

Rapport de Soutenance 1

- Bagarena -

O.R.E.P.A



Mathis BRAHIM
Arthus DAIMEZ
Marine ROY
Léo WEBER

03 Mars 2021

Table des matières

1	Introduction	3
2	À propos du cahier des charges	3
3	Conception	4
3.1	Graphismes	5
3.1.1	Personnages	5
3.1.2	Map	8
3.2	Animations	11
3.3	Sons	12
3.4	Menus	13
3.5	Réseau	15
3.5.1	Photon Unity Network	15
3.5.2	Hall (Lobby)	16
3.5.3	Room	16
3.5.4	Instantiation du Player dans la room	17
3.6	I.A.	17
3.7	Mécaniques de jeu	18
3.8	Site Web	19
4	Avances et retards	23
4.1	Les réussites	23
4.2	Les retards	24
5	Prévisions	24
5.1	Graphismes	24
5.2	Animations	25
5.3	Sons	25
5.4	Menus	25
5.5	Réseau	25
5.6	I.A.	26
6	Conclusion	26

1 Introduction

Tout d'abord, laissez moi vous rafraîchir la mémoire! Bagarena est un jeu de combat en arène, amusant et divertissant, il saura vous faire passer d'agréables moments pleins d'actions dans ce monde fantastique. Préparez vos combattants pour faire des combats excitants au sein des arènes de Bagarena!

Depuis le début du projet notre équipe composée de Léo, Arthus, Marine et Mathis, n'a cessé d'apprendre à travailler hors de sa zone de confort. En effet, nous n'avions aucune expérience dans la conception, réalisation d'un jeu-vidéo, cependant notre motivation et nos capacités d'adaptation vont nous permettre de mener à bien ce projet. Nous sommes tous les quatre passionnés, et c'est cette passion de vouloir apprendre qui va nous permettre de mener ce projet à bien. Cette première partie de développement s'est avérée pour une grande partie être de l'apprentissage et de la recherche, elle va cependant nous permettre d'acquérir quelque peu des bases solides. La création d'un système de multijoueur stable grâce à Photon. L'apprentissage de la modélisation d'objets 3D via Blender. L'élaboration d'une intelligence artificielle. La vocation a diriger et organiser une équipe. Et nous en passons.

Et pourtant, un si long voyage encore à parcourir avant de voir ce jeu abouti. Car même si la forme globale du jeu commence à se dessiner, notre persistance nous fait remarquer de nombreux points à perfectionner.

2 À propos du cahier des charges

Nous n'avons que très peu modifié le cahier des charges depuis sa validation. Il n'a s'agit de revoir seulement la partie réseau à la hausse pour cette première soutenance, suivant les conseils du jury. Nous avons également échangé quelques rôles, mais uniquement entre responsable et suppléant de la tâche. Il s'agit de la conception de l'environnement, la physique du jeu, et la création du site web.

Voici le nouveau tableau de répartition des 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

Outre ces changements, nous nous sommes tenus à atteindre les objectifs d'avancement que nous nous étions fixés initialement dans le cahier des charges. Après une première période de développement, nous pouvons dire que nous sommes globalement satisfaits des objectifs que nous nous sommes fixés et que ces derniers sont dans l'ensemble réalistes. Il n'a pas forcément été évident de les fixer, ne pouvant savoir que certaines tâches allaient être plus difficiles que prévues, ou plus simples.

3 Conception

Notre but le plus profond à O.R.E.P.A est de rendre nos joueurs heureux quand ils joueront à notre jeu et qu'ils passent un agréable moment. L'idée d'un jeu final vide et sans amusement nous rebute au plus au point. D'après nous le plus gros facteur de réussite d'un jeu est le fait que si le joueur appui sur une touche il faut au moins que cela engendre deux, voire trois réactions visuelles et ceci dans le but de garder l'attention du joueur et de lui donner une certaine satisfaction.

Nous nous sommes donc réparti les tâches afin de mettre en place les bases nécessaire, et pouvoir ensuite le perfectionner et exploiter l'effet décrit précédemment.

Du à notre expérience manquante, le début a été difficile, il a fallu apprendre à manier des concepts ainsi que des logiciels nouveaux, cependant avec du travail et de la persévérance nous arrivons et arriverons à surmonter ces difficultés.

