

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique



MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Technologies de L'information et de La communication

THEME

**Conception et Réalisation d'un jeu de tir à la troisième
personne *FBG* « Forest Battle Ground »**

Présenté par :

OUANOUGH IMENE

LAIEB HEYTHEM

Soutenu publiquement le : jj/06/2023

Devant le jury composé de:

Président : Pr. Mostefai Messaoud

Examineur : Dr. Benouada Nadjib

Encadrante : Dr. Boutouhami Sara

2022/2023

Dédicace

Je dédie ce mémoire de fin d'études à ma maman, « Tria Rachida Nadia », qui m'a soutenu et encouragé tout au long de mes années d'études et plus spécialement durant cette période intense de travail. Tes encouragements et ton amour inconditionnel ont été ma source de force. Que tu trouves ici le témoignage de ma profonde reconnaissance.

Je souhaite exprimer ma reconnaissance à mes tantes, Tria Naima et Soria, ainsi qu'à leurs maris et enfants, pour leur soutien et leur encouragement constants.

Imene

À mes parents, Laieb Mabrouke et Ahlem, je dédie ce mémoire avec gratitude. Votre soutien inébranlable et vos sacrifices m'ont été essentiels pour dans la réalisation de ce travail.

Je tiens à remercier mon frère, Laieb Nidhal, et ma sœur, Laieb Ines, ainsi que tous ceux qui ont partagé les moments émotionnels de ce parcours. Votre présence et votre soutien m'ont énormément encouragé à poursuivre chemin.

Heythem

Une dédicace spéciale va à la famille Laieb et Tria, ainsi qu'à nos proches, pour leur amour et leur encouragement tout au long de cette aventure académique.

Nous souhaitons adresser nos remerciements à tous nos amis qui nous ont soutenu et encouragé. Votre amitié a rendu cette expérience encore plus précieuse.

Que cette dédicace témoigne de ma reconnaissance envers tous ceux qui ont joué un rôle important dans la réalisation de ce mémoire de fin d'études.

Remerciement

Tout d'abord, nous tenons à remercier chaleureusement notre directrice de mémoire, Madame Boutouhami Sara, pour son encadrement, ses conseils avisés, son soutien constant tout au long de notre recherche. Sa présence et ses orientations précieuses ont été d'une aide inestimable.

Nous sommes également reconnaissants envers Monsieur Benouada Nadjib pour ses conseils durant notre travail de recherche. Ses connaissances et ses recommandations ont grandement enrichi notre étude.

Nous remercions chaque membre du jury pour avoir accepté de consacrer leur temps et leurs connaissances afin d'évaluer notre travail.

À nos enseignants, nous adressons nos remerciements les plus sincères et notre reconnaissance pour leur contribution précieuse à notre formation académique.

Nous adressons nos sincères remerciements à nos très chers parents, qui ont été une source inépuisable d'amour, de soutien et de motivation tout au long de notre parcours universitaire. Leur présence et leurs encouragements ont été une force essentielle dans notre réussite.

Enfin, nous tenons à remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire. Que ce soit par leurs encouragements, leur soutien logistique ou moral, vous avez tous joué un rôle important dans notre parcours académique.

Résumé

Le but de notre travail est la création d'un jeu de tir à la troisième personne dans deux milieux différents principalement une forêt et un deuxième enneigé. Le jeu comporte les deux modes de jeu en solo et le mode multijoueur à ce jeu. Le choix d'un environnement de forêt pour un jeu de tir à la troisième personne est motivé par la richesse des éléments visuels et interactifs qu'un tel environnement peut offrir. La nature même de la forêt est un terrain de jeu riche en éléments visuels et sonores, qui offre de nombreuses possibilités de gameplay intéressantes, telles que la furtivité, la chasse, la collecte de ressources, ainsi que la planification et l'exécution de tactiques de combat.

Nous avons opté pour l'utilisation d'Unreal Engine pour de nombreuses raisons. Unreal Engine est connu pour sa qualité graphique exceptionnelle, sa flexibilité et sa robustesse. La prise en charge multiplateforme, ce qui signifie que le jeu peut être développé pour une grande variété de plates-formes, y compris les ordinateurs, les consoles de jeux, les appareils mobiles et même les casques de réalité virtuelle. Aussi notre choix s'est porté sur le développement d'un jeu de tir sur PC plutôt que sur mobile pour plusieurs raisons telle que les performances graphique.

Mots clés : Jeu de tir à la troisième personne, Mode solo, Mode multijoueur, Unreal Engine, Blueprint.

Abstract

The goal of our work is to create a third-person shooter game set in two different environments, mainly a forest and a snowy one. The game includes both single-player and multiplayer modes. The choice of a forest environment for a third-person shooter game is motivated by the richness of visual and interactive elements that such an environment can offer. The very nature of a forest provides a playground rich in visual and auditory elements, offering numerous interesting gameplay possibilities such as stealth, hunting, resource gathering, as well as planning and executing combat tactics.

We have opted to use Unreal Engine for several reasons. Unreal Engine is known for its exceptional graphical quality, flexibility, and robustness. It supports multi-platform development, which means the game can be developed for a wide variety of platforms, including computers, game consoles, mobile devices, and even virtual reality headsets. Additionally, we have chosen to develop a PC shooter game instead of a mobile one for several reasons, such as graphical performance.

Keywords: Third-person shooter, Single-player mode, Multiplayer mode, Unreal Engine, Blueprint.

ملخص

الهدف من عملنا هوا انشاء لعبة اطلاق نار من منظور الشخص الثالث في بيئتين مختلفتين الأولى عبارة عن غابة و الأخرى منطقة مغطاة بالثلوج، تتضمن اللعبة وضعين هما اللعب الفردي و اللعب الجماعي

اختيار بيئة الغابة محفز وذلك لوجود عناصر مرئية وتفاعلية التي يمكن ان توفرها مثل هذه البيئة. يوجد أيضا في مثل هذه البيئة عناصر مرئية و سمعية مختلفة حيث انها توفر العديد من الفرص المثيرة للعب مثل الاختباء، الصيد، وجمع الموارد وكذلك التخطيط و تنفيذ التكتيكات الخاصة بالقتال

لقد اخترنا Unreal Engine لعدة أسباب فهو يعرف بجودته الرسومية الاستثنائية و مرونته وقوته لأنه يدعم التطوير

مما يعني أنه يمكن تطوير اللعبة لمجموعة واسعة من المنصات، بما في ذلك الكمبيوترات وأجهزة الألعاب والأجهزة المحمولة وحتى نظارات الواقع الافتراضي. كما اخترنا تطوير لعبة إطلاق النار على أجهزة الكمبيوتر بدلاً من الهواتف المحمولة لأسباب عديدة

الكلمات المفتاحية: لعبة إطلاق نار من منظور الشخص الثالث، وضع لاعب واحد، وضع اللعب الجماعي، محرك . ، النموذج التصميم Unreal

Table des matières

Liste des abréviations	x
Liste des figures.....	xi
Introduction générale	1
1. Contexte et problématique.....	1
2. Objectifs et contribution	1
3. Plan du mémoire	3
Chapitre 01 : Les jeux vidéo	5
1.1. Introduction.....	5
1.2. Jeux vidéo.....	5
1.2.1. Histoire des jeux vidéo	5
1.2.2. Types de jeux vidéo.....	6
1.2.3. Réalisation d'un jeu vidéo	7
1.3. Moteur de jeu	8
1.3.1. Exemples de moteurs de jeu populaires	9
1.3.2. Choix du moteur de jeu	10
1.3.3. Gameplay dans les jeux de tir à la troisième personne	11
1.3.3.1. Le système de couverture	11
1.3.3.2. Les mécanismes de visée	11
1.3.4. Les difficultés à surmontées.....	12
1.4. Conclusion	13
Chapitre 02 : Conception du jeu FBG	14
2.1. Introduction.....	14
2.2. Synopsis du jeu	14
2.2.1. Les différents armes	15
2.3. Document de conception du jeu FBG.....	16
2.3.1. Nom du jeu	16

2.3.2.	Nom du Héros	16
2.3.3.	Motivation du joueur.....	16
2.3.4.	Caractéristiques	16
2.3.5.	Plateforme ciblée	17
2.3.6.	Les contrôles et les interactions.....	17
2.3.7.	Conception graphique : modélisation des éléments du jeu	17
2.3.7.1.	Les personnages.....	17
2.3.7.2.	Les armes	20
2.3.7.3.	Les niveaux.....	21
2.4.	Les modes de jeu FGB	23
2.4.1.	Le mode solo.....	23
2.4.2.	Le mode solo.....	23
2.5.	Conception informatique.....	24
2.5.1.	Les types d'objets manipulés dans le jeu FBG	25
2.5.2.	Conception informatique du jeu FBG dans Unreal	26
2.6.	Conclusion	31
Chapitre 03 : Réalisation du jeu FBG.....		32
3.1.	Introduction.....	32
3.2.	Environnement et outils de développement.....	32
3.2.1.	Environnement et outils de développement.....	32
3.2.1.1.	Moteur Unreal Engine.....	32
3.2.1.2.	Fonctionnalités et caractéristiques d'Unreal	33
3.2.1.3.	Epic Games	34
3.2.1.4.	Visuel Scripting 'Blueprint'	35
3.2.1.5.	L'intelligence artificielle.....	36
3.3.	Les interfaces du jeu FBG	37
3.3.1.	Menu principal.....	37
3.3.2.	FBG Lobby.....	38
3.3.3.	Actions des touches (Actions mappings).....	41
3.4.	Les blueprints.....	42
3.4.1.	Game instance blueprint.....	42
3.4.2.	Player controller blueprint.....	43

3.4.3. Widget blueprint character	44
3.4.4. Widget blueprint in Game	47
3.4.5. Widget blueprint Lobby	48
3.4.6. AI blueprint	51
3.5. Les scripts C++	51
3.6. Conclusion	54
Conclusion générale et perspectives	55
Bibliographie	58

Liste des abréviations

FBG : Forest Battle Ground

IA : intelligence Artificielle

2D : Deux Dimensions

3D : Trois Dimensions

UE : Unreal Engien

UI : Interface d'Utilisateur

TPS: Third Person Shooter

FPS: First Person Shooter

PNJ: Personnage Non Joueur

UML: Unified Modeling Language

PC: Personal Computer

HUD: Heads Up Display

PvP: Person Vs Person

Liste des figures

Figure 1. La forêt et le milieu enneigé.....	15
Figure 2 : Le Héro (King et Queen).....	18
Figure 3. Les différents matériels de la modélisation du personnage héros (King et Queen)	18
Figure 4. Les différentes couleurs de l'uniforme du héros	19
Figure 5. Le personnage ennemi.....	19
Figure 6. La conception d'un pistolet en tant qu'arme	20
Figure 7. La conception d'un AR Rifle en tant qu'arme.....	21
Figure 8. La conception d'un Shotgun en tant qu'arme.....	21
Figure 9. Les niveaux du jeu FBG de 1 à 10	22
Figure 10. Les XP	22
Figure 11. Le jeu en mode solo.....	23
Figure 12. Le jeu en mode multi-joueurs.....	24
Figure 13. Exemple d'environnement composé d'objets statique	25
Figure 14. Exemple d'objets dynamiques formant une image	25
Figure 15. Hiérarchie de quelques classes importantes dans le jeu FBG	26
Figure 16. Diagramme de classe de caractère FBG.....	29
Figure 17. Diagramme de classe de la classe Héro dans le jeu FBG.....	30
Figure 18. Diagramme de cas d'utilisation de l'utilisateur.....	31
Figure 19. Logo d'Unreal Engine	33

Figure 20. Logo d'Epic Games	34
Figure 21. Exemple de Blueprint	35
Figure 22. Un Design de "Intelligence Artificielle" AI	36
Figure 23. Le menu principal « FBG Main Menu »	37
Figure 24. Main Menu Design	38
Figure 25. FBG Lobby	39
Figure 26. FBG Lobby Design	39
Figure 27. Le démarrage du jeu	40
Figure 28. Un Design "In Game"	40
Figure 29. Les actions de mappings.....	41
Figure 30. Partie du blueprint 'Game instance'	42
Figure 31. Partie du blueprint 'player controller'	44
Figure 32. Partie du widget blueprint Character	46
Figure 33. Partie de widget blueprint in Game	48
Figure 34. Partie du widget blueprint Lobby	50
Figure 35. AI blueprint	51
Figure 36. Le code C++ Build .Cs	51
Figure 37. Le code C++ Engine ini.....	52
Figure 38. Le code C++ interface Session	52
Figure 40. Le code C++ Target Cs	53

Introduction générale

1. Contexte et problématique

Les jeux vidéo sont devenus une forme populaire de divertissement dans le monde entier, offrant des expériences interactives et immersives aux joueurs. Les jeux de tir à la troisième personne en particulier, offrent une expérience unique en permettant aux joueurs de contrôler un personnage dans un environnement virtuel tout en offrant une vue de derrière. L'objectif de notre projet est de créer un jeu de tir à la troisième personne qui offrira une expérience de jeu immersive et divertissante aux joueurs. Ce jeu sera développé en utilisant des technologies modernes et des méthodes de développement de jeux avancées pour assurer une jouabilité fluide et une qualité graphique élevée. Ce projet nous permettra d'explorer et de relever les défis associés au développement de jeux de tir à la troisième personne, tels que la conception de niveaux, le gameplay et les mécanismes de jeu. En développant ce jeu, nous pourrions explorer des méthodes intéressantes pour générer des niveaux de manière procédurale, créer une intelligence artificielle pour les ennemis, et implémenter des mécanismes de jeu qui offrent une expérience de jeu plus variée et intéressante.

Ce projet de développement de jeu de tir à la troisième personne est une opportunité passionnante de créer une expérience de jeu immersive et divertissante pour les joueurs.

2. Objectifs et contribution

Le but de notre travail est la création d'un jeu de tir à la troisième personne dans deux milieux différents principalement une forêt et un deuxième milieu enneigé. Le jeu comporte les deux modes de jeu : Le mode solo et le mode multi-joueurs. Dans notre jeu deux modes de jeu sont disponibles : le multijoueur "P vs P"(Person VS Person) et le mode solo. Nous avons deux environnements : un milieu en forêt et un deuxième milieu enneigé.

Le choix s'est porté en premier temps sur un environnement de forêt pour un jeu de tir à la troisième personne. Ce choix est motivé par la richesse des éléments visuels et interactifs qu'un tel environnement peut offrir. Les forêts sont des environnements naturels riches en végétation, en relief, en faune et en flore. Ce type d'environnement est donc propice à l'exploration et à l'aventure, tout en offrant un terrain de jeu intéressant pour les joueurs. Une forêt peut offrir une grande variété d'éléments visuels et sonores qui peuvent contribuer à créer une atmosphère immersive et engageante pour les joueurs. Par exemple, les joueurs pourront être immergés dans des scènes de forêt denses avec des sons de bruissement des feuilles, des oiseaux, et des bruits d'animaux.

En choisissant un environnement de forêt pour notre jeu de tir à la troisième personne, nous pouvons créer un monde virtuel captivant et réaliste pour les joueurs. De plus, la nature même de la forêt offre de nombreuses possibilités de gameplay intéressantes, telles que la furtivité, la chasse, la collecte de ressources, ainsi que la planification et l'exécution de tactiques de combat. Nous avons ensuite développé le jeu sur un autre environnement complètement différent c'est le milieu enneigé qui offre une autre expérience aussi intéressante que le milieu de la forêt.

Nous avons opté pour l'utilisation de Unreal Engine pour de nombreuses raisons : UE est un moteur de jeu très populaire et utilisé par de nombreux développeurs de jeux professionnels. Il est connu pour sa qualité graphique exceptionnelle, sa flexibilité et sa robustesse. La prise en charge multiplateforme, ce qui signifie que le jeu peut être développé pour une grande variété de plates-formes, y compris les ordinateurs, les consoles de jeux, les appareils mobiles et même les casques de réalité virtuelle. C'est pourquoi le choix d'Unreal Engine pour le développement de ce jeu est un choix judicieux et efficace.

Aussi notre choix s'est porté sur le développement d'un jeu de tir sur PC plutôt que sur mobile pour les raisons suivantes :

- Performances graphiques : Les jeux de tir peuvent être très exigeants en termes de performances graphiques, nécessitant une grande puissance de traitement et de graphismes pour offrir une expérience de jeu fluide et immersive.

Les PC sont souvent plus puissants que les appareils mobiles en termes de matériel, ce qui permet aux développeurs de créer des jeux avec des graphismes plus avancés et des effets visuels avancés.

