

# DARK WAFFLE

## RAPPORT DE PROJET

---

### THE FALLEN ONE

---

*Les développeurs :*

ROSENKRANZ Hugo

HASSELMANN-HEBERT Romain

MINARY Théo

MATEOS Guillaume

*Site web :*

<https://thefallen.one>



Mai 2018

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Présentation du projet</b>	<b>4</b>
1.1 Origine . . . . .	4
1.2 Scénario . . . . .	4
1.3 Gameplay . . . . .	6
<b>2 Présentation du groupe</b>	<b>7</b>
2.1 Création du groupe . . . . .	7
2.2 Membres du groupe . . . . .	8
2.2.1 Hugo Rosenkranz . . . . .	8
2.2.2 Romain Hasselmann-Hébert . . . . .	9
2.2.3 Guillaume Mateos . . . . .	10
2.2.4 Théo Minary . . . . .	11
2.3 Esprit Startup . . . . .	12
2.3.1 Travailler en équipe . . . . .	12
2.3.2 Stratégie de communication . . . . .	13
<b>3 Réalisation du projet</b>	<b>15</b>
3.1 Répartition des tâches . . . . .	15
3.2 Les réalisations personnelles . . . . .	16
3.2.1 Hugo Rosenkranz . . . . .	16
3.2.2 Romain Hasselmann-Hebert . . . . .	23
3.2.3 Guillaume Mateos . . . . .	28

3.2.4 Théo Minary . . . . .	33
3.3 Pour la dernière soutenance . . . . .	37
3.3.1 Nouveau personnage . . . . .	37
3.3.2 Ajout de PNJ non hostile . . . . .	37
3.3.3 IA des gardes . . . . .	38
3.3.4 Ajout de niveaux en solo . . . . .	38
3.3.5 Dernière carte multijoueur . . . . .	38
3.3.6 Menu d'options . . . . .	38
3.3.7 Graphisme menu et interface . . . . .	39
3.3.8 Système de traduction des menus . . . . .	39
3.3.9 Sons d'ambiance . . . . .	39
3.3.10 Bande-annonce . . . . .	39
<b>Conclusion</b>	<b>40</b>

# Introduction

187 jours plus tôt, une idée est née. Quatre étudiants de l'école Epita venaient tout juste d'entamer la réalisation du plus gros projet qu'ils avaient eu à faire jusque-là. C'était avant tout l'occasion rêvée pour nous de concevoir un jeu qui répondait le plus possible à nos attentes, un jeu qui nous correspondrait plus qu'aucun autre jeu. Nous étions alors plus occupés à laisser courir notre imagination quant au contenu et *gameplay* du jeu plutôt qu'à nous rendre compte de la tâche vertigineuse qui nous était demandée. En effet réaliser presque entièrement un jeu vidéo en un peu plus de six mois en étant quatre programmeurs novices semblait presque impossible à mener à bien. Au départ, nous avions plein d'idées les plus extravagantes les unes que les autres. Chacun avait son concept qu'il voulait intégrer dans le jeu mais ils n'étaient pas forcément cohérents entre eux. C'est dans cette optique d'effervescence et de liberté artistique presque absolue que nous avons commencé ce travail.

Petit à petit, nous avons calmé nos ardeurs et avons commencé à envisager plus sérieusement et pragmatiquement le projet. Cet exercice transdisciplinaire que représente la conception d'un jeu vidéo ne nous a pas seulement fait travailler nos compétences de programmation mais aussi celles de communication et d'organisation pour gérer une équipe et des échéances. Nous avons dû planifier notre travail pour être efficace et diviser les tâches entre les membres de l'équipe. Nous avons également fini par trouver un nom pour ce beau projet : **The Fallen One...**

Nous arrivons maintenant à la fin de ce périple qui n'aura pas été de tout repos. Ce rapport va exposer le cheminement d'une idée plutôt floue jusqu'à sa réalisation en jeu vidéo. Dans un premier temps, nous vous expliquerons l'origine de The Fallen One, le scénario sous-jacent et comment nous l'avons adapté en jeu vidéo. Dans un second temps, nous nous présenterons, d'abord en tant qu'équipe, puis individuellement et enfin sous l'aspect d'une *start-up*. Dans un dernier temps, une partie plus technique détaillera la conception du jeu vidéo, autant la partie programmation que les autres logiciels que nous avons dû utiliser pour rendre ce projet possible. Enfin, dans la conclusion vous saurez si nous avons réussi à atteindre notre objectif, à savoir, créer le jeu vidéo de nos rêves !

# 1 Présentation du projet

## 1.1 Origine

Étant tous nés à la fin des années 90, nous n'avons pas directement connu les années 80 mais elles ont très largement imprégné la culture pop dans laquelle nous baignons. Cela est encore plus vrai pour le monde des jeux vidéo qui a connu son essor durant ces années. Nous avons donc voulu rendre hommage à cette période et faire un jeu à la fois rétro dans l'esthétique de l'univers et innovant dans les idées.

Les années 80 ont aussi été l'âge d'or de la contre-culture du *hacking* avec l'envie de plus de liberté et l'idée que les données devaient être gratuites et les logiciels libres pour tous alors même que les entreprises privées de l'informatique prenaient de plus en plus d'importance.

Nous avons donc cherché comment représenter cette vision du monde dans notre jeu. Après plusieurs jours de brainstorming sans résultat, nous avons eu une idée : le joueur devra exploiter de faux *glitch*<sup>1</sup> pour pouvoir s'échapper d'une prison. En effet, les *glitch* étant des comportements imprévus par les développeurs du jeu qui permettent aux joueurs de se libérer de certaines contraintes de *gameplay*, ils sont une parfaite métaphore du *hacking* pour notre projet.

## 1.2 Scénario

Notre jeu se déroule dans une société dystopique, une dictature où les individus n'ont pas de liberté à proprement parler. Les *glitcher*, personnes ayant la maîtrise des *glitch*, sont donc traqués par la police et sont surnommés les mutants car les capacités que leur confèrent ces *glitch* les rendent surhumains.

Dans notre jeu, ces *glitch* ne seront bien entendu pas des erreurs dans le code du jeu mais bien des fonctionnalités voulu mais ressembleront dans leur uti-

---

1. Glitch (cf. Wikipédia) : Ce terme est employé pour désigner un bogue dans un jeu vidéo, où un objet animé a un comportement erroné (par exemple : passage au travers des murs, « téléportation » inattendue), qui peut être exploité pour finir un jeu le plus vite possible, comme dans les concours de speedrun et plus particulièrement ceux de tool-assisted speedrun.

lisation et dans leur fonctionnement à de vrai *glitch* (traverser un mur ou devenir invisible pour les autres joueurs). Ainsi, le joueur devra pour s'échapper d'une salle de la prison, analyser son environnement et trouver des endroits propices aux *glitch*, comme un polygone complexe dont la *hitbox* pourrait être mal codé et permettre de le propulser dans les airs par exemple. Évidemment, ces *glitch* seront partiellement cachés et le joueur devra apprendre à les repérer tout au long du jeu. Certains ne sont pas nécessaires pour finir le niveau mais facilitent leur complétion ou confèrent des bonus. La première partie du jeu se déroulera donc dans une prison où il faudra combiner plusieurs *glitch* pour passer à la salle suivante. Au fur et à mesure des salles, le niveau augmentera et il y aura aussi des gardes qu'il faudra soit tuer soit esquiver furtivement.

À la fin de cette partie, le joueur arrive à s'échapper de la prison, c'est alors que commence la deuxième partie qui, étant plus ambitieuse, ne sera présente dans la version finale seulement si nous avons le temps. Dans cette partie le joueur rejoint un groupe de résistants mutants dont le chef se révèle être très charismatique. Mais très vite, le joueur se rend compte que le chef est fou et qu'il veut tuer tous les humains non mutants.

Notre but était de ne pas faire simplement un jeu mais plus une œuvre artistique avec sa vision du monde et qui fait réfléchir, comme le jeu *BioShock* par exemple. De plus, le *gameplay* est au service du scénario, en effet, l'utilisation de *glitch* qui confère au joueur des libertés par rapport aux fonctionnalités prévues par les développeurs du jeu est finalement une mise en abyme de la liberté que recherchent les mutants par rapport à ce que la dictature leur impose. Le point culminant de l'œuvre est atteint lorsque, à la fin, le joueur est face à un dilemme entre tuer le chef des mutants qui l'a aidé à s'évader et donc trahir ses semblables en se mettant à la merci des policiers ou le laisser tuer tous les humains non mutants et devenir son bras droit pour diriger un nouveau monde.