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 2 – Répartition des tâches pour la soutenance 1

R : Responsable de la tâche

S : Suppléant

3.1 Graphismes

3.1.1 Personnages

(Marine)

Comme présenté lors du cahier des charges, dans Bagarena le joueur a le choix entre 4 combattants :

- Saadel (Combattant)
- Lasnya (Tireuse)
- Tyo (Tank)
- Nyaka (Soutien)

Je me suis rapidement lancée dans la conception des personnages, du faite que je doit en faire quatre et que cela pouvait prendre pas mal de temps. Pour pouvoir les modéliser j'ai utilisé le logiciel de modélisation 3D gratuit Blender. j'ai souhaité ne pas utiliser d'assets pour les combattants pour garder de l'authenticité dans notre jeu. Le premier combattant que j'ai modélisé est Saadel. J'ai d'abord commencé avec un cube que j'ai modifiée pour en

faire un pied puis une jambe que assemble avec le pied et ainsi de suite pour former un personnage en entier puis modéliser les armes si le combattant en as. Lorsque tous les membres du personnage sont fait il reste plus qu'a rajouter une armature pour y incorporer des os pour permettre a la modélisation d'avoir des articulations afin que mon collègue Arthus puisse animer les combattants. Pour rendre chaque personnages plus ou moins unique et pour pouvoir les différencier j'ai essayer de faire des morphologies différentes en fonction de leur genre, si le combattant est une femme par exemple lui faire des traits plus fin et un corps plus fin, et de leur rôle dans le jeu, par exemple pour le tank essayer de faire un personnage assez imposant et plus grand que les autres, et plus tard quand les combattants auront leurs textures ils seront bien plus différents.

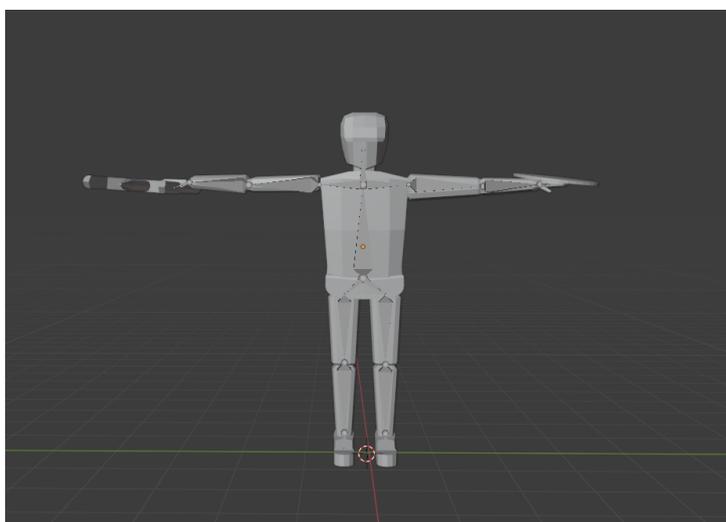


FIGURE 1 – Combattant Saadel avec son trident et son bouclier

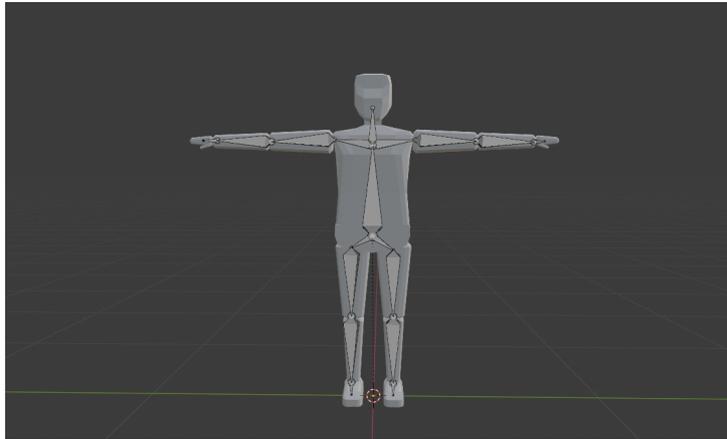


FIGURE 2 – Soutien Nyaka

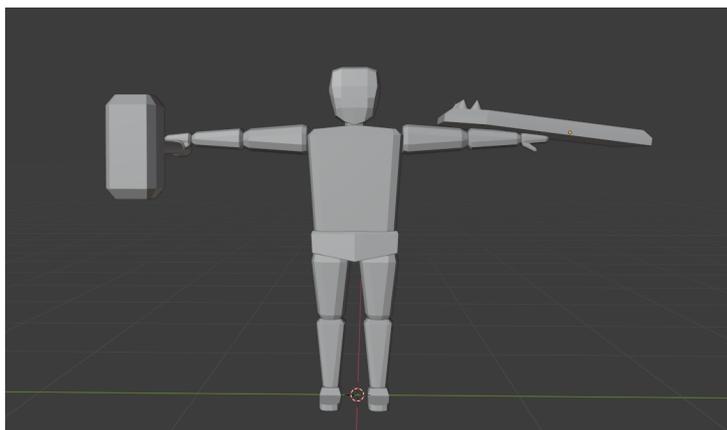


FIGURE 3 – Tank Tyo avec son marteau et son bouclier

De plus j'ai aussi modéliser les autels qui seront présents sur la map et que il faudra protéger lors du mode de jeu solo. Il est composé d'une pierre en lévitation et d'un socle.

