

PROJET S2

Nacho's Rumble

Rapport soutenance intermédiaire



Los Successos

PAUL BERTHELOT (CHEF DE PROJET)

THÉO CHARBONNIER

KILIAN HARDY

SAM ESBER

AVRIL 2022

Table des matières

1	Introduction	2
1.1	Etat actuel du projet	3
1.1.1	Historique Git	4
2	Avancement des tâches	5
2.1	Intelligence Artificielle	5
2.2	Modélisation	7
2.2.1	Map	7
2.2.2	Modèle des joueurs	8
2.2.3	Animations	8
2.2.4	Assets	8
2.3	Sons et bruitages	8
2.4	Menus	9
2.5	Données des joueurs	9
2.6	Problèmes liées à cette implémentations	11
2.7	Gestion de la partie	11
2.7.1	Statistiques de jeu (morts et éliminations)	11
2.7.2	Fin de partie	12
2.7.3	Score Board	12
2.8	ATH	13
2.9	Site internet et comptes	14
2.10	Système d'armes et de tir	14
3	Avance/retard global	16
4	Objectifs pour la prochaine soutenance	17
4.1	Mécaniques du jeu	17
4.2	Intelligence Artificielle	17
4.3	Graphismes	17
4.4	Site	17
5	Conclusion	19

1 Introduction

Notre groupe, toujours composée de Paul, Sam, Kilian et Théo, a le plaisir de continuer le développement de son projet : le jeu *Nachos Rumble*!

La création de notre jeu de tir à la première personne est toujours un projet qui nous tient à coeur, et dans lequel nous nous investissons du mieux que nous pouvons. Depuis la première phase de développement ayant mené à la première soutenance, nous avons concentré notre développement sur des tâches différentes que nous allons vous présenter dans ce rapport.

Le but du travail que nous avons mené jusqu'à la soutenance intermédiaire n'est plus le même que celui que nous avons en vue pour la première soutenance, il ne s'agit en effet plus d'implémenter le fonctionnement et les principes de base du jeu, mais plutôt de rendre le jeu jouable, d'ajouter des règles aux parties que l'on peut jouer, avancer sur la création de l'intelligence artificielle, approfondir les techniques présentes dans le jeu.

1.1 Etat actuel du projet

Le but du travail à faire pour cette première soutenance était de faire clairement un jeu jouable, avec ses axes de développement. Nous nous sommes focalisés sur la mise en place de compteurs, et statistiques pour pouvoir désigner un vainqueur. Nous avons plusieurs idées, équipe contre équipe, capture the flag ect, mais nous avons décidé de faire un Deathmatch, ou le but est de tuer le plus d'adversaire en solo. Dans l'état actuel du jeu, en plus d'avoir des mécaniques de déplacements et un système de tir, nous pouvons finir une partie en la gagnant ou en la perdant. Nous avons également créé un système de noms et de compte. En résumé : Après avoir implanter les bases du jeu lors de la première soutenance, nous avons ajouté le mode de jeu, et son système de victoire ou de défaite. déroulement de ce projet, Grâce au travail fait jusqu'à maintenant, nous sommes prêt et motivé à engager une phase de travail plus affinée, sur l'aspect du jeu, sur ses fonctionnalités plus particulières.