*Que choisirez vous ?*

## 1.3 Gameplay

Notre projet sera donc un jeu orienté aventure/puzzle. Aventure pour le côté exploration des niveaux et puzzle pour la certaine réflexion que le joueur devra mener pour parvenir jusqu'à la fin du niveau. En effet il devra penser à comment franchir des obstacles grâce aux différents glitches que le gameplay lui permettra. Ainsi vous pouvez terminer les différents niveaux de plusieurs manières, évidemment foncer dans le tas est rarement la meilleure solution. De plus quelques petits bonus sont cachés dans chaque niveaux, afin d'en pousser l'exploration au maximum. Comme l'orientation du projet a changé en cours de route, nous avons décidé de moins nous pencher sur la partie hors-ligne pour approfondir la partie en ligne. Ainsi cette partie a pris un tournant bien plus important durant le développement pour proposer une expérience multiple aux joueurs.

Pour le multijoueur, nous proposerons un mode de jeu du type dynamique/course à l'objectif. Les joueurs au nombre de 4 maximums devront parvenir le plus rapidement sur un objectif mais aussi empêcher les autres joueurs d'y parvenir grâce aux armes qui seront à leurs disposition. Le premier à atteindre ce dernier remportant la partie. La plupart des glitches proposés dans le mode solo seront accessibles en multijoueur. Vous pourrez tout savoir concernant vos statistiques sur notre site internet une fois connecté.

## 2 Présentation du groupe



### 2.1 Création du groupe

**La formation du groupe et son origine :** Dark Waffle est le nom que nous avons sélectionné pour notre groupe. Pourquoi ce nom ? Et bien l'explication est très simple ! Rapidement après avoir fait connaissance, nous avons pris l'habitude de faire des soirées gaufres de temps en temps après les cours chez Hugo Rosenkranz. C'est une tradition dans sa famille de faire des gaufres ou des crêpes les mercredis soir. Ainsi nous avons pu renforcer nos liens et créer le petit groupe que nous sommes actuellement grâce aux soirées gaufres. Ce nom de groupe nous est donc venu naturellement. Mais pourquoi Dark ? Car il fallait un adjectif imposant et que c'était la période de Star Wars 8...

Nous avons tous les quatre de grandes motivations pour réaliser ce projet car nous sommes tous passionnés de jeux vidéo. Chacun à un univers qu'il affectionne ainsi, nous pouvons apporter notre vision afin de réaliser un jeu original et complet.

## 2.2 Membres du groupe

### 2.2.1 Hugo Rosenkranz

Bien qu'ayant un cursus très généraliste, Bac S SVT, j'ai très tôt eu envie de faire de la programmation mon métier. Tout a commencé durant mes années de collège lorsque j'ai découvert le mythique jeu Minecraft. À première vue, on fait vite le tour de ce jeu aux graphismes peu attrayants mais c'était sans compter sur l'immense communauté qui publie des améliorations ou des ajouts gratuits appelés *mods*. J'ai donc voulu à mon tour modifié le code du jeu pour créer un nouveau bloc, une nouvelle épée ou même un nouveau monstre... C'est alors que j'ai appris sans m'en rendre compte à *coder*.

Cinq années plus tard, je me retrouve à Epita tout excité à l'idée de développer mon propre jeu et prêt à tout donner pour réaliser l'idée que mes collègues et moi avons en tête ! Pour cela, je peux m'appuyer sur mes compétences acquises durant ces dernières années. J'avais notamment fait un jeu 2d en python pour le projet final d'ISN en terminale. Mais surtout j'ai une certaine expérience avec la programmation en général et particulièrement la programmation orientée objet ce qui me permettra d'aborder plus sereinement la multitude de problèmes que nous allons certainement rencontrer durant notre périple.

De plus, mon amour pour l'univers incroyable des années 80 et sa vision du *hacking*, que l'on peut retrouver dans le film Matrix par exemple, trouve son écho dans notre jeu, The Fallen One, ce qui n'est pas pour me déplaire !

### 2.2.2 Romain Hasselmann-Hébert

Je suis issue de terminal S SVT, et j'ai été rapidement attiré par les différents outils informatiques. En effet, dès mes 14 ans, je me renseignais sur les nouvelles technologies et sur l'informatique en général, certes en ayant une vision naïve de ces éléments. Ainsi j'ai fait l'option ISN en Terminale pour apprendre les bases de l'informatique, et améliorer les quelques notions que j'avais. En effet j'avais fait quelques mois d'apprentissage de Java lors de mon séjour de 10 mois au Canada, et j'avais quelques notions en HTML5/CSS3 que j'avais faits en autodidacte. Je souhaitais déjà intégrer une école d'ingénieries informatiques dès le collège, et mon choix s'est orienté vers EPITA lors de la sélection post-bac effectué en Terminale.

C'est l'école que je trouvais le plus attrayant, et qui à mon sens pouvait m'apporter le plus, et répondre à mes attentes en matière d'apprentissage informatique. En projet de fin d'année j'ai déjà réalisé un jeu en Python, un petit Zelda-Like en 2 D, le fait est que l'on avait aucun bagage informatique, ou du moins bien moins solide que celui qu'on a en ayant passé un semestre à EPITA. Ce projet sonne comme une façon de réaliser un vrai projet à mes oreilles, de manière professionnelle et complète, et en tant que grand gamer, quoi de rêver de plus que de réaliser son propre jeu . Mes collègues et moi sommes beaucoup motivé pour réaliser ce projet, et j'espère que l'on pourra faire tout ce que l'on souhaite.

### 2.2.3 Guillaume Mateos

Je suis passionné d'informatique depuis mes 12 ans environ. C'est d'ailleurs à ce moment-là que j'ai commencé à regarder et apprendre les bases de la programmation Web(HTML/CSS entre-autres). Après avoir de bonnes base, je me suis très vite rendu compte que ce dernier language me limitait et c'est pour cela que j'ai decider d'apprendre le PHP/MySql. Et très vite j'ai developper des petits site dynamiques composés de plusieurs systemes assez basiques(petit espace membres, systeme d'article). C'est donc tout naturellement que je me suis tourné vers des études d'informatique.

Venant d'une terminale S SVT option ISN j'ai decider de candidater à l'EPITA car mon frère y était déjà et ce qu'il faisait à l'école à l'époque me fascinait. De plus la réalisation d'un projet de fin d'année en ISN m'a permis de confirmer mon choix concernant mes études dans le supérieur. une fois de plus, lorsque j'ai appris que nous devions réaliser un projet en C# et que nous pouvions réaliser un jeu en 3D, c'était le rêve d'enfant à porter de main. L'idée de developper un jeu en 3 dimensions m'a toujours fasciné et aujourd'hui c'est ce que je dois faire avec l'aide mes camarades. Je suis donc tout naturellement excité et motivé par ce projet qui s'annonce riche en rebondissements.

## 2.2.4 Théo Minary

J'ai toujours été attiré par l'informatique, mes parents ne sont pas du tout familier avec ce milieu et ont par conséquent été étonné lorsque qu'à tout juste 3 ans j'arrivais déjà à allumer et éteindre l'ordinateur de mon père. Puis ma passion pour les jeux-vidéos est arrivé. Je passais pas mal de temps devant mon écran, c'est pour cette raison que j'ai commencé à m'intéresser au fonctionnement de mon ordinateur, que ce soit physique ou immatériel. J'ai commencé par apprendre le Python car on m'avait dit que pour commencer c'était le plus adapté, puis j'ai obtenu quelques notions en HTML5/CSS3...