- Contrôles : Les jeux de tir nécessitent souvent des contrôles précis et réactifs pour offrir une expérience de jeu satisfaisante. Les PC ont souvent des claviers et des souris, qui sont plus adaptés aux contrôles de jeu précis et réactifs que les écrans tactiles des appareils mobiles.
- Public cible : Les joueurs de jeux de tir sont souvent plus nombreux sur PC que sur mobile. Les joueurs de PC sont souvent des joueurs plus hardcore, qui cherchent des jeux avec des graphismes avancés, des contrôles précis et des expériences de jeu plus immersives.

Notre projet porte sur la réalisation d'un jeu de tir à la troisième personne FBG. Une fois le jeu téléchargé, l'utilisateur accède à l'interface où il se connecte en cliquant sur des boutons et en sélectionnant son personnage parmi le choix du mâle ou de femelle. Il a également la possibilité de choisir l'environnement de jeu, soit la forêt (Forest) soit l'environnement enneigé (Snowfloor). Notre héros utilise diverses armes telles qu'un pistolet pour éliminer ses ennemis. À mesure qu'il élimine des ennemis, le joueur accumule de bonnes notes (XP) qui lui permettent de progresser vers des niveaux supérieurs nous avons dix niveaux de Niv10. Dans notre jeu, l'utilisateur a également la possibilité de définir un temps " 3 à 15 minute".

3. Plan du mémoire

Le mémoire se structure de la manière suivante :

Le chapitre 1 explore le monde des jeux vidéo, leur histoire, les différents types de jeux, la réalisation d'un jeu vidéo, les moteurs de jeu et l'importance du scripting et des blueprints.

Le chapitre 2 est consacré à la modélisation et à la conception du système à travers la présentation des différents diagrammes UML.

Dans le chapitre 3, nous présentons les différents outils et langages de programmation que nous avons utilisés pour réaliser notre jeu ainsi que ces différentes interfaces.

Finalemant, nous terminons le mémoire par une conclusion générale qui résume les contributions de notre travail ainsi que certaines perspectives futures.

Chapitre 01 : Les jeux vidéo

1.1. Introduction

Un jeu vidéo est un type de jeu électronique qui se joue sur un ordinateur, une console de jeux, un smartphone, une tablette ou toute autre plateforme de jeu électronique. Les jeux vidéo peuvent être joués seul ou en ligne avec d'autres joueurs, et offrent souvent une expérience interactive et immersive. Les jeux vidéo peuvent varier en genre, en style de jeu, en mécaniques de jeu et en durée de jeu. Les genres courants incluent les jeux de tir à la première personne, les jeux de rôle, les jeux de stratégie en temps réel, les jeux de sport, les jeux de simulation, les jeux de course et les jeux de puzzle. Les jeux vidéo sont devenus une forme de divertissement populaire, attirant des millions de joueurs du monde entier. Ils ont également évolué en tant que média artistique, avec des jeux vidéo primés pour leur narration, leur conception de jeu, leur graphisme et leur bande sonore. Les tournois de jeux vidéo ont également émergé, offrant aux joueurs professionnels l'opportunité de rivaliser pour des prix en argent.

Dans ce chapitre nous allons explorer le monde des jeux vidéo, leur histoire, les différents types de jeux, la réalisation d'un jeu vidéo, les moteurs de jeu et l'importance du scriptage et des blueprints.

1.2. Jeux vidéo

1.2.1. Histoire des jeux vidéo

L'histoire des jeux vidéo remonte aux années 1950, lorsque les premiers ordinateurs ont été développés. Cependant, ce n'est qu'au début des années 1970 que les premiers jeux vidéo ont été commercialisés. Le jeu Pong, sorti en 1972, est souvent considéré comme le premier jeu vidéo grand public. Il a été suivi par des jeux tels que Space Invaders, Pac-Man et Donkey Kong, qui sont devenus des classiques du jeu vidéo (1).

Au fil des années, les jeux vidéo ont évolué et se sont développés en tant que forme d'art et de divertissement de plus en plus populaires. Les années 1980 ont vu l'essor des jeux de rôle

(RPG) et des jeux d'aventure, tandis que les années 1990 ont vu l'arrivée des jeux de tir à la première personne (FPS) tels que Doom et Quake..

Dans les années 2000, les jeux vidéo ont connu une explosion de popularité avec l'arrivée des consoles de jeu de nouvelle génération telles que la Xbox, la PlayStation et la Nintendo Wii.

Les jeux vidéo sont devenus de plus en plus sophistiqués et réalistes, offrant aux joueurs des expériences de plus en plus immersives.

Aujourd'hui, les jeux vidéo sont une industrie en plein essor, générant des milliards de dollars de revenus chaque année. Les jeux sont disponibles sur une variété de plates-formes, notamment les consoles de jeu, les ordinateurs, les téléphones portables et les tablettes. Ils sont devenus un élément culturel important, influençant la musique, les films et la mode. L'histoire des jeux vidéo est riche et complexe, et continue d'évoluer à mesure que de nouvelles technologies et de nouveaux styles de jeu sont développés. C'est un domaine fascinant et en constante évolution qui a une grande influence sur la culture populaire (2).

1.2.2. Types de jeux vidéo

Les jeux vidéo peuvent être classifiés de différentes manières en fonction de leur genre, de leur plateforme, de leur public cible, de leur mode de jeu, de leur mécanique de jeu, de leur objectif, de leur niveau de difficulté, de leur histoire, de leur graphisme, de leur technologie, de leur année de sortie, etc. Il n'y a pas de classification universelle ou définitive des jeux vidéo, mais plusieurs systèmes de classification existent en fonction des besoins et des contextes (3) (4).

- Action : ce genre met l'accent sur les compétences physiques et la réactivité des joueurs, qui doivent réagir rapidement à des stimuli pour gagner.
- Aventure : ce genre se concentre sur l'exploration et la découverte, souvent avec une histoire complexe et des personnages intéressants.
- Stratégie : ce genre met l'accent sur la planification, la gestion de ressources et la prise de décisions stratégiques pour atteindre des objectifs spécifiques.

- Simulation : ce genre vise à reproduire un environnement ou une situation réelle avec un haut degré de précision, souvent utilisé dans les jeux de simulation de vol, de conduite, de gestion de ville, etc.
- Jeux de rôle : ce genre implique que les joueurs jouent le rôle de personnages et les font évoluer dans un monde virtuel, souvent avec un système de progression de niveaux.
- Sports : ce genre se concentre sur la simulation de sports réels ou fictifs, où les joueurs peuvent concourir contre des adversaires ou des équipes contrôlées par l'ordinateur.
- Jeux de tir : ce genre se concentre sur la combat, où les joueurs doivent tirer sur des ennemis ou des cibles, souvent en utilisant des armes fictives ou réelles.

Voici les chiffres d'affaires des plus grandes entreprises de jeux vidéo en 2021, selon les données publiées par Statista (5):

- Tencent - 83,2 milliards de dollars
- Sony Interactive Entertainment - 24,4 milliards de dollars
- Microsoft - 15,5 milliards de dollars
- Apple - 14,8 milliards de dollars
- Activision Blizzard - 8,09 milliards de dollars
- Google - 7,6 milliards de dollars
- Electronic Arts - 6,19 milliards de dollars
- NetEase - 5,95 milliards de dollars
- Nintendo - 5,81 milliards de dollars
- Bandai Namco - 5,74 milliards de dollars

1.2.3. Réalisation d'un jeu vidéo

Selon Ernest W. Adams un célèbre game designer et consultant en développement de jeux vidéo (considéré comme une référence dans le domaine). Selon lui, la réalisation d'un jeu vidéo nécessite plusieurs étapes, allant de la conception à la commercialisation. Voici les grandes étapes qu'il recommande (6) (2) (7) :

- La conception : Cette étape consiste à définir le concept du jeu, ses objectifs, ses mécaniques de jeu, son univers, ses personnages, son histoire, etc. Il s'agit d'établir une vision claire du jeu et de définir les bases de son développement.
- La pré-production : Cette étape consiste à réaliser les premiers prototypes du jeu, à créer des maquettes et des concepts artistiques, à définir les spécifications techniques, à établir un planning de développement et à recruter les membres de l'équipe de développement.
- La production : Cette étape est la plus longue et la plus importante. Elle consiste à créer le jeu à proprement parler, en développant les éléments de gameplay, les graphismes, les sons, les musiques, les dialogues, etc. Il faut également tester régulièrement le jeu pour s'assurer qu'il fonctionne correctement et qu'il est plaisant à jouer.
- La post-production : Cette étape consiste à finaliser le jeu en le peaufinant, en corrigeant les bugs, en améliorant les performances, en ajoutant des fonctionnalités, etc. Il s'agit également de préparer le jeu pour sa commercialisation en créant des campagnes de marketing, en négociant avec les éditeurs, en créant une communauté de joueurs, etc.
- La commercialisation : Cette étape consiste à mettre le jeu sur le marché, en le publiant sur des plateformes de distribution en ligne ou en le proposant à des éditeurs. Il s'agit également de créer des stratégies de promotion pour attirer les joueurs, de gérer les relations avec les médias et les critiques, et de suivre les ventes et les retours des joueurs pour améliorer le jeu ou proposer des suites.

1.3. Moteur de jeu

Le moteur de jeu est un logiciel spécialisé qui fournit les outils et les fonctionnalités nécessaires pour développer un jeu vidéo. C'est un ensemble de bibliothèques, d'outils et de systèmes intégrés qui facilitent le processus de développement d'un jeu vidéo. Il fournit une base solide pour la création des graphismes, de la physique, de l'intelligence artificielle, de l'audio, de l'animation, de la gestion des ressources et d'autres éléments clés d'un jeu (8) (9).

Les fonctionnalités courantes d'un moteur de jeu comprennent :

- Graphismes : le moteur de jeu offre des capacités de rendu graphique avancées, permettant la création de mondes virtuels, de modèles 3D, d'effets visuels et de textures.
- Physique : il permet la simulation de lois physiques réalistes, comme la gravité, les collisions, la cinématique inverse, etc.
- Intelligence artificielle : le moteur de jeu propose des outils et des algorithmes pour la création de comportements et de décisions autonomes pour les personnages non-joueurs.
- Audio : il permet la gestion des effets sonores, de la musique et du mixage audio pour une expérience immersive.
- Animation : le moteur de jeu prend en charge les animations des personnages et des objets, qu'il s'agisse d'animations préenregistrées ou générées en temps réel.
- Gestion des ressources : il facilite le chargement, la gestion et l'optimisation des ressources du jeu, tels que les modèles 3D, les textures, les sons, etc.

1.3.1. Exemples de moteurs de jeu populaires

Il existe de nombreux moteurs de jeu disponibles sur le marché, chacun avec ses propres caractéristiques et avantages (10) (11).

Voici quelques exemples de moteurs de jeu populaires :

- Unity : c'est l'un des moteurs de jeu les plus utilisés, offrant une grande flexibilité et une compatibilité multiplateforme.
- Unreal Engine : ce moteur de jeu puissant est connu pour ses graphismes de haute qualité et ses outils de développement avancés.
- CryEngine : il se distingue par ses capacités graphiques impressionnantes et son rendu réaliste.
- Godot : un moteur de jeu open-source et léger, idéal pour les projets de petite à moyenne envergure.
- GameMaker Studio : un moteur de jeu facile à prendre en main, adapté aux développeurs débutants ou à ceux qui privilégient la simplicité.

1.3.2. Choix du moteur de jeu

Le choix du moteur de jeu dépend des besoins spécifiques du projet. Les critères tels que la plateforme de déploiement, la complexité du jeu, les compétences de l'équipe de développement, le budget et les performances souhaitées doivent être pris en compte. Il est important de bien évaluer les fonctionnalités offertes par chaque moteur de jeu et de considérer les ressources disponibles avant de faire un choix (7) (12) (10).

Dans la suite de ce mémoire, nous nous concentrerons sur l'utilisation d'Unreal Engine pour la réalisation de notre jeu vidéo. Nous explorerons les fonctionnalités spécifiques de ce moteur et les raisons qui ont motivé notre choix.

Le scriptage et les blueprints sont des aspects importants du développement de jeux vidéo, notamment dans le cadre de la programmation des mécanismes de jeu et des comportements des personnages.

- Le scriptage : Le scriptage est le processus de création de scripts ou de lignes de code qui contrôlent le fonctionnement du jeu. Il permet de définir les règles du jeu, de gérer les interactions entre les différents éléments du jeu, de mettre en place des mécanismes de jeu spécifiques et bien plus encore. Le scriptage peut se faire dans différents langages de programmation, tels que C++, C#, Python, JavaScript, etc., en fonction du moteur de jeu utilisé.
- Les blueprints : Les blueprints, quant à eux, sont un système de programmation visuelle qui permet aux développeurs de créer des fonctionnalités et des comportements de jeu sans avoir à écrire du code. Les blueprints utilisent une interface graphique conviviale qui permet de connecter des nœuds représentant des actions et des conditions pour créer des séquences logiques. Cela facilite la création rapide de prototypes, l'itération et la collaboration entre les membres de l'équipe de développement (13).

Dans Unreal Engine, par exemple, le système Blueprint permet de créer des scripts visuels en utilisant des nœuds et des connexions pour définir les comportements des objets, des personnages, des ennemis, etc. Il permet aux développeurs de définir des événements déclencheurs, des conditions, des actions et des réactions pour rendre le jeu interactif et réactif.

L'utilisation des blueprints présente plusieurs avantages, notamment une approche visuelle qui facilite la compréhension et l'itération, un gain de temps dans le développement, la possibilité d'impliquer des membres de l'équipe qui n'ont pas de compétences en programmation, et une flexibilité pour expérimenter et ajuster rapidement les mécaniques de jeu. Le scriptage et l'utilisation des blueprints sont des aspects essentiels du développement de jeux vidéo. Ils permettent de créer des mécanismes de jeu, de gérer les interactions et de définir les comportements des personnages.

1.3.3. Gameplay dans les jeux de tir à la troisième personne

Le gameplay d'un jeu de tir à la troisième personne est un élément central qui définit l'expérience de jeu et engage les joueurs dans des actions captivantes. Il se compose de différents aspects, dont deux éléments clés sont particulièrement significatifs : le système de couverture et les mécanismes de visée (13) (1).

1.3.3.1. Le système de couverture

Une caractéristique essentielle des jeux de tir à la troisième personne est la possibilité pour les joueurs de se mettre à l'abri derrière des objets ou des murs pour éviter les tirs ennemis. Ce système de couverture offre aux joueurs un moyen stratégique de se protéger tout en planifiant leurs actions. En utilisant des mécanismes de déplacement fluides, les joueurs peuvent passer rapidement d'une couverture à l'autre, créant ainsi des moments de tension et de prise de décision tactique. Le système de couverture permet également de jouer avec des notions de risque et de récompense, où les joueurs doivent évaluer le moment opportun pour sortir de leur couverture et attaquer leurs ennemis.

1.3.3.2. Les mécanismes de visée

Les jeux de tir à la troisième personne impliquent des mécanismes de visée précis pour permettre aux joueurs de cibler avec précision leurs ennemis. En utilisant des curseurs ou des viseurs à l'écran, les joueurs peuvent ajuster leur visée pour atteindre des ennemis spécifiques ou des points faibles. Certains jeux offrent des mécanismes de visée avancés, tels que le tir à la tête

pour infliger des dégâts accrus. Les mécanismes de visée bien calibrés et réactifs sont essentiels pour offrir aux joueurs une expérience de jeu satisfaisante et immersive.

Ces deux éléments, le système de couverture et les mécanismes de visée, travaillent de concert pour créer des moments d'action intenses et stratégiques. Les joueurs doivent planifier leurs mouvements, choisir judicieusement leurs positions de couverture et viser avec précision pour réussir leurs attaques. Cette combinaison de stratégie, d'adresse et de réflexes fait du gameplay des jeux de tir à la troisième personne une expérience dynamique et engageante (12).

1.3.4. Les difficultés à surmontées

Dans le développement d'un jeu de tir à la troisième personne, plusieurs difficultés peuvent se présenter. Voici quelques-unes des principales difficultés auxquelles les développeurs peuvent être confrontés et qu'ils doivent surmonter pour créer un jeu de qualité (6) (7) (13) :

- La conception des niveaux : Concevoir des niveaux captivants et bien équilibrés est un défi majeur dans le développement d'un jeu de tir à la troisième personne. Les développeurs doivent créer des environnements intéressants, avec des obstacles, des cachettes et des chemins diversifiés pour offrir aux joueurs une expérience de jeu engageante. Équilibrer la difficulté et le rythme du jeu pour maintenir l'intérêt des joueurs tout en évitant la frustration est un équilibre délicat à trouver.
- L'intelligence artificielle (IA) des ennemis : Créer une IA réaliste et compétente pour les ennemis est un défi important. Les ennemis doivent réagir de manière intelligente aux actions du joueur, prendre des décisions tactiques, utiliser des stratégies d'attaque et de défense, et offrir un défi satisfaisant. Le développement d'une IA complexe et réactive peut demander du temps et de l'expertise pour assurer des interactions ennemies convaincantes.
- L'équilibrage des armes et des compétences : Dans un jeu de tir à la troisième personne, l'équilibrage des armes et des compétences des personnages joue un rôle crucial dans l'expérience de jeu. Les développeurs doivent s'assurer que chaque arme a sa propre utilité et son impact spécifique, tout en évitant qu'une arme ne soit trop puissante ou

qu'une compétence ne soit trop dominante. Cela nécessite des tests rigoureux, des ajustements et un rééquilibrage constant pour assurer un gameplay équilibré et gratifiant.

