie multijoueur. Cette partie est celle qui m'a pris le plus de temps lors de l'élaboration des menus. La première difficulté étant qu'il fallait comprendre le fonctionnement de Photon, le framework que nous avons choisi d'utiliser pour la partie en réseau. J'ai donc regardé énormément de vidéos et de forum pour comprendre comment s'occuper de la gestion des rooms (salons de jeu), allant de la création de celles-ci, la gestion d'entrée de joueurs, de changement de masterclient (le propriétaire de la room). Dès lors que j'avais compris le fonctionnement j'ai pu commencer à créer le menu permettant de choisir son surnom sur le réseau, et les menus qui permettent de lister les rooms existantes sur le serveur, ou

d'en créer une. Ces menus étaient présents pour la première soutenance. Pour en revenir à l'explication du joueur qui vient de sélectionner le mode multijoueur, il va donc pouvoir choisir un nom d'utilisateur sur le réseau, de rejoindre ou créer une room. Par ailleurs, si il souhaite créer une room, il va choisir entre les deux modes de jeu disponibles (à savoir "Altar Defense", qui n'est autre que "Défense d'autel", et "Free For All" qui est le mode dit "chacun pour soit").



FIGURE 32 – Menu choix de pseudo - Menu sélection (création de salon, ou rejoindre un salon)

Ensuite de ça, il m'a fallu pour la deuxième soutenance m'occuper de faire les menus de choix de personnages des différents modes multijoueurs, le menu de choix d'équipe du mode "Altar Defense", ainsi que le lancement de ces menus lorsque le masterclient décide de lancer la partie avec les joueurs présents dans la room. Première complication dans la conception de ces menus, envoyer chaque joueur sur le menu de choix d'équipe lorsque le propriétaire de la room actuelle décide de lancer la partie. J'ai beaucoup cherché, et finalement, les PunRPC (Remote Procedure Calls) m'ont permis de lancer pour tous les joueurs le menu, les méthodes RPC permettent en effet de définir la target sur Target.ALL pour l'appeler pour chaque joueur, cependant, il a fallu que j'ajoute un PhotonView component à mon GameObject englobant le menu. Est venue la deuxième complication, l'implémentation d'un menu de choix d'équipe pour le mode multijoueurs "Altar Defense" de notre jeu. En effet, il a fallu que je passe également beaucoup de temps à rechercher comment gérer la synchronisation de chaque changement d'équipe de joueur. Après de nombreuses recherches et de nombreuses tentatives ratées, je me suis rabattu sur l'utilisation des CustomProperties de Photon, pour chaque joueur, nous pouvons leur définir une Hashtable contenant les informations que l'on souhaite stocker pour chaque joueur. Et donc pour pouvoir faire la modification sur l'écran de chaque joueur, j'ai surchargé la méthode OnPhotonPlayerPropertiesChanged(), pour que lorsqu'un joueur modifie sa customProperties, l'appel de la méthode OnPhotonPlayerPropertiesChanged() se fasse auto-

matiquement et donc que la méthode surchargée fasse l'action souhaitée. Ci-dessous un exemple d'une room avec deux joueurs ayant choisi un pseudo aléatoire et ayant choisi d'être dans une équipe différente.



FIGURE 33 – Menu de choix de choix d'équipe

Lorsque les joueurs ont choisi leur équipe, le propriétaire de la partie peut donc cliquer sur "Ready" (si une des deux équipes a plus de 3 joueurs, un message d'erreur apparaîtra), grâce à une méthode PunRPC les joueurs vont tous sur un menu de choix de personnage. Est venue alors une autre difficulté, à savoir que nous avons décidé que lorsqu'un joueur d'une équipe choisi un combattant, les autres joueurs de son équipe ne pourront plus le choisir, nous avons également décidé que lorsqu'un joueur verrouille un choix, il ne pourra plus revenir en arrière et faire un autre choix. Nous nous sommes inspirés de League Of Legends pour cette règle. J'ai donc du faire en sorte de bloquer les combattants déjà choisis par des joueurs de l'équipe, pour les autres joueurs de cette même équipe, j'ai eu pas mal de difficultés et de problèmes liés à cette fonctionnalité. De plus, il a fallu que je fasse en sorte que tous les joueurs voient le personnage que vient de choisir le joueur.

Voici donc une capture d'écran depuis l'écran d'un joueur, de la même équipe qu'un joueur venant de choisir son combattant, avec un joueur de l'équipe adverse ayant choisi un combattant. (Les joueurs n'ayant pas choisi ont une croix à la place de l'image, et lorsqu'un joueur de la même équipe a choisi un combattant, celui-ci n'est plus cliquable et son image est remplacée par une croix).



FIGURE 34 – Menu choix de combattant pour le mode "Altar Defense"

Nous avons également un deuxième mode multijoueur, "FFA", celui-ci n'a pas d'équipe et n'a pas de règle spécifique concernant le choix de combattant hormis le verrouillage définitif d'un choix, son menu de choix de combattant a donc été moins laborieux à créer puisque je connaissais la gestion des difficultés de synchronisation des choix.

Ci-dessous, une capture d'écran de ce menu.



FIGURE 35 – Menu choix de combattant pour le mode "FFA"

4.3.2 Menus durant une partie

Concernant les menus durant une partie, à savoir les menus "pause" et de fin de partie. Voici des captures d'écran des différents menus.



FIGURE 36 – Menu pause du mode Solo (gauche) - Écran de fin de partie en mode Solo (droite)



FIGURE 37 – Menu Pause des modes FFA et Defense d'Autel (vue depuis le mode Defense d'Autel, ils sont identiques)

Il y a également les écrans de fin de parties qui ont été fait. Ceux-ci permettent simplement d'afficher soit l'équipe gagnante et le nom des joueurs de cette équipe pour le mode Défense d'Autel, soit d'afficher le grand vainqueur de la partie en mode FFA.