1.1.1 Historique Git

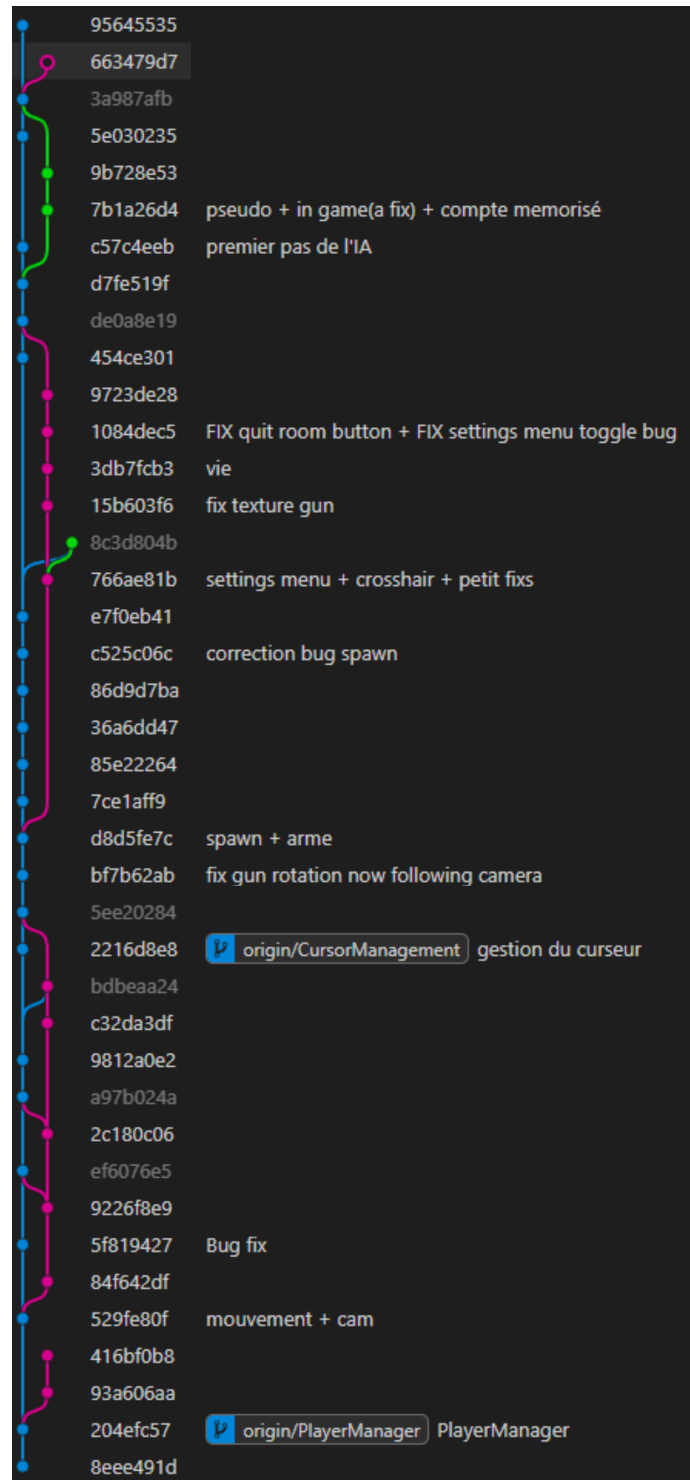


Figure 1: Historique des commits git du projet

2 Avancement des tâches

2.1 Intelligence Artificielle

Paul a commencé à implémenter l'IA sur la base des recherches de Théo. Pour rappel, le but de l'IA que nous voulions créer dans le jeu est de jouer le rôle d'un ennemi en le cherchant, puis en lui tirant dessus à une certaine distance. Nous avons donc dû trouver un moyen de communiquer à l'IA à la fois, où elle peut se déplacer, et dans quelle destination elle doit se rendre.

Pour cela, nous avons eu recours au 'NavMesh' d'Unity. Il s'agit d'une mesh permettant à l'IA de dicter où elle peut se diriger.

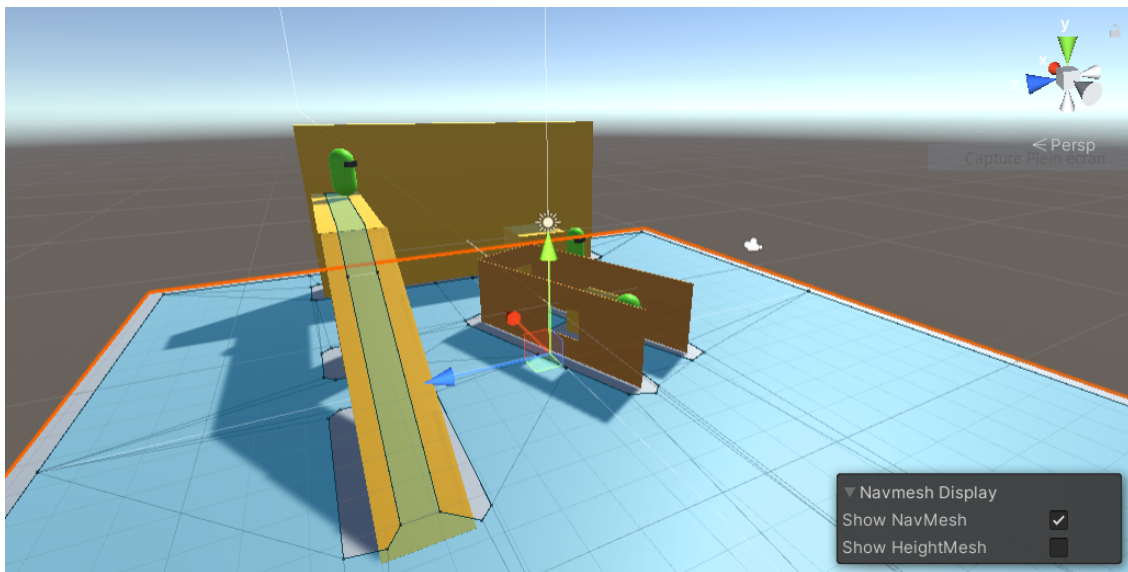


Figure 2: Première création de la NavMesh.

Avec cela, nous avons aussi utilisé un algorithme de PathFinding pour permettre à l'IA de savoir dans quelle direction se diriger pour se déplacer.