Arrivé au lycée il était naturel pour moi de prendre l'option ISN, mais lors de mon projet de SI j'ai eu l'occasion d'utiliser et programmer des cartes ARDUINO. Je trouvais super intéressant de voir immédiatement le résultat de mes actions, j'ai même fini par créer un moyen d'allumer la lumière de ma chambre grâce à une télécommande et un capteur infrarouge programmé sur ARDUINO. J'ai maintenant acquis une beaucoup plus de connaissance grâce à l'EPITA qui est un choix plus que logique pour moi. J'ai de grandes ambitions certes mais je suis prêt à tout donner pour y arriver et je suis sûr que l'EPITA sera à la hauteur de celles-ci. Ce premier projet sera sûrement très intéressant que ce soit pour moi ou pour mes camarades je suis sûr qu'on se donnera à fond pour réussir ce premier jeu.

## 2.3 Esprit Startup

Compte tenu de notre effectif et du délai qui nous était imposé pour réaliser notre jeu vidéo, nous nous sommes à bien des égards comportés comme une startup.

### 2.3.1 Travailler en équipe

*L'union fait la force* est une expression connue de tous, et pourtant, travailler en équipe n'est pas forcément innée, certainement parce que depuis tout petit, l'école valorise principalement le travail individuel. Nous avons donc dû apprendre à se partager les tâches au lieu de coder seul dans son coin, ce qui est d'autant plus vrai pour moi, le chef de projet. Je vais ici vous expliquer comment j'ai organisé ce travail en équipe.

Tout d'abord, il a fallu avoir un moyen de communiquer facilement, nous avons sans hésiter utilisé le logiciel *Discord* qui permet de créer des salons de chat et de faire des appels groupes. Nos réunions en dehors de l'école se sont presque exclusivement déroulé grâce à cet outil. J'ai par la suite voulu nous permettre de facilement communiquer les tâches que nous étions en train de réaliser ainsi que les potentielles difficultés que nous rencontrions. Cela permet à tout le monde d'aider à la complétion d'une tâche à laquelle il n'était pas assigné initialement s'il en a l'aptitude. J'ai donc implémenté un cahier de bord sur notre site, je détaille plus cela par la suite dans mes réalisations personnelles.

Ensuite, pour partager facilement notre travail à l'ensemble du groupe, nous avons utilisé la très connue plateforme *Github* qui permet de créer et de maintenir facilement des projets informatiques collectifs. Malgré quelques cafouillages lorsque deux personnes mettaient à jour le code en même temps, cet outil a été très pratique et je dirais presque indispensable à la réalisation d'un tel projet. En effet, même pour une petite équipe, avoir chacun une version différente du code sur son ordinateur peut vite tourner en désastre si deux personnes modifient des parties similaires.

Enfin, avant la deuxième soutenance, nous avons pu faire progresser le projet encore plus vite en se réunissant tous dans mon appartement pour coder

du matin au soir pendant deux jours. Je considère cette expérience comme un avant-goût de la piscine en première année du cycle ingénieur.



### 2.3.2 Stratégie de communication

Pour jouer encore un peu plus le jeu de cet exercice, nous avons voulu faire un peu de communication sur le jeu. Nous avons donc créé une page *Facebook* ainsi qu'un site internet. Nous comptons également faire une bande-annonce pour présenter le jeu.

Pour nos publications, nous jouons sur le concept inattendu de mettre intentionnellement des *glitch* dans notre propre jeu. *Les glitch comme vous ne les avez jamais vus* est le slogan de notre jeu.



Nous comptons également sur une communauté de bêta-testeurs qui peuvent dorénavant télécharger une version en développement du jeu pour nous donner leur retour sur le jeu. Ils peuvent, en effet, nous écrire directement dans la rubrique *Feedback* de notre site les changements qu'il voudrait voir dans notre jeu tel que des problèmes rencontrés ou même des fonctionnalités à ajouter.

Malheureusement, le manque de temps à consacré dans ce domaine somme toute secondaire comparé au développement du jeu ne nous a permis d'exploiter pleinement ces idées.

# 3 Réalisation du projet

## 3.1 Répartition des tâches

Tâches	Membre de l'équipe			
	Hugo R.	Romain H.	Théo M.	Guillaume M.

Graphisme	-	R	S	-
Level design	S	R	-	S
Multijoueur	R	S	S	-
Interface	-	-	R	S
Joueur et PNJ	R	S	-	S
Gameplay (Interaction)	S	-	S	R
Audio	-	S	R	-
Site Web	S	-	-	R

**R** : Responsable

**S** : Suppléant

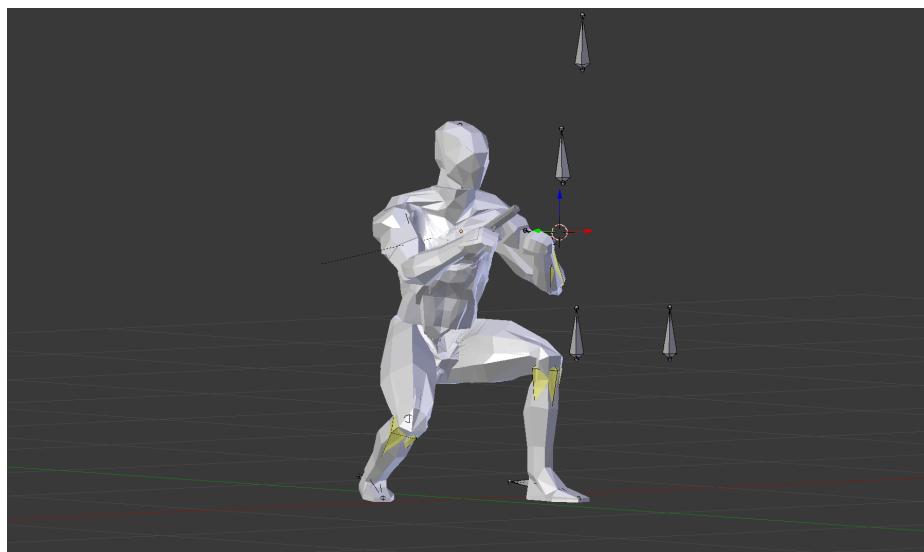
## 3.2 Les réalisations personnelles

### 3.2.1 Hugo Rosenkranz

#### Animations des modèles 3d

Le personnage principal qu’incarnerait le joueur a très vite été au cœur de la conception du jeu. La méthode la plus simple aurait été de trouver un modèle sur la plateforme d’échange des *Assets Unity* tout fait avec les animations préexistantes. Cependant, aucun modèle gratuit ne nous convenait. Romain a finalement trouvé un modèle 3d qui correspondait à nos attentes sur un site de partage libre de droits mais il n’était pas animé. Je l’ai donc importé sur le logiciel de modélisation *Blender* pour le mettre en scène afin de le rendre plus vivant. Par exemple, pour que le joueur est l’impression de faire marcher son personnage, il ne suffit pas de déplacer le modèle 3d sur le terrain mais il faut aussi que ses jambes se balancent d’une certaine façon, ainsi que ses bras et le reste de son corps.

Pour effectuer cette tâche essentielle que je n’avais jamais réalisée auparavant, j’ai décidé d’utiliser la technique la plus courante : *rigger* le modèle 3d. Derrière ce terme intrigant, se cache une méthode plutôt simple qui consiste à créer un squelette virtuel sur le modèle 3d qui va contraindre les mouvements. Par exemple, quand je bouge le pied du modèle en avant, toute la jambe va venir avec, mais pas le bras, également si je le lève, le genou du modèle va se plier. Cette technique fonctionne très bien dans la plupart des cas, mais nécessite parfois de s’y reprendre à plusieurs fois pour effectuer des contorsions plus complexes, comme pivoter une certaine partie du bras seulement.



### L'animation du personnage accroupi

[voir la vidéo](#)

Une fois le squelette terminé, il n'y a plus qu'à enregistrer les différentes séquences d'animations (marcher, courir, sauter, ...), puis importer le tout dans *Unity* et utiliser son gestionnaire d'animation *Mecanim*. On peut ainsi connecter les animations entre elles et les lancer en fonction de variable qui représente les actions du joueur : quand ma variable booléennes COURIR est activée, l'animation de course est jouée. Ces variables sont contrôlés dans le code C# du joueur. Au fur et à mesure que le nombre d'animations du joueur augmentait, le schéma *Mecanim* devenait de moins en moins lisible, j'ai donc utilisé les *Blending Tree* qui permettent de grouper les animations ensemble. par exemple le *Blending Tree* MOUVEMENT va contenir courir et marcher, en fonction de la valeur de la variable flottante VITESSE il va lancer l'une ou l'autre animation.