- Les contrôles et la jouabilité : Offrir des contrôles fluides et réactifs est essentiel pour une expérience de jeu satisfaisante. Les développeurs doivent concevoir des mécanismes de contrôle intuitifs et faciles à prendre en main, en tenant compte des différentes plateformes de jeu et des périphériques disponibles. Ils doivent également
- s'assurer que la jouabilité du jeu est fluide, sans bugs ni problèmes de collision ou de mouvement qui pourraient altérer l'expérience de jeu

- Les performances et l'optimisation : Les jeux de tir à la troisième personne peuvent être visuellement exigeants, avec des environnements détaillés, des effets spéciaux et des animations complexes. Les développeurs doivent donc s'assurer que le jeu fonctionne de manière fluide et sans problème sur différentes configurations matérielles. Cela implique l'optimisation des performances, la réduction de la consommation des ressources système et la gestion efficace de la mémoire pour offrir une expérience de jeu stable et agréable.

1.4. Conclusion

Les jeux vidéo sont une forme de divertissement populaire qui continue d'évoluer et d'influencer la culture populaire. Leur réalisation implique un processus complexe et utilise plusieurs outils. Dans ce chapitre, nous avons exploré l'univers des jeux vidéo en examinant leur histoire, leurs différents types et le processus de réalisation d'un jeu vidéo. Nous avons également abordé le concept de moteur de jeu, en mettant l'accent sur ses fonctionnalités et en présentant quelques exemples populaires. Enfin, nous avons discuté du scriptage et des blueprints en tant qu'outils importants pour la programmation des mécanismes de jeu et des comportements des personnages.

Le chapitre suivant portera sur la conception de notre jeu de tir à la troisième personne en forêt FBG.

Chapitre 02 : Conception du jeu FBG

2.1. Introduction

Les jeux de tir à la troisième personne (TPS) sont des jeux vidéo où le joueur voit son personnage à l'écran en vue de dos, ou de côté, et doit le contrôler pour combattre des ennemis à l'aide d'armes à feu ou d'autres types d'armes. La conception d'un jeu a principalement comme objectif d'une part d'avoir une idée précise et solide du jeu qui sera créé, la manière dont il sera joué, les mécaniques récurrentes, du cœur de gameplay et de son scénario, et d'autre part de poser les bases de l'architecture logicielle sur laquelle il va s'articuler.

Dans ce chapitre nous allons introduire la conception de notre jeu FBG.

2.2. Synopsis du jeu

FBG est un jeu de tir à la troisième personne dans deux milieux différents principalement une forêt et un deuxième enneigé. Le jeu comporte les deux modes de jeu en solo et le mode multijoueur à ce jeu. Notre héros, que ce soit King ou Queen, est capable de vaincre ses ennemis grâce à une variété d'armes différentes. Il peut utiliser des pistolets, des AR rifles et des Shotgun qu'il ramasse dans les différents niveaux pour infliger des dégâts à ses ennemis.

Le monde dans lequel évolue notre héros est divisé en deux environnements totalement différents : une forêt dense et un paysage enneigé. Ces deux milieux sont remplis de défis qui mettent à l'épreuve les compétences de notre héros, et il devra être prêt à utiliser toutes les armes et compétences à sa disposition pour survivre.



Figure 1. La forêt et le milieu enneigé

2.2.1. Les différents armes

- **Pistolets** : Dans les jeux de tir, les pistolets sont des armes à feu courantes et populaires. Ils sont souvent utilisés comme arme secondaire par les personnages, en complément d'une arme principale plus puissante, comme un fusil d'assaut ou un fusil à pompe. Les pistolets sont généralement compacts, légers et faciles à manier, ce qui les rend utiles pour les combats rapprochés ou les situations où une arme plus lourde serait inutile ou encombrante
- **AR rifles** : Les AR rifles (ou fusils d'assaut) sont des armes à feu très populaires dans les jeux de tir. Ils sont conçus pour être utilisés à moyenne ou longue portée, et sont souvent utilisés comme arme principale par les personnages. Dans les jeux de tir, les AR rifles ont une capacité de munitions élevée, ce qui permet au joueur de tirer plusieurs coups avant de devoir recharger. Ils ont également un recul modéré, ce qui signifie que le joueur peut tirer rapidement tout en conservant une bonne précision. Les AR rifles sont souvent utilisés pour les combats à moyenne et longue portée, où leur précision et leur puissance leur donnent un avantage sur les ennemis.

- **Shotguns** : Les Shotguns (ou fusils à pompe) sont des armes à feu puissantes et très appréciées dans les jeux de tir. Ils sont généralement conçus pour les combats à courte portée, où leur puissance de feu leur donne un avantage sur les ennemis. Dans les jeux de tir, les Shotguns sont souvent utilisés pour éliminer rapidement plusieurs ennemis à courte portée.

Nous avons aussi un autre élément :

- **Les bonus** que le joueur obtient pendant les combats sont appelés XP (des points). Ces points permettent au joueur de progresser vers des niveaux supérieurs dans le jeu. Par exemple, lorsqu'il tue un ennemi, le joueur peut gagner 1000 XP. Grâce à ces points, il peut atteindre des niveaux plus avancés dans le jeu.

2.3. Document de conception du jeu FGB

2.3.1. Nom du jeu

- Forest Battle Ground

2.3.2. Nom du Héros

- Héros King et Queen

2.3.3. Motivation du joueur

Le joueur se bat dans la forêt ou dans le milieu enneigé contre des ennemis et utilise une variété d'armes pour les éliminer. À chaque ennemi vaincu, il gagne des points (XP) qui lui permettent de progresser vers des niveaux supérieurs.

2.3.4. Caractéristiques

- Un jeu totalement en 3D inspiré de jeux de tir avec des personnages détaillés et personnalisables.
- Un temps de jeu infini avec des sessions de 15 min pour chaque partie.
- Des dizaines d'améliorations et de tenues avec des couleurs différentes.
- Un jeu destiné à toutes les tranches d'âge.

- Des ennemis dotés des dernières avancées en intelligence artificielle.
- Des bonus (XP).
- 02 environnements totalement différents modélisés en 3D.

2.3.5. Plateforme ciblée

Le Jeu est disponible en Windows.

2.3.6. Les contrôles et les interactions

Le joueur peut interagir avec le caractère via des mécanismes et actions. La liste des actions que le joueur peut exécuter sont :

- Le joueur peut se déplacer dans l'univers.
- Le joueur peut sauter avec (ALT dash).
- Le joueur peut attaquer avec des armes.
- Le joueur peut récupérer les armes.

2.3.7. Conception graphique : modélisation des éléments du jeu

Le jeu FBG est principalement constitué des objets suivants :

- Les personnages, le héros (King ou Queen) et les ennemis.
- Les armes (pistolet, AR rifle et le Shotgun).
- Autres éléments des niveaux : maisons, arbres, neige ...etc.

2.3.7.1. Les personnages

• **Le héros** est le personnage jouable par l'utilisateur. Nous avons téléchargé son modèle graphique à partir de Unreal Engine. La modélisation est basée sur des paquets disponibles dans l'Epic Game (un store). La figure suivante représente le personnage héro (King or Queen)



Figure 2 : Le Héro (King et Queen)

Les options de personnalisation telles que le choix du matériel et des couleurs de l'uniforme sont possibles grâce à l'utilisation d'Unreal Engine.

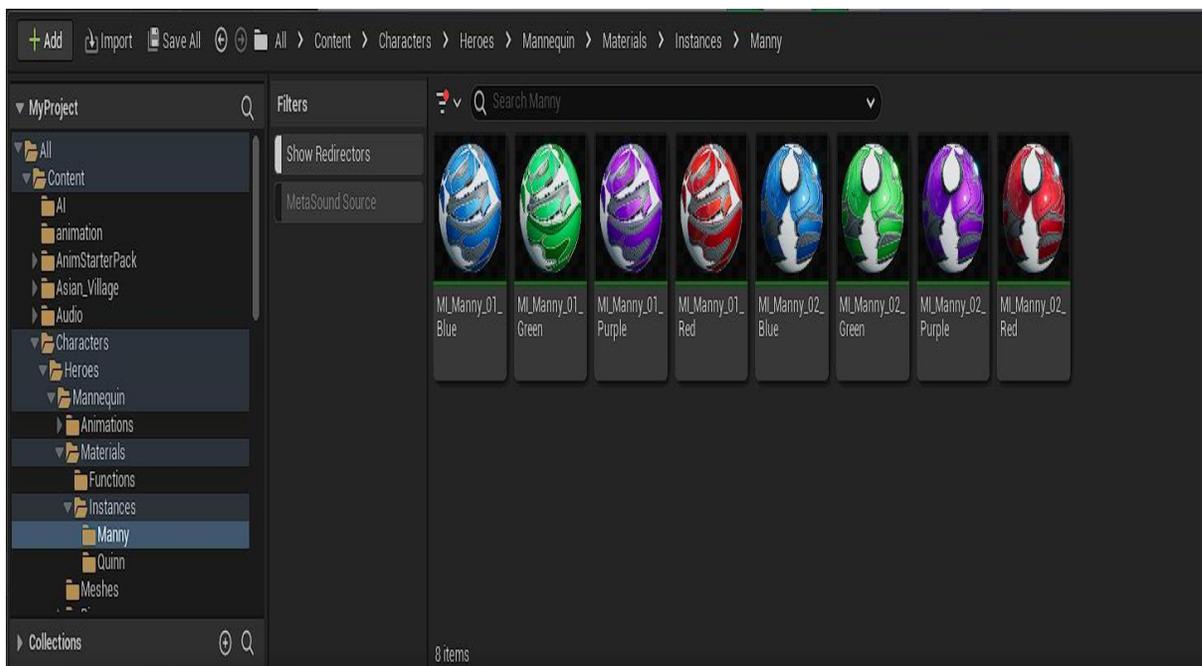


Figure 3. Les différents matériels de la modélisation du personnage héros (King et Queen)

À chaque niveau, la couleur de l'uniforme du personnage change. Par exemple, au premier niveau, seule la couleur bleue est disponible. En progressant jusqu'au deuxième niveau, le joueur a la possibilité de choisir librement une autre couleur pour son uniforme.



Figure 4. Les différentes couleurs de l'uniforme du héros

- **L'ennemi** peut prendre différentes formes, qu'il s'agisse d'une personne ou d'un monstre robotique. Toutefois, sa modélisation de base reste la même que celle du personnage Principal, ou du héros. La modification est sur l'animation.

Ce personnage est autonome et indépendant du joueur, et il est conçu pour être géré automatiquement par le système du jeu. On lui prévoit des capacités de combat ainsi que des habilités spécifiques.



Figure 5. Le personnage ennemi

2.3.7.2. Les armes

Toutes les armes utilisées par le joueur pour affronter les ennemis ont été obtenues à partir d'autres jeux créés avec le moteur de jeu Unreal Engine. Le moteur de jeu Unreal Engine est largement utilisé dans l'industrie des jeux vidéo en raison de ses capacités de développement de jeux rapides et efficaces. Les armes peuvent être conçues avec des modèles 3D détaillés et des textures réalistes, offrant des expériences de jeu immersives pour les joueurs. Le choix d'utiliser des armes provenant d'autres jeux peut offrir des avantages en termes de développement de jeux, car les armes peuvent être pré-conçues et optimisées pour une utilisation dans le moteur Unreal Engine. Cela nous a permis d'économiser du temps et des ressources, tout en offrant des expériences de jeu de qualité pour les joueurs.

a) Pistolet



Figure 6. La conception d'un pistolet en tant qu'arme

b) AR Rifle



Figure 7. La conception d'un AR Rifle en tant qu'arme

c) Shotgun



Figure 8. La conception d'un Shotgun en tant qu'arme

2.3.7.3. Les niveaux

Dans notre jeu FBG, il existe dix niveaux différents (niveaux 1 à 10). Chaque niveau correspond à un certain nombre de points (XP) que le joueur peut gagner en battant des ennemis. En atteignant un certain nombre de points, le joueur peut passer au niveau suivant et bénéficier

de divers avantages. Par exemple, en passant à un niveau supérieur, le joueur peut obtenir la possibilité de personnaliser la couleur de son uniforme.



Figure 9. Les niveaux du jeu FBG de 1 à 10



Figure 10. Les XP

2.4. Les modes de jeu FGB

Le jeu comporte deux modes de jeu : Le mode solo et Le mode multi-joueurs

2.4.1. Le mode solo

Le mode solo dans un jeu de tir à la troisième personne est un mode de jeu dans lequel le joueur peut jouer seul, sans la présence d'autres joueurs. Le joueur affronte des ennemis contrôlés par l'ordinateur et doit progresser dans le jeu en passant différents niveaux, accomplissant des objectifs et en surmontant des défis. Le mode solo permet aux joueurs de s'immerger dans l'histoire et le monde du jeu, en affrontant des adversaires contrôlés par l'ordinateur et en explorant des environnements sans avoir à se soucier des autres joueurs.

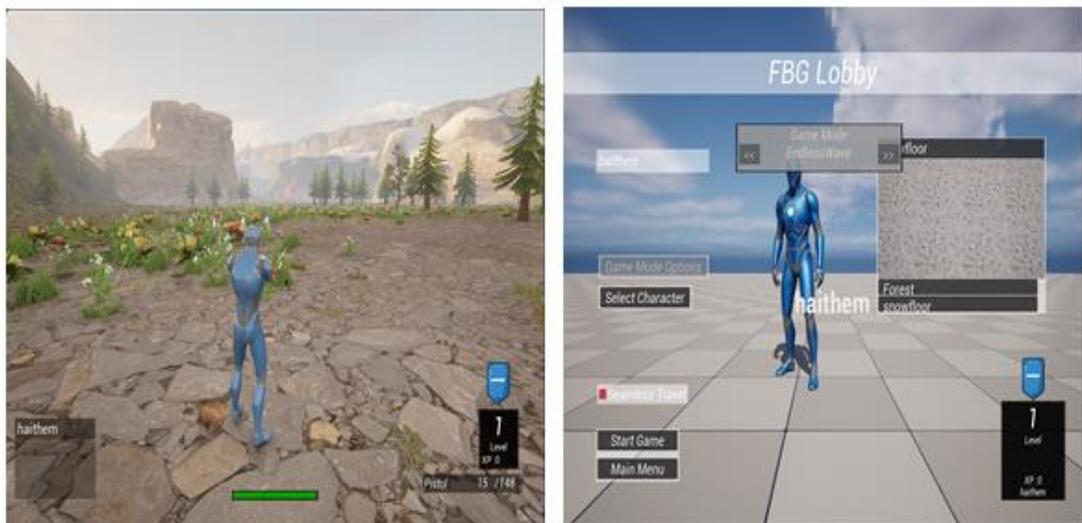


Figure 11. Le jeu en mode solo

2.4.2. Le mode solo

Le mode multijoueur dans un jeu de tir à la troisième personne est un mode de jeu dans lequel plusieurs joueurs peuvent jouer ensemble en temps réel. Dans ce mode, les joueurs peuvent s'affronter les uns les autres ou coopérer pour atteindre des objectifs communs. Les joueurs peuvent former des équipes et jouer contre d'autres équipes ou jouer individuellement en affrontant d'autres joueurs.

Le mode multijoueur permet aux joueurs de défier leurs amis ou des joueurs du monde entier, de tester leurs compétences et de découvrir de nouvelles stratégies pour progresser dans le jeu



Figure 12. Le jeu en mode multi-joueurs

2.5. Conception informatique

Dans cette partie, nous allons concevoir la hiérarchie des différentes classes du jeu qui représentent les différents objets utilisés dans la création du jeu FBG. Nous allons ensuite définir et présenter les diagrammes de classe de quelques objets du jeu ainsi que le diagramme de cas d'utilisation de l'utilisateur.