Le principe actuel, que nous pensons sûrement améliorer pour la soutenance finale, consiste à déterminer pour chaque IA une cible (un joueur) en fonction de la distance qui les sépare. Cette recherche de cible n'ayant pas besoin d'être faite à chaque fois qu'une image est générée, elle est gérée dans une 'Coroutine' avec un intervalle de temps donné, qui est pour le moment de 0,1 secondes, pour rendre le déplacement de l'IA assez fluide.

La gestion du calcul de la distance nous est apparue simple, avec un calcul de distance euclidienne basique, mais plusieurs cas sont en fait à gérer. Notre map possédant plusieurs niveaux, une IA se trouvant à un étage en dessous d'un joueur se considérerait proche de celui-ci, et arrêterait donc de suivre un autre joueur étant sur le même niveau, pourtant juste en face.

Après avoir créé les principes de base de l'IA tels que décrits plus haut, il nous manquait

l'implémentation sur les serveurs multijoueur Photon, que nous utilisons pour notre jeu. Nous avons rencontré plusieurs problèmes quant à la synchronisation des IA entre les clients, mais aussi quant à leur capacité à prendre des dégâts, et donc mourir et réapparaître. En effet, la manière d'implémenter notre système de tir entre joueurs qui consiste à appeler des méthodes RPC (Remote Procedure Calls) sur tout les joueurs dans la partie, ne marche pas sur les IA, qui ne sont pas des joueurs en soi.

Nous avons donc réorganisé la gestion des IA dans la partie, pour qu'elles soient en fait instanciées par le MasterClient, c'est-à-dire, le joueur ayant les droits sur la partie. Nous avons donc dû gérer le fait que le Master puisse se déconnecter ou quitter la partie, et pour cela, il a fallu transmettre la propriété des IA au nouveau Master, pour qu'elles restent dans la partie avec les joueurs.

Cette partie du projet est encore en plein développement, l'implémentation de l'IA en multijoueur produisant de multiples bugs que nous fixons au fil du temps.

2.2 Modélisation

2.2.1 Map

Pour donner suite à la première soutenance, nous avons décidé de ne pas trop avancer les différents modèle 3D des joueurs, cependant Nous avons tout de même fait le choix de faire une carte viable, quitte à ce qu'elle ne soit pas magnifique, pour l'améliorer en esthétique par la suite.

Pour faire cette carte, Sam s'est inspiré de nombreuses cartes disponibles sur internet, mais il à surtout repris les designs que nous avons conçu. Les principaux murs, et délimitation étaient déjà faite mais on fait le choix de rajouter des ponts, ce qui permet de donner à la carte l'avantages d'avoir plusieurs étages, ce qui rajoute de nombreuses possibilités, car en plus de devoir faire attention à ce qu'il l'entoure, le joueur devra regarder au-dessus ou en dessous de lui. Nous avons également étudié la carte de manière à avoir de nombreuses lignes de tir disponible, des endroits à couverts, et d'autres plus exposés, ce qui permet au joueurs et à l'intelligence artificielle de pouvoir surprendre n'importe quel autre personne.

En plus d'avoir rajouter ses ponts, Il a été fait quelques modifications minimales (ajout de fenêtres, rajout de quelques murs) Mais dans l'ensemble, la carte reste la même. J'ai également mis des cactus pour faire esprit Mexique, les couleurs du sol et des murs sont également présents dans cette optique.