Le dernier menu en partie est celui du marchand du mode solo, au début du projet nous étions indécis sur la possibilité de le faire, cependant après la deuxième soutenance je me suis rendu compte que j'ai bien progressé et que les menus ne sont plus un soucis pour moi, j'ai pu donc ajouter un menu de boutique, accessible en appuyant sur la touche "E" lorsque notre personnage est assez proche du marchand, cette boutique permet d'augmenter ses dégâts d'attaques (légère et lourde), et d'augmenter sa vie maximale. Permettant ainsi de pouvoir continuer de résister aux vagues ennemies. Pour augmenter ses pouvoirs la seule condition requise est d'avoir assez de pièce de monnaie,

à chaque vague le joueur gagne une pièce de monnaie. Il a fallu également, en parallèle de ce menu, que je code tout le système de gain de monnaie, et d'augmentation de point de vie ou d'attaque, cependant avec l'expérience que j'ai acquise j'ai pris beaucoup de plaisir à le faire fonctionner ce système.



FIGURE 38 – Menu de boutique du marchand

4.3.3 Interface pendant la partie

Pour les interfaces en partie, elles sont quasiment toutes identiques, ci-dessous un exemple de l'interface en mode solo :



FIGURE 39 – Interface du mode solo

Au dessus de la barre de vie en bas, il y a 2 images représentant les 2 attributs du personnage, à gauche l'attaque légère, et à droite l'attaque lourde, lorsqu'une attaque vient d'être utilisée l'image se "recharge" de manière circulaire, permettant de savoir quand l'attaque va pouvoir être réutilisée (c'est à dire lorsque l'image redevient pleine). Pour les autres interfaces, il s'agit de la même barre d'attribut niveau design, et il y a juste des différences en fonction du mode au niveau des autres informations. Pour le mode Solo comme vous pouvez le voir sur la capture d'écran, on voit en temps réel la vie de l'autel, le temps avant la prochaine vague, le numéro de la vague actuelle, ainsi que le temps écoulé.

Pour le mode FFA, il y a uniquement la barre d'attribut et de point de vie du personnage, et le nombre de parties gagnées affiché en haut à droite.

Pour le mode Défense d'Autel, il y a toujours cette barre d'attribut et de point de vie, en plus de cela il y a un affichage du temps de la partie en cours, et il y a les infos de points de vie de l'autel adverse, ainsi que ceux de l'autel de l'équipe.

Ces interfaces sont rattachées au prefab de chaque personnage, et ont donc des scripts qui permettent d'actualiser les infos sur la vie du personnage, de l'autel, les attributs utilisés, etc.. Comme ils sont attachés directement au prefab du personnage, accéder aux informations du personnage et de la partie n'a pas été très compliqué à faire. Pour les attributs des personnages, j'ai été sélectionner des images qui ont plu au groupe.

Ci-dessous toutes les images choisies pour représenter les attributs dans l'interface. L'image la plus haute de chaque colonne représente l'attribut de l'attaque légère, celle en dessous la plus lourde, et en dessous le nom du personnage ayant ces attributs.

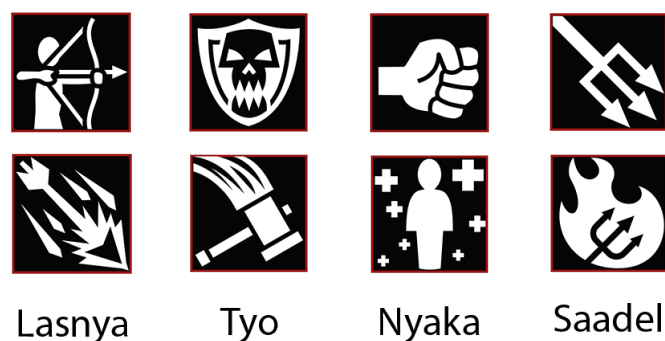


FIGURE 40 – Logos des attributs

Pour conclure sur les interfaces, j'ai bien aimé faire les interfaces, celles-ci étaient bien moins compliquées à faire fonctionner que les menus, ça a donc moins été un casse tête, une fois que j'avais fait une interface pour un personnage j'ai pu refaire pour les autres sans trop de difficultés. Cependant les interfaces sont arrivées assez tard dans le jeu car j'ai voulu avancer d'abord sur les menus. Les interfaces du mode Solo étaient les premières faites, présentes à la deuxième soutenance, tandis que les interfaces des modes multijoueurs ont entièrement été faites entre la deuxième soutenance et le rendu final.

5 Intelligence Artificielle

(Mathis)

Pour les intelligences artificielles, il fallait en faire deux "types" :

- Une IA pour les ennemis des vagues du mode Solo (squelette et orc, avec squelette les ennemis les moins forts, et les orcs les plus forts).
- Une IA pour les bots en mode multijoueur (qui remplacent les joueurs manquants dans la room pour faire un vrai 3 VS 3).

Avant de décrire le développement de chacune des IA, je précise que j'ai d'abord commencé par coder les IA du mode solo, qui étaient terminés pour la deuxième soutenance, et je me suis occupé dans un second temps des IA du mode multijoueur pour cette dernière soutenance.

5.1 IA en mode solo

Dans l'optique de débiter les tests concernant les mécaniques de jeu, j'avais commencé à implémenter une IA minimaliste, elle se contentait de vérifier sa distance avec le joueur placé en paramètre ainsi que celle avec l'Autel et de les comparer pour choisir le chemin qu'elle doit suivre. Pour se diriger dans la scène et prendre des chemins, elle se base sur un NavMesh (Cette classe nous permet de définir le coût de recherche de chemin pour des types de zone spécifiques et d'ajuster le comportement global de la recherche de chemin et de l'évitement). Pour ce début de développement de l'IA j'avais rencontré des difficultés au niveau de la gestion de l'arrêt des IA lorsqu'elles sont à une certaine distance pour pouvoir attaquer, je n'ai d'ailleurs jamais réussi à régler ce problème, mais en cherchant sur de nombreux forums il se trouvait que ce problème était récurrent et lié également à un problème de manque de précision dans les destinations données. Le problème étant que lorsque l'on arrête un NavMeshAgent, il veut continuer à se diriger à travers le chemin qu'on lui a donné, de ce fait il y a un effet de tremblement à l'écran. De plus, lorsqu'une autre IA s'approchait de lui la collision les faisaient se repousser l'un l'autre. Ces IA étaient développés sur un modèle 3D de capsule (modèle disponible sur unity).