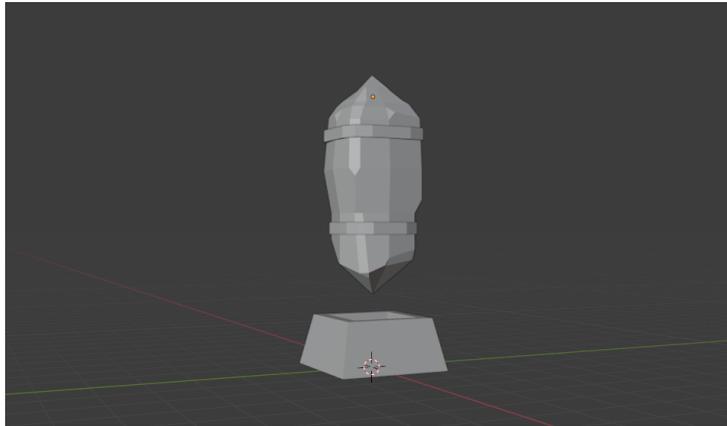


FIGURE 4 – Autel a défendre

3.1.2 Map

(Arthus)

Pour ce qui est de la map, c'est la partie qui m'a tout de même pris beaucoup de temps. Faire une map bâclée m'était totalement inconcevable. Pour le design de celle-ci, j'ai utilisé un logiciel que je n'avais jamais utilisé jusque là : Unity. Pour comprendre son fonctionnement je me suis beaucoup renseigné sur divers sites dont YouTube, et j'ai créé plusieurs maps de test avant d'arriver à cet aboutissement. 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 sur ce logiciel il n'existe pas d'outils pour faire des pentes ainsi nous devons jouer avec l'extrusion petit à petit pour créer celle désiré.
- L'ajout d'assets nécessite aussi pas mal de documentation pour le seul fait de les ajouter.

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, ou encore un étang. Enfin j'y ai incorporé des infrastructures

et j'ai créé des hitbox sur tout mes objets pour éviter que le joueur ne les traverse. Enfin, j'ai mis des murs invisible tout autour de la map pour éviter que les joueurs ne tombent dans le vide, ce qui bloque l'extract du joueur, les 3 entrées de mobs et leur laisser de la place pour que mon collègue Mathis puisse les faire spawner.

Cette map est constituée d'assets récupérés principalement sur le site "<https://assetstore.unity.com>". Ces assets sont par exemple des texture pour les montagne, des types d'herbes, des fleurs, des préfabriqués (comme des bâtiments ou des objets à placer soi-même (comme pour faire une table sur laquelle il y aura des chopes par exemple)). Pour l'instant j'ai utilisé un personnage et un "autel" récupéré sur ce site dans le but de se déplacer, de tester la carte, et d'avoir un premier aperçu, mais par la suite nous mettrons bien entendu nos modèles de personnage et d'autel. Quelques images pour imager mes propos.