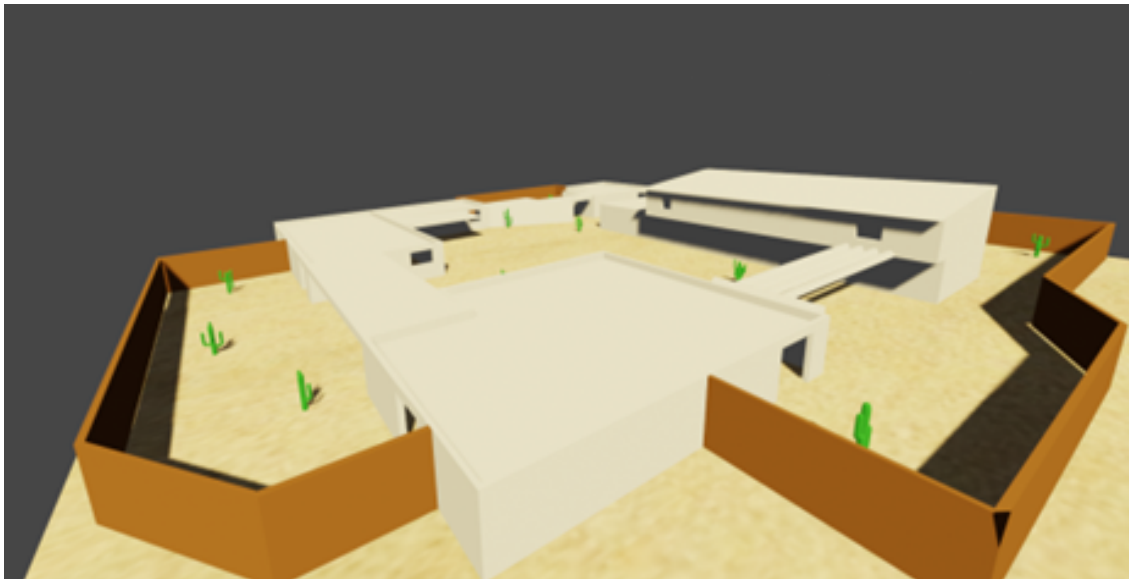


Figure 3: Map du jeu à l'état actuel.

2.2.2 Modèle des joueurs

Au niveau du playermodel, Sam à également fait une simple maquette de ce à quoi pourrais ressembler pour la troisième soutenance, cependant on hésite encore sur le personnage, pourquoi pas mettre quelque chose qui a une forme très lisse et basique qui ferait vraiment Cartoon, et low poly. Mais peut importe le style exact, nous comptons mettre des signes distinctifs pour faire penser à ce côté hispanique (Sombrero, moustache, poncho...) Nous n'avons pas fait ni implémenté de modèle 3D pour les joueurs pour le moment, car nous pensons le faire pour la fin du projet.

2.2.3 Animations

Les animations sont assez complexes à implanter étant donné que nous n'avons pas encore de playermodel choisi, mais nous avons tout de même lister les différentes animations possibles (saut, marcher, courir ect...) . Nous n'avons toujours pas d'animations disponibles en jeu, mais Sam c'est bien renseigner sur le sujet, et c'est la première chose que nous implémenterons une fois que le playermodel sera fixe, et que le jeu sera fonctionnel.

2.2.4 Assets

Au niveau des assets en général, il nous reste à implémenter un couteau, que nous envisageons de remplacer par un maracas pour l'esprit Mexique, cela rendrai notre jeu plus fun et moins dans l'esprit compétitif, et c'est ce que nous recherchons.

2.3 Sons et bruitages

Au niveau des bruitages, on a affiné ma sélection et nous avons maintenant un dossier avec des bruitages précis (coup de feu, bruit de pas...). Mais pour cette soutenance ce n'est pas la chose sur laquelle on à décider de passer le plus de temps, donc nous travaillons sur cet aspect et il est bientôt prévu d'avoir de nombreux bruitages disponibles en jeu.

Au niveau des musiques, Sam à lui moi-même composé une musique d'ambiance thème hispanique, que nous avons implémenter en jeu. Cette musique n'est que dans les menus, et malgré le fait que cela ne soit pas la choses la plus importante du projet (et de loin). Il est quand même agréable de ce faire recevoir dans les menus principaux par une musique dans le thème du jeu.

2.4 Menus

Il est évidemment que les menus du jeu sont encore très simple, c'est pourquoi nous sommes en train de créer de nouveaux boutons pour rendre le jeu plus accueillant qu'il ne l'est actuellement.

L'identité visuelle de notre jeu est en pleine confection par Paul et Sam, et les menus seront donc changés une fois le nouvel aspect visuel du projet choisi.

Cependant, des points restent à éclaircir, comme par exemple la police d'écriture de notre jeu, qui est encore un débat qui attise de grands conflits entre les membres du groupe, mais nous avons plusieurs candidats.

2.5 Données des joueurs

Pour notre jeu, nous voulions que chaque joueur de notre jeu ait un ensemble de données qui lui est propre, comprenant le nom d'utilisateur, son niveau, etc.

Pour gérer cela, on a commencé par créer un nouveau Input Field (champ de saisie) dans le menu principale qui permet aux joueurs de pouvoir avoir leur propre pseudo dans le jeu. Le nombre de caractères est pour l'instant limité à seize. Si, dans le cas où le joueur ne saisit pas de pseudo, un pseudo aléatoire lui sera attribué.

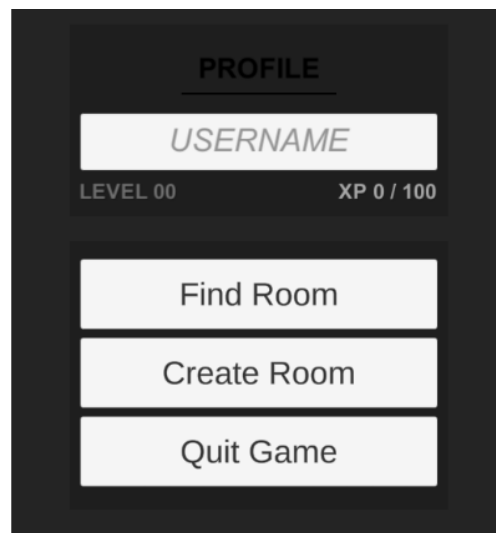


Figure 4: Champ de saisie et affichage du profil du joueur

Pour faciliter le développement du jeu, on est passé par la création d'une nouvelle classe Profil qui est associée à chaque joueur qui est dans une partie. La classe est pour l'instant constituée de trois variables : username, level et XP (l'expérience). Grâce à cette classe on a donc pu attribuer le pseudo saisi par l'utilisateur (présenté ci dessus), à chaque joueur distinct dans la partie en accédant simplement à la variable username dans ma classe.