Pour la deuxième soutenance, ayant globalement le système pour se diriger à l'endroit le plus avantageux en terme de distance, j'ai eu juste à optimiser un peu plus ce système, mais j'ai pu me concentrer sur l'implémentation du système d'attaque de ces bots. Étant donné que j'avais déjà codé le système d'attaque des personnages que le joueur peut contrôler, j'ai pu m'inspirer de ce modèle tout en l'adaptant aux scripts différents de l'IA. La gestion des cooldowns (temps de "gèle" entre 2 utilisations d'une attaque) a été moins laborieuse puisque les bots ont uniquement une seule attaque. J'en ai profité aussi pour changer le prefab des bots en lui attribuant le modèle 3D que nous avons sélectionné (le squelette), afin de remplacer la capsule qui était utilisée jusqu'à présent. A la deuxième soutenance j'avais donc les bots squelette fait avec l'IA complète.

Et donc mon objectif à la suite de cette soutenance a été de créer le prefab de l'orc, notre deuxième ennemi possible, j'ai donc juste utilisé le modèle 3D de l'orc, je lui ai mis les différents scripts de l'IA, et mis juste des dégâts d'attaque plus élevé que celui du squelette. Il me restait désormais plus qu'à ajouter les animations, et intégrer le code nécessaire à l'exécution de ces animations. Les IA du mode solo sont donc désormais entièrement finies, et je suis satisfait du résultat. J'ai un seul regret c'est de ne pas avoir réussi à

résoudre le problème des légers tremblements.

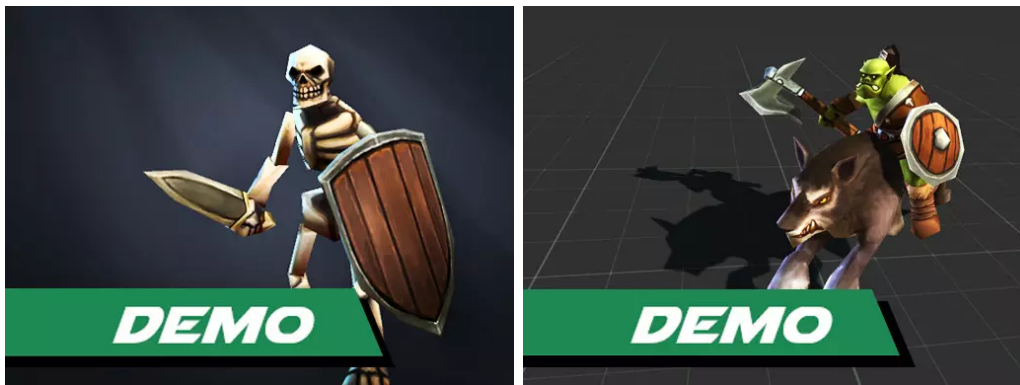


FIGURE 41 – Les deux modèles 3D utilisés pour les bots

5.2 IA en mode multijoueur

Les IA du mode multijoueur sont présentes uniquement dans le mode Défense d'Autel (3 VS 3), elles ont pour but de remplacer les joueurs manquants pour avoir à chaque fois 3 joueurs dans chaque équipe. Nous avons fait le choix de ne pas en mettre dans le mode FFA pour que les joueurs ayant envie uniquement de faire un 1 contre 1 ne se retrouve pas avec des bots dans leur partie. Cette partie sur les bots multijoueurs fait écho avec la partie multijoueur décrite plus bas. J'avais d'abord codé les programmes des deux modes multijoueurs pour qu'ils fonctionnent avec un joueur humain avant de m'attarder sur l'intégration de bots en ligne.

Pour cette IA j'ai dû m'inspirer de ce que j'avais pu faire avec l'IA du mode Solo, néanmoins plusieurs nouvelles difficultés sont apparues, la première étant qu'il fallait rajouter une deuxième attaque à l'IA, et que celle-ci est différente en fonction du personnage que l'IA incarnera (dans ce mode l'IA incarne soit Saadel, soit Lasnya, soit Tyo, ou soit Niaka), la deuxième difficulté, et la plus conséquente, est de récupérer tous les joueurs et bots présent sur la carte. Le problème étant que la récupération de ces joueurs doit se faire en cherchant tous les GameObject ayant le tag Manager, puis il faut regarder si ce manager appartient à un joueur ou à un bot, ensuite comparer une à une les positions de l'avatar de chaque manager, pour enfin pouvoir calculer si il doit aller vers le joueur le plus proche ou l'autel ennemi. Ce programme est assez lourd et causait des temps où le bot n'avancait plus le temps de trouver son chemin, j'ai trouvé un moyen de l'alléger ensuite, cependant un problème persiste et je n'ai trouvé aucun moyen de le régler. Il s'agit du fait que les calculs de positions ne sont pas parfaits, et causent

certain ralentissement ou bugs parfois. Ces ralentissements sont liés au fait que les bots sont localisés et appartiennent au MasterClient, or pour recevoir les informations de position de chaque joueur il y a une latence lié au réseau et au léger temps d'échange des informations entre les clients.

Cette partie m'a pris beaucoup de temps et a été assez difficile, certes j'ai gagné du temps en sachant comment faire avancer et suivre une cible à une IA, mais j'ai du me confronter à tous les problèmes d'une IA fonctionnant sur un réseau et comment les ajouter à ce réseau, cependant cette partie a été assez intéressante et venait en complément de toute la partie réseau.

6 Mécaniques de jeu

(Mathis)

Je me suis occupé de l'entièreté des mécaniques de jeu et j'ai fortement apprécié m'occuper de toutes ces parties au fur et à mesure. Ça m'a pris du temps mais réparti sur l'ensemble de la durée du projet, j'avancais petit à petit en parallèle des développements des modes de jeux, menus, etc..

