

# Rapport de Soutenance 2

## - Bagarena -

### O.R.E.P.A



Mathis BRAHIM  
Arthus DAIMEZ  
Marine ROY  
Léo WEBER

29 Avril 2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Conception</b>	<b>4</b>
2.1	Graphismes . . . . .	4
2.1.1	Personnages . . . . .	4
2.1.2	Map . . . . .	9
2.2	Animations . . . . .	15
2.3	Sons . . . . .	15
2.4	Bande d'annonce . . . . .	16
2.5	Menus et Interface . . . . .	17
2.6	Réseau . . . . .	23
2.7	I.A. . . . .	23
2.8	Mécaniques de jeu . . . . .	25
2.9	Site Web . . . . .	25
<b>3</b>	<b>Avances et retards</b>	<b>26</b>
3.1	Les réussites . . . . .	26
3.2	Les retards . . . . .	27
<b>4</b>	<b>Prévisions</b>	<b>27</b>
4.1	Graphismes . . . . .	27
4.2	Animations . . . . .	27
4.3	Sons . . . . .	28
4.4	Menus . . . . .	28
4.5	Réseau . . . . .	28
4.6	I.A. . . . .	28
<b>5</b>	<b>Conclusion</b>	<b>28</b>

# 1 Introduction

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'avons 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 deuxième partie de développement a représenté une charge de travail bien plus conséquente, en effet, la première partie s'était avérée pour une grande partie être de l'apprentissage et de la recherche, elle avait cependant pour but d'acquérir quelque peu des bases sur les différents logiciels et manière de développer. Maintenant que les logiciels ont été globalement pris en main nous avons vraiment pu nous concentrer sur le développement du jeu plutôt que sur la compréhension du fonctionnement des outils. Le jeu a donc vraiment commencé à prendre forme lors de cette période.

Ainsi les éléments précédents ont pu être mis en commun, et améliorés afin d'avoir un début de jeu fonctionnel. On commence donc à imaginer de plus en plus à quoi notre jeu va ressembler en version finale.

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 – Répartition des tâches pour la soutenance 2

**R** : Responsable de la tâche

**S** : Suppléant

## 2 Conception

### 2.1 Graphismes

#### 2.1.1 Personnages

(Marine)

Comme depuis le début du projet je suis responsable des modélisations 3D du projet. Lors de la dernière soutenance j'avais présenté les modélisations de trois combattants que le joueur peut choisir et aussi l'autel que les joueurs vont devoir défendre ou attaquer. Cependant les choses ne se sont pas passées comme prévue, en effet lors de l'exportation des personnages dans Unity, les modélisations étaient corrompues, elles apparaissaient avec des trous. J'ai donc du retravailler mes modélisations pour que l'exportation se passe correctement. J'en ai aussi profité pour les rendre plus esthétiques et plus jolies et faire quelques modifications. Après avoir régler ce problème j'ai pu commencer le dernier combattant qu'il me restait à faire, "Lasnya" il s'agit d'une combattante avec un arc qui pourra attaquer à distance. Comme pour les autres personnages, j'ai du rajouter un squelette à la modélisation pour que mon camarade Arthus puisse l'animer au mieux.

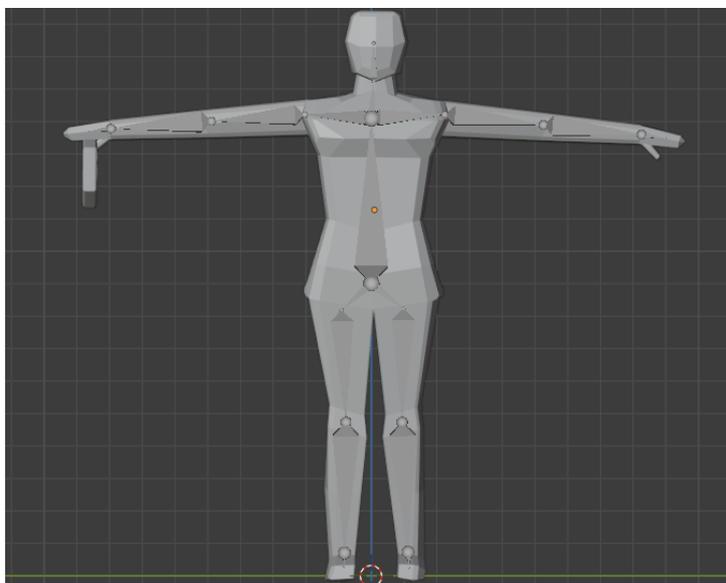


FIGURE 1 – Le combattant Lasnya avec son arc

Pour le mode Solo (défense d'autel) nous avons décider de rajouter un marchand auquel le joueur pourra acheter des bonus tels qu'une amélioration

de la vitesse ou des bonus d'attaques ou de vie durant sa partie, il sera positionné à côté de l'autel. J'ai du donc modéliser ce marchand en ajoutant aussi un squelette parce qu'il sera légèrement animé aussi.