Ensuite, pour faire en sorte que les utilisateurs puissent connaître "l'identité" de chaque joueur dans la partie on a implémenté un texte au-dessus de chaque personnage qui correspond à leurs pseudo respectifs.

```
public class ProfileData
{
    public string username; // Serializable
    public int level; // Serializable
    public int xp; // Serializable

    [2 usages] Kilian
    public ProfileData() //profile par default
    {
        this.username = "DEFAULT USERNAME";
        this.level = 0;
        this.xp = 0;
    }
}
```

Figure 5: Classe ProfileData de chaque joueur

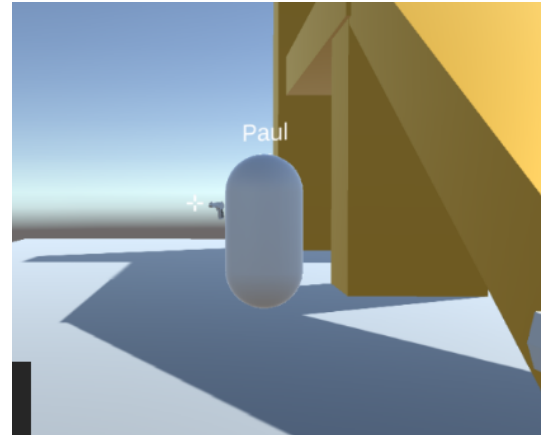


Figure 6: Affichage en jeu des noms d'utilisateurs

Pour finir sur cette partie, afin de faciliter la connexion de nos utilisateurs on a veillé à qu'il puisse rester connecté sur leurs profils même après avoir quitté le jeu. Pour permettre cela on a créé un script à partir de la classe Profil et de ses informations, qui récupère les datas d'un profil.

A chaque fois qu'une personne rejoint ou commence une partie, un fichier qui contient toutes les informations de son profil s'installe sur son ordinateur et permet donc de récupérer les renseignements dès que le jeu se relance. Un fichier est créé à chaque fois qu'un nouveau pseudo est saisi. Si un fichier existe déjà, il est donc supprimé et remplacé. Les sauvegardes de ces données sont stockées sur l'ordinateur de l'utilisateur dans un emplacement précis, dépendant de son système d'exploitation. Ce chemin est représenté en C par Application.persistentDataPath.

```
public static void SaveProfile(ProfileData t_profile)
{
    try
    {
        string path = Application.persistentDataPath + "/profile.dt";

        if (File.Exists(path))
        {
            File.Delete(path);
        }

        FileStream file = File.Create(path);

        BinaryFormatter bf = new BinaryFormatter();
        bf.Serialize(file, t_profile);
        file.Close();
    }
}
```

Figure 7: Enregistrement des données dans un fichier

Pour la manipulation des fichiers, Kilian s'est inspiré d'un des TP fait en cours d'informatique pratique.

2.6 Problèmes liées à cette implémentations

Nous avons rencontré plusieurs difficultés sur cette partie mais notamment sur la gestion des noms d'utilisateurs :

- lorsqu'il y avait plusieurs personnes dans la partie, les joueurs avaient tous le même pseudo.
- quand un nouveau joueur rejoignait une partie déjà en cours, il ne voyait pas les bons pseudos des autres utilisateurs (au dessus du personnage)
- également, dans le menus pour rejoindre une room où il y a la liste des joueurs, les pseudos n'étaient pas les bons.

Tous ces problèmes on était un peu compliqué à gérer mais au final on était résolu.

Le deuxième problème rencontré ici, c'est que nous nous sommes rendu compte à la suite du projet que la création de cette classe Profil n'était pas forcément la meilleure solution pour manipuler les propriétés des joueurs. Notamment quand Kilian s'est occupé du système pour compter les kills et morts, que nous allons présenter dans la prochaine partie.

Un autre problème va également intervenir quand il va falloir synchroniser les comptes créés sur notre site web avec ceux du jeu. C'est pour ça que nous allons devoir sûrement trouver une nouvelle méthode pour la suite du projet.

2.7 Gestion de la partie

Afin d'obtenir une fin à notre jeu, nous avons dû réfléchir par quel moyen une partie pouvait se terminer. Nous avons donc opté pour une fin de partie basée sur le nombre de kills. Autrement dit, le premier à un certain nombre de kills gagne et la partie s'arrête. Kilian s'est occupé de cette partie.