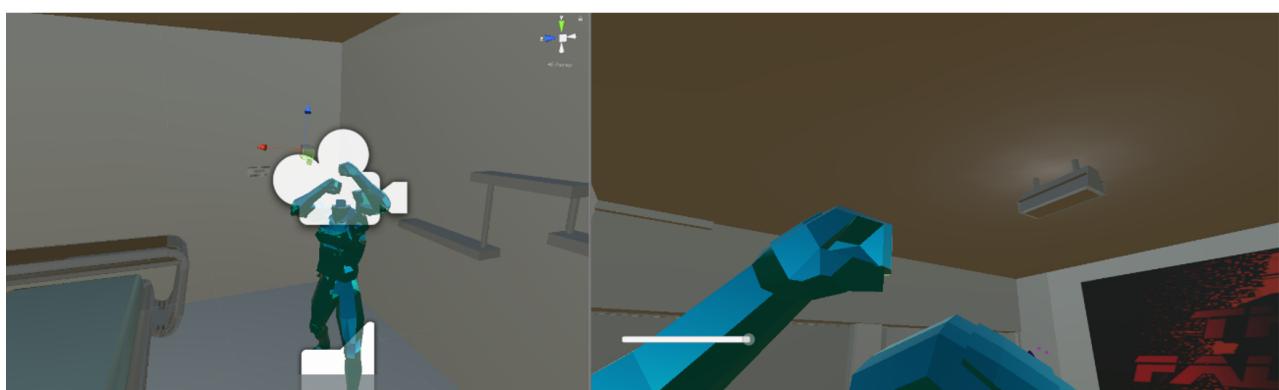
Finalement, j'ai réalisé une dizaine d'animation pour le joueur, ce qui représente une certaine quantité de travail. Alors, lorsque par la suite, j'ai dû implémenter un PNJ (personnage non joueur) dans le jeu, Nous avons décidé de réutiliser le même modèle 3d avec les mêmes animations que pour le joueur, en ne changeant que les couleurs et éventuellement certaines petites animations. Cela a grandement accéléré la création du PNJ et nous a permis de nous concentrer davantage sur le code C#.

### Classe du joueur et du PNJ

Ayant commencé par le joueur, sans envisager tout de suite le PNJ, j'ai

tout d'abord créé des scripts uniquement pour le joueur. Mais dès lors que la conception du PNJ a eu lieu, je me suis rendu compte que le code source aurait beaucoup de redondance avec celui du joueur. J'ai donc décidé de créer une classe abstraite *Human* dont hériterait le joueur et le PNJ ce qui a également accéléré l'implémentation du PNJ. Cette classe s'occupe principalement de lancer les animations comme vu précédemment mais aussi des déplacements, des sons et plus généralement des différentes actions que peuvent faire les personnages dans le jeu. Cependant, les contrôles des personnages sont gérés par d'autres scripts respectivement *BotController.cs* et *PlayerController.cs* pour le PNJ et le joueur. De plus, ils implémentent tous deux l'interface *ITarget* qui permet de recevoir un coup (ou une balle) et de mourir.

Le joueur est en vue à la première personne, c'est-à-dire que la caméra se trouve au niveau de la tête du modèle 3d. La tête est donc invisible pour le joueur bien que visible pour les autres joueurs en multijoueur. Lorsque le joueur tourne la tête, c'est-à-dire la caméra, horizontalement, tout le corps suit le mouvement. Par contre, lorsque le joueur regarde vers le haut ou vers le bas, il faut que seuls ses bras suivent tourne avec la tête car les pieds doivent rester au sols. Pour ce faire, j'ai utilisé qui consiste à importer un deuxième fois les bras seulement du modèle 3d et à les *accrocher* à la caméra plutôt qu'au corps du joueur. Les bras initiaux du modèle 3d sont donc eux-aussi invisibles en vu à la première personne et inversement pour les autres joueurs. Ses bras supplémentaires ne sont pas animés par la classe *Human* car ils constituent une particularité du joueur et sont donc régis par sa classe fille *Player*. Quant à elle, la classe *PlayerController* transcrit les frappes du clavier en action et modifier parfois certain réglage de la caméra. Par exemple pour s'accroupir, le joueur appuie sur CTRL, *PlayerController* abaisse la caméra et execute la fonction *Player.Crouch()* qui renvoie vers *Human.Crouch()* pour mettre à jour les animations et diminuer la taille de la *hitbox*.



Les bras du joueur en vue à la première personne

Le PNJ est un élément incontournable pour presque tous les jeux vidéo. Il permet de rendre l'aventure solo du joueur plus vivante. Il ne fallait donc pas minimiser cette partie. Les PNJ les plus important de notre jeu sont les gardes de la prison, les autres n'interagissent pas directement de joueur mais servent à rendre le décor plus réaliste. les gardes sont hostiles, ils doivent donc repérer le joueur, le suivre et lui tirer dessus. L'intelligence artificielle que j'ai faite est très rudimentaire, elle se trouve dans *BotController.cs*. Guillaume m'a un peu aidé en ajoutant notamment la notion de distance de repérage : lorsque le joueur est plus loin que cette distance, le garde ne doit pas aller vers lui ni lui tirer dessus et s'il est trop près le garde s'arrête à une certaine distance du joueur. J'ai par la suite ajouté que s'il y avait un obstacle entre le joueur et le garde, ce dernier ne tirer pas inutilement. Bien que l'intelligence du garde soit très basique, cela reste plutôt convainquant en jeu.

## Multijoueur

Pour rendre l'expérience de jeu plus intéressante, en plus des niveaux solo, notre jeu contient également des terrains jouables à plusieurs en ligne. Les joueurs peuvent se tirer dessus, ramasser des bonus et évidemment utiliser des *glitch* afin d'arriver au drapeau pour remporter la manche. Au fur et à mesure du développement du jeu, le multijoueur qui était à l'origine secondaire a pris une place de plus en plus importante dans le jeu si bien que lors de la première soutenance, seul le niveau multijoueur était complètement jouable.

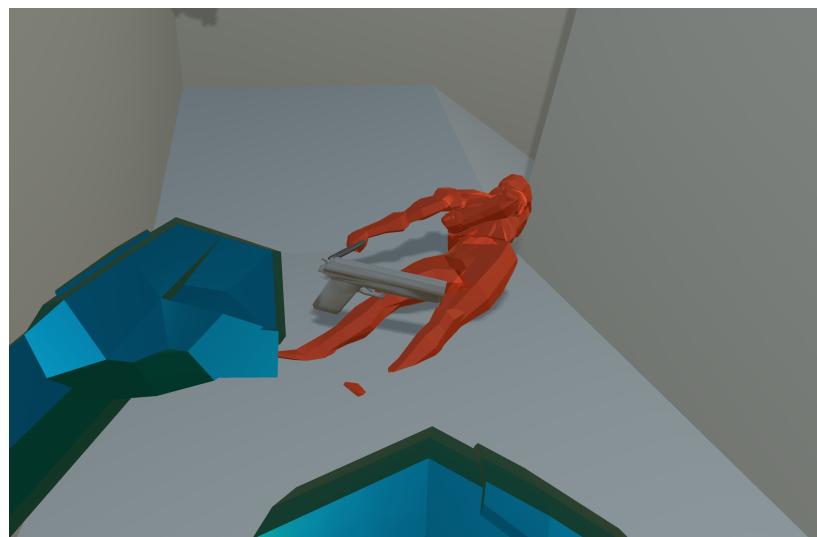
Etant le responsable de cette partie, j'ai donc dû apprendre très vite les mécaniques réseaux de *Unity*. Pour me faciliter un peu la tâche j'ai décidé d'utiliser le *NetworkManager* de *Unity* qui synchronise automatiquement les déplacements et les animations entre les joueurs mais permet moins de flexibilité que d'autres solutions. Les scripts pour le joueur sont ainsi séparés en deux : Solo et Network héritant respectivement de *MonoBehaviour* et de *NetworkBehaviour*. Il a également fallu faire la distinction entre le code qui devait s'exécuter sur le client et celui sur le serveur. Par exemple, lorsqu'un joueur meurt, il faut mettre à jour la vie du joueur sur le serveur, puis jouer l'animation pour tout le monde et enfin marquer un message sur le client qui vient de mourir.