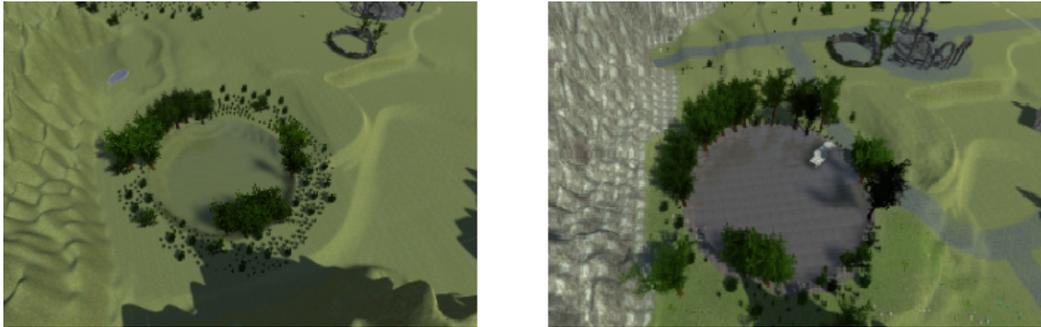


FIGURE 5 – Évolution de la map : avant / après

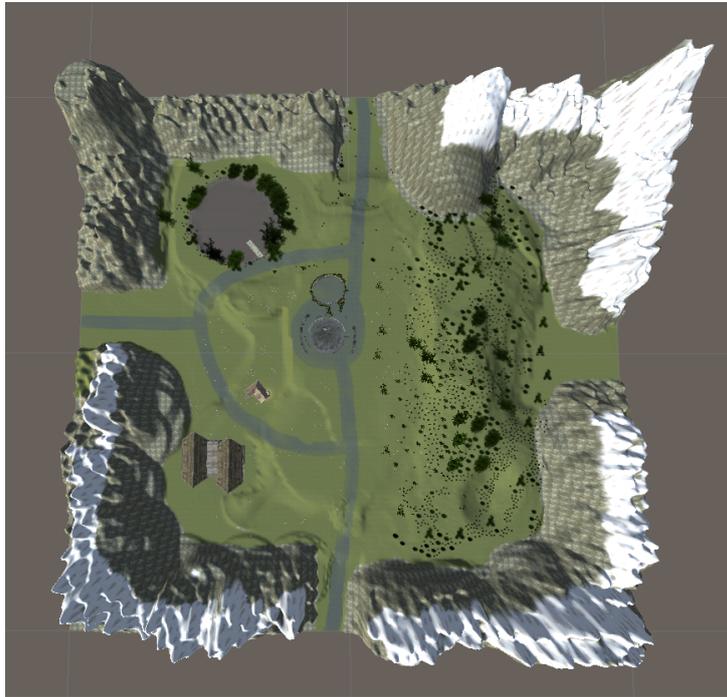


FIGURE 6 – Carte de jeu dans son ensemble

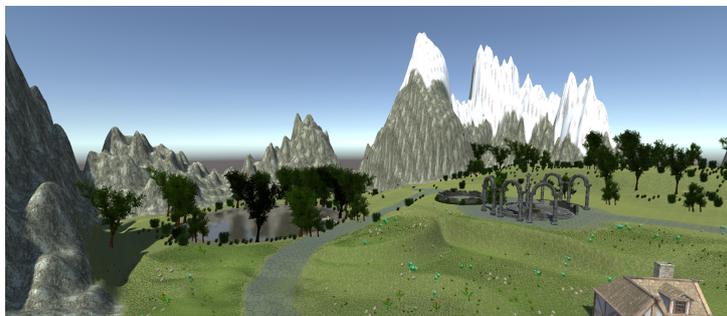


FIGURE 7 – Photo de la zone à défendre et de l'étang

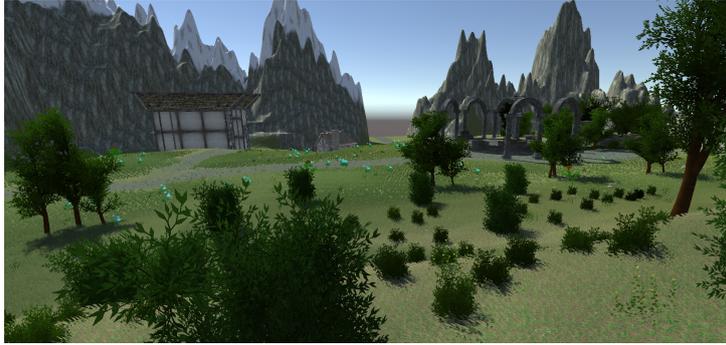


FIGURE 8 – Photo depuis la colline

3.2 Animations

(Arthus)

Dès le début du projet, je suis allé sur un site stockant une grande quantité d'animations de bases comme "marcher" ou bien "s'accroupir". Sur celui-ci je me suis fait un fichier répertoriant les plus basiques. Néanmoins, suite à la réception du modèle de Saadel, j'ai commencé à créer l'attaque lourde de ce dernier se battant au trident. Pour ce faire, je me suis servi du squelette du personnage que ma collègue Marine nous a créé ainsi que du logiciel Blender.

Plus en détails, j'ai modifier les positions et les rotations des os à des instants précis en les marquant avec des "Valeurs Clés". Blender ensuite fait pour que le personnage bouge d'une clé valeur à une autre de manière uniforme. Néanmoins il nous est possible d'accélérer/ralentir la rotation et/ou la position.

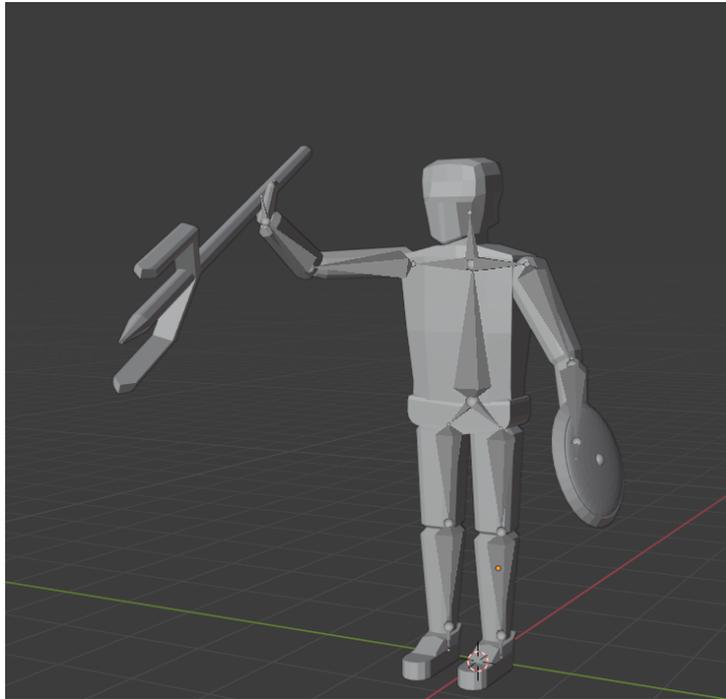


FIGURE 9 – Attaque lourde de Saadel

Pour ce qui est du décor, j'ai récupéré sur le site d'assets, des animations que j'ai calqué sur les arbres, ou encore sur l'eau, qui se déplacent avec le vent.