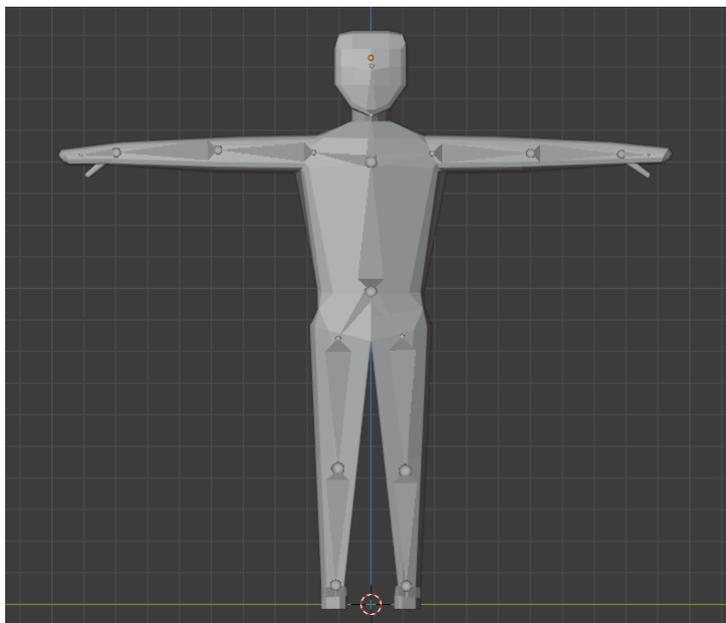


FIGURE 2 – Le marchand

Après avoir finalisé toutes mes modélisations je me suis attaquée aux textures et encore une fois j'ai du faire face à quelques problèmes. J'ai appliqué des couleurs et des textures aux autels. Il me fallait trois autels de couleurs différentes, un vert pour le mode solo, un rouge pour l'équipe rouge en mode multijoueur et un bleu pour l'équipe bleu en mode multijoueur. Après avoir appliqué les couleurs et les textures sur mes autels je les exporte en .FBX pour pouvoir les utiliser sur Unity. Malheureusement les choses n'étaient pas aussi simple, lors de l'exportation sur Unity les couleurs et les textures n'apparaissent pas. Pour pouvoir régler ce problème j'ai fais le tour des tutos disponibles sur internet et j'ai fais appel a des amis qui s'y connaissent en modélisation et en texture, et après tout fonctionnait comme prévu, j'ai pu tout exporter correctement.



FIGURE 3 – Lasnya (à gauche) et le marchand (à droite) avec texture dans Unity



FIGURE 4 – Niaka (à gauche) et Tyo (à droite) avec texture dans Unity



FIGURE 5 – Saadel avec texture dans Unity

Pour les autels, on a fait d'une manière différente, en effet lorsque j'exporte un autel avec ses couleurs et ses textures celle-ci apparaissent lisses alors que les pierres sont censés être brillantes. Nous avons du donc retravailler les textures des autels directement sur Unity pour avoir un résultat plus satisfaisant.