Le multijoueur est certainement la partie qui m'a donner le plus de fils à retordre. En effet, mis à part le modèle 3d et les animations du joueur, tout a dû être modifié. C'est donc également la partie qui contient le plus de comportants inattendus comme des animations qui ne se jouent que pour le client par exemple. En plus de mes autres travaux au sein du projet, l'essaie donc de revenir régulièrement sur le multijoueur pour corriger des problèmes. Finalement, j'ai essayé d'implémenter une redirection automatique de port avec la technologie très répandue *UPnP* pour que le joueur puisse très facilement héberger une partie et inviter ses amis. Cela fonctionne sans problème sur ma box SFR mais apparemment pas aussi bien sur celles de mes coéquipiers. Le problème n'est pas très pratique à régler puisqu'il faudrait que je me déplace chez eux pour analiser la situation.

### Interaction avec le Gameplay

Une fois les niveaux finis par Romain, nous avons, avec Guillaume le responsable des interactions avec le joueur, décidé où placer les gardes que j'avais créés. J'ai ainsi configuré les différentes distances de détection en fonction de leur place sur le terrain, certains gardes attaquent de près et d'autres de loin. Ce travail, bien qu'un peu aliennant quand il y a beaucoup de PNJ à placer comme dans le niveau 2 est très important car il va très largement influencer la difficulté du niveau. De plus, les gardes permettent intuitivement du guider le joueur qui va penser qu'il est sur le bon chemin quand il les croise. Cela peut également être utilisé pour piéger les joueur et le diriger vers la mauvaise direction, c'est le cas dans le niveau 1.

J'ai également utilisé la classe *box* de Guillaume pour en créer deux nouvelles. La première est un pistolet car vous pouvoir être ramasser par le joueur. Il apparait quand ce dernier tue le premier garde dans le niveau 1, cela donne l'illusion que le garde lache son arme en mourant. La deuxième *box* est le matérialise la fin du niveau, c'est un drapeau qui passe au niveau suivant lorsque le joueur entre en collision avec. Ce même drapeau est utilisé pour terminer les parties en multijoueur, le joueur l'attrapant étant le gagnant !



Pistolet ramassable par le joueur

### Synchronisation jeu-serveur web

Ayant ressenti une légère baisse de motivation de mes troupes après la première soutenance, j'ai décidé d'introduire les FALLENCOIN. Accouplés à un cahier de bord qui permettait aux développeurs d'inscrire leurs tâches, de notifier leur avancement et d'être récompensé en FALLENCOIN. De plus, ces pièces sont aussi trouvables dans les niveaux et pourront récompenser les meilleurs joueurs du jeu. Leur utilité n'est pas encore clairement définie mais elles serviront sûrement à débloquer des bonus.



Le FALLENCOIN<sup>©</sup>

Le site web ayant entre-temps été ouvert à tous, j'ai décidé de l'utiliser pour afficher aux joueurs toutes sortes de statistiques sur sa progression dans le jeu comme

son nombre de parties gagnés/perdues ou le nombre d'élimination qu'il a pu réaliser. Avec l'aide de Guillaume, nous avons implémenté cette fonctionnalité sur le site web dans la partie Dashboard accessible à tous à condition de créer un compte.

Pour rendre le tout cohérent, il fallait que le joueur puisse se connecter à son compte depuis le jeu et que ce dernier envoie les différentes statistiques au serveur web. Au lancement du jeu, le joueur à le choix entre se connecter avec son compte, en créer un s'il n'en a pas, ou jouer en hors ligne mais sa progression ne sera pas sauvegardée. Il peut donc gagner des FALLENCOIN s'il trouve ceux cachés dans les niveaux solos et augmenter son nombre d'élimination et de parties gagnés en ligne.

### 3.2.2 Romain Hasselmann-Hebert

#### Recherche d'Asset

Dès le début il fallut rechercher des assets car il aurait été trop difficile de tout faire par nous-mêmes. Ainsi, en trainant sur des sites qui partagent des objets 3D gratuitement et libre de droits, j'ai pu trouver notre bonheur. Quelques modifications sont parfois nécessaire pour que l'objet choisi colle plus à un style graphique low poly (peu de polygones). C'est de cette manière que j'ai pu trouver Bluzz, le personnage que nous incarnons, qui sert également de modèle pour les ennemis, le fusil à pompe et le décor des niveaux. De plus certains modèles peuvent donner des inspirations pour la création de niveaux, ou même la réalisation d'asset. Effectivement, les modèles trouvés n'étaient pas tous de mon goût, j'ai ainsi réalisé les miens lorsqu'il sagissait de réalisation plutôt simple, tel qu'une étagère.

#### Création des cartes

Le jeu, tel qu'on le voulait à la base, se concentrer sur une histoire narrative importante. Cette idée a été remplacé par un mode solo sous forme de puzzle et l'émergence du mode multijoueur. Ainsi j'ai décidé de créer les niveaux en essayant de respecter le background (arrière-plan) du projet. Effectivement, pour un joueur qui connaît l'histoire que nous avons créée autour de The Fallen One, il pourra clairement identifier les différents niveaux comme les stades majeurs de l'aventure. De ce fait, les niveaux ont une connexion logique entre eux. Pour créer les niveaux, j'ai tout simplement fait appel à mon imagination. Evidemment j'essaye de respecter au mieux un level design compréhensible par le joueur, pour savoir où se diriger. Mais les niveaux sont totalement originaux et uniques. Malgré une apparence visuelle basique il est très long de créer un niveau pour notre jeu, ou pour n'importe quel jeu en général. Effectivement penser la carte et la créer, la modifier, apporter des textures ou des décors, voir comment le joueur évolue dans cette dernière, etc ... Tout ceci fait qu'il est laborieux et long de créer un niveau dans un jeu vidéo.

Pour le multijoueur quand à lui, je me suis clairement inspiré de carte déjà existante dans le monde du jeu vidéo. En effet notre deuxième map correspond

tout à fait à la map Rust que l'on peut retrouver sur Call of Duty Modern Warfare 2. Ces inspirations ne sont pas anodines, puisqu'il sagit d'une carte que nous aimons mais aussi qui relève d'un dynamisme évident. La première est une sorte d'essai, de carte de tests pour le multijoueur, que nous laissons à disposition pour les joueurs qui aiment le défi, car elle est bien plus compliquée que les autres. Quant à la dernière carte, il s'agit également d'une inspiration à la série Call of Duty, une carte également connue pour son dynamisme, Dôme. Pourquoi choisir ces cartes ? Tout simplement car c'est le style de gameplay que nous souhaitons mettre en place pour The Fallen One, c'est-à-dire une partie rapide, sans ennui et avec un fonctionnement relativement binaire. La plupart des jeux gratuits multijoueur basent le temps de vie de leurs jeux sur ce fonctionnement. Car pour nous, The Fallen One est un projet de fin d'année, mais nous voulons aussi qu'il soit un jeu à part entier.

## Logo et images

Mon rôle de graphiste fut de créer l'aspect visuelle du jeu, interne ou externe. C'est-à-dire qu'il a fallu créer le logo de notre équipe Dark Waffle ainsi que celui du jeu. Evidemment nous nous sommes consulté pour savoir quel aspect donner à ces différents logos. Toutes les images que j'ai réalisées pour le projet ont étaient faites avec Photoshop, car c'est l'outil de retouche d'image que je manie le mieux. Ainsi les différentes images que vous trouverez dans le jeu ou bien sur notre site internet sortent de mon photoshop, qu'elles soient retouchées ou créées. J'ai mis à disposition les différents PSD si un membre de l'équipe souhaite faire des modifications que se sent capable de faire, ou bien de regarder comment j'ai réalisé les images.

