

# Projet Unity : TPS Shooter

Mohamed Ghacham Amrani

Janvier 2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Présentation du Jeu</b>	<b>4</b>
2.1	But du Jeu . . . . .	4
2.2	Contrôles . . . . .	4
<b>3</b>	<b>Spécifications du Projet</b>	<b>5</b>
3.1	Modalités et Contraintes . . . . .	5
3.2	Mécaniques de Jeu . . . . .	5
<b>4</b>	<b>Développement du Projet</b>	<b>6</b>
4.1	Moteur de Jeu et Outils Utilisés . . . . .	6
4.2	Structure du Code . . . . .	6
<b>5</b>	<b>Résultats et Conclusion</b>	<b>7</b>
<b>6</b>	<b>Illustrations et Captures d'Écran</b>	<b>8</b>
6.1	Aperçu du Jeu . . . . .	8
6.2	Interface et Gameplay . . . . .	8
6.3	Prefabs . . . . .	9
6.4	Animations . . . . .	9
6.5	Scripts utilisés . . . . .	10
<b>A</b>	<b>Appendices</b>	<b>11</b>
A.1	Exemple de Code . . . . .	11

# Table des figures

6.1	Aperçu du jeu en action . . . . .	8
6.2	Menu et interface du jeu . . . . .	9
6.3	Liste des prefabs utilisés dans le projet . . . . .	9
6.4	Organisation des animations dans Unity . . . . .	9
6.5	Scripts principaux du projet . . . . .	10

# Chapitre 1

## Introduction

Le but du projet est de mettre en application les techniques vues au cours du semestre pour réaliser un mini-jeu vidéo de type TPS Shooter sous Unity. Ce document détaille le développement, les choix d'implémentation, et les défis rencontrés.

# Chapitre 2

## Présentation du Jeu

### 2.1 But du Jeu

L'objectif principal du jeu est de survivre le plus longtemps possible face à des vagues d'ennemis de plus en plus coriaces. Le joueur doit utiliser son environnement pour se protéger, récupérer des soins et améliorer ses compétences de combat afin d'atteindre le meilleur score possible.

### 2.2 Contrôles

- **Déplacement** : *ZQSD* pour se déplacer.
- **Actions** :
  - *Espace* : Sauter
  - *Ctrl gauche* : Roulade (invincibilité temporaire)
  - *Maj gauche* : Courir
- **Combat** :
  - *Clic gauche* : Tir
  - *Clic droit* : Viser
  - *1, 2, 3* : Changer d'arme (AK, Pistolet, Shotgun)
- **Caméra** : Vue à la troisième personne contrôlée par la souris.

# Chapitre 3

## Spécifications du Projet

### 3.1 Modalités et Contraintes

- Utilisation d'**Unity 2022.3 LTS**.
- Gestion de projet avec **Git**.
- Rendu final le **10/01/2025 à 22h**.
- Soumission par email avec le dépôt Git et une description du projet.

### 3.2 Mécaniques de Jeu

- **Système de vagues** : Les ennemis apparaissent progressivement avec une difficulté croissante.
- **Amélioration du joueur** : Augmentation de la vitesse, des dégâts et de la cadence de tir.
- **Gestion des obstacles** : Permet de se cacher et d'éviter les tirs ennemis.
- **Système de santé** : Ramassage de soins pour restaurer la vie.

# Chapitre 4

## Développement du Projet

### 4.1 Moteur de Jeu et Outils Utilisés

Nous utilisons Unity avec le langage C# pour implémenter les différentes mécaniques de jeu. Le contrôle de version est géré via Git.

### 4.2 Structure du Code

Le projet est organisé en plusieurs scripts gérant :

- **Le mouvement du joueur** : gestion du déplacement, des actions (saut, roulade, course).
- **Le système de tir** : gestion des projectiles et de leurs impacts.
- **L'intelligence artificielle** : comportement des ennemis et augmentation de la difficulté.
- **L'interface utilisateur** : affichage des barres de vie et score.