Celle qui a subi pas mal de modifications est le système de vagues d'apparition d'ennemis en mode Solo. Au début, et ce jusqu'à la deuxième soutenance, celle-ci se contentait uniquement de faire apparaître des ennemis en en faisant apparaître un de plus que la vague précédente toutes les 20 secondes. A chaque ennemi qui apparaissait, le suivant apparaissait sur le point d'apparition suivant. Ce système a du changé par la suite puisqu'il fallait intégrer un deuxième type d'ennemis (les orcs), et nous nous sommes aussi rendu compte que de ne pas attendre que la vague se termine rendait le jeu très difficile. J'ai donc entièrement changé le système, en ce qui concerne les différents ennemis qui peuvent apparaître, je me suis basé sur une probabilité de 70% de chance d'apparition de squelette, et un taux d'apparition de 30% pour les orcs. J'ai également changé le système de gestion des points d'apparitions de chaque ennemi, désormais chaque ennemi apparaît sur un point d'apparition aléatoire (il y en a 3 possibles). Et enfin, pour qu'une autre vague se déclenche il faut que le joueur ait éliminé tous les ennemis de la vague précédente.

Concernant les déplacements du joueur, j'ai utilisé le component "Character Controller" de Unity, car je n'avais pas besoin de gérer un Rigidbody et que celui-ci me permettait de gérer plus facilement les déplacements des joueurs, et de savoir grâce à l'attribut "isGrounded" si le personnage est bien

au sol avant de faire un saut. Les déplacements ne sont pas latéraux, le personnage fera des rotations si les touches de déplacements latéraux sont utilisées.

Les attaques, les attributs, la gestion de la vie, et la gestion des autels ont été continuellement améliorés, tout a commencé par le mode Solo, puis j'ai pu m'en inspirer dans les modes multijoueurs, même si la contrainte de l'utilisation du réseau a changé ma façon de les faire.

La caméra qui permet de suivre le joueur a subi des modifications constantes, voire même a été souvent complètement refaite. Au tout début nous avons utilisé un asset, puis j'ai voulu la faire moi même, cependant ce fut assez compliqué, au début la caméra suivait trop le joueur et lorsque le joueur attaquait un ennemi et qu'il devait s'orienter face à cet ennemi la caméra "sautait", rendant le suivi très désagréable à regarder. J'ai donc essayé plusieurs techniques de suivi, et je suis arrivé à un script globalement convenable, celui-ci fait une transition beaucoup plus fluide lorsque le personnage effectue une rotation, ce qui s'avère pratique lorsque le personnage se tourne face à son ennemi rapidement. Vous remarquerez que la hauteur de la caméra est différente dans le mode "Défense d'Autel" puisque je me suis rendu compte qu'avec la forte présence d'arbres sur la map cela rendait difficile de pouvoir voir son personnage ou les autres joueurs. La caméra est donc plus haute et moins à l'arrière du personnage dans ce mode. De plus, j'ai ajouté des scripts sur les canvas au dessus des personnages des autres joueurs et bots ,permettant d'afficher la vie dans un slider ainsi que le pseudo du joueur sur le réseau, en utilisant la méthode LateUpdate() afin d'orienter ces canvas toujours face à la caméra.

Et enfin, dernière mécanique : les animations. Arthus s'est occupé de les créer, et a créé les animators pour chaque animation. Cependant pour les intégrer au projet nous avons eu des difficultés, en effet pour activer les animations d'attaques il fallait activer le boolean correspondant et le désactiver rapidement pour qu'il puisse revenir à l'animation qu'il faut à la suite de cette animation, sauf que lorsqu'on met le boolean à true, et dans la ligne suivante à false, l'enchaînement est beaucoup trop rapide et le changement d'animation n'était pas pris en compte. L'autre problème étant qu'il fallait que les points de vie de l'ennemi prennent ses dégâts à peu près au moment où l'arme d'attaque se retrouvait au niveau du corps de l'ennemi pendant l'animation. Avec Arthus nous avons donc cherché des solutions pour les utiliser dans les scripts et éviter ces problèmes. Après quelques heures à essayer

des choses, nous avons réussi à pallier ce problème en utilisant des coroutines pour attaquer, et donc pouvoir mettre le boolean à false au bout d'un temps défini, et que lorsque ce temps est écoulé dans la coroutine le reste des lignes servant à réaliser le retrait des points de vie peut s'exécuter, et correspondre à peu près au moment où l'arme touche l'ennemi.

7 Multijoueur

(Mathis)

Pour rappel, nous avons choisi d'utiliser l'asset "Photon Unity Network"(PUN) pour gérer la partie multijoueur du jeu qui permet avec la version gratuite d'avoir jusqu'à 20 joueurs en même temps sur un serveur. Cette partie a été très difficile et m'a pris énormément de temps, en effet, il a fallu que je me documente énormément sur l'utilisation de Photon, qui gère quoi, qui a quoi sur sa scène en local, comment synchroniser les mouvements, les actions, quels objets sont indépendants du PC du joueur. Plein de notions à apprendre et comprendre avant de se lancer dans la conception.

J'ai d'abord commencé par la conception des menus multijoueurs, où vous avez l'explication plus haut, les joueurs se connectent en premier lieu au serveur photon, puis, ils doivent rejoindre le Hall (Lobby) qui permet ensuite d'accéder à plusieurs rooms.

Pour rappeler le fonctionnement du menu et du réseau lié à celui-ci, un script Launcher.cs s'occupe de gérer la connexion du joueur au serveur, de créer ou rejoindre une salle et de charger la scène correspondant au bon mode de jeu (3vs3 ou FFA), celui-ci est rattaché à un GameObject "Canvas" qui contient différent GameObject correspondant aux différents sous-menu. La classe Launcher hérite de MonoBehaviourPunCallbacks qui permet d'implémenter des fonctions qui seront appelés spécifiquement par Photon lors de certains évènements. Par exemple la fonction OnConnectedToMaster() est appelée par Photon lorsque le jeu est connectée au serveur et permet ainsi d'afficher les options permettant de rejoindre ou créer une salle à l'écran. La classe Launcher implémente des fonctions similaires comme OnJoinedRoom() ou OnJoinedLobby().