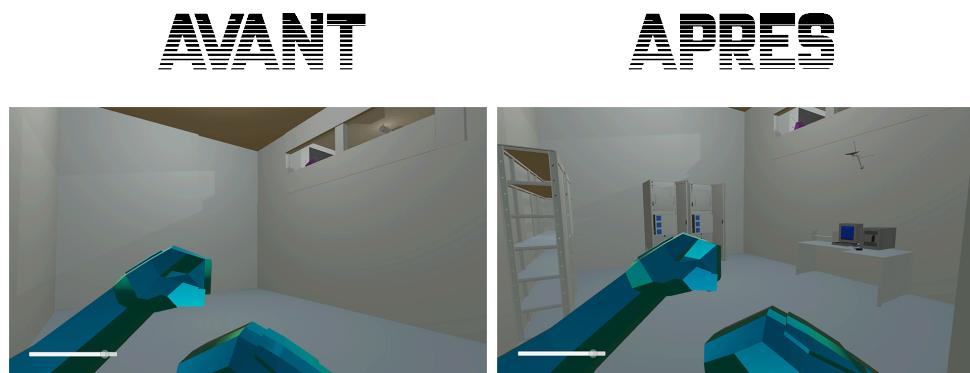
## Logo de rank

Comme nous avons décidé de créer différents rangs selon le nombre de Fallen Coin, il fallait bien évidemment créer les logos associés. De ce fait nous avons choisi des noms qui évoquent un personnage marquant fictif ou non des années 80, car c'est de cette époque que nous tirons notre style graphique. Une fois les noms sélectionnés, il ne manquait plus qu'à réaliser les logos correspondants, ce qui évoquerait le plus le nom. J'ai donc réalisé les différents logos sur Photoshop en m'inspirant d'images trouvées sur internet, et voici le résultat :



Logos des différents rangs

Comme promis à la fin de la première soutenance, The Fallen One aura plusieurs niveaux jouables. En effet, une fois les fonctionnalités souhaitées implémenté, je pouvais finir le niveau 1. Tous les niveaux sortent de mon imagination ou de celle d'un membre de la DarkWaffle. Une fois les niveaux finis, il fallait implémenter un décor, c'est pour cela que j'ai parcouru internet en quête d'asset qui colleraient à l'ambiance du jeu. De cette manière j'ai pu habiller les niveaux pour les rendre moins vides.



Avant / Après le décor

## Niveau 1 et Niveau 2

De plus nous avons ajouté quelques petites surprises dans chaque niveau ! N'oublions pas que notre ambiance tourne autour du glitch, ainsi un joueur qui cherche suffisamment trouveras quelques petits passages secrets !

Une fois le niveau 1 fini comme il faut, je me suis attelé à la conception du niveau 2. Sans perdre de vue l'histoire qu'il y a derrière le jeu, le joueur, sur ces deux niveaux disponibles, réussira à échapper aux mains des humains qui le retenait dans un bâtiment étrange. Ainsi l'échappé se fait au travers d'un parking remplis d'ennemis. Nous avons testé plusieurs fois les niveaux pour savoir comment les équilibrés. Nous voulons que nos niveaux soit dur tout en restant accessible, pour assurer un minimum de temps de vie pour le jeu. Ainsi nous avons placé les gardes et différentes box de maniére méthodique basé sur notre expérience du jeu.



Aperçu du niveau 2

Je me suis également longuement attaché à la modification des lumières pour avoir une ambiance sympa, et des ombres qui rendent un aspect plus esthétique au jeu. Selon l'environnement, les lumières ne sont pas les mêmes. Par exemple, dans le parking elles sont moins lumineuses pour donner un aspect sombre qui correspond mieux à une ambiance sous terraine.

Vous pourrez également retrouver quelques textures originales, même si nous souhaitons garder cet aspect brut. J'ai actuellement rencontré beaucoup de soucis avec l'importation de textures que je réalise sur Cinéma 4D. Ainsi à l'avenir j'essayerais de me pencher sur ce problème pour avoir une amélioration du visuel.

Nous avons au fur et à mesure que le projet avance, décidé de s'orienter vers une partie multijoueur plus importante. De ce fait il nous faut plusieurs cartes

pour garantir un renouvellement de l'expérience de jeu. Ainsi je me suis inspiré d'une célèbre carte, que tous les membres de la DarkWaffle apprécient, Rust de Call of Duty Modern Warfare 2. Ce remake à la The Fallen One permet des mouvements dynamiques et une partie sans repos ! Le but étant toujours celui qui atteint le drapeaux le premier.



Aperçu map multijoueur 2

## Tâches Annexes

Mis à part les tâches importantes où j'ai évolué seul, j'ai réalisé quelques tâches annexes qui ne nécessite pas à elles seules une partie à part entière. Nous avons souhaité changer le fond du site, car le soleil nous dérangeait, alors comme Théo avait du mal, je l'ai aidé pour avoir le résultat obtenu actuellement. De plus j'ai aidé guillaume dans la traduction des différentes variables pour le site internet, afin d'avoir les deux langues disponibles. Sans oublier de passer par la réalisation et recherche de quelques bruitages. A la demande de mes camarades je réalise également quelques images, comme celle de la Newsletter que vous pouvez voir sur le site. J'essaye avec les autres membres de la DarkWaffle de maintenir la section MAJ de notre site à jour, ainsi que notre page facebook pour maintenir notre faible communauté au courant des avancements du projet.