FIGURE 6 – L'ancien autel avec des textures lisse

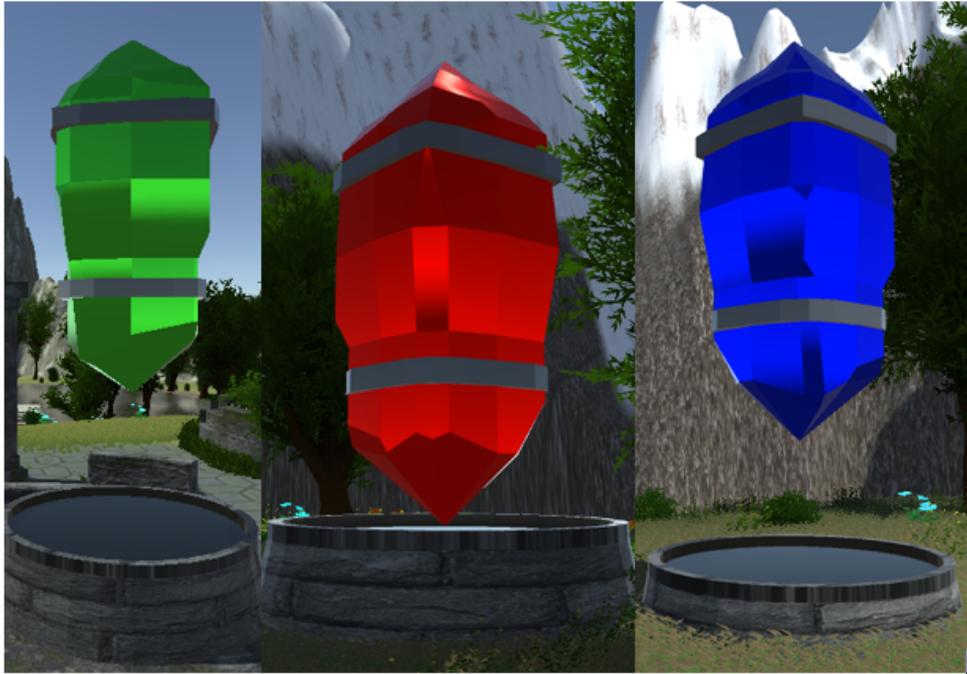


FIGURE 7 – Les autels avec différentes couleurs sur Unity

Et enfin pour finir, nous avons choisi d'utiliser des modèles 3D de l'asset store de Unity pour les ennemis qui vont attaquer le joueur ou attaquer l'autel. Il y a deux types de monstres, les "Toon RTS Units - Undead Demo" qui représentent les ennemis les plus faibles, et les "Toon RTS Units - Orcs Demo" qui sont des ennemis plus puissants.



FIGURE 8 – Toon RTS Units - Undead Demo



FIGURE 9 – Toon RTS Units - Orcs Demo

### 2.1.2 Map

(Arthus)

Pour ce qui est des cartes de jeu, lors de la dernière soutenance la map du solo était quasiment finie, enfin c'est ce que je pensais. Effectivement, sur

celle-ci, j'ai bien vérifié que tout les objets aient un collider et donc qu'on ne puisse pas passer au travers, et j'ai ajouté tout un décor pour un marchand proche de l'autel.



FIGURE 10 – Ancienne zone de marchand



FIGURE 11 – Nouvelle zone de marchand



FIGURE 12 – Intérieur de la nouvelle zone de marchand

De plus, 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 le présente l'image ci-dessous, cela ne rendait pas bien du tout.



FIGURE 13 – Pente faites à la main



FIGURE 14 – Pente refaites vu du dessus

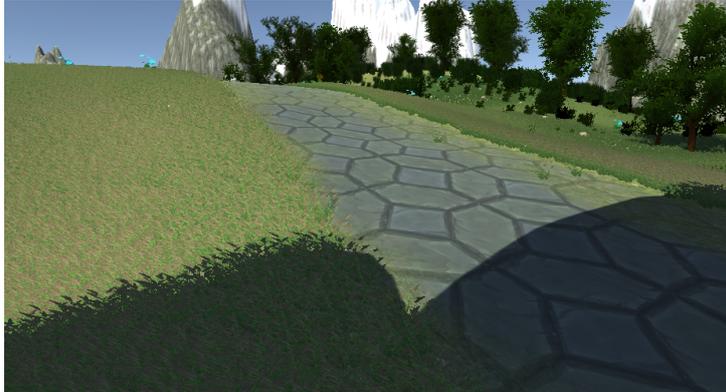


FIGURE 15 – Pente refaites vu de coté

Une nouvelle map est aussi disponible ! Je me suis occupé de faire la map du 3vs3, c'est une map plus en long que la première pour que les joueurs doivent faire du chemin avant d'attaquer l'autel adverse. 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.



FIGURE 16 – Deuxième map



FIGURE 17 – La crevasse et les montagnes

Au tout début 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 18 – Début de partie

Il y a un point commun à ces deux maps, le fait que pour un souci de place encore une fois, j'ai dû "épurer" les assets utilisés. De manière plus précise, j'ai retiré chaque fichier non utilisé car cela prenait beaucoup de place pour rien. Néanmoins, ce n'est pas une tâche facile et rapide ! Cela m'a pris plusieurs heures, car nous devons nous 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 corrompu.

## 2.2 Animations

(Arthus)

Étant donné que je suis le responsable des animations, je suis celui qui s'est le plus penché sur ce domaine. Comme depuis la soutenance dernière ma collègue Marine a fini de modéliser les autels, j'ai pu leur créer des animations, deux plus précisément. Une qui tourne en boucle, qui fait "vivre" l'autel à l'aide d'un changement de hauteur et d'une rotation de 360°, et une autre lorsque l'autel est détruit, dans laquelle le cristal tombe dans le néant qui se trouve dans le socle en dessous de lui.

Pour ces animations, j'ai eu quelques difficultés comme par exemple le fait que : j'ai fait ces animations une première fois sur Blender, mais elles s'exportaient très mal sur Unity. Du coup je les ai refaites sur l'animateur intégré dans Unity.