2.7.1 Statistiques de jeu (morts et éliminations)

La première étape a été de pouvoir récupérer le nombre de kills et de mort de chaque joueur dans la partie.

Pour cette partie, nous n'avons pas utilisé la classe Profil décrite dans la partie précédente car c'est ici qu'elle nous a posé problème mais nous nous sommes aidé de Photon. Photon nous a permis de créer des propriétés personnalisées beaucoup plus aisément à chaque joueur puisqu'il va gérer tout seul le stockage et donc de faciliter la gestion des propriétés pour le multijoueur.

La première propriété dont nous nous sommes occupés était celle de la mort. Nous avons fait une fonction "ApplyDeath" qui instancie la propriété "Death" à chaque joueur de la partie, la variable deathOfParent y est associée, elle commence à zéro et est incrémenté de un chaque fois qu'on appelle la fonction. Pour cette partie là ça n'as pas trop posé de problème puisqu'il a seulement fallu qu'on appelle "ApplyDeath" dans notre fonction qui

détruit un joueur quand il meurt.

Pour la seconde propriété ça était un peu différent. Le processus de création de propriété est tout de même semblable, on a fait une fonction "ApplyKill" qui instancie la propriété "Kills" à chaque joueur de la partie avec également une variable associée. Cependant on ne savait pas comment appeler la fonction puisqu'on avait aucun moyen de savoir qui a tué qui ou qui a été tué par qui. Donc ce qu'il a fallu faire, c'est dans notre fonction qui applique les dégâts à un joueur (voir image ci dessous), ajouter une variable qui indique quel ennemi a fait des dégâts à notre joueur que nous avons donc pu récupérer sur la fonction "ApplyKill"

```
void RPC_TakeDamage(float damage, Player opponent)
{
    if (!PV.IsMine)
        return;

    currentHealth -= damage;

    if (currentHealth <= 0) {
        ApplyKill(opponent);
        Die();
    }
}
```

Figure 8: Fonction RPC

2.7.2 Fin de partie

Quelque chose qui manquait à notre jeu à la fin de la première soutenance, est une fin à la partie. Pour cela, nous avons suivi l'idée que nous avions en tête depuis la création du cahier des charges : faire un DeathMatch (match à mort en français). C'est à dire que le joueur qui atteint un nombre éliminations donné gagne la partie. Pour cela, Photon a été très utile, car il permet d'appeler une fonction à chaque fois qu'une propriété d'un joueur change. Cela a permis à Paul d'implémenter facilement une fin de partie en vérifiant qu'aucun joueur ne dépassait la limite de kill donnée, et si c'était le cas, d'afficher le nom du gagnant sur l'écran de chacun, puis de quitter la partie au bout de quelques secondes, grâce à l'utilisation d'une coroutine.

2.7.3 Score Board

Après avoir donc fait le système d'évaluation du nombre de kills et de mort, Kilian a pu implémenter un tableau des scores qui permet à tous les joueurs de la partie de voir le nombre de kills et de mort de chacun.

Le fait d'avoir créer des propriétés spéciales avec Photon pour chaque joueur nous a permis de récupérer facilement les données et donc de pouvoir les afficher.

Nous avons fait pour l'instant un design très simple mais fonctionnel pour ce score board.

La difficulté ici était de créer une nouvelle case pour chaque joueur dans la partie contenant le bon nombre de kills et le bon nombre de morts.

Ce qui à était rencontré ici comme problème c'est qu'au début, à chaque nouvelle personne dans la partie, les cases correspondants aux joueurs se superposaient, ce qui rendait le tableau des scores complètement illisible pour l'utilisateur.

Après avoir corrigé ce problème, pour rendre le jeu plus agréable, nous avons fait en sorte que ce soit la personne ayant le plus de kills qui soit tout en haut du tableau des scores.

Username	Kills	Deaths
bob	1000	14345
AAAAAA	3	0
BBBBBB	2	0
CCCCCC	1	0
quill18	0	1

Figure 9: Inspiration utilisée pour le score board

2.8 ATH

Afin de rendre le jeu plus agréable et jouable, Kilian a créé un ATH (Affichage tête haute) dans le jeu, ce qui correspond à tout ce que l'utilisateur peut voir comme information sur son écran pendant qu'il joue. Ici l'ATH comporte pour l'instant :

- La vie, pour l'instant initialisé à 100 mais qu'on compte peut-être modifier
- Le pseudo du joueur (en bas à droite)
- Le nombre de kills du joueur
- Le nombre de morts du joueur
- Le crosshair