2.5.1. Les types d'objets manipulés dans le jeu FBG

Notre jeu se compose de plusieurs types d'objets, qui peuvent être classés en 2 catégories :

- **Objets statiques** : Ce sont des objets qui composent l'environnement du jeu, ils n'ont aucune interaction avec le joueur. Ces objets sont souvent issus de logiciels de modélisation 3D ou de logiciels 2D et sont placés dans les différents niveaux, à l'exemple des maisons, arbres, fleurs, Montagnes



Figure 13. Exemple d'environnement composé d'objets statique

- **Objets dynamiques** : ce sont tous les éléments qui ne font pas partie de l'environnement et qui accomplissent ou subissent des actions préalablement configurées. Ces objets sont souvent composés d'une partie visible qui est l'objet en soit et d'une partie implémentée qui est le code, qui définit les actions de cet objet.



Figure 14. Exemple d'objets dynamiques formant une image

2.5.2. Conception informatique du jeu FBG dans Unreal

Tous les objets utilisés dans le jeu FBG sont classifiés selon une classification établie par le moteur Unreal. Dans cette classification, les objets dynamiques sont représentés avec plusieurs classes, chaque classe dispose de ses propres caractéristiques. Toute ces classes sont étendues d'une classe mère appelé 'Object'. La figure suivante représente la hiérarchie des classes dynamiques qui définissent notre jeu. Cette Hiérarchie est obtenue par extension d'une hiérarchie de base proposée par le moteur de jeu Unreal.

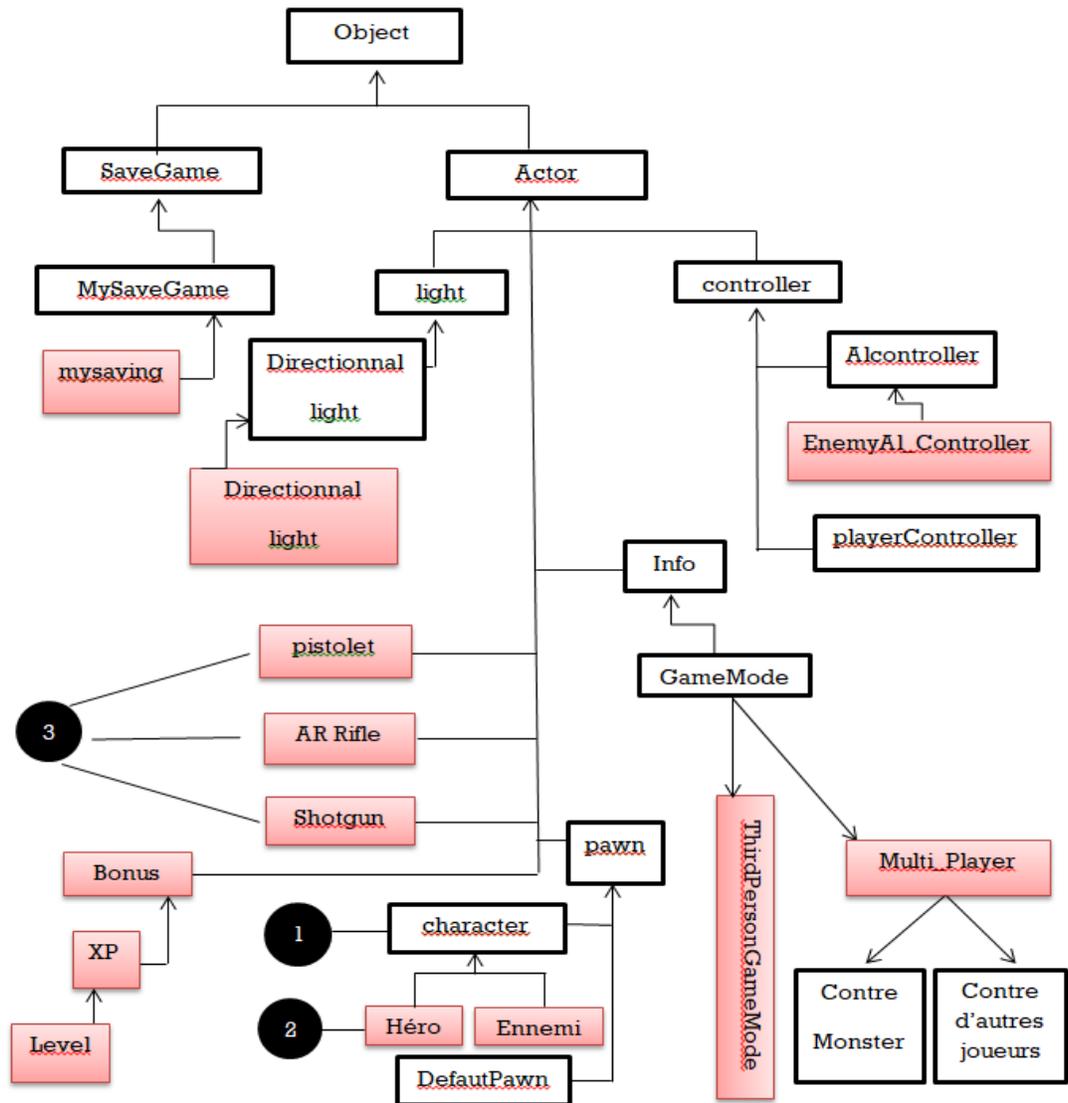


Figure 15. Hiérarchie de quelques classes importantes dans le jeu FBG

La classe la plus importante est la classe 'Actor', cette classe représente tous ce qui peut être placé, ou engendré par le moteur dans une partie et qui est contrôlé par l'utilisateur ou le moteur lui-même. Les éléments de la classe 'Actor' jouent des rôles dans la phase de jeu et plusieurs autres classes héritent de cette dernière :

1. **Classe Pawn** : Tous les éléments jouables par l'utilisateur ou l'AI, héritent de cette classe.
2. **Classe Character** : Classe fille de la classe Pawn, elle englobe tous les caractères ayant un mesh, un système de collision et une logique de mouvement pouvant être joués par le joueur ou l'AI.
 - **Héro** : C'est le caractère jouable par l'utilisateur.
 - **Ennemi** : C'est le caractère qui représente les ennemis du héros.
3. **Classe AIController et PlayerController** : Ce sont des classes non physique composées seulement de scripts, appelées respectivement par les caractères gérés par l'AI et le joueur pour exécuter certaines tâches avec des fonctions particulières comme 'EventPossess' et 'Event Unpossess' dès que L'AI ou le joueur prennent possession de leurs caractères ou perdent le control de ces derniers.
 - **EnemyAI_Controller** : C'est le contrôleur utilisé dans le jeu FBG pour permettre aux ennemis d'utiliser leurs arbres de décisions.
4. **Classe Light** : Comme son nom indique cette classe qui hérite de la classe Actor, elle gère toutes les lumières disponibles dans le moteur.
 - **Directionnal light** : Elle représente le système de lumière utilisé dans les différents niveaux pour simuler la lumière du soleil fixé.
5. **Classe Game Mode** : Cette classe hérite de la classe 'info'. Toutes les classes de cette catégorie ne sont pas physique et servent en premier lieu aux échanges, communication d'informations pendant les parties de jeu. La classe Game Mode est utilisé fréquemment pour définir les règles de la partie, gérer les scores des joueurs, et gérer les actions des joueurs. L'appel à une instance de cette classe se fait automatiquement lors de la création d'un niveau de jeu

- **ThirdPersonGameMode** : C'est une instance de la classe Game Mode utilisé dans le jeu.
 - **Multi_player** : Il s'agit d'une occurrence de la classe "Mode de jeu" qui sert à définir les deux modes de jeu disponibles : solo et multijoueurs.
- 6. Classe Xp** : C'est la classe qui représente tous les bonus disponibles dans le jeu. Il s'agit des points que le joueur acquiert en éliminant des ennemis.
- 7. Classe Level** : Cette classe hérite de la classe 'XP' Le joueur peut atteindre de nombreux niveaux en accumulant des points (XP) qu'il gagne en éliminant des ennemis.
- 8. Classes pistolet, Shotgun, AR Rifle** : sont les classes qui représentent les armes utilisées par le caractère du héros ainsi que par l'ennemi, dans le but d'infliger des dégâts.

La classe 'SaveGame' hérite directement de la classe 'Object', elle permet de créer des sauvegardes persistantes sur le support de stockage de l'ordinateur. La classe Mysaving est l'instance de la classe SaveGame dans FBG et stocke des informations telles que le nombre de XP gagnées, et le nombre d'ennemis vaincus. Les éléments statiques comme les couleurs des vêtements du personnage héro n'héritent pas de la classe Actor, et peuvent être rattachés à des éléments dynamiques comme les caractères.

La figure suivante détaille la partie de la hiérarchie définissant les caractères du jeu.

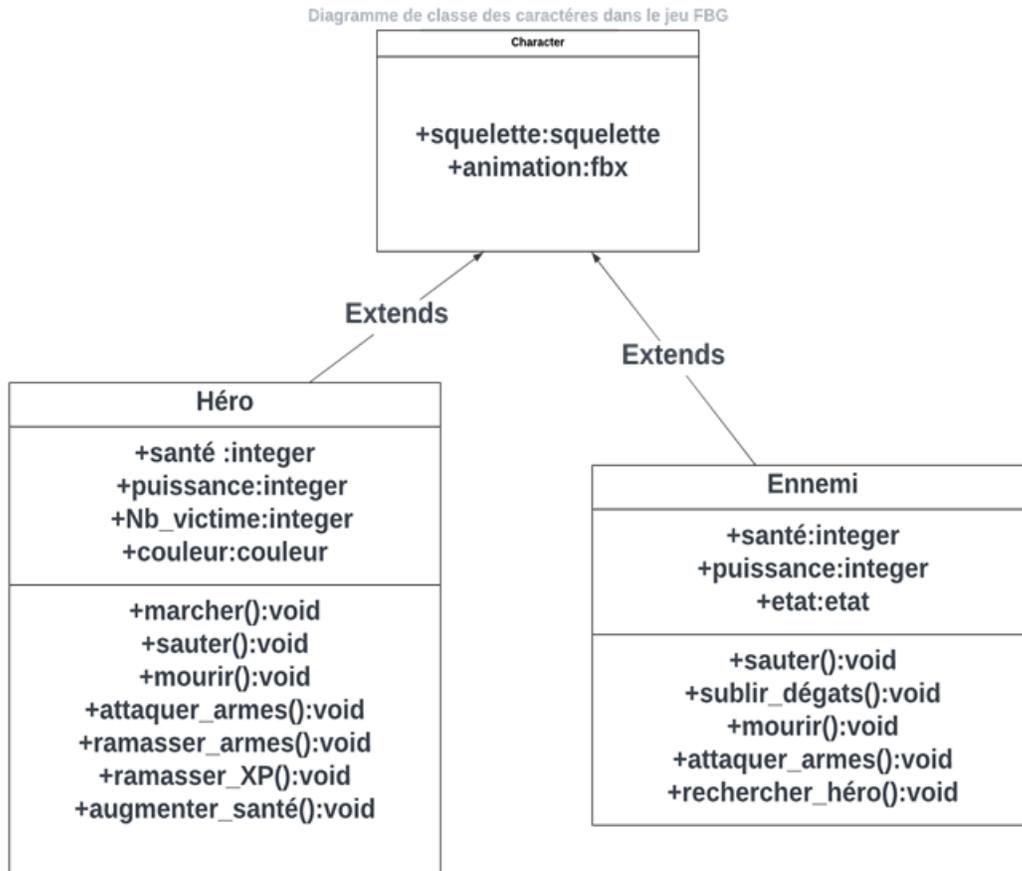


Figure 16. Diagramme de classe de caractère FBG

Le diagramme de classe de la classe "Héros" est représenté dans la figure suivante, qui établit le lien avec la classe "Couleur" qui définit toutes les couleurs disponibles pour la création des vêtements du personnage.

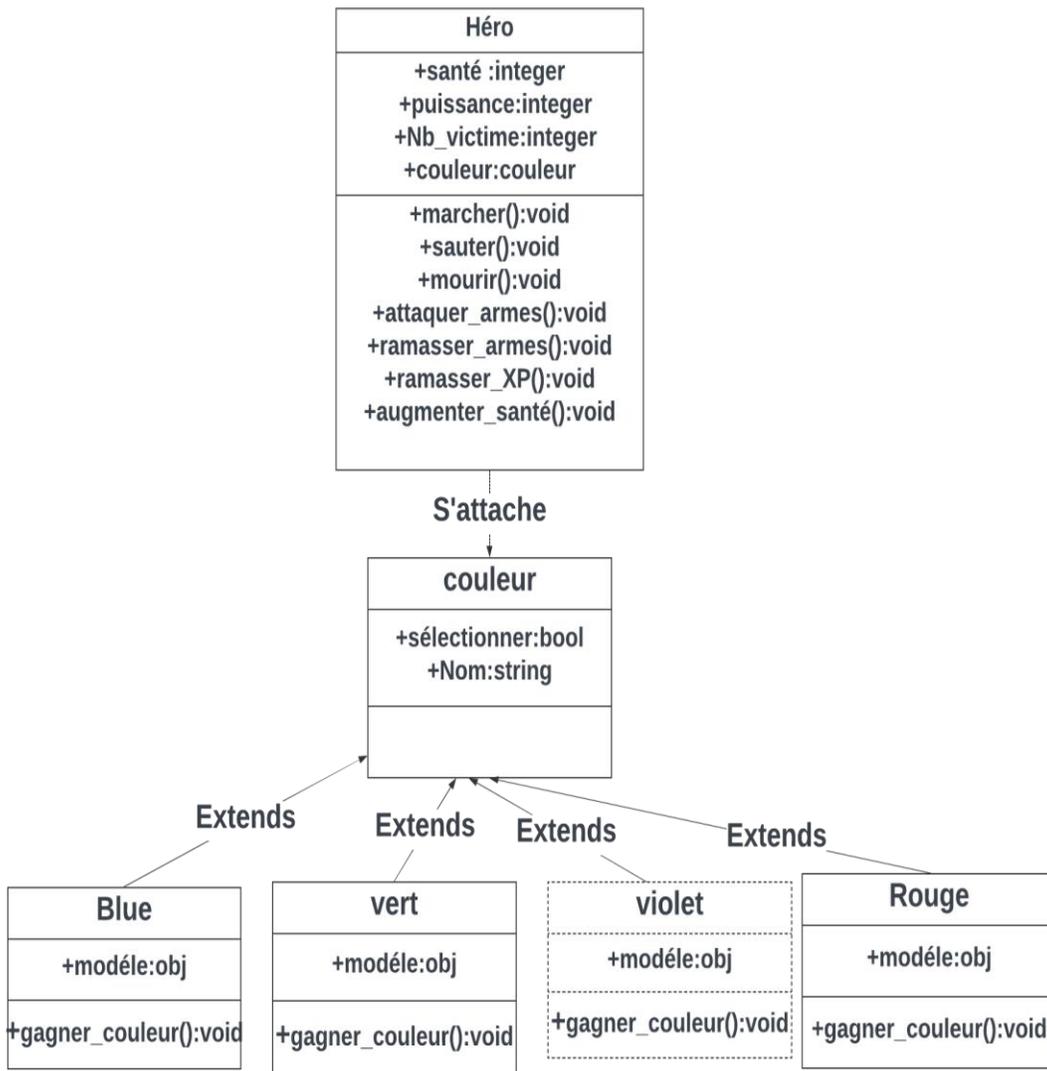


Figure 17. Diagramme de classe de la classe Héro dans le jeu FBG

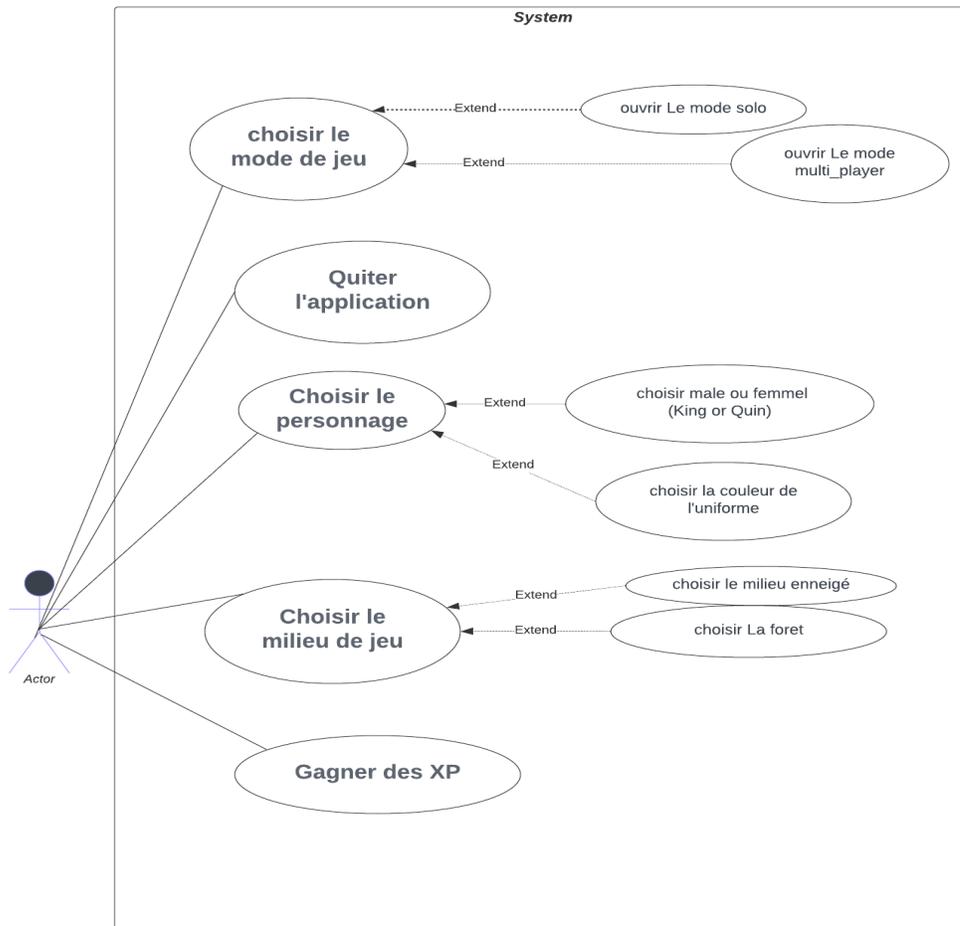


Figure 18. Diagramme de cas d'utilisation de l'utilisateur

2.6. Conclusion

Dans ce chapitre, nous avons détaillé le document de conception du jeu FBG et ses différentes parties. Ensuite nous avons présenté une hiérarchisation des différentes classes de notre jeu ainsi que les diagrammes de classe de quelques classes les plus importantes du jeu. Enfin on a présenté un diagramme de cas d'utilisation pour l'utilisateur.

