

Rapport de la soutenance intermédiaire Projet S2



par

LES RADIANTS

Membres du groupe:

Aymeric HESNARD

Chiyong LEE

Ryan PRAK

Professeurs:

Mme CHAOUED

M SABER



Lyon 69003, FRANCE.

2022-2023

Contents

1	Introduction	2
1.1	Rappels généraux	2
1.1.1	Répartition des tâches	2
1.1.2	Ce que nous avons	3
2	Etat actuel du projet	4
2.1	Nouveautés	4
2.1.1	Sur l'interface	4
2.1.2	Sur le joueur et son environnement	5
2.2	Problèmes rencontrés	8
2.3	Corrections apportées	8
2.4	Bilan	8
3	Etat actuel du Site Web et du Launcher	9
3.1	Nouveautés	9
3.2	Problèmes rencontrés	9
3.3	Correction apportées	10
3.4	Bilan	10
4	Conclusion	11
4.1	Récit de réalisation	11
4.2	Ce qu'il nous reste à faire	13
4.3	Ressources	13

Chapter 1

Introduction

1.1 Rappels généraux

Pour rappel, notre projet est de réaliser un Jeu de Tir à la Première personne (ou FPS en anglais). Le but du jeu est d'éliminer les membres de l'équipe adverse afin d'obtenir des points. Nous souhaitons que les équipes soient composées de 5 joueurs et qu'il y en ai deux au total.

1.1.1 Répartition des tâches

Nous nous étions répartis les tâches de cette manière:

Tâches	Aymeric	Chiyoun	Ryan
Intelligence Artificielle	10%	80%	10%
Site Web	20%	60%	20%
Interface et Launcher	10%	40%	50%
Game Design et Environnement	60%	30%	10%
Programmation interaction objets	40%	25%	35%

1.1.2 Ce que nous avons

Lors de la première soutenance nous vous avons présenté une ébauche de notre projet.

Sur le projet en lui-même

A ce moment là nous n'avions qu'une petite scène où le joueur pouvait évoluer librement, sauter et courir, il était muni d'un prototype d'une arme qui pouvait tirer mais sans animations pour en attester.

Sur le site web

Nous avons également un site web presque entièrement fonctionnel (presque car les liens de téléchargement ne fonctionnaient pas) avec une page principale, une page de téléchargement, une page de représentation du groupe et du projet, et une page des ressources que nous avons pu utilisé.

Sur l'interface

L'interface utilisateur était assez simple, composé de deux boutons: l'un pour accéder à notre site web, l'autre pour lancer un exécutable de notre projet.

Chapter 2

Etat actuel du projet

2.1 Nouveautés

2.1.1 Sur l'interface

Afin d'améliorer l'expérience de jeu de l'utilisateur nous avons amélioré son "interface". En particulier, nous avons rajouté :

- Un pointeur, au centre de l'écran afin que le joueur sache plus précisément l'endroit vers lequel il va tirer.
- Une liste dynamique des joueurs connectés, avec précision de leur nom, du nombre d'élimination qu'ils ont pu effectuer et le nombre de mort à leur compte (fig.6, fig.7).
- Une zone dans l'angle supérieur droit, qui annonce à tous les joueurs des dernières éliminations qui ont eu lieu dans les 5 dernières secondes, afin que ceux-ci se fasse une idée de qui peu être à craindre ou au contraire semble être une cible facile.
- Un menu "pause", qui permet à un joueur de quitter la partie en cours de route (fig.5, fig.8).
- Une barre de vie, qui permet au joueur de savoir plus précisément à quel point il peut être proche de la mort.

2.1.2 Sur le joueur et son environnement

Dans le but de renforcer la qualité de notre jeu nous avons rajouté sur différents composants, animations et effets sonores, ainsi que d'autres particularités.

Arme

Du côté de l'arme, nous avons pu trouver des modèles 3D prêt à être utilisé car nous avons pensé que le niveau de détail nécessaire à un rendu "correct" nous aurait pris beaucoup de temps et nous n'étions même pas certains que le résultat nous plaise donc dans l'immédiat nous avons opté pour cette solution (fig.1 et fig.11). De plus, nous avons animé l'arme que se soit avec des animations sonores(fig.14) et également des animations visuelles au niveau du canon de l'arme et également au niveau de l'endroit de l'impact du tir.