3.3 Sons

(Léo et Arthus)

Concernant les sons du jeu, on ne pouvait pas vraiment les choisir avant de connaître la texture des matériaux et l'ambiance de notre univers. Ainsi, il a fallu attendre qu'une map soit fini pour décider des sons qu'on aurait besoin.

C'est donc après la réalisation de la map qu'Arthus et moi avons décidé des sons dont nous auront besoin. en voici une liste, à priori non exhaustive :

- Bruit de feu
- Bruit de vent
- Bruit d'arc qui décoche une flèche
- Bruit de flèche qui atteint son objectif (bois, métal, brique, ...)

- Bruit de monstres
- Bruit de pas (dans l'eau, dans l'herbe, ...)
- Bruit pour annoncer une nouvelle vague de monstre
- Bruit de sorts
- Bruit qui annonce que l'autel est attaqué
- musique d'ambiance (calme et stressante)

Après avoir établi nos besoins en terme de sons, nous avons donc commencé nos recherches. Pour le moment, nous n'avons pas trouvé tous les sons mais une grande partie était trouvable sans grande difficultés, avec la bibliothèque audio libre de droit de YouTube.

3.4 Menus

(Mathis et Léo)

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 10 – Menu Principal

Pour la partie Options nous avons décidé d'ajouter la possibilité de diminuer le volume du jeu avec un Slider, de changer la résolution (dans le

Dropdown il y aura 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.

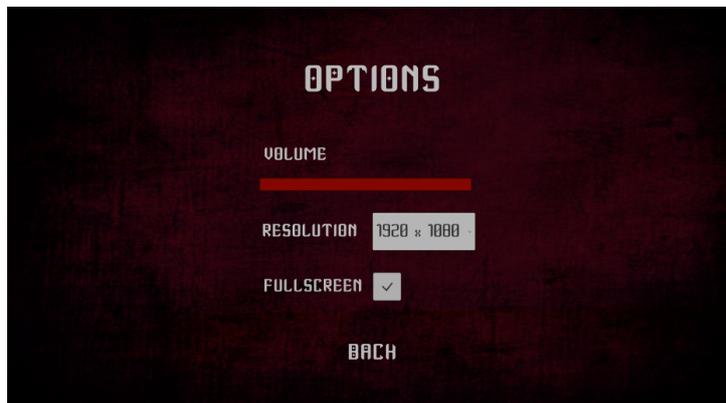


FIGURE 11 – Menu des options

Lorsque l'utilisateur sélectionne le mode "Solo", il va être redirigé vers la scène "Solo" où il va pouvoir choisir son combattant. S'il sélectionne le mode "Multiplayer", il va être redirigé vers la scène "Multiplayer", celle-ci contient les menus propres à la partie multijoueur. Il va pouvoir choisir un nom d'utilisateur, 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"). Pour plus de détails sur le menu multijoueur voir la partie "Réseau" ci-dessous.

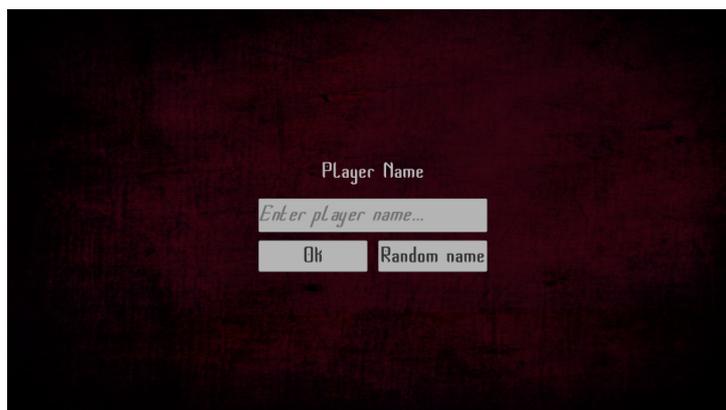


FIGURE 12 – Menu de choix du pseudo



FIGURE 13 – Menu de connexion

Chaque bouton a une liaison vers un objet, contenant un script avec diverses fonctions. Allant d'un simple affichage d'un menu en retirant la boîte contenant l'ancien menu, aux lancements des fonctions de connexions multijoueurs.

3.5 Réseau

(Mathis et Léo)

3.5.1 Photon Unity Network

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. Pour ce faire, 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 expliquer brièvement 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 différents 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 événements. 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().