Dans le prochain chapitre nous présentons les différents outils et langages de programmation que nous avons utilisés pour réaliser notre jeu ainsi que ces différentes interfaces.

Chapitre 03 : Réalisation du jeu FBG

3.1. Introduction

Le chapitre que nous abordons est consacré à la réalisation du jeu "FBG" et à la conception de ses interfaces. Nous allons présenter en détail les étapes de développement du jeu, ainsi que la création des environnements, les modes de jeu disponibles et la conception des interfaces, quelques blueprints et scripts. L'objectif de ce chapitre est de fournir un aperçu approfondi du processus de création de notre jeu et des éléments clés qui le composent.

3.2. Environnement et outils de développement

Dans cette partie du chapitre, nous allons présenter les différents logiciels et outils utilisés pour la création des éléments et objets présents dans le jeu FBG, et enfin nous détaillerons les étapes de création et de mise en place de ces éléments dans notre jeu.

3.2.1. Environnement et outils de développement

3.2.1.1. Moteur Unreal Engine

Le moteur Unreal Engine est un moteur de jeu développé par Epic Games. Il est largement utilisé dans l'industrie du jeu vidéo pour la création de jeux, mais il est également utilisé dans d'autres domaines tels que la réalité virtuelle, la réalité augmentée, les films d'animation et les visualisations architecturales (14) (9) (15).

L'Unreal Engine est connu pour sa puissance, sa flexibilité et ses fonctionnalités avancées. Il offre un large éventail d'outils et de fonctionnalités pour créer des mondes virtuels, des graphismes 3D, des effets spéciaux, des animations, des simulations physiques, des systèmes de particules, des effets sonores, etc.

Le moteur Unreal Engine utilise un langage de script visuel appelé Blueprint, qui permet aux développeurs de créer des fonctionnalités et des comportements de jeu sans avoir à écrire de

code. Cependant, il prend également en charge la programmation traditionnelle en utilisant des langages tels que C++.

L'Unreal Engine est réputé pour ses performances graphiques exceptionnelles, avec des capacités de rendu en temps réel haut de gamme. Il prend en charge diverses plateformes, notamment les PC, les consoles de jeux, les appareils mobiles et les casques de réalité virtuelle.

En plus de sa puissance technique, l'Unreal Engine propose également un écosystème riche comprenant une vaste documentation, une communauté active, des ressources de formation et un marché en ligne où les développeurs peuvent acheter et vendre des actifs, des modèles 3D, des personnages, des effets spéciaux, etc (15).



Figure 19. Logo d'Unreal Engine

3.2.1.2. Fonctionnalités et caractéristiques d'Unreal

L'Unreal Engine offre un large éventail de fonctionnalités et de caractéristiques qui en font l'un des moteurs de jeu les plus populaires de l'industrie. Voici quelques-unes de ses principales fonctionnalités :

- **Graphismes de pointe** : L'Unreal Engine propose un rendu graphique avancé avec des capacités de rendu en temps réel, des effets d'éclairage dynamiques, des ombres réalistes, des matériaux physiquement basés, des particules et des effets spéciaux de haute qualité.
- **Outils de création de niveau** : Il offre des outils puissants pour la conception de niveaux, y compris un éditeur de terrain, un système de placement d'objets, des outils de modélisation et

de sculpting, des outils de peinture et de texturage, et des fonctionnalités de génération procédurale de contenu.

- **Système de physique réaliste** : L'Unreal Engine dispose d'un moteur de physique robuste qui permet la simulation précise des interactions physiques, y compris la gravité, les collisions, la destruction d'objets, les véhicules, les ragdolls, etc.
- **Blueprints** : Il propose un système de script visuel appelé Blueprints, qui permet aux développeurs de créer des fonctionnalités de jeu et de définir des comportements en utilisant une interface graphique intuitive, sans avoir à écrire de code. Cela facilite la création rapide de prototypes et l'itération des idées.
- **Langage de programmation C++** : L'Unreal Engine prend également en charge la programmation en utilisant le langage C++. Les développeurs peuvent écrire du code pour personnaliser et étendre les fonctionnalités du moteur, créer des classes de jeu, optimiser les performances, etc (14).

3.2.1.3. Epic Games

Epic Games est une société américaine de développement de jeux vidéo et de logiciels basée à Cary, en Caroline du Nord. Elle est connue pour avoir créé et publié des jeux populaires tels que Fortnite, Unreal Tournament, Gears of War et Infinity Blade. Epic Games est également le développeur du moteur de jeu Unreal Engine, utilisé par de nombreux développeurs de jeux à travers l'industrie. En plus de ses activités de développement de jeux, Epic Games exploite également la plateforme de distribution de jeux vidéo appelée Epic Games Store, qui concurrence les plateformes telles que Steam (16).



Figure 20. Logo d'Epic Games

3.2.1.4. Visual Scripting 'Blueprint'

L'une des caractéristiques les plus importantes d'Unreal c'est la possibilité de programmer les différentes classes et logiques du jeu grâce au langage de programmation C++, ainsi qu'avec un système de script visuel appelé 'Blueprint'. Les blueprints permettent de faire appel à des fonctions, des macros et des parties du programmes grâce à des nœuds qui sont connectés entre eux par des connections sous forme de ligne. Ils peuvent remplacer la programmation traditionnelle en C++ dans presque tous les aspects de la création d'un jeu sur Unreal, mais la programmation en C++ est actuellement nécessaire pour implémenter des fonctionnalités plus avancées comme le multi-joueurs en ligne (16).

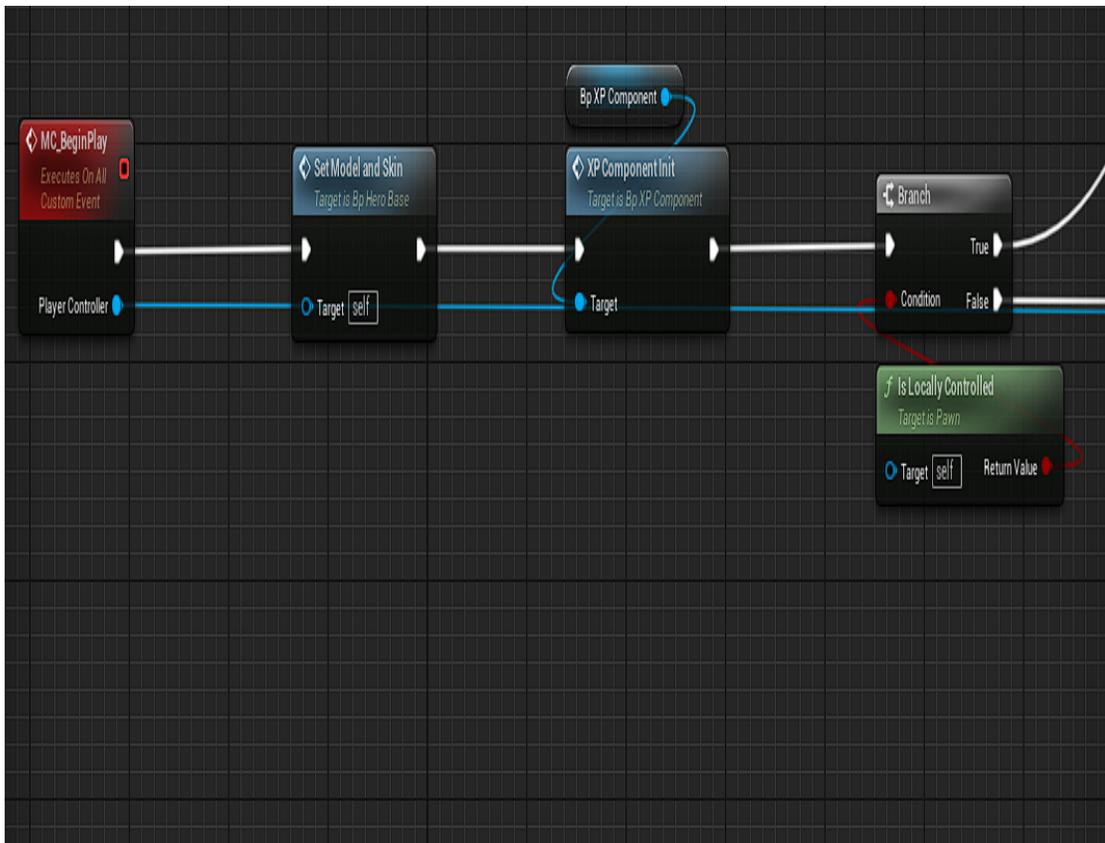


Figure 21. Exemple de Blueprint

3.2.1.5. L'intelligence artificielle

L'Unreal Engine offre des fonctionnalités intégrées pour l'intégration de l'intelligence artificielle (IA) dans les jeux. Voici quelques-unes des fonctionnalités d'IA proposées par l'Unreal Engine :

- Behavior Trees : L'Unreal Engine utilise un système appelé Behavior Trees (arbres de comportement) pour définir les comportements des PNJ. Les Behavior Trees permettent de représenter visuellement les états et les transitions entre les différents comportements des PNJ, facilitant ainsi la création de comportements complexes et adaptatifs.
- Blueprints et C++ : L'Unreal Engine permet d'utiliser à la fois le langage de script visuel, les Blueprints, et la programmation en C++ pour implémenter des comportements d'IA personnalisés. Les développeurs peuvent créer des Blueprints ou écrire du code pour définir les décisions, les actions et les réactions des PNJ en fonction des conditions de jeu.

L'intelligence artificielle donne un comportement logique à un caractère basé sur une imagination préalable du développeur.

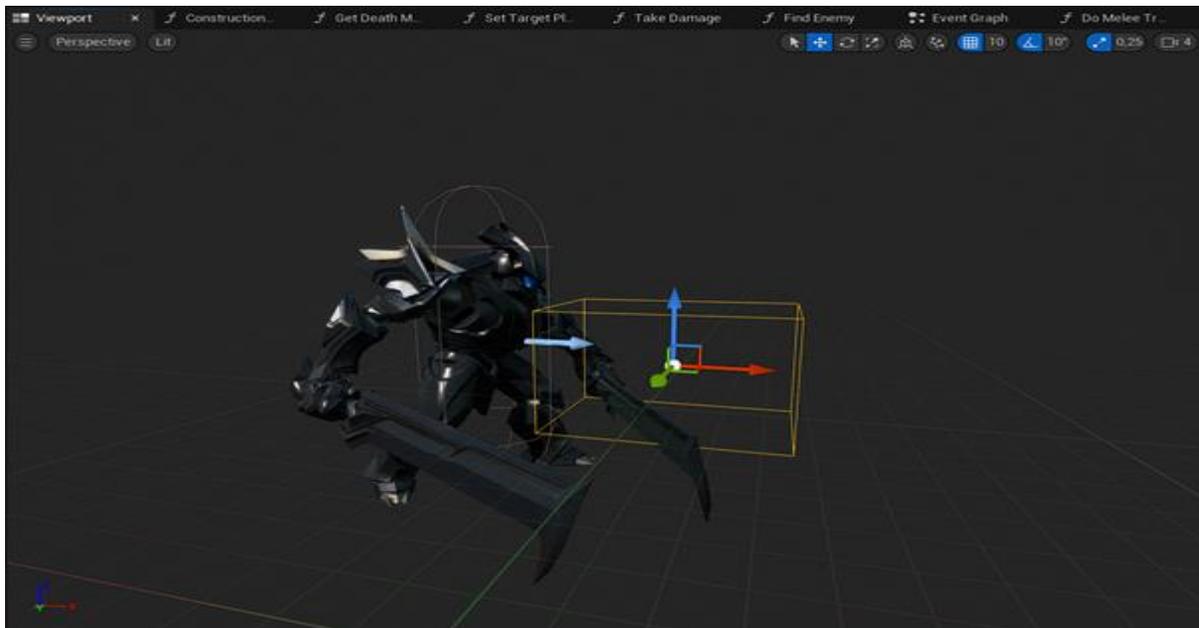


Figure 22. Un design de l'intelligence artificielle « AI »

3.3. Les interfaces du jeu FBG

3.3.1. Menu principal

Lorsque le jeu démarre, le menu principal est affiché en premier. C'est l'interface initiale où les joueurs peuvent sélectionner le mode de jeu en cliquant sur les boutons « Solo Game » ou « Multiplayer ». Il est également possible de créer un nouveau profil en cliquant sur le bouton « Profiles » et en entrant un nom pour commencer à jouer. Si besoin, les joueurs peuvent quitter le jeu en cliquant sur le bouton « Quit Game » dans cette interface utilisateur.

Sur le côté droit de l'interface, vous pouvez voir le niveau actuel du joueur « Level » ainsi que son nombre d'XP.