Pour finir avec cette partie nous avons mis en place trois particularités majeures avec l'ajout d'une classe d'arme (fig.2):

- La possibilité à une arme de tirer soit en mode "automatique", soit en "coup par coup"
- Un système de rechargement de l'arme, car dans un jeu où il peut y avoir différents types d'armes comme des mitrailleuses ou des pistolets, un système de munitions peut rendre une arme plus intéressante en fonction de certaines situations.
- Un système de dégât des armes couplé avec un système de porté, afin de permettre à notre projet d'être plus attractif, la création de différents types d'arme (pistolet, mitrailleuse, fusil, et bien d'autre) est un bon choix car chacun des utilisateurs peut avoir des goûts différents et préférer tel ou tel mode de jeu (plutôt à distance, plutôt en combat rapproché), de ce fait il paraît plus cohérent que les armes aient des portés différentes et des dégâts infligés différents.

Lorsque le chargeur est vide, on recharge automatiquement l'arme du joueur avec la méthode `Reload()` et on impose qu'on ne puisse pas tirer pas pendant le rechargement avec un `return`(fig.12). Il est possible de recharger manuellement en appuyant sur la touche R. On impose un temps de rechargement. On évite de recharger lorsqu'on est déjà en train de recharger(fig.13).

Joueur

Au niveau du joueur en lui-même, nous n'avons pas encore décidé de son apparence définitive et nous n'avons rien changé à part l'affichage de la barre de vie(fig.15). Dans PlayerUI.cs, on ajoute 2 images, une pour le fond de la barre de vie et une autre par dessus qui indique la proportion de points de vie par rapport au nombre maximum de points de vie. On ajoute une icône pour indiquer explicitement au joueur que c'est sa barre de vie. On crée dans Player.cs une méthode GetHealthPct() qui renvoie un float qui est la proportion de la quantité actuelle de points de vie et la quantité maximum de points de vie. En effet l'échelle initiale de l'objet Image de UI est à 1 au maximum et à 0 lorsque le joueur meurt (plus de points de vie). Cela va permettre de baisser la barre de vie par rapport à la barre maximum de vie. Nous avons rendu cette méthode "public" afin de pouvoir la récupérer dans le fichier PlayerUI.cs. On cherche à récupérer la quantité de points de vie actuelle du joueur Player, c'est-à-dire les attributs currentHealth et maxHealth. On ajoute les attributs HealthFill (qui représente la barre de vie rouge) et player (qui va permettre de récupérer la méthode GetHealthPct()). On implémente une méthode SetHealthAmount qui va permettre de modifier l'échelle de la barre de vie rouge. Dans la méthode Update, on modifie l'affichage de la barre de vie en fonction des points de vie actuelles du joueur.

IA

Nous avons pu concevoir une première version de l'Intelligence Artificielle. Elle a pour apparence pour l'instant, d'un simple sphère avec des yeux (fig.3). Elle se déplace librement de façon aléatoire sur toute la plateforme du jeu. Lorsqu'elle rencontre un joueur ennemi, elle le poursuit et si elle arrive à rentrer en contact avec le joueur, elle explose (fig.9). Si le joueur parvient à lui infliger assez de dégâts avant qu'il se fasse toucher, l'IA explose sur place. A tire indicatif, les fonctions auxiliaires `Patrolling()` permet à notre IA de se déplacer librement sur la carte, `ChasePlayer()` permet de poursuivre le joueur, et enfin `Explode()` comme son nom l'indique, fait exploser l'IA. Pour l'instant, nous n'avons pas encore mis de système de réseau pour que l'IA soit synchronisée sur toutes les instances, ni le système de dégâts sur les joueurs par explosion. A travailler prochainement.

Son fonctionnement est basé sur un asset Unity appelé NavMesh et du module Unity AI. Grâce à ces derniers, nous avons pu délimiter les zones de la map où l'IA peut se déplacer, et de pouvoir définir le layer à prendre en compte par l'IA. En ce qui concerne le déplacement aléatoire de l'IA, nous avons du coder le système aléatoire du calcul de la prochaine destination de l'IA. Plus précisément, tant qu'il n'y a pas de joueur dans sa zone de détection, elle calcule la prochaine destination aléatoirement en utilisant le `Random` et le `Vector3` (fig.10). Sinon, elle poursuit le joueur détecté. Comme expliqué auparavant, si elle touche le joueur, elle explose avec l'animation de l'explosion, mais elle peut également se faire détruire par le joueur.