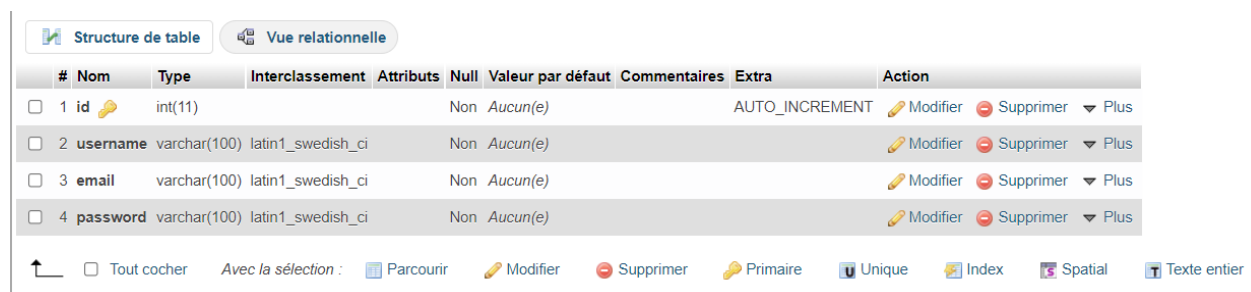
En plus de ce que Kilian à ajouter (Crosshair, vie...), Sam à rajouté une petite icône de l'arme que nous sélectionnons, pour rendre plus agréable et fluide la compréhension du jeu. Pour L'ATH nous avons décidé d'implémenter les principales fonctionnalités, pour avoir un jeu fonctionnel, quitte à ne pas avoir un maximum d'esthétique, que nous sommes petit à petit en train d'affiner.

Une barre de vie est en cours de création, pour l'instant même s'il n'est pas encore dans le jeu elle sera présente pour la soutenance numéro deux.

Un timer est également implémenté dans le jeu, il fait un décompte de dix minutes mais n'a pour l'instant aucune utilité. Il sera sans doute par la suite un autre moyen de finir une partie.

2.9 Site internet et comptes

Au terme de la première soutenance, le site était statique. C'est à dire que les utilisateurs ne pouvaient pas interagir avec. Depuis, je l'ai rendu dynamique grâce à un système de compte. Pour ce faire, j'ai utilisé Xampp (application permettant de mettre en place un serveur web local) pour pouvoir accéder à PHPMyAdmin (application permettant la création d'une base de données).



#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	username	varchar(100)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	email	varchar(100)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
4	password	varchar(100)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus

Figure 10: Base de données du projet.

Grace a cette base de données, il est désormais possible de s'enregistrer sur le site en entrant son nom d'utilisateur, email et mot de passe, l'ID servant simplement de repère à l'intérieur de la base de données, pour pouvoir se connecter au lieu de s'enregistrer par exemple

Le but à terme est de relier le compte sur le site avec le jeu afin de pouvoir afficher un leaderboard, ou des records personnels par exemple.

J'ai aussi reconstruit le site de manière à ce qu'il soit plus lisible : page d'accueil principale avec une présentation générale du projet, puis boutons vers les autres pages (présentation du groupe, sources, lien vers nos différents travaux...). Pour cette deuxième soutenance, je me suis plutôt concentré sur les mécaniques du site (découpage en plusieurs pages, connexion...) plutôt que sur la forme (contenu et design). Pour la suite, il faudra donc soigner la présentation et finaliser les contenus explicatifs.

2.10 Système d'armes et de tir

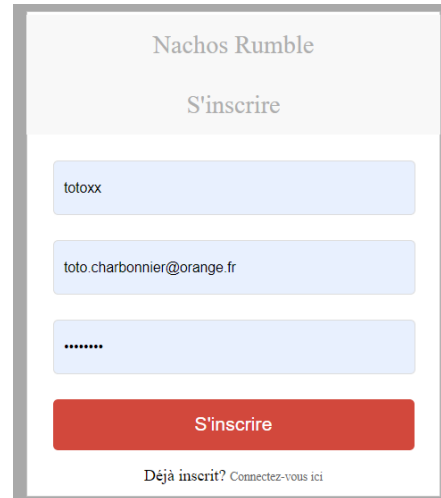
Nous avons aussi légèrement amélioré le système d'armes. En effet, après avoir effectué des tests sur le jeu, il est apparu que la précision des armes (parfaite et égale sur chaque arme) n'était pas très réaliste.

Nous avons donc implémenté un système de précision par arme, ainsi qu'une modification lors que le personnage est en mouvement ou qu'il court. Pour faire cela j'ai créé une variable "spread" qui gère l'amplitude, elle augmente lorsque le personnage bouge et diminue lorsqu'il



The image shows a login form for 'Nachos Rumble'. At the top, the text 'Nachos Rumble' is centered, followed by the title 'Connexion'. Below this, there are two light blue input fields: the first contains the text 'totoxx' and the second contains seven dots '.....'. A red button with the text 'Connexion' is positioned below the input fields. At the bottom, a link reads 'Vous êtes nouveau ici? S\'inscrire'.