Plus tard, Marine m'ayant fourni le modèle du marchand, je lui ai fait une animation de respiration pour qu'il soit vivant et non statique. Ici, le problème s'est inversé, j'ai voulu créer cette animation sur Unity, mais cela ne prenait pas en compte les os du marchand. Je suis ainsi retourné sur l'outil d'animation de Blender, et ensuite exporter via le format FBX sur Unity. Suite à cela, j'ai créé un "Animator Controller" (qui gère l'ordre de lancement des animations) que j'ai rattaché au modèle et ainsi lui donner vie via une animation simulant une respiration !

Maintenant, la partie un peu triste de l'histoire, les animations des personnages... Comme Marine l'a précisée précédemment, nous avons dû refaire les modèles des personnages, ainsi les quelques animations que j'avais faites plus tôt sur Blender ont été supprimées. Ainsi, je dois reprendre depuis le début les animations des héros...

## 2.3 Sons

(Léo)

Depuis la dernière soutenance, il y a eu beaucoup d'avancées concernant le son. Là où je n'avais choisi que les sons que j'allai utiliser pour notre jeu, j'ai maintenant implémenté ces sons dans certaines parties du jeu.

Pour commencer, j'ai dû regarder des vidéos pour voir comment fonctionne les implémentations de son sur Unity. Il s'avère que c'est relativement simple, il m'a donc été aisé de faire le son de tous les boutons du menu dans un premier temps. Pour ce qui est des boutons, il y a un son différent selon si l'on passe sur le bouton, ou si l'on clique dessus. Pour l'instant, les deux sons sont identiques, mais il est possible de les changer très facilement.

Après le son des boutons, il m'a semblé judicieux d'ajouter une musique d'ambiance pour les menus. Un problème est vite survenu, si on met simplement un Audio Source sur le menu principal, la musique s'arrête lorsque l'on change de scène. Pour pallier ce problème, il est nécessaire de faire un script qui permet de ne pas détruire l'objet qui contient le son bien que la scène change.

Enfin, lorsque nous lançons le jeu, il faut arrêter la musique du menu et en lancer une autre qui correspond mieux à l'ambiance du jeu en elle-même. Pour cela, sachant que la caméra utilisée pour suivre le personnage et celle de l'animation de mort et de fin ne sont pas les mêmes, j'ai dû créer un autre objet qui ne se détruira qu'à la fin de la partie pour que le son ne soit pas coupé quel que soit le plan.

Finalement, pour résumer ce qui a été fait, je peux avancer le fait que les sons du menu sont finis, et que le début des sons pour le mode solo a été entamé.

## **2.4 Bande d'annonce**

(Léo)

En ce qui concerne la bande d'annonces, il y a encore peu de matière pour faire des plans pour la vidéo, mais un début a été fait. Le logiciel de montage a été choisi et j'ai pu expérimenter quelque temps celui-ci afin d'être à l'aise lors de la création du produit final.

Des idées ont déjà été trouvées pour le choix des plans, l'ambiance qui va être installée et la présentation du jeu de manière générale.

## 2.5 Menus et Interface

(Mathis)

Pour rappel, lors de la première soutenance, nous avions déjà le menu principal, le menu d'options (avec la possibilité de régler le volume, choisir une résolution, et choisir si le jeu doit être en plein écran ou non), le menu multijoueur (contenant le choix du pseudo, le menu de liste de rooms, et celui de création de room), ainsi qu'un menu pause pour le mode solo.

Notre jeu ayant différents personnages possibles à choisir, je me suis donc concentré sur l'implémentation d'un menu de choix de personnage tout d'abord pour le mode solo. En effet, celui-ci n'avait pas les complications de synchronisation entre joueurs qu'il allait me falloir gérer dans les menus multijoueurs. Son implémentation n'a pas été très compliquée grâce à l'expérience que j'avais acquise pour la première soutenance dans la création de menu. Notez également que les images contenues dans les différents choix possibles ont vocation à être remplacés par les images de nos personnages lorsqu'ils seront totalement finis (pour l'instant, il s'agit d'image représentant leur attribut principal).