Map

Afin que le joueur ait plus de surface de jeu, nous avons agrandi la carte, avec plus d'obstacles, d'endroits où se couvrir des tirs ennemis, de tour de guet qui pourraient par exemple aux joueurs de prendre un avantage par la hauteur et deux batiments centraux afin de rajouter encore plus de possibilités. De plus nous avons mis en place deux zones d'apparitions opposées qui correspondront aux zones d'apparitions respectives des 2 camps adverses. Afin de tester notre début d'intelligence artificiel nous avons placé des prototypes aux quatres coins de la carte mais ce n'est que temporaire car nous envisageons que ce soit les joueurs qui puissent les invoquer, modification que nous comptons apporter d'ici la prochaine soutenance(fig.4).

2.2 Problèmes rencontrés

Lors de la réalisation des modifications apportées nous avons rencontré de nombreux problèmes:

- Le premier étant que la majorité des tutoriels que nous avons pu regarder sont sur une version antérieure de unity que laquelle nous utilisons pour notre projet. Ainsi, parfois certaines commandes n'existent plus ou ont été remplacé par d'autre.
- Ensuite, nous avons remarqué que lorsque l'un des joueurs est éliminé son tableau des scores et complètement effacé sans raisons apparentes.
- Bien sûr avec l'ajout d'animation, nous avons rencontré des problèmes d'esthétique car nous avons du faire en sorte que chaque animation, quelles soient visuelles ou sonores, ne soient ni trop pesante, ni trop minime.

2.3 Corrections apportées

Auparavant, nous avions eu quelques problèmes avec la physique du joueur, par exemple lorsque le joueur se déplaçait dans le vide il tombait indéfiniment. A présent, nous avons rajouté un mur qui empêche le joueur de sauter dans le vide et qui définit mieux les limites de la zone de jeu.

De plus nous avons remarqué lors de test que la zone qui permet au système de prendre en compte la physique du joueur était mal configuré, c'est-à-dire que lorsque nous tirions sur un joueur adverse, celui-ci ne prenait pas toujours en compte le fait d'avoir été touché par un tir. Par conséquent nous avons redimensionné cette composante du joueur afin que cela fonctionne mieux.

2.4 Bilan

Notre projet a avancé, plus encore de ce que nous aurions pu pensé. Le jeu est fonctionnel, nous avons les effets visuels et sonores prévus, nous avons commencé la création d'une zone de jeu plus confortable pour le joueur, et enfin nous avons commencé la création d'un système d'arme plus poussé. Ce qu'il nous resterait à faire serait d'améliorer l'apparence de nos composants, améliorer certains côtés de notre jeu et pourquoi pas ajouter un système de "compétences".

Chapter 3

Etat actuel du Site Web et du Launcher

3.1 Nouveautés

Une nouvelle page "Gameplay" a été rajoutée à l'en-tête de notre site web. Sur cette page, nous envisageons de mettre une vidéo extraite de notre jeu en fond de la page. Pour l'instant, comme notre jeu n'est pas encore terminé, nous avons mis une vidéo quelconque juste pour donner une idée du résultat. En plus de cela, nous avons placé notre titre de jeu "Invasion" en grand au milieu de la page. Lorsqu'on place la souris dessus, l'intérieur du titre devient transparent. Enfin, nous avons également mis en disposition un bouton "Télécharger" qui ramène directement sur l'onglet "Télécharger".

3.2 Problèmes rencontrés

Nous avons rencontrés certains problèmes :

- Le bandeau du copyright qui est censé se situer tout en bas, se retrouvait sur certains onglets au milieu de la page.
- La position de notre logo n'était pas bien centrée sur toutes les pages.
- L'image du fond se répétait si on dépassait une certaine limite de hauteur.
- Le chemin du .exe de notre jeu pour le lancer diffère en fonction du matériel.

3.3 Correction apportées

Nous avons pu régler les problèmes suivants :

- Nous avons pu placer le bandeau du copyright tout en bas de chaque page.
- La position de notre logo a été fixée au centre sur toutes les pages.
- Le problème du chemin du .exe n'a pas été réglé encore, mais nous avons une alternative qui évite à chaque fois de recopier le chemin absolu local, en revanche il est nécessaire de placer le dossier du projet dans un dossier utilisateur (exemples : Documents, Bureau ou Desktop, etc...)