# Chapitre 5

## Résultats et Conclusion

Le jeu est fonctionnel et répond aux objectifs fixés. Des améliorations possibles incluent l'ajout d'animations plus fluides et d'une IA plus complexe.



# Chapitre 6

## Illustrations et Captures d'Écran

### 6.1 Aperçu du Jeu

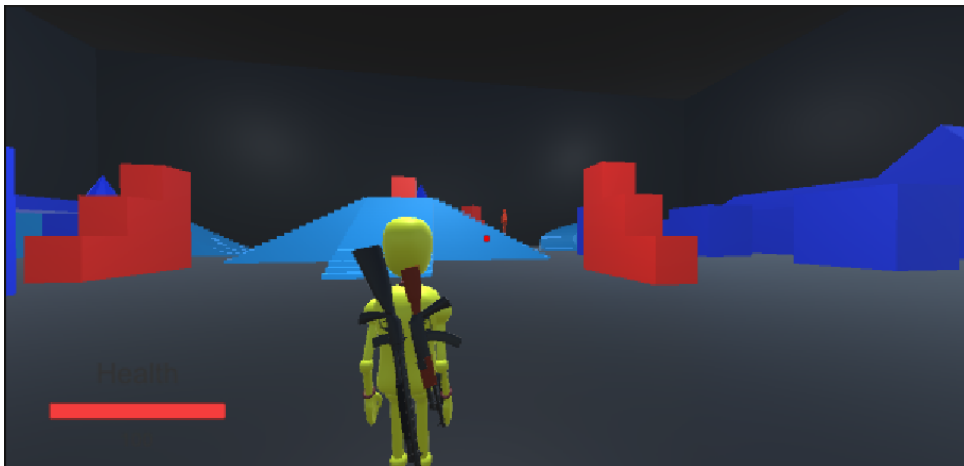


FIGURE 6.1 – Aperçu du jeu en action

Cette section présente une vue générale du jeu en action. Le personnage du joueur évolue dans un environnement en 3D avec des éléments interactifs. L'objectif est de combattre les ennemis tout en utilisant les obstacles pour se protéger. La perspective à la troisième personne permet une meilleure immersion et un contrôle optimal du personnage.

### 6.2 Interface et Gameplay

L'interface utilisateur comprend plusieurs éléments essentiels, tels que les menus, les options et le HUD en jeu. Le joueur peut naviguer dans le menu principal pour ajuster les paramètres, commencer une nouvelle partie ou quitter le jeu. En cours de partie, l'affichage des informations comme la barre de vie et les munitions permet une gestion efficace des ressources.