Voici le résultat du menu de choix de personnage du mode solo :

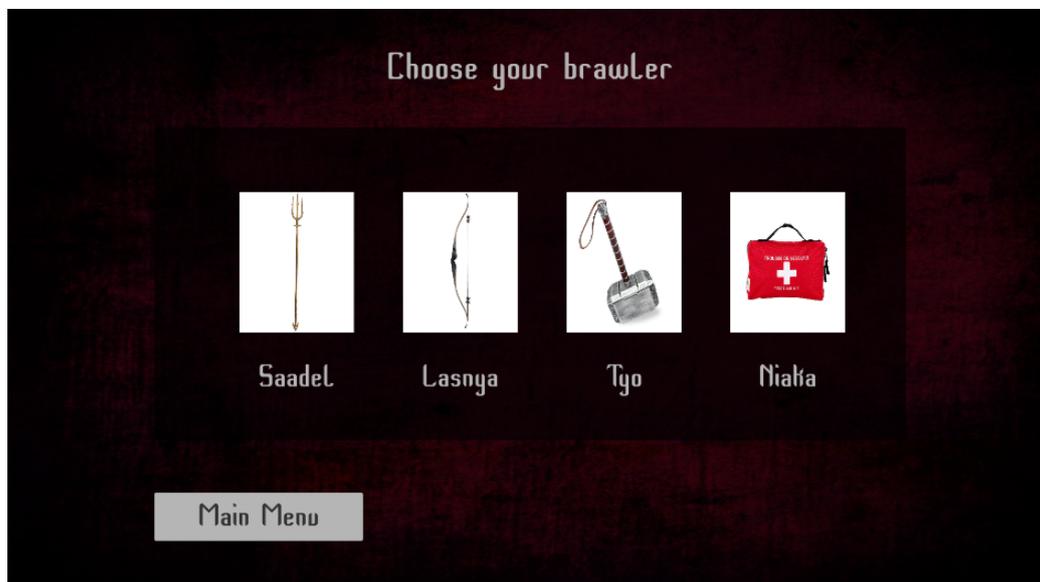


FIGURE 19 – Menu choix de personnage Solo

Première complication dans la conception des 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. Viens 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 automatiquement et donc que le contenu surchargé fasse l'action souhaitée. Ci-dessous, un exemple du menu avec deux joueurs ayant décidé d'aller dans deux équipes différentes.

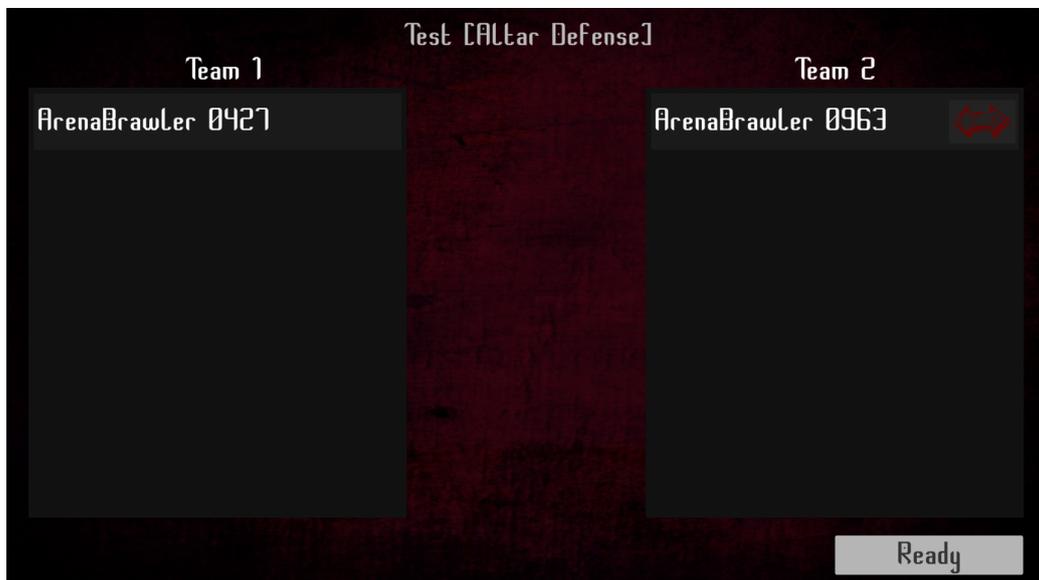


FIGURE 20 – Menu choix d'équipe

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 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 qu'a choisi 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).