Cependant, nous n'avons pas encore pu régler le problème du fond qui se répétait.

3.4 Bilan

Nous avons pu améliorer notre site web en rajoutant une page avec une vidéo "Gameplay" de notre jeu (à venir). Mais comme notre site web était déjà bien avancé pour la première soutenance, nous n'y avons pas amené de majeures modifications. En ce qui concerne le Launcher, il était déjà fonctionnel et relativement complet. Ce qu'il restera toujours à modifier sera son esthétisme et possiblement son ergonomie afin de le rendre toujours plus attractif et simple d'utilisation.

Chapter 4

Conclusion

4.1 Récit de réalisation

Ryan

J'ai été très enthousiaste à l'idée de pouvoir enfin avancer la programmation de notre projet sur le moteur Unity. J'ai eu l'occasion de découvrir pléthore de fonctionnalités intéressantes, notamment les animations et les objets UI. C'est notamment la création de nouvelles armes avec leurs spécificités qui m'a amusé. J'ai observé la partie de l'Intelligence Artificielle que j'ai trouvé intéressante. Malheureusement, je n'ai pas encore participé à la création de la carte. Je suis ravi de pouvoir exploiter ma créativité dans ce projet. Au début, j'ai eu quelques difficultés à comprendre certaines parties du code, il est facile d'être étourdi lorsqu'on programme intensément. J'espère qu'on mènera ce projet à bien et qu'il permettra de nous enrichir de connaissances.

Chiyoung

Pour cette deuxième soutenance, j'ai eu l'occasion de découvrir et apprendre le HTML et le CSS. Je suis assez satisfait de mon travail concernant la nouvelle page qui contiendra la vidéo gameplay de notre jeu. Cependant, par manque d'inspiration, je n'ai pas pu apporter d'importantes modifications. En effet, notre site web était déjà bien complété et réalisé par mes camarades. D'autre part, j'ai été chargé de la partie de l'intelligence artificielle. Comme il s'agit d'un sujet qui m'intéresse fortement, j'ai eu beaucoup d'idées pour notre IA, tout en s'inspirant des autres jeux vidéos. Cependant, comme tout le monde est au courant, créer une intelligence artificielle n'est pas une simple chose. En effet, cela m'a pris un temps assez conséquent pour pouvoir comprendre et l'implémenter et d'appliquer sur notre projet. Enfin, j'ai également pu contribuer à la réalisation de notre première carte du jeu. Cette partie est très intéressante puisque la carte reflète purement notre imagination et créativité. Je suis satisfait du travail de notre groupe en général. J'ai hâte de pouvoir finaliser notre jeu, mais en même temps inquiet de ne pas pouvoir obtenir le résultat attendu de notre part dû à un manque de temps.

Aymeric

Ayant été assigné principalement, à la partie création de notre site web lors de la première partie de la réalisation de notre projet, je découvre l'utilisation de Unity. C'est un logiciel à la fois puissant et amusant à utiliser, mais dont l'apprentissage peut s'avérer relativement difficile et assez laborieux. En effet, tous les composants ne sont pas évidents à utiliser et parfois les tutoriels sur internet peuvent manquer. Néanmoins j'éprouve un certain plaisir à ressentir la capacité de créer tout ce dont je pourrais souhaiter enfin dans la plupart des cas. J'ai hâte de continuer à travailler sur la suite de notre projet même si je pense que nous ne sommes pas à la fin de tous nos soucis.

4.2 Ce qu'il nous reste à faire

Dans l'ensemble le projet à bien avancé, toute la partie fonctionnelle semble fonctionner. Par conséquent il ne nous resterait qu'à:

- Donner la possibilité de télécharger notre projet via notre site web,
- Ajouter de nouvelles armes, de nouvelles compétences, de nouvelles animations.
- Synchroniser l'IA sur toutes les instances et l'améliorer pour la rendre plus intéressante
- Améliorer le rendu visuel du joueur, des animations, de la carte en général

En bref de continuer de nous amuser à la création de ce merveilleux projet qui nous permet de développer nos compétences d'endurance et de cohésion d'équipe.