Hall (Lobby)

Dans le Hall il est possible de :

- Créer une room
- Joindre une room en ayant accès à la liste des rooms existantes
- Connaitre le nombre de joueurs présents dans les différentes rooms

Même s'il est possible de joindre manuellement le Hall, Photon Unity Network propose dans le "PhotonServerSettings" de pouvoir le joindre automatiquement, nous nous en sommes donc servis pour le projet.

Pour pouvoir se balader dans les différents menus multijoueur, il a fallu ajouter dans le script des fonctions permettant d'effectuer les différentes actions de connexion et d'interactions avec le serveur principal, nous avons donc ajouté des scripts permettant de lister les rooms, et de gérer les cas éventuels de room vide qui restaient présentes dans la liste, ainsi que des fonctions qui permettent de joindre une room spécifique d'après la liste des rooms existantes, ou d'en créer une si la room n'existe pas.

Room

Dans une room il est possible de

- Lister les joueurs de la room
- Quitter la room
- Lancer la partie (uniquement pour le créateur de la salle)

Pour pouvoir joindre et ajouter quelques fonctionnalités au sein d'une room, les différents scripts ont été modifiés. A la scène "Multiplayer" a été ajouté des inputfield pour pouvoir renseigner le pseudo du joueur, avec la possibilité éventuelle de laisser un pseudo aléatoire. De plus, nous avons ajouté la fonctionnalité de lister les joueurs de la room dans laquelle le joueur est et également la présence de "[Owner]" à côté du pseudo du joueur propriétaire de la room. Et enfin, lorsque le propriétaire décide de lancer la partie, tous les joueurs seront redirigés vers la scène correspondant au mode de jeu choisi par le propriétaire.

Lors de la deuxième soutenance j'avais uniquement fait les menus de choix d'équipe, et de choix de personnage, je m'étais concentré sur le mode Solo. Ces modes avaient nécessité l'apprentissage et la maîtrise des PhotonPUN RPC.

Et donc, pour cette dernière partie de développement de la partie réseau,

j'ai du me concentrer sur l'implémentation du mode "Défense d'Autel" et du mode "FFA". Il s'agit d'une des parties les plus longues de mon travail sur cette dernière partie de développement. J'ai rencontré beaucoup de problèmes, de bugs, et j'ai effectué beaucoup d'essai afin de bien comprendre comment bien gérer les actions effectués sur la scène du mode de jeu en question.

Pour expliquer un peu le fonctionnement de mon implémentation du mode "Défense d'Autel", au lancement de la scène (donc de la partie), chaque joueur va instancier un `playerManager` (un `GameObject` contenant des scripts utile à la gestion de son joueur), en utilisant `PhotonNetwork.Instantiate()` cela permet à ce que tous les joueurs de la room ait un accès à ce `playerManager` et ait les infos necessaire sur le joueur en question. Ce `playerManager` va avoir comme information le personnage et l'équipe du joueur, et tout un tas d'autres informations. Pour savoir l'équipe et le personnage que le joueur a choisi il suffit juste qu'au moment où le client instancie son `playermanager` sur le réseau, au début de la partie, celui-ci récupère les `CustomProperties` du `LocalPlayer` qui auront été définies dans les menus précédents. Ensuite de ça il a fallu créer de nouveaux scripts spécifiques à ce mode et créer de nouveaux prefabs de personnages. Sur ces prefabs la nouveauté était qu'il fallait mettre un `PhotonView` component, celui-ci est très utile puisqu'il permet de synchroniser des informations avec le serveur et de savoir à qui appartient l'objet. De plus, ce `PhotonView` component permet de synchroniser les animations ainsi que les mouvements, il fallait juste ajouter un "Photon Animator View" component ainsi qu'un "Photon Transform View" component afin que le `PhotonView` puisse synchroniser ces deux éléments. Pour synchroniser les éléments non visuels entre script je me suis servi des `Photon Remote Procedure Call (RPC)`, qui permettent d'appeler une fonction sur des clients spécifiques et mettre à jour ce que l'on souhaite.

Concernant l'intégration des bots (décrite dans la partie IA), Photon ne met pas à disposition des librairies permettant de faciliter la gestion des bots sur un serveur. De plus, ne trouvant rien sur internet à ce sujet, j'ai du complètement mettre en pratique les connaissances que j'avais acquise auparavant. Cependant, je me suis retrouvé rapidement ralenti par les nombreuses contraintes de gestion d'un serveur. Par exemple, le `MasterClient` supprime des objets instanciés (en arrière plan sans action humaine) quand il estime qu'ils ne sont pas utiles, ceci est très problématique puisque les bots n'étant pas des `Player` dans la `Room` ils se font très rapidement supprimer de la scène. J'ai donc testé beaucoup de choses. Pour en arrivant à l'idée de créer un objet `BotManager` qui sera instancié sur la scene, et ayant les infos d'équipes et de personnage que le bot doit prendre. Ce `BotManager` ressemble quasiment à celui du `PlayerManager`, sauf que celui-ci est instancié

par le MasterClient et que le choix du personnage est fait en fonction des choix restants dans l'équipe dans laquelle il sera affecté. Globalement le reste à savoir sur cette partie concernant les bots a été décrite dans la partie "IA multijoueur".

Et donc pour le mode de jeu "FFA", j'ai eu moins de mal à le faire étant donné que les règles sont beaucoup plus simple, et qu'il n'y a pas d'IA à intégrer. Cependant j'ai rencontré des difficultés pour faire terminer la manche, je me suis beaucoup retrouvé avec soit la fin de manche qui ne se déclenchait pas, soit le dernier joueur restant qui ne se détruisait pas et qui restait pour la manche suivante. Comme j'ai été plus rapide à concevoir ce mode de jeu, je me suis permis d'ajouter une petite fonctionnalité de pouvoir suivre la caméra d'un joueur lorsque l'on vient de mourir et que ce n'est pas la fin de la manche, on peut changer de joueur à suivre en cliquant juste sur un bouton.

8 Son et bruitages

(Léo)

Nous en arrivons alors aux sons et aux bruitages du jeu.