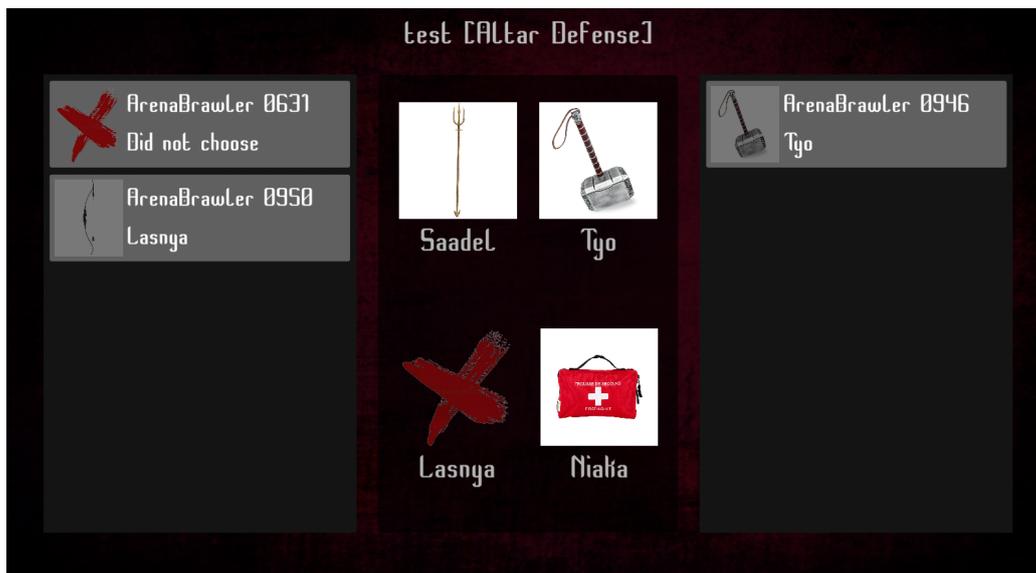


FIGURE 21 – Menu choix de combattant pour le mode "Altar Defense"

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.

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

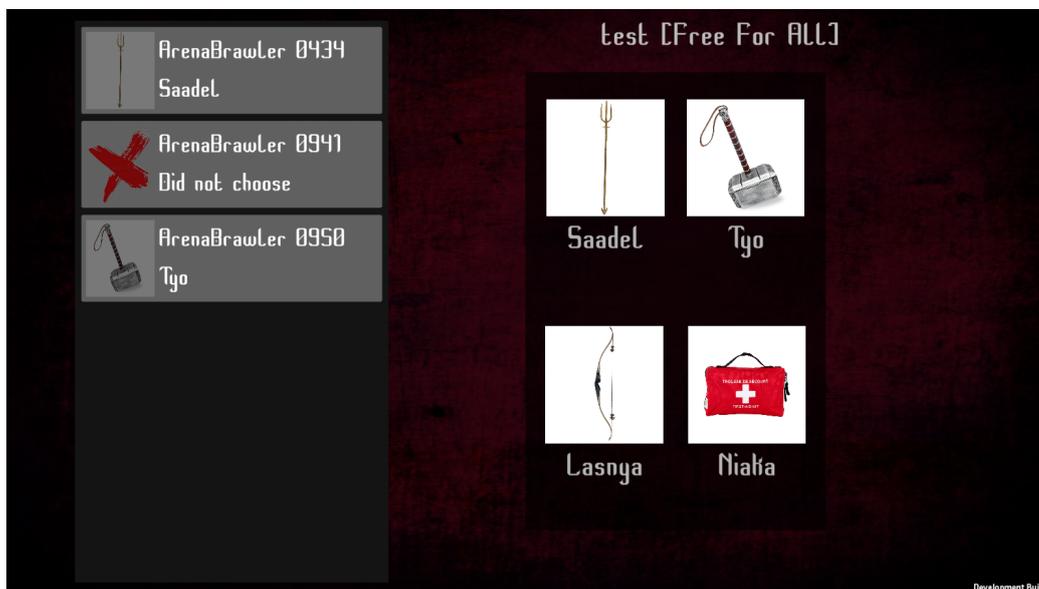


FIGURE 22 – Menu choix de combattant pour le mode "FFA"

Pour le menu pause du mode solo voici une capture d'écran :



FIGURE 23 – Menu pause du mode solo

Et ci-dessous une capture d'écran de l'écran de fin de partie du mode solo :



FIGURE 24 – Écran de fin de partie en mode "Solo"

Et enfin, viens la création de l'interface d'un joueur pour le mode solo. Il a accès à la vie de l'autel en haut à gauche (au sein d'un slider) mais également d'un texte plus précis. Il a également accès à sa barre de vie en bas et le détail. Il peut savoir combien de temps sa partie dure, à quelle vague il en est et le temps qu'il reste avant la prochaine vague. De plus, le combattant ayant deux attributs représentés sur l'interface au-dessus de sa barre de vie par deux images, celle-ci deviennent noires lorsqu'il vient d'utiliser son pouvoir, et ont une petite animation d'apparition circulaire du pouvoir, symbolisant la durée restante avant de pouvoir réutiliser ce même pouvoir. Ci-dessous, une capture d'écran depuis une partie



FIGURE 25 – Interface joueur en mode "solo"

Les points de vie figurant sur cette capture d'écran différeront sûrement pour la version finale du jeu. De plus, le combattant n'a pas encore ses textures, car comme dis plus haut dans la partie "modèle 3D", nous avons rencontrés des difficultés avec l'exportation de modèle Blender.