En espérant que vous aurez apprécié lire notre rapport, bien à vous :

LES RADIANTS

4.3 Ressources

- <https://www.youtube.com/@TUTOUNITYFR> : Tuto FPS Unity
- <https://www.youtube.com/@davegamedevelopment> : Tuto IA Unity
- <https://stackoverflow.com/questions/3736462/getting-the-folder-name-from-a-full-filename-path> : recherche pour régler le problème du chemin du .exe
- <https://stackoverflow.com/questions/6817639/get-filenames-without-path-of-a-specific-directory> : recherche pour régler le problème du chemin du .exe
- <https://devassets.com/assets/multiplayer-fps-assets/> : Lien de téléchargement du pack visuel de notre arme

Annexe



Figure 1: L'arme principale

```
public class PlayerWeapon
{
    public string name = "Gun name";    🔗 Serializable
    public float damage = 10f;          🔗 Serializable
    public float range = 100f;          🔗 Serializable
    public float fireRate = 0f;          🔗 Serializable
    public GameObject graphics;          🔗 Serializable
}
```

Figure 2: L'arme principale

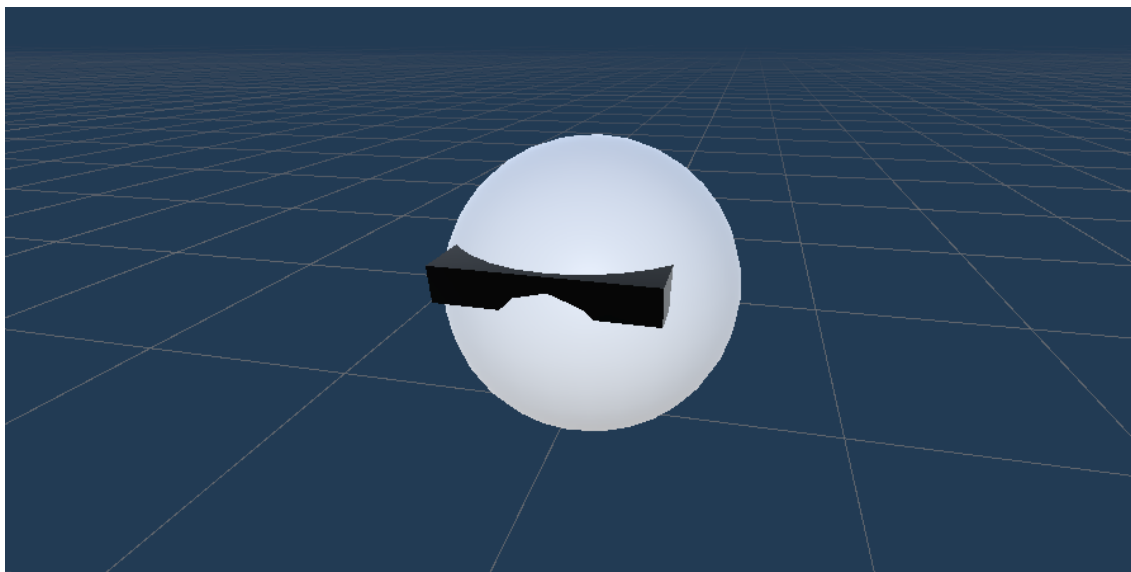


Figure 3: L'intelligence artificielle

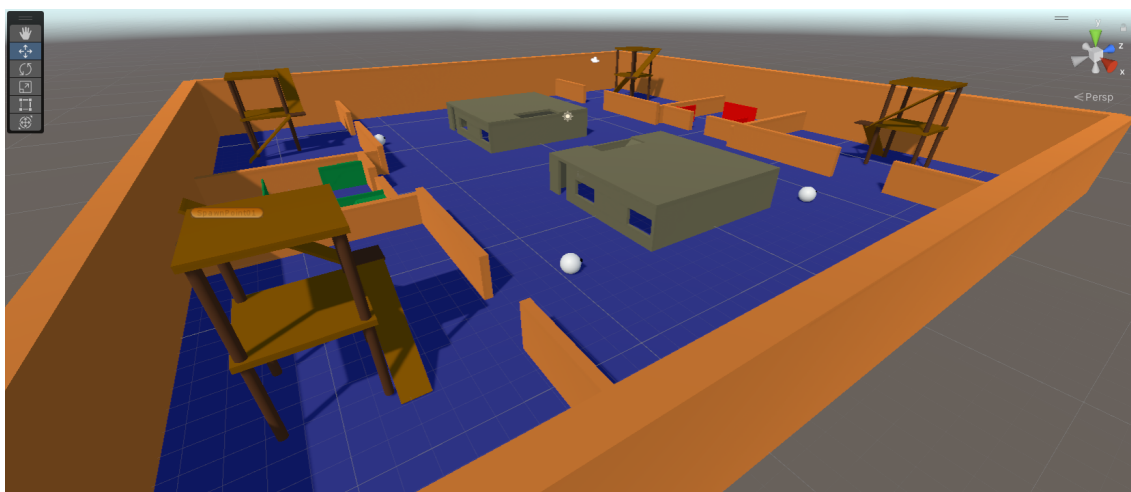


Figure 4: La carte

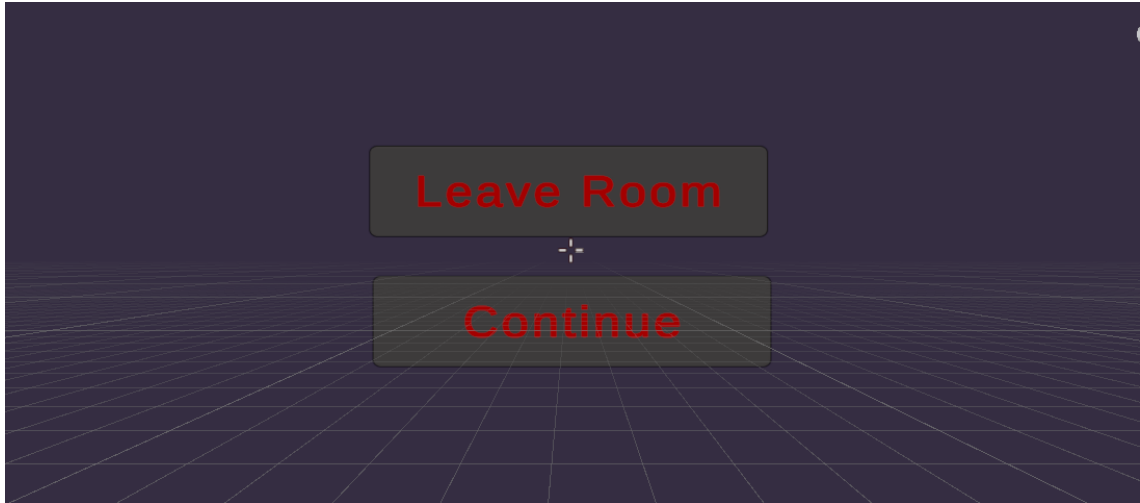


Figure 5: Le menu pause

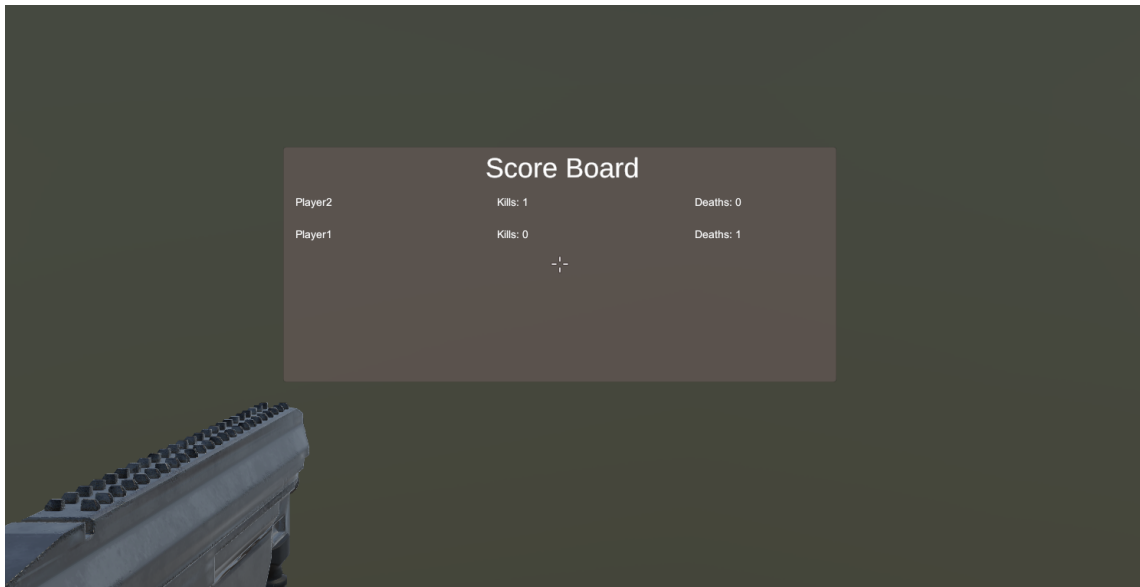


Figure 6: Le tableau des scores

```

private void OnEnable()
{
    //recuperer une array de tout les joueurs du serv
    Player[] players = GameManager.GetAllPlayers();