Le premier élément pour lequel nous avons du mettre un son est le bouton. Quand je parle du bouton, je parle en fait de tous les boutons du jeu. Ceux du menus comme ceux qu'il y a dans la boutique du mode de jeu solo. Pour cela, il a fallu créer des "AudioSource" sur chacun des boutons, ainsi qu'un script permettant de différencier le son selon si on passe dessus ou si on clique dessus. Ainsi sur chacun des boutons nous avons un bruit en passant et en cliquant sur celui-ci.

Ensuite, il a fallu (par préférence) ajouter une musique d'ambiance dans les menus, pour rendre le tout moins vide. Pour cela, sachant que les scènes changent lorsque nous passons d'un menu à un autre (par exemple du menu principal au menu du mode individuel) il a fallu créer un script "DontDestroyOnLoad" afin d'empêcher la musique de s'arrêter lors d'un changement de scène. Il y a eu cependant un problème : la musique se dupliquait lorsque nous revenions sur le menu principal. Ainsi nous avons changer le script initial pour empêcher l'apparition d'un nouveau GameObject qui diffuse la musique.

venons en aux bruitage utilisé pour le mode solo. Pour ce mode, il a fallu trouver les bruitages suivant :

- Une musique d'ambiance
- Un bruitage pour chaque compétence de personnage
- Un bruitage pour les attaques de monstres

En ce qui concerne la musique d'ambiance, elle a été simplement activé quand la partie se lançait, et arrêté lors d'une fin de partie. Pour les bruitages des compétences et des monstres, il a fallu rajouté un script sur les prefabs des personnages et des monstres de telle sorte que lorsqu'une compétence est activé, l'"AudioClip" associé soit lancé. Il était plutôt aisé t'intégrer ces sons au mode de jeu solo, mais l'histoire à été différente pour le multijoueur.

Finalement, il nous a fallu implémenter les sons pour le multijoueur. Je vais expliqué comment nous avons fait pour un des deux modes de jeu, sachant que le même procédé a été utilisé pour l'autre. Le problème pour les sons en multijoueur, c'est le fait que si nous faisons comme pour le mode solo, le son ne sera jouer qu'en local. ainsi, on n'aura pas les sons ni des opposant, ni des alliés. Il a donc fallu utilisé photon pour que tout le monde entende chaque sons. Par ailleurs, aucune musique d'ambiance ne sera ajouter dans ces modes de jeu dès lors qu'avec chaque son de compétence des personnages, on se perdrait facilement, et trop de sons serait jouer à la fois. Ainsi pour chaque compétence nous avons cette fois-ci du utilisé une fonction "RPC-PlayAudioClip" qui prend en argument un entier correspondant à la capacité utilisé (forte ou faible). Ainsi avec une telle fonction, chacun des joueurs entendra les bruitages de chacune des compétences utilisé sur le terrain de jeu.

9 Bande Annonce

(Léo)

Lorsque le jeu fût enfin fini, il a fallu commencer la bande annonce. En effet, on ne pouvait pas la commencer sans avoir de plans du jeu à afficher. Cependant, avant cela, il a été préférable d'établir les idées qui ont été les suivantes :

- afficher le logo au début de la bande annonce
- mettre une musique entraînante
- faire un plan d'apparition des personnages
- mettre en avant les modes de jeu disponibles

Ainsi, même si le début du montage ne se fait qu'à la fin, les idées étaient déjà toutes faites et il n'a pas été long d'obtenir les plans voulus. Finalement, la bande annonce a vu le jour et correspond de près à ce que nous nous attendions.

10 Bilan du projet

10.1 Bilan général

10.1.1 Les points positifs

Premièrement, le plus gros point positif est le fait que ce projet est majoritairement libre, nous avons donc pu entièrement décider quels chemins entreprendre et concevoir à notre goût. Ensuite, ce projet fut bénéfique pour tous les membres du groupe. En effet, chacun a pu développer des compétences différentes à travers les tâches qu'il a pu effectuer. Comme nous apprenons le C# en Informatique Pratique, ça nous a grandement aidé pour faire toute la partie code du jeu sur Unity. Aimant tous les jeux vidéos, en faire un était agréable, intéressant, et amusant à faire (en dehors des difficultés rencontrées tout le long). Nous avons donc pris plaisir tout au long, avec l'envie de se surpasser pour être fier du rendu. Ce projet étant en groupe, nous avons également pu progresser tout au long en terme de travail de groupe. En effet, organiser un travail de groupe est une chose difficile, et sur un projet d'une telle ampleur il a fallu revoir nos méthodes tout au long pour essayer de rendre efficace notre travail, et ne pas ralentir l'avancée du groupe par une mauvaise organisation. Nous sortons aujourd'hui de ce projet en ayant une meilleure idée de ce qu'est un travail en groupe sur longue durée. La bonne entente du groupe fut également un aspect positif puisque ça a permis à tout le monde de se sentir bien au sein du groupe, et de prendre plaisir à mener à bien ce projet. Ce projet nous a permis aussi à tous de progresser à l'oral, et surtout d'apprendre à gérer correctement un oral en groupe, les oraux en groupe étant beaucoup plus difficile à gérer en si peu de temps de passage.

10.1.2 Les points négatifs

Beaucoup de points positifs ont été évoqués concernant ce projet. Cependant certains aspects négatifs sont présents. Le premier étant que même si devoir se documenter soi-même est bénéfique puisque c'est une qualité non négligeable, certaines fois lorsque nous avons des difficultés sur certaines choses, notamment l'utilisation de git en groupe où nous avons pas réellement d'aide lorsqu'on ne s'en sortait pas, ce qui nous faisait perdre énormément

ment de temps sur l'avancée, et qui n'était pas du temps rentable. Un autre point négatif, plutôt lié à la situation sanitaire de la Covid-19, est que comme nous étions une majeure partie du semestre en distanciel, nous n'avons pas pu échanger facilement sur le projet, nous avons mis en place un serveur Discord où nous avons essayé de communiquer le maximum d'informations, cependant ce n'était pas toujours facile de travailler totalement en groupe à travers uniquement un partage d'écran. Et enfin, le dernier point négatif est que comme le projet se déroule parallèlement aux cours, ça a été souvent difficile d'avancer à la fois sur le projet et s'en sortir pendant les cours, ce qui a nui à notre productivité, et fait que le jeu n'est pas entièrement parfait. La plupart des tâches ont été réalisées entièrement par le responsable de la tâche à cause de ce manque de temps, puisque le suppléant de la tâche ne pouvait pas à la fois faire sa tâche où il était responsable et aider sur une autre. Nous avons donc une petite frustration de ne pas avoir eu le temps de faire mieux, même si nous sommes fiers du résultat.