Et enfin, nous avons décidé de changer le curseur de souris pour lui donner un aspect qui correspondait plus à l'univers de notre jeu. J'ai donc cherché des curseurs style RPG, malheureusement il n'y avait pas énormément de choix, et surtout, ils étaient en majorité payants, nous avons donc choisi d'utiliser le "Basic RPG Cursors" de l'asset store de Unity.



FIGURE 26 – Curseur choisi

## 2.6 Réseau

(Mathis)

Pour la partie réseau, nous avons choisi d'utiliser l'asset "Photon Unity Network"(PUN). La partie réseau n'a pas beaucoup évolué depuis la dernière soutenance, j'ai choisi de me concentrer d'abord sur la partie solo, car j'avais rempli pleinement mes objectifs d'avancement.

Pour la partie réseau le joueur peut donc actuellement retourner au Hall depuis une room (salle de partie), lister les joueurs d'une room, lister les salles depuis un hall, connaître le nombre de joueurs présents dans les différentes rooms, et donc pour cette deuxième soutenance, j'ai uniquement rajouté la gestion des équipes, et la gestion des choix de personnages grâce aux Player-CustomProperties de Photon. Celle-ci me servira lors de la gestion d'une partie pour pouvoir instancier les joueurs avec le bon prefab du combattant qu'ils ont choisi ainsi qu'au bon endroit en fonction de leur équipe.

## 2.7 I.A.

(Mathis)

Pour le développement de l'IA, j'ai commencé par l'IA de notre mode Solo, pour rappel le mode Solo se déroule en système de vagues. J'ai donc dû implémenter un système de vague en même temps que de développer l'IA. L'IA est basée sur un système de navmeshagent, qui permet grâce à un fichier navmesh calculé automatiquement sur la map, d'avoir les différents chemins empruntables par le navmeshagent. Il va calculer automatiquement

le meilleur chemin pour aller à une destination qu'on lui définit dans un script attaché au gameobject contenant le composant NavmeshAgent.

L'IA est actuellement capable de :

- Suivre le joueur.
- Stopper sa course lorsqu'il est à une distance suffisante pour attaquer le joueur ou l'autel (en fonction de sa destination).
- Choisir sa direction entre l'autel ou le joueur (en fonction du plus proche).
- Attaquer un joueur, ou l'Autel.

L'IA est muni de points de vie, de dégâts d'attaque, d'un cooldown (temps de "rechargement" entre deux attaques). Pour l'instant, ses valeurs sont temporaires, nous les définirons par la suite lorsque nous aurons créé tous les joueurs et leurs attributs pour équilibrer le jeu. Chaque bot aura au-dessus de lui un slider représentant sa vie, ce slider sera toujours face à la caméra peu importe l'angle de vue, il y a un script attaché à ce GameObject.

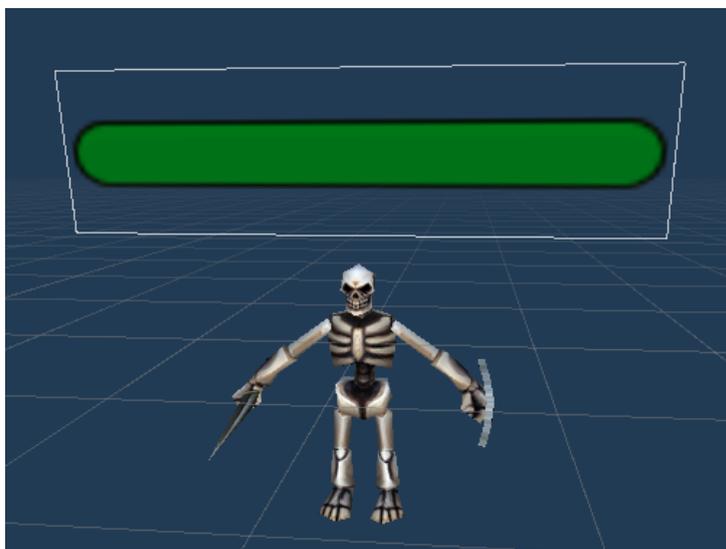


FIGURE 27 – Prefab de l'IA

Pour l'instant, le système de vague n'est pas très sophistiqué, mais il fonctionne, à chaque vague le nombre de bot à faire apparaître augmente de 1, et le script WaveSpawner.cs qui s'occupe de la gestion des vagues, va attendre un certain temps avant de refaire apparaître une vague d'ennemis. Il y a trois points d'apparitions.