    // mise en place d'une ligne par joueur

    foreach (var player in players)
    {
        GameObject itemgo = Instantiate(playerScoreBoardItem, playerScoreBoardList);
        ScoreBoardItem item = itemgo.GetComponent<ScoreBoardItem>();
        if (item != null)
        {
            item.Setup(player);
        }
    }
}

Event function Hesnard Aymeric
private void OnDisable()
{
    // vider la liste des joueur
    foreach (Transform child in playerScoreBoardList)
    {
        Destroy(child.gameObject);
    }
}

```

Figure 7: Le code du tableau des scores

```

private void Update()
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        TogglePauseMenu();
    }

    //affichage du tableau des scores
    if (Input.GetKeyDown(KeyCode.Tab))
    {
        scoreBoard.SetActive(true);
    }else if (Input.GetKeyUp(KeyCode.Tab))
    {
        scoreBoard.SetActive(false);
    }
}

1 asset usage 1 usage Hesnard Aymeric
public void TogglePauseMenu()
{
    pauseMenu.SetActive(!pauseMenu.activeSelf);
    PauseMenu.isOn = pauseMenu.activeSelf;
}

```

Figure 8: Le code du UI

```

private void Update()
{
    player = FindLocalPlayer().transform;

    //check for sight and attack range
    playerInSightRange = Physics.CheckSphere(transform.position, sightRange, (int) whatIsPlayer);
    playerInAttackRange = Physics.CheckSphere(transform.position, attackRange, (int) whatIsPlayer);

    if (!playerInSightRange)