Figure 11: Formulaire de connexion



The image shows a registration form for 'Nachos Rumble'. At the top, the text 'Nachos Rumble' is centered, followed by the title 'S\'inscrire'. Below this, there are three light blue input fields: the first contains 'totoxx', the second contains 'toto.charbonnier@orange.fr', and the third contains seven dots '.....'. A red button with the text 'S\'inscrire' is positioned below the input fields. At the bottom, a link reads 'Déjà inscrit? Connectez-vous ici'.

Figure 12: Formulaire d'inscription

est à l'arrêt. Cela permet de tirer plus ou moins précisément. A terme, il serait intéressant d'instaurer un système de visée pour tirer encore plus précisément, ou faire plus de dégâts...

3 Avance/retard global

A l'heure actuelle, beaucoup de travail a été fourni pour créer un jeu fonctionnel et complet, avec de multiples fonctionnalités que nous améliorons toutes au fil du temps.

Il est possible pour tout joueur de créer sa propre partie et de la lancer, ou bien de rejoindre une partie créée par un autre joueur. Contrairement à la première soutenance, ces parties ont maintenant un but et une fin, ainsi qu'un gagnant.

La partie intelligence artificielle du jeu est en plein développement et est presque fonctionnelle en multijoueur, même s'il reste quelques points à clarifier sur son fonctionnement et sa manière d'être gérée par le service Photon.

En vue de la dernière soutenance, nous allons enfin rendre le jeu agréable visuellement, en implémentant les animations des joueurs, de tir, de rechargement, mais aussi finir de rendre belle et amusante le map de notre jeu, finir rajouter tout les effets sonores (car c'est une tâche déjà en cours), mais aussi les menus, etc.

4 Objectifs pour la prochaine soutenance

Pour la troisième et dernière soutenance, nous comptons évidemment finir le jeu. Nous devons rattraper le peu de retard que nous avons pris sur certains points, en profitant de l'avance pris sur le développement d'autres fonctionnalités.

Désormais, les bases du jeu sont en place : fonctionnement d'une partie, menu, site. Il nous reste donc à peaufiner le travail sur tous les aspects. Le but étant d'obtenir un jeu agréable à jouer tant par les graphismes que par la qualité du gameplay. Pour ce faire, plusieurs pistes sont avancées :

4.1 Mécaniques du jeu

Il nous faut implémenter un système de tir plus précis : dispersion des balles, visée et précision des tirs. Le but ici est de rendre le jeu plus réaliste et détaillé, donc plus agréable à jouer.

4.2 Intelligence Artificielle

Grace au gros travail de Paul, nous avons déjà une IA, l'objectif pour la troisième soutenance est simple : la rendre 100% fonctionnelle. Nous entendons par cela que l'on puisse jouer en solo, et de s'amuser tout comme en multijoueur. Il faut également qu'elle soit un minimum compétitif, pour que le jeu ne soit pas trop simple, nous tâcherons de doser cela.

4.3 Graphismes

Pour la troisième soutenance, nous prévoyons d'avoir fini la map du jeu en plus des modèles 3D d'armes que nous avons maintenant. Cela permettrait d'avoir un cadre clair aux parties du jeu. Pour ce qui est du Player Model, Il est certain que nous aurons un, et il est prévu qu'il soit animé (au moins sur les animations basiques). Nous sommes également en train d'implanter des menus plus jolis, et de rajouter de l'audio et des bruitages. Pour ce qui est des graphismes, notre objectif est clair : avoir un jeu visuellement attractif.

4.4 Site

Pour le site, l'objectif est bien sûr de le mettre sur internet, néanmoins il ne faut pas perdre de vue l'aspect visuel (toute la partie CSS). Il manque aussi de contenu car nous nous sommes focalisés sur les mécaniques du site (création de compte, répartitions des pages...). Il faut donc le rendre plus complet, plus beau et aussi essaye de l'adapter à tous les types d'écran, moitié d'écran...

Tâche	Avances	Dans les temps	Retards
Graphisme			
Map		X	
Armes		X	
Personnages		X	
Animations		X	
Audio			
Musiques		X	
Bruitages		X	
Réseau			
Multijoueur	X		
IA			
Algorithme		X	
Mécanique de jeu			
Physique		X	
Caméra		X	
Mécanique de tir		X	
Interface			
Menu		X	
ATH		X	
Site web			
Création du site internet	X		

5 Conclusion

Nous sommes contents de la tournure que prend le développement de notre jeu à l'heure actuelle. Même si nous avons eu du mal à gérer le travail sur le jeu en parallèle avec les autres projets et travaux à l'EPITA, nous sommes excités de pouvoir profiter de la fin du semestre pour pouvoir travailler sur le jeu, qui a bien pris forme.