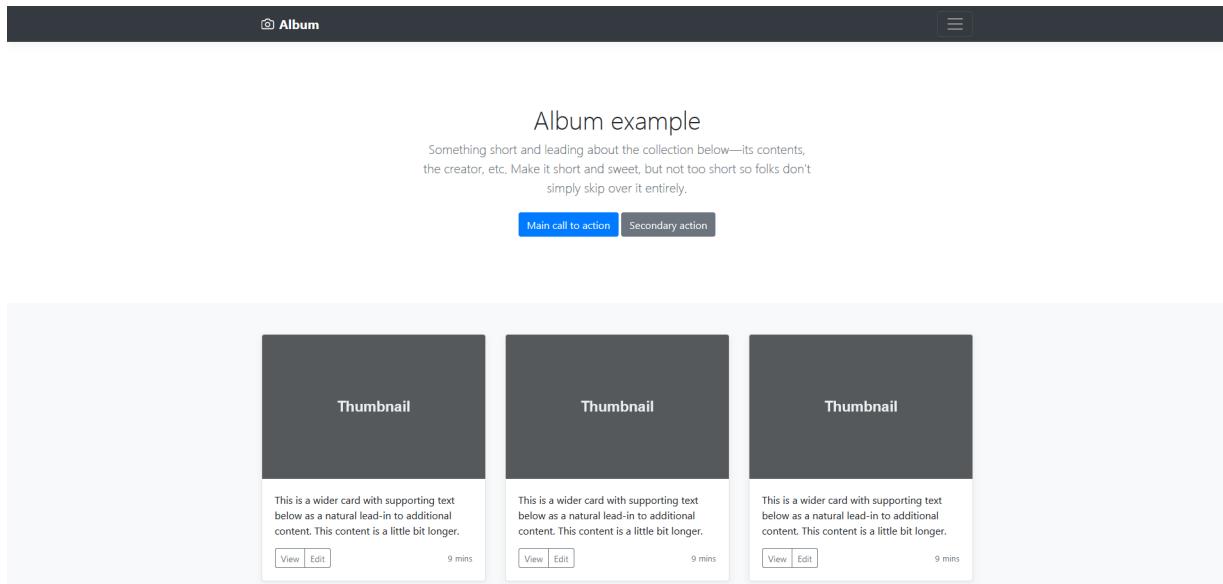
### 3.2.3 Guillaume Mateos

#### Mécaniques associées au joueur

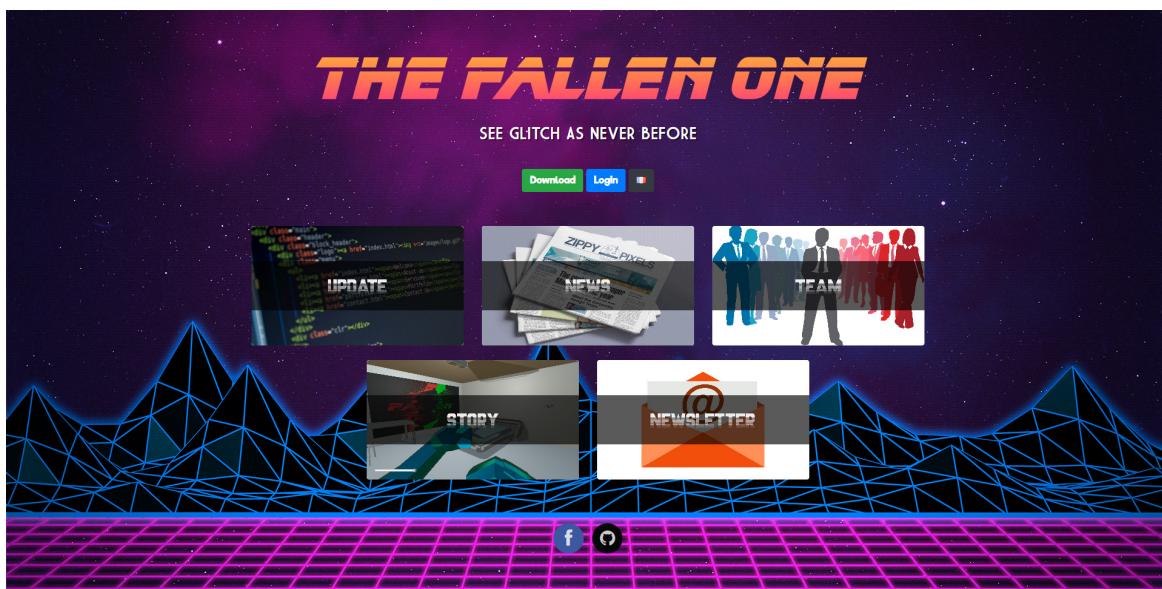
Mon premier travail a été de rechercher la méthode qui me permettrait de mettre en place les mécaniques de bases du joueur : se déplacer dans un environnement et sauter. J'ai alors commencé mes recherches sur internet et c'est alors que je me suis aperçu qu'il y avait beaucoup de façons différentes d'aborder ces mécaniques au niveau du code. J'ai alors décidé de prendre une méthode qui consiste à appliquer une force à un objet physique UNITY en fonction du déplacement désiré par l'utilisateur. Par cette technique il a fallu ajouter un composant UNITY sur le joueur : un *rigidbody*. Ce dernier permet entre autres de contrôler un objet grâce à des propriétés physiques (lui appliquer une gravité par exemple). Cependant, lorsque Romain nous a partagé la map test, il y a eu des problèmes indésirables notamment lorsque l'on essayait de se déplacer avec ce que je venais de produire. C'est pourquoi Hugo a pris les choses en main et a décidé de revoir le code composant les mécaniques de déplacement du joueur tout en conservant la technique choisie.

#### Site Internet

Étant le membre le plus à l'aise avec les langages de programmation web, j'ai très vite proposé de réaliser ce dernier. Je voulais que le site soit totalement automatisé afin que nous puissions créer et administrer les pages de ce dernier sans avoir à (re)mettre en ligne sur le serveur les pages à chaque mise à jour de l'une d'entre elles. Pour le design du site, j'ai décidé de m'aider d'une librairie CSS libre : Bootstrap. Ce fut un très bon compromis cette dernière propose un large choix de composants simples d'utilisation permettant de réaliser rapidement le template d'un site. Ainsi, grâce à cet outil notre site est d'un style sobre, propre et épuré qui permet une navigation facile entre les différentes parties de notre site internet. Il est bien évidemment personnalisé pour accrocher avec le thème de notre projet.



Template original

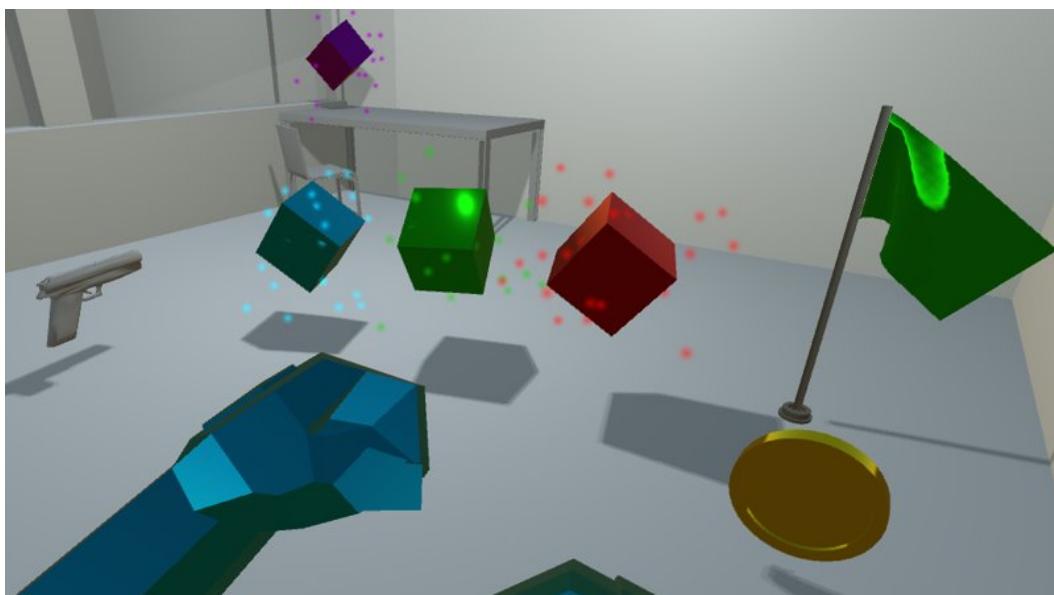


Le site aujourd’hui

Après avoir achevé la partie design du site, c'est avec l'aide du langage PHP couplé à une base de données MySQL que j'ai commencé à réaliser, sans partir d'aucune base, les différents systèmes et parties qui allaient composer notre site et nous permettre de le gérer dynamiquement. Espace membre, actualités et commentaire pour les membres, newsletter, tableau des statistiques personnels, classements du jeu et administration du site en ligne : tous ces systèmes sont en ligne et accessible à l'adresse suivante : [thefallen.one](http://thefallen.one). Tout ce qui était prévu au début du projet a vu le jour niveau site internet et il représente une bonne partie de ce que j'ai pu réaliser.

## Interaction entre l'environnement et le joueur

Très grosse composante des jeux vidéo, nous voulions bien évidemment ajouter des interactions entre le joueur et l'environnement qui l'entoure tout au long de son périple. C'est dans cette optique que j'ai implémenté le système dit de « box ». Pour mieux comprendre ce dernier, il faut imaginer ces fameuses box comme des boîtes qui lorsqu'elles sont ramassées/traversées par le joueur, déclenchent un événement prédéfini par un script. Niveau code, il faut minimum deux scripts afin d'ajouter un objet de type « box » : un premier qui décrit le comportement général d'une box et un deuxième décrivant l'événement que l'on veut déclencher et qui hérite des propriétés du script précédent. Ainsi tous les box auront le même comportement, tout en ayant un événement différent. De plus, deux comportements de box ont été implémenté : celle qui se détruise totalement lorsqu'elles sont prises et celle qui se désactive le temps qui l'effet activé s'estompe. Ainsi tout objet physique UNITY (cubes, cylindre, etc) peut devenir une "box". Nous n'avons qu'à simplement posé l'objet n'importe où sur un niveau, y attacher le script qui déclenche l'événement et lorsque le joueur « traversera » la box, l'événement qu'elle contient se déclenchera.