    {
        Patrolling();
    }

    if (playerInSightRange && !playerInAttackRange)
    {
        ChasePlayer();
    }
    if (playerInSightRange && playerInAttackRange)
    {
        Explode();
    }
}

```

Figure 9: Fonctionnement principal de l'IA

```

Frequently called 1 usage Chiyong
private void SearchWalkPoint()
{
    //calculate random point in range
    float randomZ = Random.Range(-walkPointRange, walkPointRange);
    float randomX = Random.Range(-walkPointRange, walkPointRange);

    walkPoint = new Vector3(transform.position.x + randomX, transform.position.y, transform.position.z + randomZ);

    if (Physics.Raycast(origin: walkPoint, direction: -transform.up, maxDistance: 2f, (int) whatIsGround))
    {
        walkPointSet = true;
    }
}

```

Figure 10: Calcul de la prochaine destination



Figure 11: Exemple d'une nouvelle arme

```
private void Shoot()
{
    if (!isLocalPlayer || _weaponManager.isReloading)
    {
        return;
    }

    if (_weaponManager.currentMagazineSize <= 0) //Si on n'a plus de balles
    {
        _weaponManager.Reload();
        StartCoroutine(_weaponManager.Reload());
        return; //parce qu'on ne veut pas le code dessous (particules,tir, ...) vu qu'on n'a plus de balles
    }
    _weaponManager.currentMagazineSize--; //decremente une balle dans le chargeur

    Debug.Log("Il nous reste " + _weaponManager.currentMagazineSize + " balles dans le chargeur.");
}
```