3.5.2 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.

3.5.3 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.

3.5.4 Instantiation du Player dans la room

Pour que les joueurs puissent se voir dans la partie il faut tout d'abord instancier les joueurs. En attendant la réalisation du menu de choix de personnage et l'implémentation de celui-ci, nous avons donc réalisé une instantiation des joueurs très simplifiée, pour ensuite la compléter.

Nous avons créé un GameObject "RoomManager", il prend le relais pour superviser les actions relatives au réseau. Il s'occupe tout d'abord d'instancier le joueur sur le réseau en appelant :

```
PhotonNetwork.Instantiate(playerPrefab.name, startPosition, Quaternion.identity);
```

Cette méthode permet d'instancier le joueur dans le jeu de chaque joueur et dans la scène. Un PhotonView component doit être attaché au GameObject que l'on veut instancier afin de lui donner une identité sur le réseau. Enfin, c'est également le Photon View component qui s'occupe de synchroniser la position et la rotation du GameObject (à l'aide d'un Photon Transform View component) ainsi que d'autres valeurs spécifiées dans le script attaché au GameObject, qui doit implémenter l'interface IPunObservable. Le Game-Manager permettra également au joueur de quitter la partie depuis le menu Pause en appelant la méthode **PhotonNetwork.LeaveRoom()**.

Et enfin nous avons créé un script PlayerManager.cs, qui sera attaché au GameObject du personnage que l'on voudra implémenter. Dans ce script pour l'instant, seule une initialisation d'un attribut PhotonView est implémenté, ainsi qu'un appel à une fonction CreateController() (qui ajoutera les différents contrôles du joueur sur le personnage instancié) lorsque la condition IsMine est vérifiée (IsMine est un booléen permettant de savoir quel est le GameObject instancié par le joueur et lui appartenant).

3.6 I.A.

(Mathis et Léo)

Nous avons commencé par l'IA de notre mode Solo, pour rappel le mode Solo se déroule en système de vague. Nous avons donc du implémenter un système de vague que nous développons avec l'IA.

L'IA est actuellement capable de :

- Suivre le joueur placé en paramètre.
- Stopper sa course lorsqu'il est à une petite distance du joueur ou de l'autel.

— Choisir sa direction entre l'autel ou le joueur.

Dans l'optique de débiter les tests concernant les mécaniques de jeu, nous avons commencé à implémenter une IA pour l'instant minimaliste, elle se contente 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 Nav-Mesh (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).

Concernant le système de vague, celui-ci se contente d'instancier le GameObject de l'IA dans les 3 différents points d'apparition. Le nombre d'ennemis apparaissant par vagues augmente au fur et à mesure.

Lors de la création de ce personnage non joueur, plusieurs problématiques se sont posées :

Premièrement pour le système de vague il a fallu utiliser une Coroutine, car lorsque les ennemis sont instanciés en même temps ils se superposent dans la scène, on ne distingue donc pas le vrai nombre d'ennemis. Le fonctionnement des Coroutines nous était inconnu, nous avons donc dû nous renseigner sur celles-ci et ce fut la première difficulté.

La seconde difficulté rencontrée, fut la gestion de l'arrêt des ennemis lorsqu'ils sont à une certaine distance pour pouvoir les attaquer. En effet, 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'un autre ennemi s'approche de lui la collision va les faire se repousser l'un l'autre.

3.7 Mécaniques de jeu

(Mathis)

Le jeu possède actuellement les mécaniques suivantes :

- Mouvement du joueur
- La caméra peut suivre le joueur
- Collisions entre le joueur, les IA, ainsi que l'Autel

Lors de la mise en place des quelques mécaniques pour l'IA, nous avons besoin de pouvoir contrôler un joueur, ainsi que de pouvoir le suivre avec une caméra. Nous avons donc créé un script qui permet à la caméra de suivre le GameObject du joueur. Puis la gestion des mouvements, il s'agit d'une simple gestion des entrées claviers qui vont effectuer une Translation de l'objet suivant un Vecteur de direction.

Pour gérer les collisions nous avons simplement ajouté aux GameObject des IA, et du joueur un Rigidbody, auquel on pourra s'occuper par la suite de la gestion des forces et de la gravité par exemple.

3.8 Site Web

(Mathis)

Pour cette soutenance, avoir un site web ne faisait pas partie des actions demandées. Cependant, j'ai souhaité le réaliser 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 connaissances dans le 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 anciens. Ci-dessous le design de ma première version.