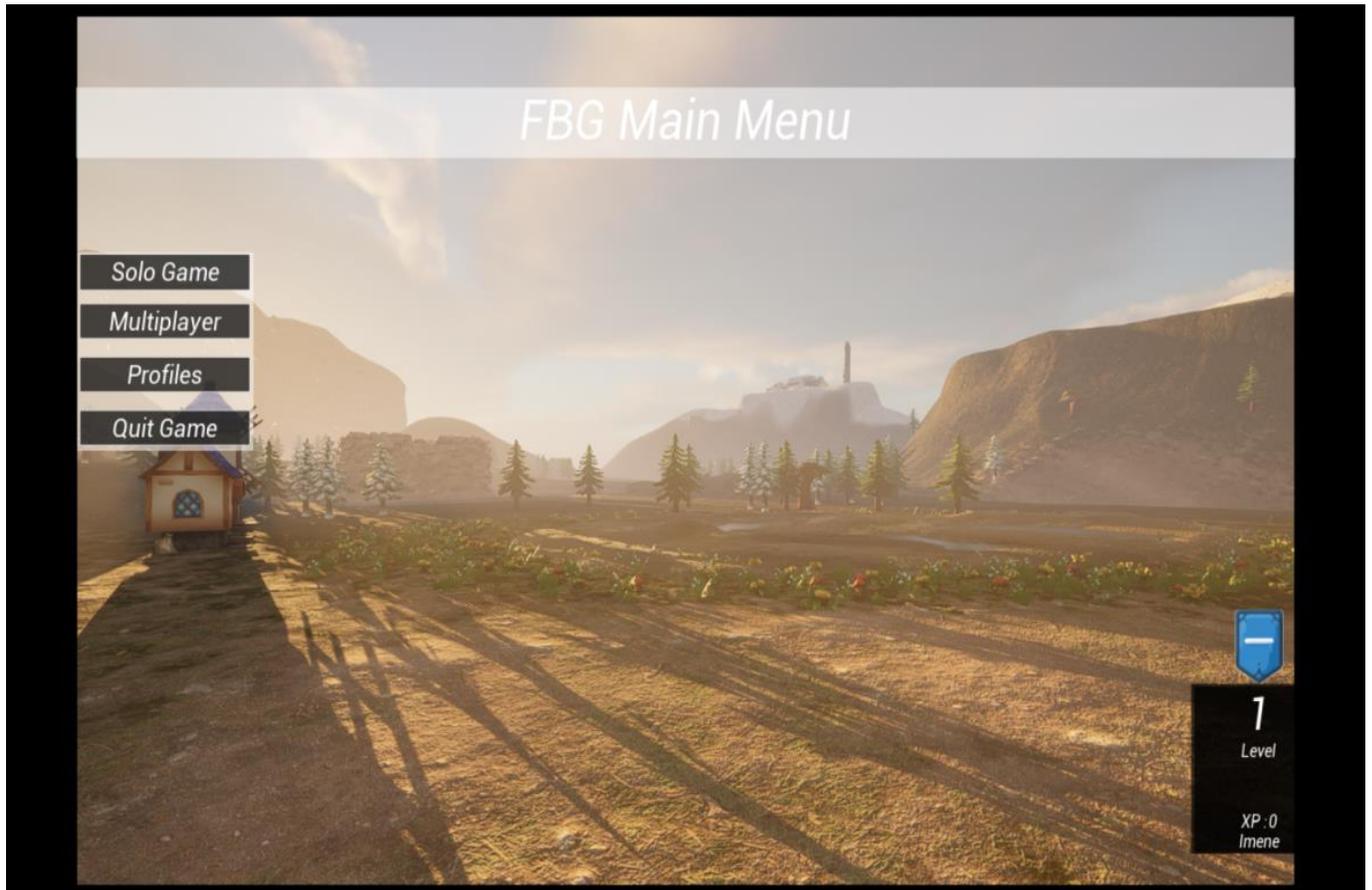


Figure 23. Le menu principal « FBG Main Menu »

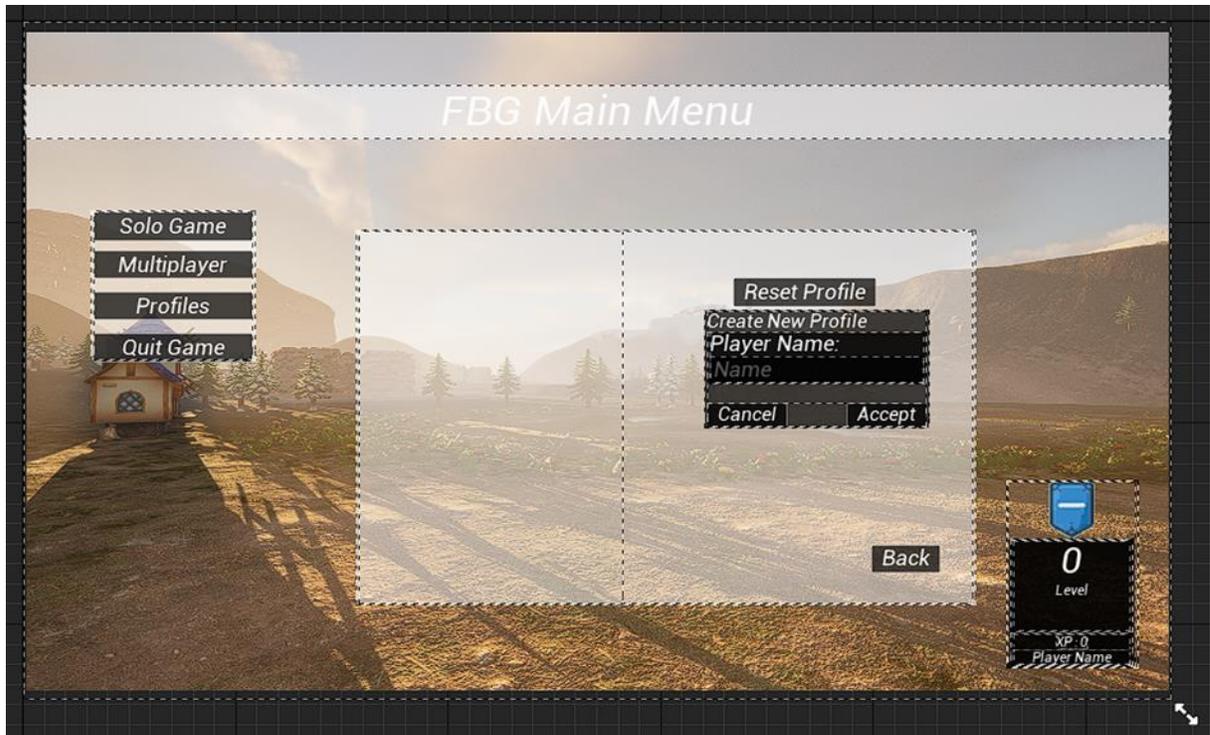


Figure 24. Main Menu design

3.3.2. FBG Lobby

Dans l'interface FBG Lobby, l'utilisateur a la possibilité de sélectionner l'environnement dans lequel il souhaite jouer en faisant un choix entre « Forest » et « Snowfloor » situés du côté droit. Il peut également choisir le mode de jeu en cliquant sur l'option à gauche « PvP » (joueur contre joueur), soit « Endlesswave ».

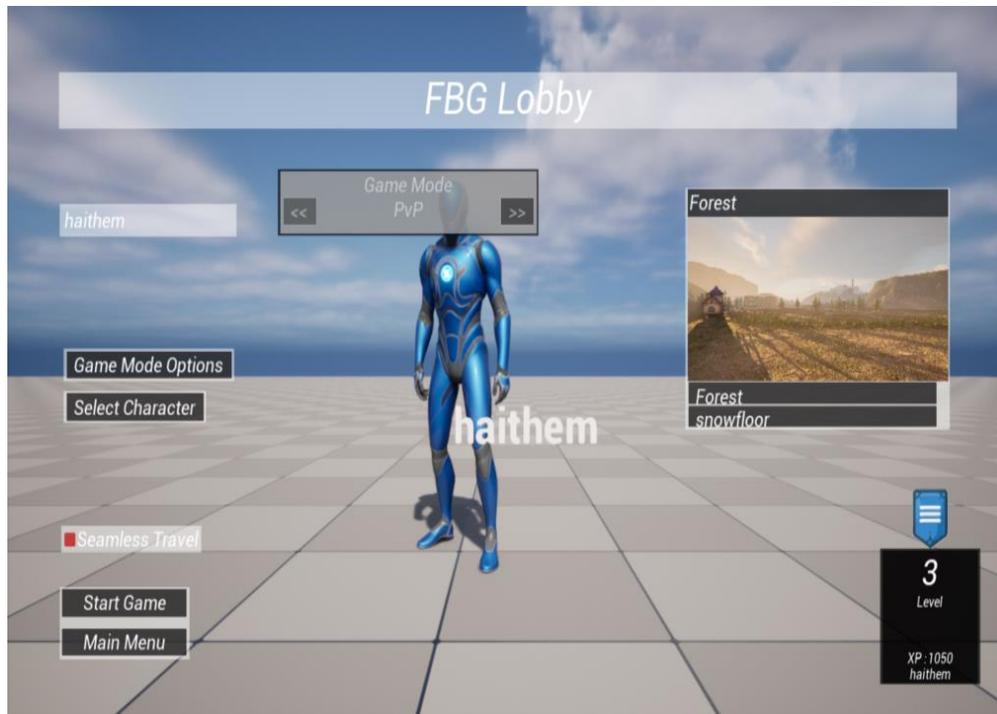


Figure 25. FBG Lobby

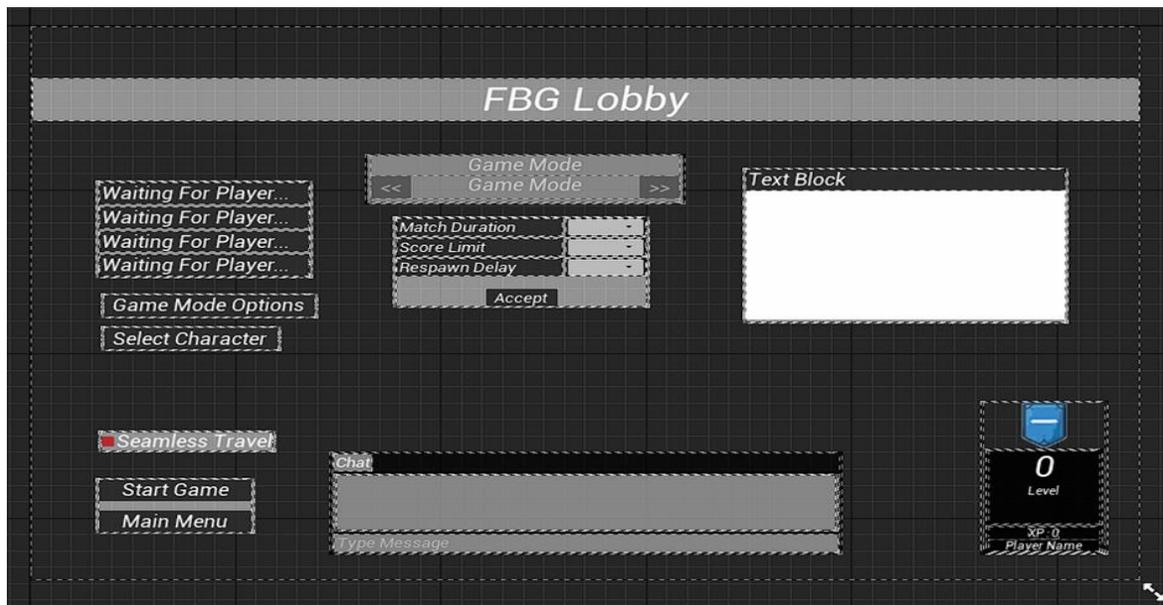


Figure 26. FBG Lobby design

Une fois que l'utilisateur presse le bouton « Start Game » situé en bas, il peut lancer le jeu

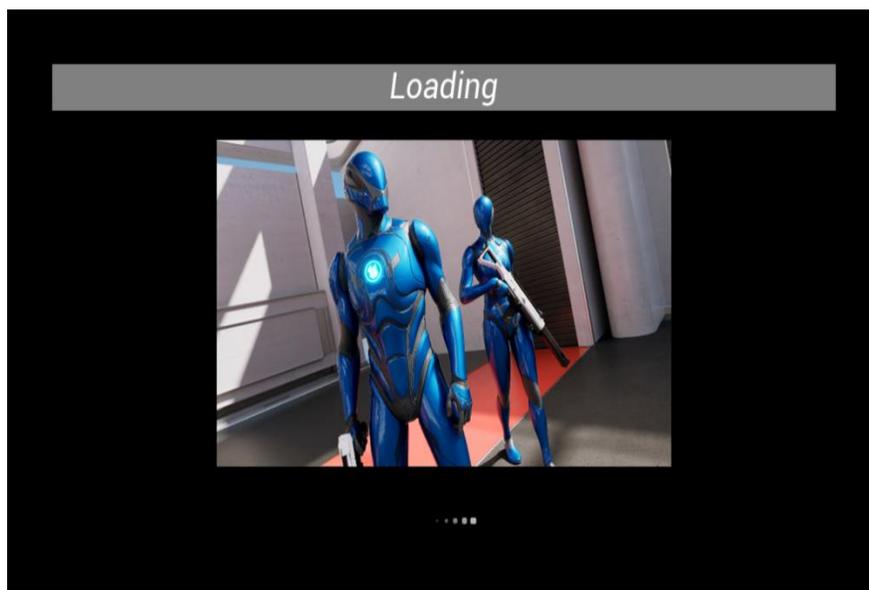


Figure27. Le démarrage du jeu

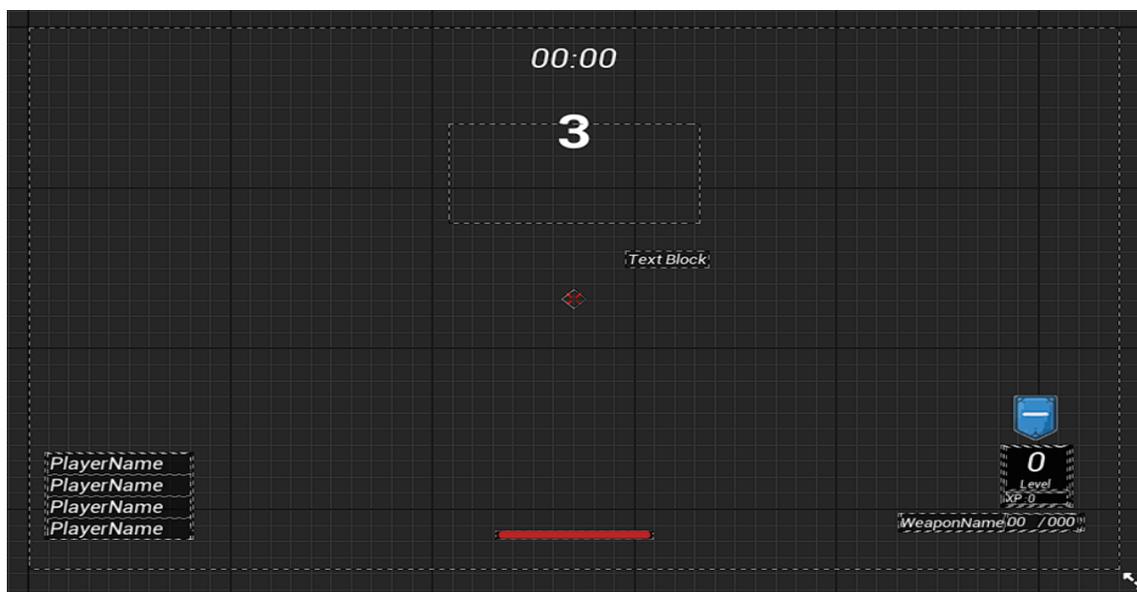


Figure 28. Le Design « In Game »

3.3.3. Actions des touches (Actions mappings)

Les Commandes ou les actions interactives disponibles pour le joueur ou l'utilisateur telles que : Sauter, courir et bouger dans la forêt, obtenez des armes.....

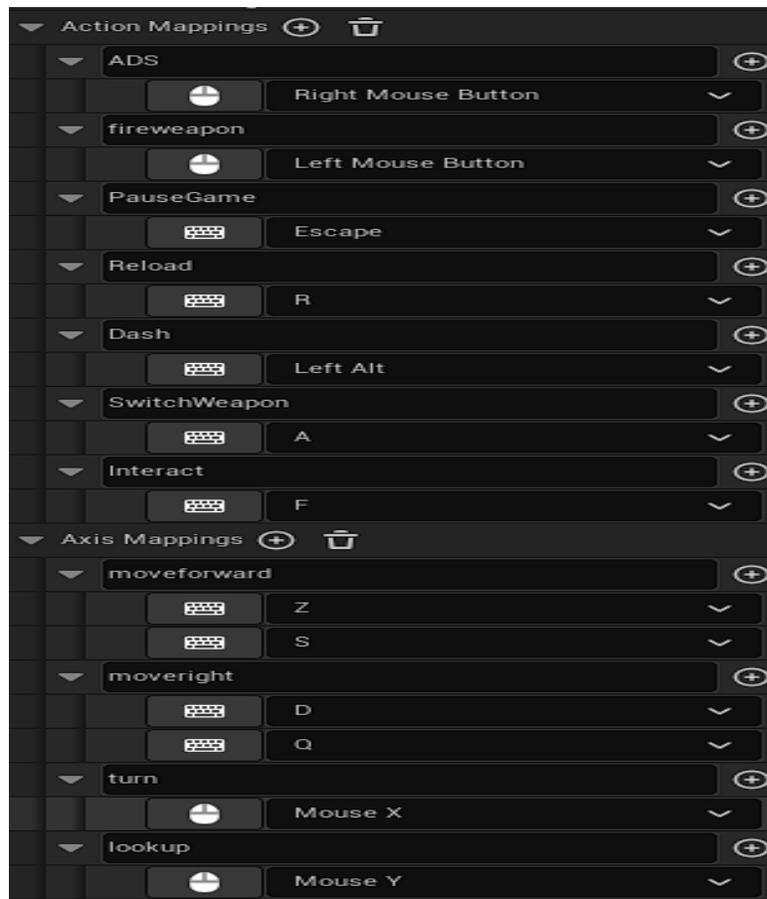


Figure 29. Les actions de mappings

3.4. Les blueprints

3.4.1. Game instance blueprint

Dans Unreal Engine, une Game Instance Blueprint est une classe Blueprint spécialisée qui sert d'objet persistant tout au long d'une session de jeu. Elle nous permis de stocker et de gérer des données qui doivent persister à travers plusieurs niveaux ou modes de jeu.