Figure 12: Restrictions du rechargement

```

public IEnumerator Reload()
{
    if (isReloading)
    {
        yield break; //return ne marche pas avec Coroutine
    }

    Debug.Log("Reloading ...");

    isReloading = true;

    CmdOnReload();

    yield return new WaitForSeconds(_currentWeapon.reloadTime); //on cree un delai de rechargement

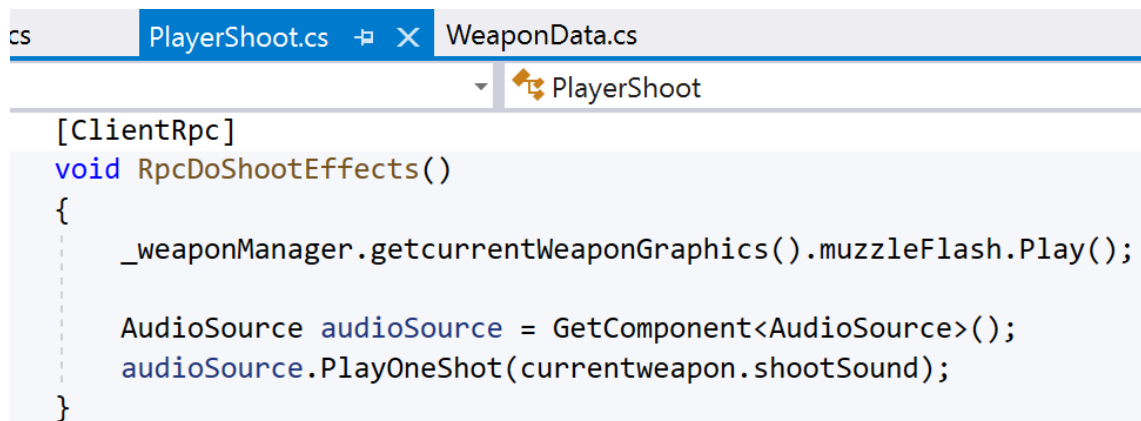
    currentMagazineSize = _currentWeapon.magazineSize;

    isReloading = false;

    Debug.Log("Reloading finished ...");
}

```

Figure 13: Fonctionnement du rechargement



```

cs PlayerShoot.cs X WeaponData.cs
▼ PlayerShoot

[ClientRpc]
void RpcDoShootEffects()
{
    _weaponManager.getCurrentWeaponGraphics().muzzleFlash.Play();

    AudioSource audioSource = GetComponent();
    audioSource.PlayOneShot(currentweapon.shootSound);
}

```

Figure 14: Effets sonores des tirs

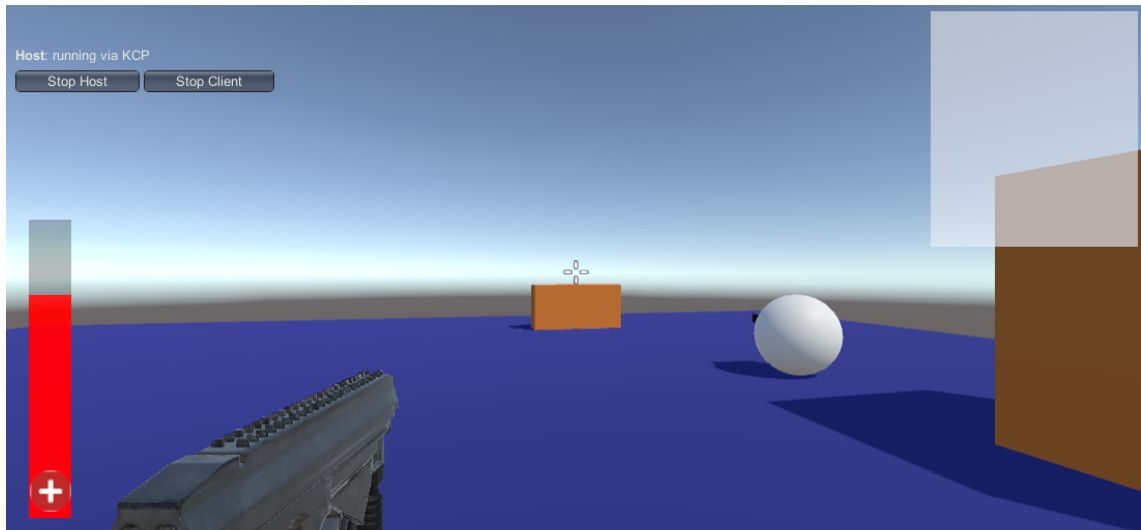


Figure 15: Barre de vie visible pour le joueur