10.2 Les problèmes rencontrés

Au niveau des problèmes rencontrés, nous avons eu des difficultés à configurer correctement notre repository git sur GitHub, la création n'a pas été un problème, mais l'utilisation en groupe a été compliquée à prendre en main. Ces difficultés nous ont ralentis, en effet la mauvaise utilisation de notre repository git nous a coûté de nombreuses heures de réparations d'objets s'étant détruits ou modifiés. De plus, lorsque les éléments volumineux en terme d'espace de stockage ont dû être ajoutés (notamment les maps et les modèles 3D de personnage) nous avons dû utiliser git LFS (Large File Storage) puisque git a une limite de stockage de fichiers, la configuration de celui-ci a été laborieuse également et nous a provoqué de nombreux problèmes ralentissant l'avancement général du projet.

Nous avons également eu des problèmes au niveau des modèles 3D, en effet Marine a utilisé Blender pour faire les modélisations, il fallait donc les importer sur Unity ensuite, malheureusement lors de l'importation des personnages sur Unity les modélisations étaient corrompues, elles apparaissaient avec des trous. Marine a donc dû retravailler les modélisations pour que l'exportation se passe correctement, causant un retard sur certaines tâches nécessitant l'accès aux modèles 3Ds sur Unity.

10.3 Bilans personnels

10.3.1 Léo

Ce projet a été une expérience pour moi. Ayant pour habitude de rester dans mon coin pour travailler, j'ai dû faire un effort pour communiquer avec mon groupe pour bien établir ce qu'il fallait faire et améliorer, selon les préférences de chacun dès lors que la partie sur laquelle j'ai travaillé peut faire débat. Intégrer l'intégralité de la bande sonore dans le jeu a été un plaisir pour moi, par le plaisir que j'ai du domaine audio. Découvrir comment fonctionne le système Unity avec les sons et apprendre à gérer les quelques difficultés rencontrées a été une expérience enrichissante, et tout du moins sympathique. Le rendu du projet est finalement satisfaisant, ce qui m'a permis de faire une bande d'annonce qui m'a donné du plaisir à faire. D'autant plus qu'initialement, l'audiovisuel est un domaine sur lequel j'aurais pu m'orienter. Ainsi, le projet dans sa globalité fût un plaisir à réaliser, et si l'expérience que nous avons vécue lors de celui-ci était à refaire, je n'hésiterai pas.

10.3.2 Mathis

C'était un projet très sympa à accomplir, surtout avec ce groupe. Ce projet a créé de belles amitiés. Nous nous sommes bien entendus et entraïdés. Nous avons décidé ensemble à quoi le jeu devrait ressembler à la fin, et nous avons réussi, à quelques exceptions près. C'était malgré tout plutôt plaisant de rechercher par soi-même des solutions à des problèmes que nous rencontrions. Le fait aussi de travailler en groupe, en apprenant les forces et faiblesses de chacun, faire confiance au travail d'autrui, respecter les deadlines que nous nous étions fixées, était agréable. J'ai fortement apprécié le travail de groupe. Ce projet a été vraiment bénéfique pour moi puisque j'ai codé 95% du jeu, et grâce à tout ce travail j'ai pu vraiment consolider mon niveau en programmation, ceci m'a également conforté dans mon choix de faire des études dans l'informatique, j'ai pris énormément de plaisir à programmer le jeu. J'en ressors en sachant vraiment utiliser Unity et ses différentes caractéristiques. Grâce à ce projet, j'ai pu également concevoir mon premier site web, et j'ai vraiment pris plaisir à le faire, je voulais un beau site, c'est pourquoi j'ai passé beaucoup de temps à apprendre et à le réaliser, je suis très content du résultat. J'ai également acquis des connaissances dans la gestion d'un jeu en réseau, grâce à Photon j'ai économisé du temps sur de nombreuses contraintes, mais j'ai pu me confronter à une majorité des contraintes que l'on peut rencontrer dans l'élaboration d'un jeu en réseau. La seule compétence que je n'ai pas acquise c'est au niveau des modélisations et des animations, je me suis uniquement occupé de la partie code concernant les animations.

Je tiens donc à remercier Marine et Arthus qui ont fait un travail excellent sur l'ensemble des parties graphiques (modélisations 3Ds et animations), où sans eux le jeu n'aurait pas été si beau, et tout le code que j'ai pu produire n'aurait pas servi à grand-chose. C'est un travail d'équipe avant tout, et on a donc besoin de chacun pour avoir un résultat final qui nous convient. Merci à toute l'équipe ! Tout le travail effectué depuis la dernière soutenance se voit vraiment et permet d'avoir un jeu agréable à regarder et à jouer.