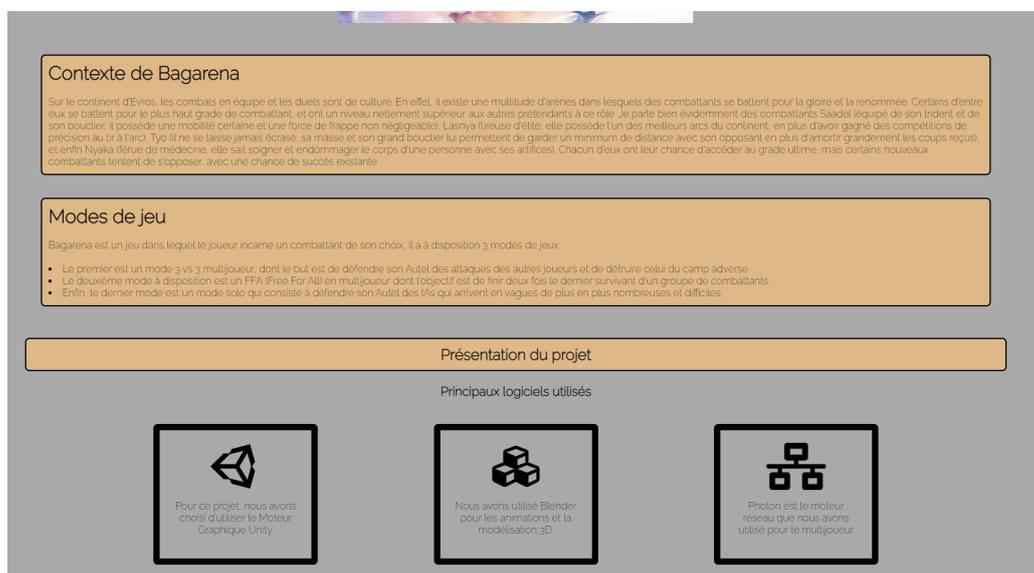


FIGURE 14 – Première version du site

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) basiques pour pouvoir tout recréer de moi-même tout en ayant quelques objets à réutiliser, notamment le menu et quelques composants CSS. De plus, Bootstrap permet d'avoir un site responsive et ceci me semblait être un atout. Le site a donc totalement été recréé, j'ai uniquement gardé le menu, la forme du footer, et réutilisé les classes CSS utilisés dans la page de démo du template pour mettre en forme la partie HTML.

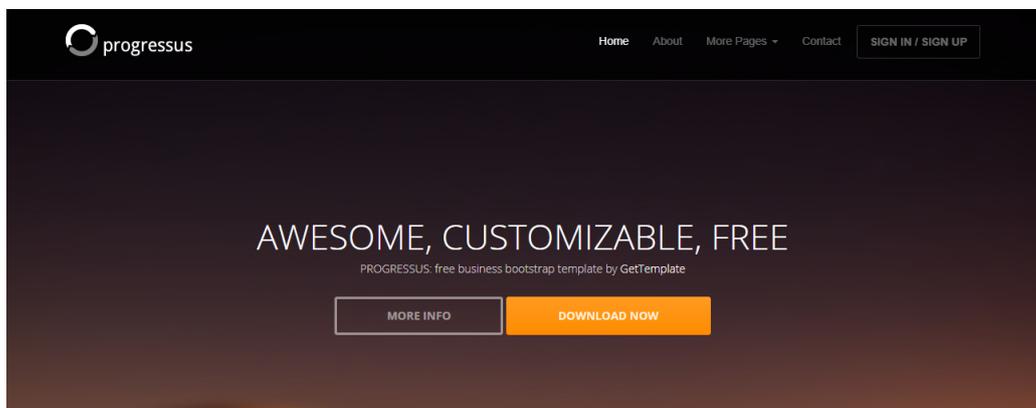


FIGURE 15 – Progressus - Template Bootstrap utilisé

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, archive en .zip).
 - 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 16 – Page d'accueil de Bagarena

Pour héberger le site, nous utilisons GitHub Pages qui permet de transformer un Projet en hébergement pour site web.
Le lien du site web : <https://skraam308.github.io/BAGARENA-website>

4 Avances et retards

Tâche	Prévu	Actuel
Edition 3D	30%	40%
Animation	20%	25%
Conception de l'environnement	25%	40%
Physique du jeu	0%	0%
I.A	20%	20%
Musiques et bruitages	30%	30%
Création du site web	33%	90%
Bande annonce	10%	10%
Mise en place multijoueur	50%	50%
Responsables Communication	33%	33%

TABLE 3 – Tableau des avances et retards

4.1 Les réussites

Pour l'instant, les problèmes que nous avons ne viennent pas de là où nous pensions. En effet, le réseau très basique (connexion/synchronisation) s'avère certes exigeant dans sa réalisation, mais pas aussi complexe que ce qui était attendu. Avec beaucoup de travail, de documentation et de tests, le réseau devrait suivre les prévisions attendues.

Ensuite viens la modélisation, Blender est un outil très puissant et avec ce dernier nous avons pu créer des modèles 3D satisfaisants.

Concernant les menus, Unity propose des outils d'interface utilisateur qui sont faciles à mettre en oeuvre. Ainsi, il n'y aucun souci à déplorer dans la création des menus.