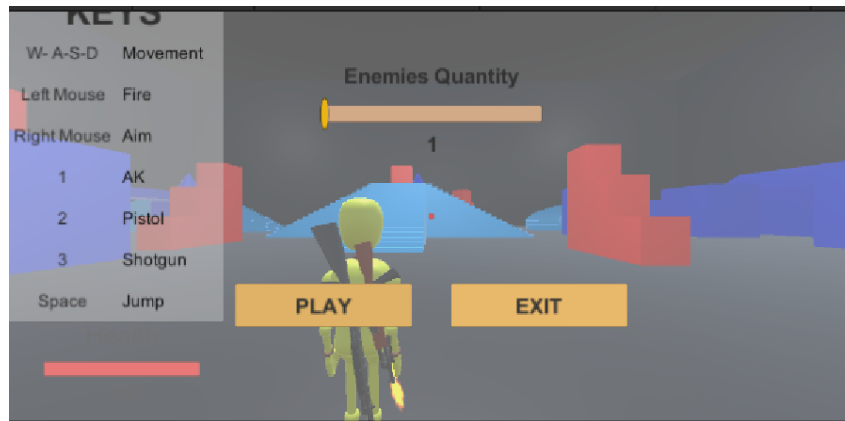


FIGURE 6.2 – Menu et interface du jeu

## 6.3 Prefabs

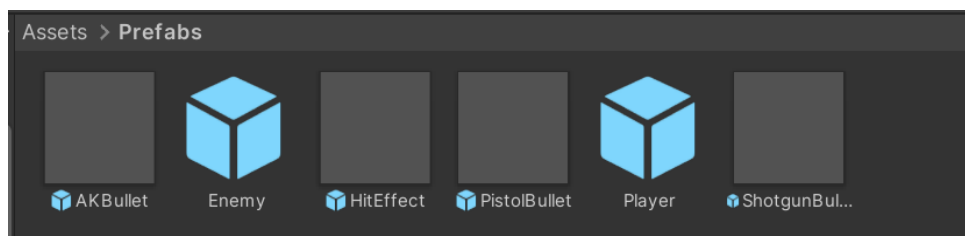


FIGURE 6.3 – Liste des prefabs utilisés dans le projet

Les prefabs sont des objets préconfigurés utilisés pour optimiser la gestion des éléments du jeu. Ils permettent d’instancier facilement des ennemis, des projectiles et d’autres objets interactifs. Grâce à cette approche modulaire, il est possible de modifier les paramètres d’un élément sans affecter l’ensemble du projet.

## 6.4 Animations

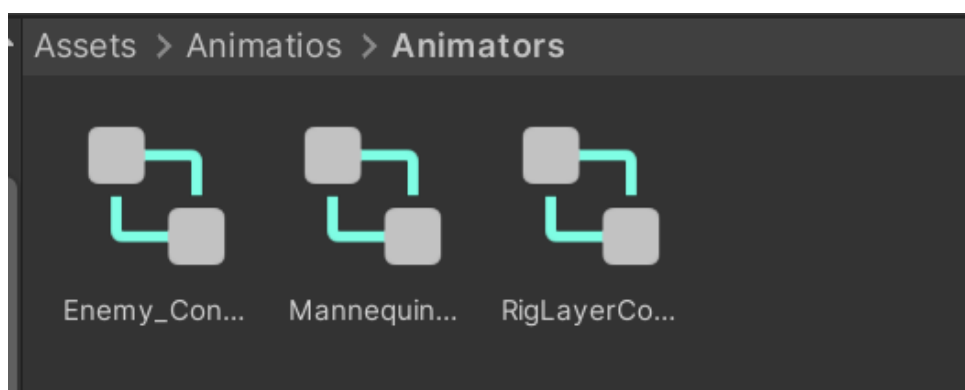


FIGURE 6.4 – Organisation des animations dans Unity

Le système d’animation repose sur des contrôleurs Animator, qui gèrent les transitions entre différentes animations. Cela permet d’obtenir un rendu fluide des mouvements du

personnage, des ennemis et des effets spéciaux. L'intégration des animations améliore le dynamisme du jeu et renforce l'expérience immersive.

## 6.5 Scripts utilisés

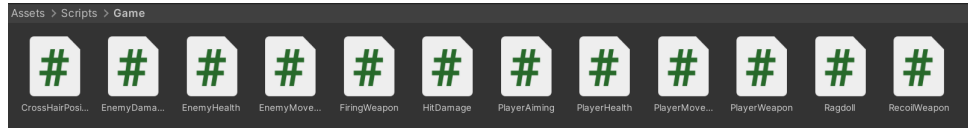


FIGURE 6.5 – Scripts principaux du projet

Le développement du jeu repose sur plusieurs scripts C assurant le fonctionnement des différentes mécaniques. Ces scripts couvrent le déplacement du joueur, la gestion des tirs, l'intelligence artificielle des ennemis et la mise à jour de l'interface utilisateur. Une architecture bien structurée facilite la maintenance et l'ajout de nouvelles fonctionnalités.

# Annexe A

## Appendices

### A.1 Exemple de Code

Listing A.1 – Gestion du tir en C#

```
void Shoot() {  
    GameObject bullet = Instantiate(bulletPrefab, firePoint.position, fireP  
    Rigidbody rb = bullet.GetComponent<Rigidbody>();  
    rb.velocity = firePoint.forward * bulletSpeed;  
}
```