10.3.3 Arthus

Personnellement, ce projet fût une expérience fantastique, il a tous d'abord permis de consolider mes relations avec les membres de mon groupe, de créer une confiance mutuelle et solide, il a aussi permis de voir ce qu'est la réalisation d'un gros projet contenant le travail de diverses personnes. De cela en découle le fait de devoir mettre les bouchés doubles sur ses parties (je veux dire par là travailler très tard le soir par exemple) pour que les autres puissent aussi s'avancer dans leurs parties. Comme partout, il y a eu des problèmes mais la cohésion de notre groupe s'est bien faite ressentir. Derrière ce projet il y a donc une grosse part de social, mais aussi un aspect plus technique. Ce projet m'a aussi permis d'élargir mes connaissances, avec la découverte des logiciels Blender et Unity et de leurs utilisations ou encore l'approfondissement d'autres logiciels comme Git. Tout au long de ce projet je me suis aussi découvert une réelle passion dans le design de carte, car je peux y laisser s'exprimer ma créativité, et les animations, via lesquelles je peux donner vie à des personnages pour le moment inanimé, ce sont des parties qui m'ont vraiment fasciné et que je vais très probablement continuer de mon côté même après la fin de ce projet. Ce projet, comme je l'ai dit précédemment, m'as parfois poussé à bout, néanmoins, je n'ai jamais abandonné, ce qui m'a confirmé à moi-même ma pugnacité et mon envie de réussir. J'aurais aimé un peu plus me pencher sur la partie code mais je voulais découvrir de nouvelles choses, de surcroît, les tâches devaient être réparties, dans le cas contraire la réalisation aurait été tout bonnement impossible. Cependant, je suis fier de Mathis qui a très bien su s'en occuper et je lui en remercie beaucoup. Je remercie aussi Marine, qui a su designer ces merveilleux personnages, sans elle je n'aurais sûrement jamais découvert cette passion dans les animations. Je finirai par remercier le groupe en général, ce fut une aventure mémorable et pleine de gaieté sur ce magnifique projet qui relève de l'informatique certes mais aussi du social et de ses capacités.

10.3.4 Marine

J'ai trouvé ce projet vraiment plaisant à réaliser, grâce à lui j'ai pu apprendre pleins de nouvelles choses. Le fait que notre groupe se soit bien entendu durant toute la durée du projet nous a été vraiment bénéfique. La complicité et la confiance présente au sein du groupe m'ont permis de m'épanouir pleinement durant la réalisation du projet. Devoir réaliser un tel projet en groupe n'a pas toujours été facile et il a fallu parfois faire des concessions et parfois s'adapter aux autres, travailler en groupe m'a permis de savoir faire face aux difficultés des autres, faire confiance aux autres et de mieux communiquer. Malgré quelques difficultés nous sommes arrivés à bout du projet et c'est grâce au travail de chacun d'entre nous. J'ai donc beaucoup aimé la partie coopérative du projet et la partie réalisation en groupe de celui-ci. Pour le côté travail personnel j'ai aussi adoré travailler sur ce projet, j'ai pu apprendre l'utilisation de nouveaux logiciels comme Blender et Unity, et aussi mieux utiliser Git et Github. J'ai pris beaucoup de plaisir à réaliser toutes mes modélisations 3Ds et aussi pouvoir participer au côté artistique du jeu, j'ai pu m'améliorer en modélisation. Mais j'ai pris encore plus de plaisir quand je les ai vu prendre "vie" directement dans le jeu, grâce aux magnifiques animations d'Arthus et au superbe travail de Mathis (je les remercie beaucoup pour ça). Du fait que les modélisations étaient une part essentielle à l'accomplissement du projet je me suis sentie utile dans ce projet. Je tiens quand même à remercier l'ensemble du groupe pour tout le travail accompli depuis le début du projet ! On a réussi à faire un jeu qui fonctionne et qui est agréable à jouer.

11 Conclusion

Bagarena est un projet qui porte en lui les espoirs et les efforts de notre groupe. Il est la chose qui nous a fait nous réunir, travailler et débattre. Bagarena est l'aboutissement d'une demi-année de labeur de quatre personnes dont la volonté et la persévérance ont permis de surmonter les obstacles.

Il y a 6 mois de cela notre projet n'était qu'une idée imaginée par quatre personnes ayant peu de connaissances et de maîtrises de ce qu'est la conception d'un jeu et la gestion d'un projet d'une telle ampleur, désormais, il représente notre détermination à réussir notre travail et ce qui s'en suit.

Notre projet est complet, tous nos objectifs primaires sont remplis. Pour ce qui est du secondaire certaines parties sont très bien développées, certaines auraient sûrement mérité plus d'approfondissement, cependant, la globalité finale de Bagarena satisfait notre groupe.

Table des figures

1	Ébauches du logo de O.R.E.P.A	5
2	Ébauches du logo du projet	6
3	Vue isométrique	9
4	Graphismes de Fortnite	10
5	Première version du site	14
6	Progressus - Template Bootstrap utilisé	15
7	Page d'accueil de Bagarena	16
8	Première version de Saadel	18
9	Version finale de Saadel dans Unity	19
10	Première version de Tyo	19
11	Première version de Niaka	20
12	Les versions finales de Tyo(droite) et Niaka(gauche)	21
13	Flèche de l'arc de Lasnya	21
14	Les versions finales du marchand et de Lasnya	22
15	Dernière version des trois autels	23
16	Animator Contrôler des Autels	24
17	Animation d'attaque lourde de Tyo sur Blender	25
18	Animator Contrôler de Unity	26
19	Marchand du solo	29
20	Magasin du Marchand	29
21	Capture d'écran de la map solo	30
22	2ème Capture d'écran de la map solo	30
23	Capture d'écran de la map 3vs3	30
24	Crevasse de la map 3vs3	31
25	Brouillard de début de la map 3vs3	32
26	Pente faites à la main	33
27	Pente faites avec l'outil Smooth Height	33
28	Les maps en résumé	34
29	Menu Principal	35
30	Menu des options	36
31	Menu de choix de personnage en Solo	36
32	Menu choix de pseudo - Menu sélection (création de salon, ou rejoindre un salon)	37
33	Menu de choix de choix d'équipe	38
34	Menu choix de combattant pour le mode "Altar Defense"	39
35	Menu choix de combattant pour le mode "FFA"	39
36	Menu pause du mode Solo (gauche) - Écran de fin de partie en mode Solo (droite)	40

37	Menu Pause des modes FFA et Defense d'Autel (vue depuis le mode Defense d'Autel, ils sont identiques)	40
38	Menu de boutique du marchand	41
39	Interface du mode solo	41
40	Logos des attributs	43
41	Les deux modèles 3D utilisés pour les bots	45

Liste des tableaux

1	Tableau de la répartition des tâches	7
2	Tableau des objectifs d'avancement	13