Un Game Instance Blueprint est généralement utilisé pour stocker et gérer des informations telles que les données des joueurs, les paramètres du jeu, les variables globales et d'autres données qui doivent être accessibles et modifiées à partir de différentes parties du jeu. Il agit comme un centre névralgique pour la gestion de l'état du jeu et le maintien des informations entre différents niveaux ou modes de jeu.

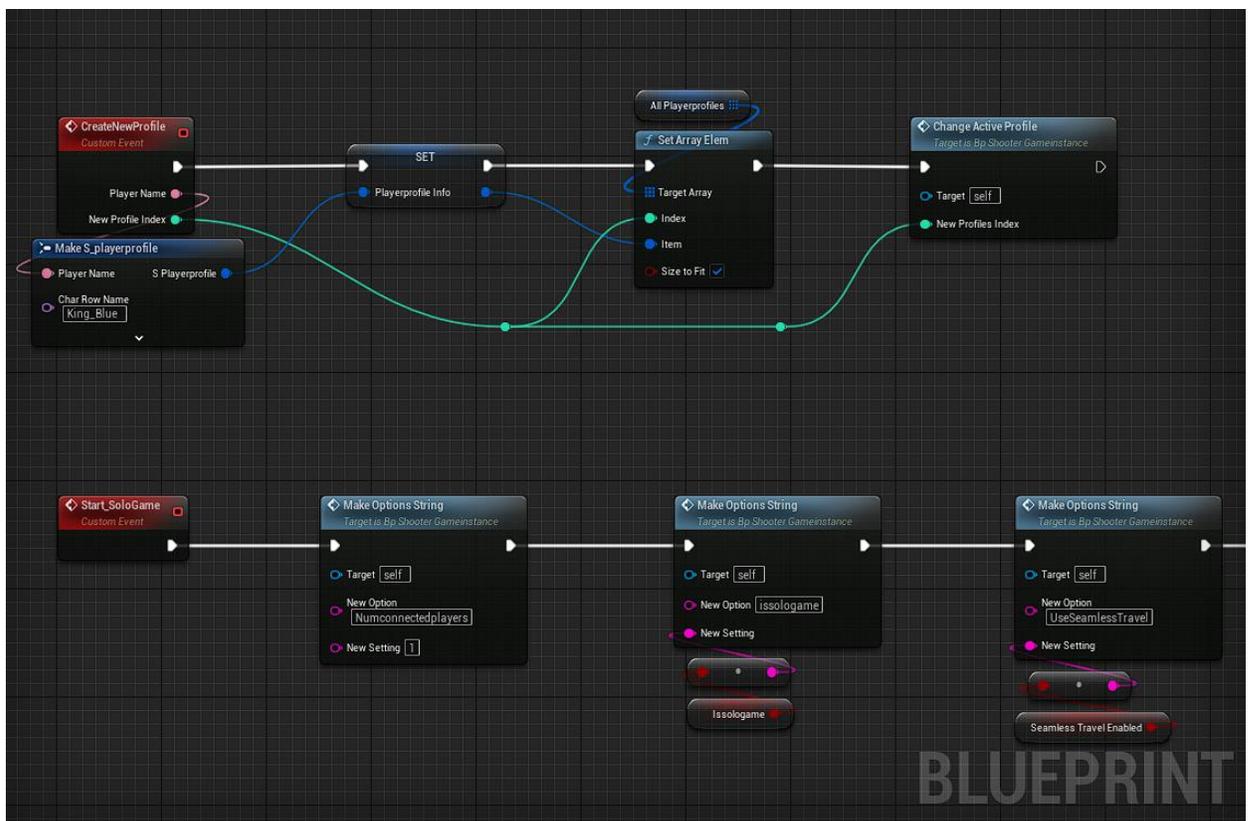


Figure 30. Partie du blueprint 'Game instance'

3.4.2. Player controller blueprint

Le Player controller Blueprint est responsable de la gestion des entrées du joueur, de la gestion des contrôles de la caméra, du contrôle du personnage ou de l'avatar du joueur, ainsi que de l'interaction avec d'autres systèmes de jeu. Il reçoit les événements d'entrée du joueur et les traduit en actions ou commandes à l'intérieur du jeu. Certaines des tâches courantes effectuées par le player controller blueprint comprennent :

- ❖ *Gestion des entrées* : Il capture et traite les événements d'entrée du joueur, tels que les touches du clavier pressées, les mouvements de la souris ou les pressions des boutons de la manette de jeu.
- ❖ *Contrôle de la caméra* : Il gère la perspective du joueur et contrôle le mouvement et la rotation de la caméra en fonction des entrées ou des événements du jeu.
- ❖ *Contrôle du personnage* : Il contrôle le personnage ou l'avatar du joueur dans le jeu, gérant les mouvements, les actions, les animations et les interactions avec le monde du jeu.
- ❖ *Interaction avec le jeu* : Il communique avec d'autres systèmes de jeu et acteurs, déclenchant des événements, interagissant avec des objets ou gérant l'état du jeu.

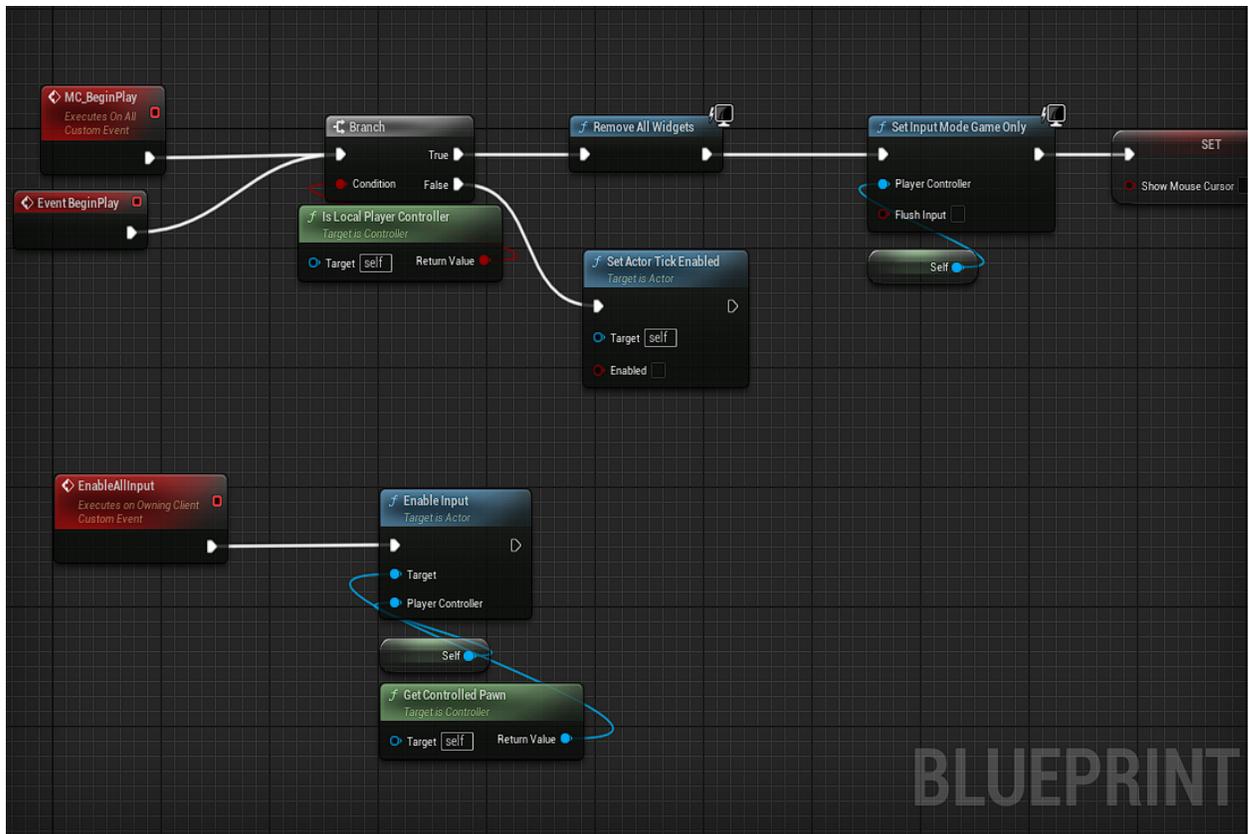


Figure31. Partie du blueprint ' player controller'

3.4.3. Widget blueprint character

Dans Unreal Engine, un Widget Blueprint fait référence à une classe Blueprint spécialisée utilisée pour créer des éléments d'interface utilisateur (UI) et concevoir des écrans d'interface utilisateur interactifs pour les jeux ou les applications. Il fournit un système de script visuel qui nous permis de créer et de personnaliser des éléments d'interface utilisateur tels que des boutons, des champs de texte, des images, des curseurs, et bien d'autres.

Les Widget Blueprints sont principalement utilisés pour concevoir l'interface utilisateur d'un jeu ou d'une application. Ils nous permis de créer des éléments visuellement attrayants et interactifs qui peuvent être affichés à l'écran et réagir aux entrées de l'utilisateur. Certaines fonctionnalités clés et composants des Widget Blueprints comprennent :

- ❖ *Widgets* : Ce sont les éléments d'interface utilisateur individuels qui constituent l'interface utilisateur, tels que des boutons, des zones de texte, etc. Les widgets peuvent être ajoutés, positionnés et personnalisés dans le Widget Blueprint.
- ❖ *Événements et fonctions* : Les Widget Blueprints prennent en charge la programmation événementielle, ce qui nous permet de définir des fonctions et des gestionnaires d'événements qui répondent aux entrées de l'utilisateur, aux changements de données ou à d'autres événements.
- ❖ *Variables et liaison de données* : on peut définir des variables dans le Widget Blueprint pour stocker et manipuler des données.

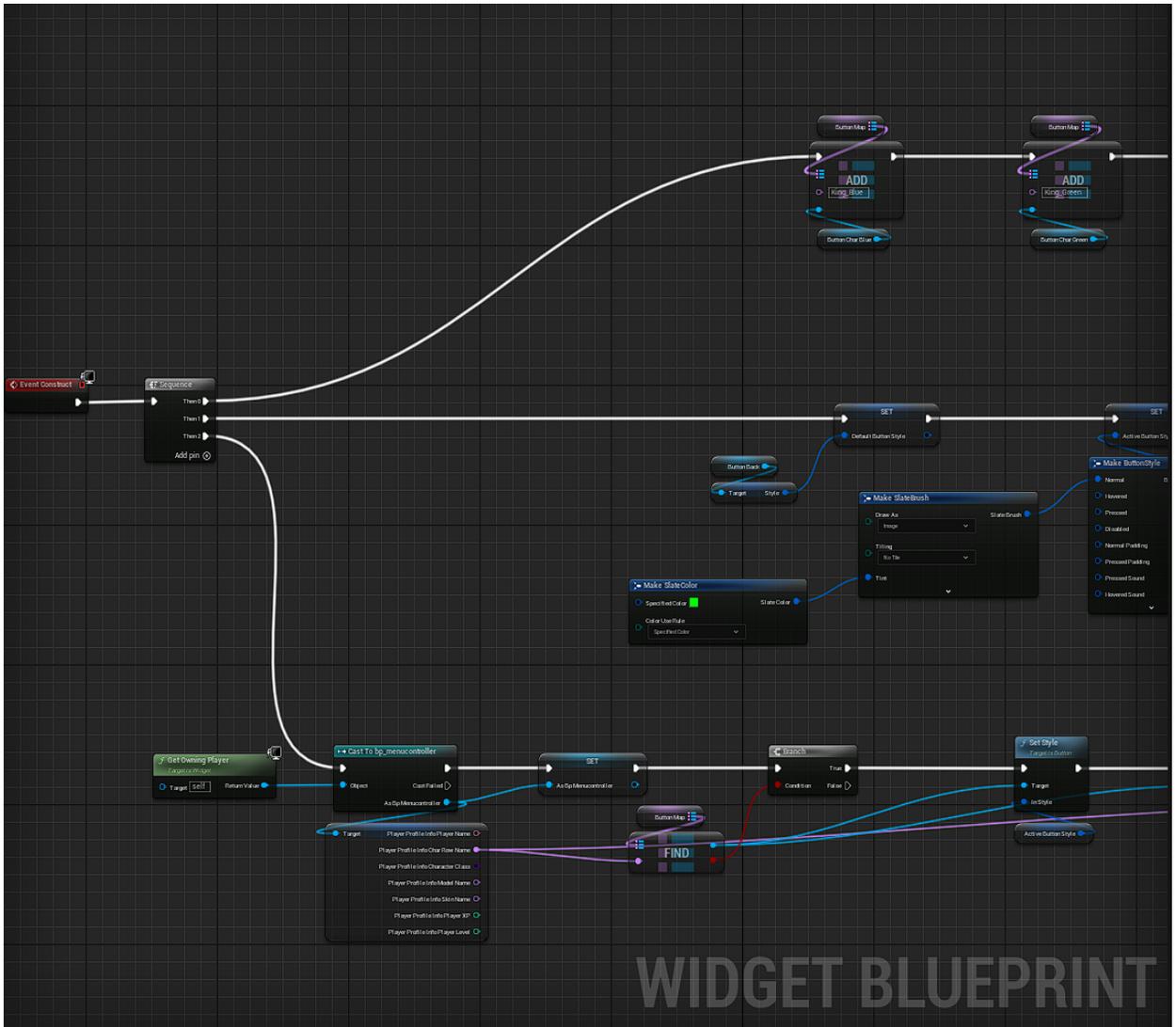


Figure32. Partie du widget blueprint Character

3.4.4. Widget blueprint in Game

Dans le contexte du développement de jeux en utilisant Unreal Engine, un Widget Blueprint dans le contexte de l'HUD (Heads-Up Display) fait référence à une classe Blueprint spécialisée utilisée pour créer et gérer les éléments de l'interface utilisateur qui s'affichent à l'écran pendant le gameplay.

L'HUD sert de superposition sur l'écran du jeu, fournissant aux joueurs des informations importantes et des éléments interactifs. Cela peut inclure des barres de santé, des compteurs de munitions, des mini-cartes, des marqueurs d'objectifs, des tableaux des scores et d'autres éléments qui améliorent la compréhension du joueur de l'état et de la progression du jeu.

Pour créer un Widget Blueprint pour l'HUD, nous suivrions ces étapes :

- ❖ *Créer un Widget Blueprint* : Dans le système de script visuel Blueprint d'Unreal Engine, vous créeriez un nouveau Widget Blueprint spécifiquement pour l'HUD.
- ❖ *Concevoir l'HUD* : Dans le Widget Blueprint, vous pouvez ajouter et organiser des éléments d'interface utilisateur tels que des zones de texte, des images, des barres de progression, des boutons, et plus encore. Personnalisez leur apparence, appliquez des animations et définissez des gestionnaires d'événements ou des fonctions pour gérer les interactions.
- ❖ *Lier les données* : Si vous souhaitez que l'HUD affiche des informations dynamiques, telles que la santé du joueur ou son score, vous pouvez lier les propriétés de données provenant de Blueprints pertinents ou de systèmes de jeu aux éléments d'interface utilisateur. Cela permet à l'HUD de se mettre à jour automatiquement en fonction des changements dans les données du jeu.
- ❖ *Afficher l'HUD* : Le Widget Blueprint de l'HUD doit être instancié et affiché pendant le gameplay. Cela se fait généralement en attachant l'HUD au viewport ou à l'écran du joueur, en veillant à ce qu'il reste visible en permanence.