Le site web est terminé, désormais il suffira de le mettre à jour, nous allons gagner du temps pour se concentrer uniquement sur le développement du jeu.

Nous avons une idée des bruitages dont nous avons besoin, nous en avons trouvé pour la plupart, il faudra maintenant voir comment l'implémenter et potentiellement en créer nous même pour qu'ils correspondent parfaitement à ce qu'on veut. Pour la bande-annonce, nous avons fait une timeline.

Et enfin pour finir nous avons donc créé un compte Twitter et un compte Instagram qui vas permettre de communiquer sur l'avancée de notre projet et de promouvoir le jeu.

4.2 Les retards

Les choses viendront à se complexifier avec notamment les mécaniques de jeu, et en particulier la physique. En effet, la cohérence et le réalisme de celle-ci est un processus de création généralement laborieux. Par exemple les collisions entre joueurs et ennemis doivent être réaliste afin de ne pas avoir une impression d'avoir des ennemis très lourds qui s'éjectent de chaque côté et loin. C'est pourquoi nous nous sommes pas encore attardé sur la physique de notre jeu, nous attendons d'avoir avancé dans le développement global de notre jeu pour s'y attarder.

5 Prévisions

Tâches	1ère Soutenance	2ème Soutenance	3ème Soutenance
Édition 3D	40%	60%	100%
Animation	25%	50%	100%
Conception de l'environnement	40%	50%	100%
Physique du jeu	0%	50%	100%
I.A	20%	60%	100%
Musiques et bruitages	30%	60%	100%
Création du site web	90%	95%	100%
Bande annonce	10%	25%	100%
Mise en place multijoueur	50%	75%	100%
Responsables Communication	33%	66%	100%

TABLE 4 – Tableau de la planification des tâches selon les soutenances

5.1 Graphismes

Pour les modélisations j'ai rencontré quelques problèmes avec l'armature, certaines articulations de fonctionnait pas correctement. Aujourd'hui il manque encore la modélisation de Lasnya la tireuse, l'armature sur Tyo

pour faire des articulations. D'ici la prochaine soutenance tous les combattants seront modélisés, certains personnages/autel auront des textures et nous auront choisis les assets pour les mobs.

5.2 Animations

D'ici à la prochaine soutenance, mon but serait de finir les animations des mobs, les animations d'attaques d'au moins 3 personnages, et enfin les animations générales comme par exemple marcher, sauter, parler, etc ...

5.3 Sons

Le problème est que les sons que nous avons trouvé ne nous conviennent pas parfaitement, nous aimerions si possible créer ces sons afin qu'ils soient parfaitement intégré dans le jeu. Mais pour le moment, notre objectif pour la prochaine soutenance sera d'implémenter les sons dans le jeu. Il est alors probable que la création des sons voulu sera pour la troisième soutenance.

5.4 Menus

Pour la prochaine soutenance, la quasi totalité des menus seront intégrés dans le jeu. A savoir, le menu de sélection du personnage, ainsi que le menu de fin de partie. On peut aussi imaginer que les menus changeront un peu pour soit apporter des nouvelles fonctionnalités, ou alors rendre le jeu plus esthétique.

5.5 Réseau

Par ailleurs, Photon View est avantageux, mais pas suffisamment. En effet, les mouvements des personnages sont saccadés et mal synchronisés. Donc, les étapes restantes à faire porteront d'autre part sur :

- La synchronisation des déplacements et de la rotation du Player
- La synchronisation des animations (PhotonAnimatorView)
- La synchronisation de la physique (PhotonRigidbodyView)

Finalement en fin de la partie réseau il on va implémenter des fonctionnalités comme :

- La gestion des SpawnPoints
- La liste des scores des joueurs
- La gestion des particules

5.6 I.A.

Nos objectifs à accomplir à terme de la seconde période sont de donner comme cible à notre IA le joueur le plus proche de sa position ou de lui faire calculer sa position par rapport à l'ensemble des joueurs de manière à l'adapter au mode multijoueur et de pouvoir commencer les tests sur ce mode et dans un second temps d'améliorer la gestion de son environnement. Nous allons également munir notre IA d'animations, de modèle 3D, ainsi que d'attributs lui permettant d'attaquer soit un joueur soit l'autel.

6 Conclusion

Le début de projet nous a permis d'apprendre beaucoup de choses allant de la gestion globale de notre projet, aux compétences spécifiques développées par chacun. La cohésion au sein du groupe étant bonne, nous avons bien pu démarrer le développement du jeu. Cette première partie nous a permis de poser les bases de notre jeu. Nous avons pris beaucoup de plaisir durant cette première partie. Nos compétences vont nous permettre de créer un jeu amusant et attractif. Cependant nous pensons objectivement que le plus dur est encore à venir, car si nous voulons arriver à un résultat satisfaisant nous allons devoir redoubler d'efforts.