## 2.8 Mécaniques de jeu

(Mathis)

Le jeu possède actuellement les mécaniques suivantes :

- Mouvements du joueur
- La caméra peut suivre le joueur
- Collisions entre le joueur, les IA, ainsi que l'Autel
- Attaquer un bot
- Temps de rechargement pour chaque attaque
- Mise en surbrillance d'un ennemi lors du passage de la souris dessus

## 2.9 Site Web

(Mathis)

Le site web n'a pas changé durant cette période, en effet, il était déjà quasiment terminé. Il a juste été mis à jour avec les derniers tweets, ainsi que la disponibilité du rapport de la première soutenance. Il est disponible à l'adresse suivante <https://skraam308.github.io/BAGARENA-website> (hébergé par GitHub Pages gratuitement).

### 3 Avances et retards

Tâche	Prévu	Actuel
Edition 3D	60%	100%
Animation	50%	30%
Conception de l'environnement	50%	85%
Physique du jeu	50%	30%
I.A	60%	60%
Musiques et bruitages	60%	60%
Création du site web	95%	95%
Bande annonce	25%	25%
Mise en place multijoueur	75%	60%
Responsables Communication	66%	66%

TABLE 2 – Tableau des avances et retards

#### 3.1 Les réussites

Les modélisations sont toutes terminées et elles sont toutes exploitables sur Unity nous avons aussi ajouté tous les assets trouvés sur l'asset store de Unity pour les ennemis.

Les maps promises sont bientôt toutes jouables avec un design agréable!

Les mécaniques du mode solo sont dans l'ensemble terminées, elles ont le fonctionnement attendues, le mode de jeu est jouable sans avoir les animations d'implémentées.

Les menus ont bien avancé, ils sont à 90% faits. Ils ont le design que nous souhaitions et fonctionnent correctement.

Les IA se comportent globalement comme nous le souhaitions, les navmesh agent de Unity facilitent la tâche pour les implémenter. Elles ont les capacités d'attaquer comme nous le voulions.

## 3.2 Les retards

Pour la physique du jeu nous avons pris du retard, en effet, nous manquons encore d'éléments ayant une physique particulière à modifier. Jusqu'à présent, les éléments de Unity se suffisaient à eux-mêmes pour avoir une physique relativement correcte.

Nous sommes légèrement en retard sur la partie réseau, hormis les menus et un script d'instanciation de prefab sur une scène, aucune autre chose n'a été faite pour la gestion d'une partie multijoueur.

La partie animation aussi a encaissé un peu de retard, dont les animations basiques, telles que marcher, et les attaques de chacun des personnages.

## 4 Prévisions

Tâches	1ère Soutenance	2ème Soutenance	3ème Soutenance
Édition 3D	40%	100%	100%
Animation	25%	30%	100%
Conception de l'environnement	40%	85%	100%
Physique du jeu	0%	30%	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%	60%	100%
Responsables Communication	33%	66%	100%

TABLE 3 – Tableau de la planification des tâches selon les soutenances

### 4.1 Graphismes

Pour les modélisations 3D des personnages nous avons rien à prévoir, car elles sont toutes finies, néanmoins, il nous reste la dernière carte à créer, pour ce faire nous allons reprendre la map du Solo et la modifier un petit peu.

### 4.2 Animations

Pour les animations, suite au retard causé par les bugs, nous allons concentrer notre temps sur celles-ci pour en faire, et qu'elles soient de qualité.

### 4.3 Sons

Pour la dernière soutenance, nous pouvons dire que tous les sons que nous voulions implémenter le seront, entre les bruits d'attaque, les musiques d'ambiance et de fin de parties, rien ne sera oublié et correspondra au mieux à l'univers de notre jeu.

### 4.4 Menus

Les menus ont très bien avancé. Pour la soutenance finale, il ne reste plus qu'à créer les menus :

- Fin de partie multijoueur
- Menus pause pour les modes multijoueurs

### 4.5 Réseau

Pour la partie réseau les objectifs sont :

- Instancier le bon personnage choisi
- Gestion des points d'apparitions
- Synchronisation des déplacements
- Synchronisation des animations
- Le leaderboard de fin de manche
- La mise en place du son en multijoueur

### 4.6 I.A.

Il va falloir créer un deuxième prefab d'IA pour le mode solo, pour avoir deux types d'ennemis possibles (l'un plus fort que l'autre). Nous allons devoir la modifier de sorte à ce qu'elle puisse être utilisée en partie multijoueur et donc manipuler deux attaques possibles au lieu d'une seule. Et nous modifierons également un peu le système de vagues ennemies du solo.

## 5 Conclusion

Même si de nombreuses difficultés ont été rencontrées, et que nous avons pris du retard dans certaines tâches, nous avons, durant cette deuxième session, réalisé un grand bond en avant. En effet, que ce soit au niveau de la jouabilité ou simplement de l'aspect visuel, la version finale du jeu approche à grand pas et les objectifs seront, sans aucun doute, atteints pour la dernière soutenance. Nous nous investissons autant qu'il le faut pour mener à terme ce projet.