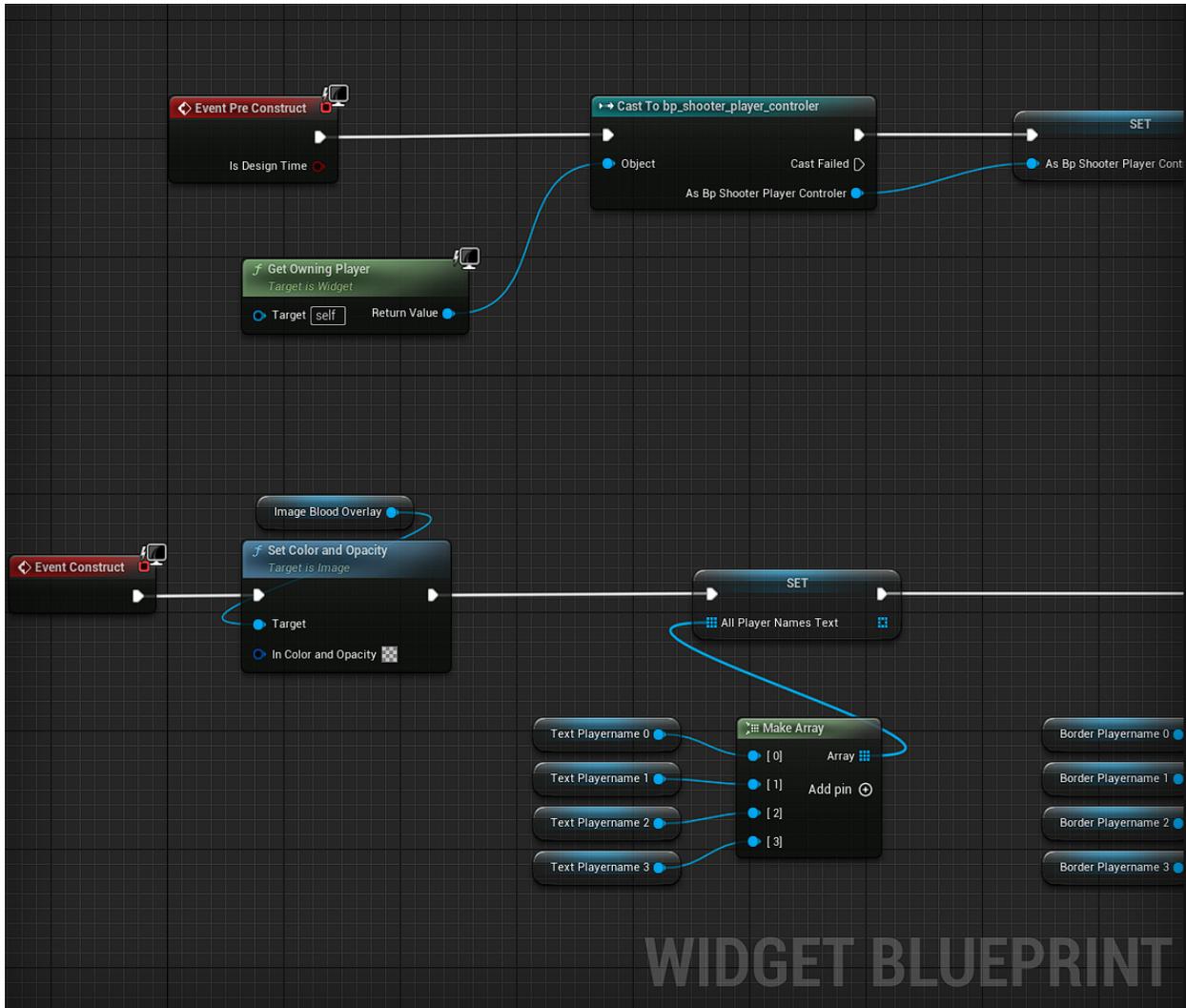


Figure 33. Partie de widget blueprint in Game

3.4.5. Widget blueprint Lobby

Dans le contexte du développement de jeux en utilisant Unreal Engine, un Widget Blueprint pour un lobby fait référence à une classe Blueprint spécialisée utilisée pour concevoir et gérer les éléments de l'interface utilisateur spécifiques au lobby ou au système de matchmaking du jeu.

Le lobby est un composant courant dans les jeux multijoueurs où les joueurs se rassemblent, communiquent et se préparent aux sessions de gameplay. Le Widget Blueprint pour le lobby nous permis de créer un écran d'interface utilisateur visuellement attrayant et interactif où les

joueurs peuvent rejoindre ou créer des sessions de jeu, inviter d'autres joueurs, définir des options de jeu et effectuer d'autres actions connexes.

Pour créer un Widget Blueprint pour un lobby, nous suivrions ces étapes :

- ❖ *Créer un Widget Blueprint* : Dans le système de script visuel Blueprint d'Unreal Engine, nous créerions un nouveau Widget Blueprint spécifiquement pour le lobby.
- ❖ *Concevoir l'interface utilisateur du lobby* : Dans le Widget Blueprint, nous pouvons concevoir la disposition et l'apparence des éléments de l'interface utilisateur du lobby, tels que la liste des joueurs, la boîte de discussion, la liste des sessions de jeu, les boutons pour rejoindre ou créer des sessions, et d'autres composants pertinents. Personnalisez leur apparence, appliquez des animations et définissez des gestionnaires d'événements ou des fonctions pour gérer les interactions.
- ❖ *Gérer la logique du lobby* : nous implémentons la logique nécessaire dans le Widget Blueprint pour gérer les fonctionnalités liées au lobby, telles que la gestion des sessions de jeu, la gestion des connexions des joueurs, la mise à jour des listes de joueurs et la gestion de la communication entre les joueurs.
- ❖ *Implémenter le réseau* : Étant donné que le lobby est généralement une fonctionnalité multijoueur, nous devons implémenter la fonctionnalité réseau nécessaire pour permettre aux joueurs d'interagir les uns avec les autres et de synchroniser les informations des sessions de jeu.

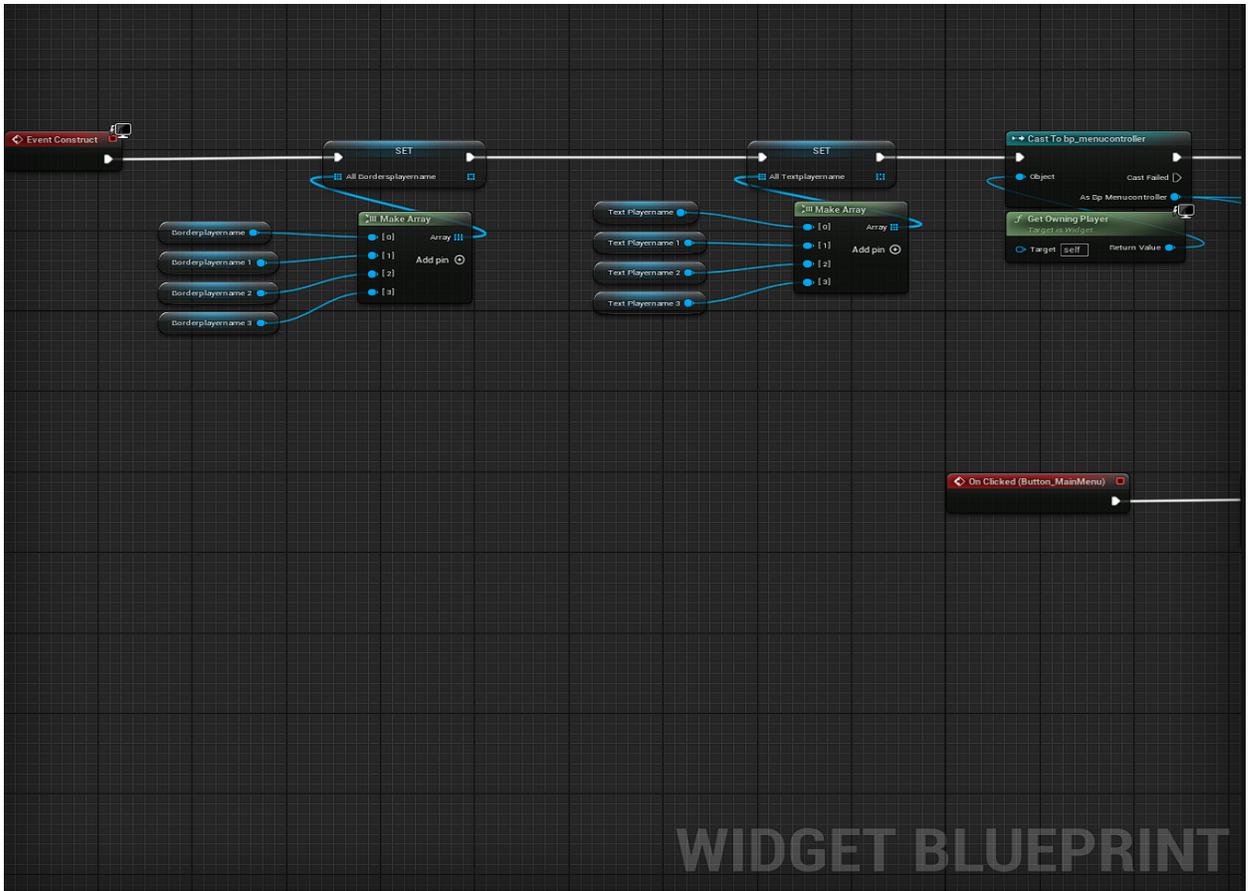


Figure 34. Partie du widget blueprint Lobby

Engine ini :

```
[/Script/Engine.GameEngine]
+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnLineSubsystemSteam.SteamNetDriver",DriverClassNameFallback="OnLineSubsystemUtils.IpNetDriver")

[OnLineSubsystem]
DefaultPlatformService=Steam

[OnLineSubsystemSteam]
bEnabled=true
SteamDevAppId=480
bInitServerOnClient=false
GameServerQueryPort=27015
bRelaunchInSteam=false
bHasVoiceEnabled=false
GameVersion=1.0.0.0
bVACEnabled=0
bAllowP2PPacketRelay=true
P2PConnectionTimeout=90

[/Script/OnLineSubsystemSteam.SteamNetDriver]
NetConnectionClassName="OnLineSubsystemSteam.SteamNetConnection"
```

Figure 37. Le code C++ Engine ini

Interface Session :

```
// Unique identifier of this build for compatibility
Session->SessionSettings.BuildUniqueId = GetBuildUniqueId();
Session->SessionSettings.bUseLobbiesIfAvailable = true;
```

Figure 38. Le code C++ interface Session

Target. Cs :

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 using UnrealBuildTool;
4 using System.Collections.Generic;
5
6 public class MyFBGTarget : TargetRules
7 {
8     public MyFBGTarget(TargetInfo Target) : base(Target)
9     {
10         Type = TargetType.Game;
11         DefaultBuildSettings = BuildSettingsVersion.V2;
12
13         bUsesSteam = true;
14
15         ExtraModuleNames.AddRange( new string[] { "MyFBG" } );
16     }
17 }
18
```

Figure 39 . Le code C++ Target Cs

3.6. Conclusion

Dans ce chapitre, nous avons présenté les différents outils et langages de programmation que nous avons utilisés pour réaliser notre jeu ainsi que ces différentes interfaces. Nous avons mis en évidence l'importance de concevoir des interfaces intuitives et conviviales pour offrir une expérience utilisateur agréable. Nous avons également exploré le visuel scripting avec Blueprint, qui nous a permis de créer des interactions et des fonctionnalités complexes sans écrire de code : Game Instance Blueprint, le Player Controller Blueprint, ainsi que les Widget Blueprints utilisés pour gérer les interactions avec les personnages, l'interface du jeu et le lobby. Enfin, nous avons abordé les scripts en C++, qui ont été utilisés pour ajouter des fonctionnalités plus avancées et pour optimiser les performances du jeu.

Ce chapitre nous a permis d'appréhender l'ampleur du travail de développement nécessaire pour créer un jeu comme FBG. Nous avons exploité les outils et les fonctionnalités d'Unreal Engine, ainsi que les concepts de scripting, pour donner vie à notre vision du jeu. Ce chapitre constitue un jalon essentiel dans notre parcours de création du jeu FBG. Nous avons exploré les différentes étapes et les technologies impliquées, tout en soulignant l'importance de la conception d'interfaces intuitives et de l'utilisation judicieuse des blueprints et des scripts en C++.

Conclusion générale et perspectives

Notre travail a consisté à développer un jeu de tir à la troisième personne à la fois en mode solo et en multijoueurs, offrant aux joueurs une expérience immersive et captivante dans deux environnements distincts : une forêt et un paysage enneigé. Nous avons entrepris ce projet avec détermination et passion.

La réalisation de ce jeu a été un véritable défi, mais cela nous a permis d'acquérir une expérience précieuse dans le domaine de développement de jeux. Nous avons utilisé le moteur de jeu Unreal Engine pour créer un gameplay fluide et des graphismes attrayants qui transportent les joueurs dans des mondes virtuels riches et détaillés.

Dans le mode solo, nous avons conçu des niveaux captivants et des défis stimulants pour offrir aux joueurs une expérience de jeu satisfaisante et divertissante. Nous avons veillé à équilibrer les mécanismes de tir, les mouvements du personnage et les obstacles pour garantir un gameplay fluide et engageant.

En parallèle, nous avons également intégré un mode multijoueurs, permettant aux joueurs de s'affronter ou de coopérer dans des batailles passionnantes. Nous avons mis en place des fonctionnalités de matchmaking et de communication en ligne pour faciliter la connectivité et la collaboration entre les joueurs, créant ainsi une expérience de jeu sociale et immersive.

Cependant, notre voyage ne s'arrête pas là. Nous avons de grandes perspectives pour l'avenir de notre jeu :

- Nous avons l'intention d'élargir davantage les fonctionnalités du mode multijoueur en introduisant des modes de jeu variés tels que la capture du drapeau, le jeu par équipe, ou des batailles épiques en ligne. Nous souhaitons offrir aux joueurs une expérience multijoueur riche et compétitive.

- Nous sommes déterminés à continuer d'améliorer et d'optimiser le gameplay et les mécanismes du jeu en tenant compte des commentaires des joueurs. Nous voulons créer une expérience de tir fluide et satisfaisante, offrant un équilibre entre l'excitation de l'action et la précision des contrôles.
- Nous envisageons également d'explorer de nouveaux environnements et niveaux pour offrir aux joueurs une plus grande diversité de décors et de défis. Que ce soit des jungles mystérieuses, des déserts arides ou des environnements futuristes, nous voulons continuer à surprendre et à divertir les joueurs avec de nouveaux mondes à explorer.

En conclusion, notre travail sur ce jeu de tir à la troisième personne en solo et multijoueurs dans des environnements forestiers et enneigés nous a permis de repousser nos limites et de développer nos compétences dans le domaine du développement de jeux. Nous espérons que notre jeu fournira aux joueurs une expérience immersive et excitante, que ce soit en jouant en solo ou en défiant leurs amis en multi-joueurs.

Bibliographie

1. gaming campus. [En ligne] <https://gamingcampus.fr/boite-a-outils/les-grandes-etapes-de-lhistoire-du-jeu-video.html>.
2. L'histoire des jeux vidéo. [En ligne] https://fr.wikipedia.org/wiki/Histoire_du_jeu_vid%C3%A9o.
3. **Murat, Mathieu.** *Conception, réalisation et évaluation d'un jeu sérieux de stratégie temps réel pour l'apprentissage des fondamentaux de la programmation.* Toulouse : Université Toulouse III- Paul Sabatier. IRIT-UPS, 2010.
4. **Baer, Ralph H.** *The Medium of the Video Game.* s.l. : Mark J. P. Wolf, 2010. EPUB 9780292786646.
5. Le marché de jeu video. [En ligne] <https://fr.statista.com/themes/9063/le-marche-du-jeu-video/>.
6. **Adams, Ernest.** *Fundamentals of Action and Arcade Game Design.* s.l. : Kindle Edition , 2014.
7. **Latorre Freixes, Marc.** Prototype of a video game for computer using Unity. *Bachelor's Thesis.* s.l. : Universitat politechnica de catalunya barcelonatech, 2014.
8. L'institut e-tribArt. [En ligne] <https://www.e-tribart.fr/blog/secteur-3d/logiciels-techniques/4-moteurs-de-jeux>.
9. Moteur de jeu. [En ligne] <https://jeux.developpez.com/livres/index/?page=Moteur-de-jeux>.
10. **SAYAD, Yasmine.** Conception et réalisation d'un jeu mobile 3D sous. *Mmeoire de Master.* s.l. : Université A/Mira de Béjaia, 2020.

11. **Szymczak, Leszek.** FIRST PERSON SHOOTER WITH PROCEDURAL. *Bachelor Thesis*. s.l. : Politechnika Wroclawska Faculty of Computer Science and Management, 2020.
12. **Sofiane, Merrad.** *Conception et réalisation d'un jeu sur Android avec le moteur de jeu Unreal*. s.l. : Université Moulou maamri de tizi ouzou, 2020.
13. Unreal Engine. [En ligne] <https://docs.unrealengine.com/5.1/en-US/blueprints-visual-scripting-in-unreal-engine/>.
14. Les Fonctionnalités d'Unreal Engine. [En ligne] <https://www.techlila.com/fr/unreal-engine-benefits/>.
15. **Peter L. Newton, Jie Feng.** *Unreal Engine 4 AI Programming Essentials*. s.l. : Packt Publishing, 2016. 978-1784393120.
16. Epic Games. [En ligne] <https://www.epicgames.com/site/fr/about>.