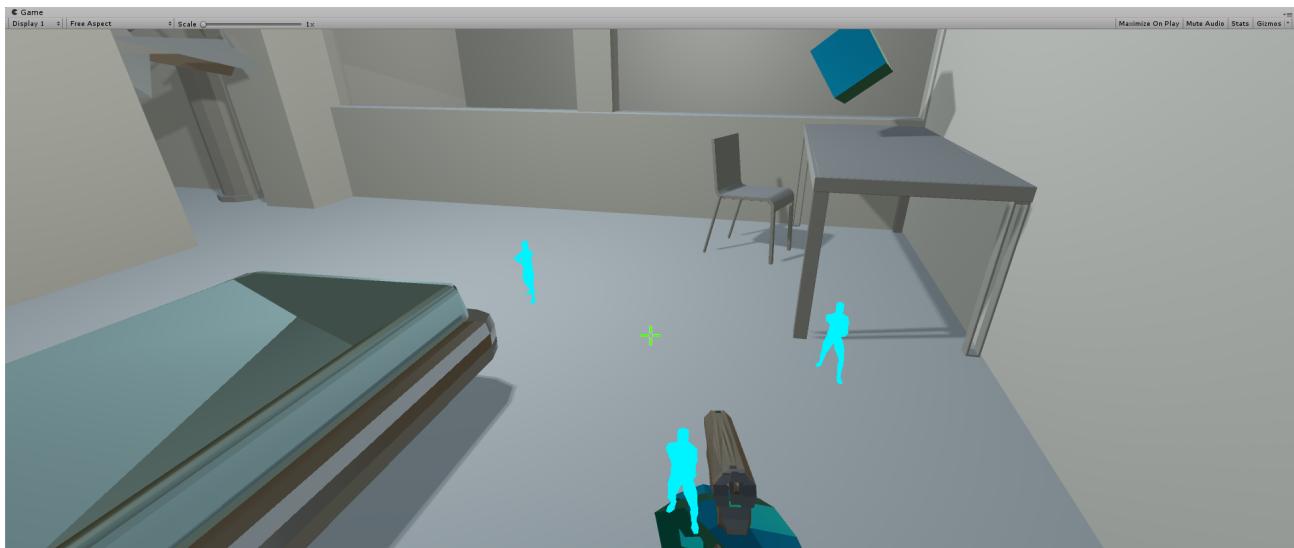


Les différentes « box's » du jeu

## La capacité dite "Wallhack"

Le principe de notre jeu était traversé les différents niveaux grâce aux différents « glitches » ou capacités permettant de contourner la difficulté que nous

proposons, nous avons voulu dès le début du projet ajouter une capacité qui permettrait au joueur de voir les ennemis à travers les murs et surfaces. Le problème était de mettre en place des conditions de déclenchement de cette capacité. En effet, si le joueur venait à abuser de cet effet, l'expérience de jeu serait beaucoup réduites et la difficulté de certains niveaux n'auraient plus de valeur. C'est pourquoi nous avons décidé que la capacité wallhack se déclencherait grâce à une "box". Au niveau du code, et afin de rendre tous les bots visibles à travers toutes les surfaces, cela a plutôt été simple. En effet, une texture me permet de rendre un objet Unity physique visible à travers toutes les surfaces. Ainsi je n'ai eu qu'à lister tous les différents ennemis et lorsque la capacité s'active, la texture des ennemis devient celle visible à travers les surfaces. Lorsqu'elle se désactive la texture des ennemis listés redeviens normal c'est-à-dire non visible à travers les murs.



**Vision lors du wallhack**

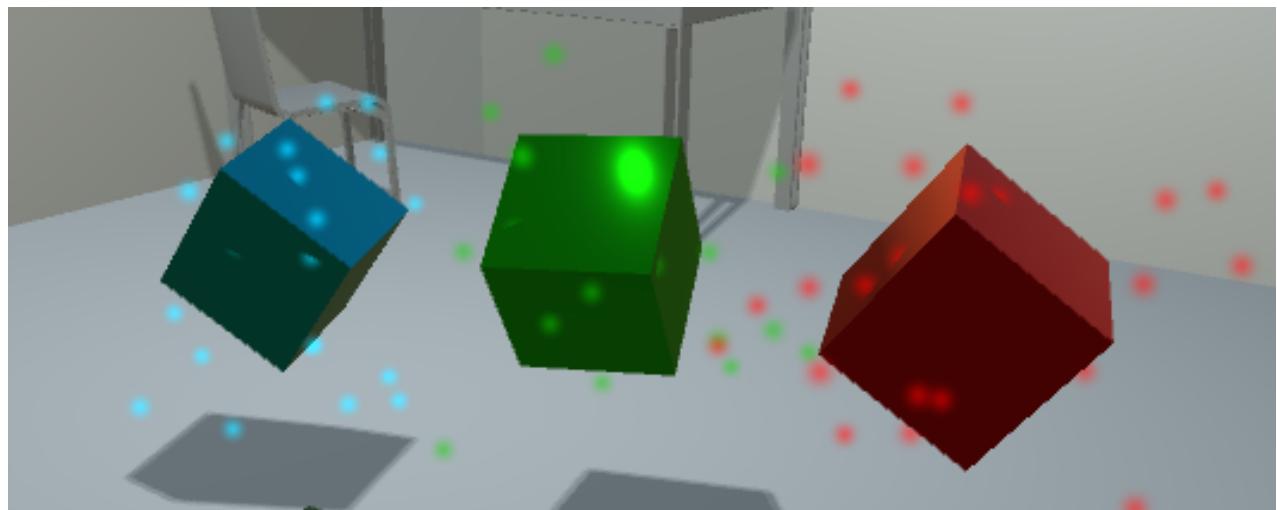
## Tâches annexes

Je considère les parties ci-dessus comme de grosses implémentations. Mais j'ai aussi fait de petites implémentations qui ne méritent pas réellement un paragraphe en entier. Je peux notamment citer les effets de particules que j'ai pu rajouter lors d'un tir dans le jeu. De plus, des effets de particules ont aussi été rajouter sur certaines box. Ainsi sur certaines boxes les particules ont été ajouté uniquement pour des raisons esthétiques mais sur celle qui active des capacités(comme la wallhack box), cela permet de savoir si le box peut être pris ou non. De plus, comme a dit Hugo plus haut, j'ai implémenté un petit script qui nous

permet de contrôler la distance à laquelle les différents PNJ remarque le joueur.



Particules « effet de tir »



Particules sur les box's

### 3.2.4 Théo Minary

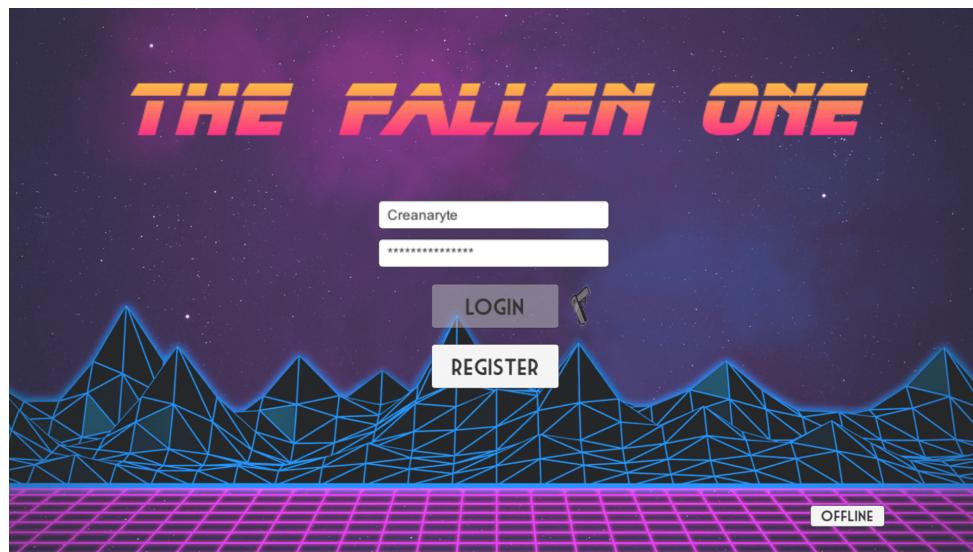
#### Menu principal

Afin d'accéder rapidement à chacunes des parties de notre jeu, par exemple aux différents niveaux solo ou au mode multijoueur, il était nécessaire de créer un menu visible et intuitif. C'était l'une des priorités lorsque l'on a commencé à coder. Afin qu'il soit parfaitement fonctionnel il a fallu remplacer le hud original intégré au script multi joueur de Unity. Il faut rajouter aussi à cela l'écran de login qui fait partie de ce menu et qui permet de se connecter afin de pouvoir récupérer les données de jeux.

Le menu est fait à base de "Canvas", c'est à dire qu'on superpose l'équivalent de plusieurs écrans mais que l'on en désactive certains. Cela permet d'obtenir différents écrans avec des boutons liés pour différentes fonctions et même avec des graphismes différents. Tout cela est géré par plusieurs script, d'abord les scripts liés au login, puis celui du menu principal qui assure les interactions entre les différents "Canvas", et si le multijoueur venait à être sélectionné alors un autre script prend le dessus afin de définir les paramètres d'ip de port etc...

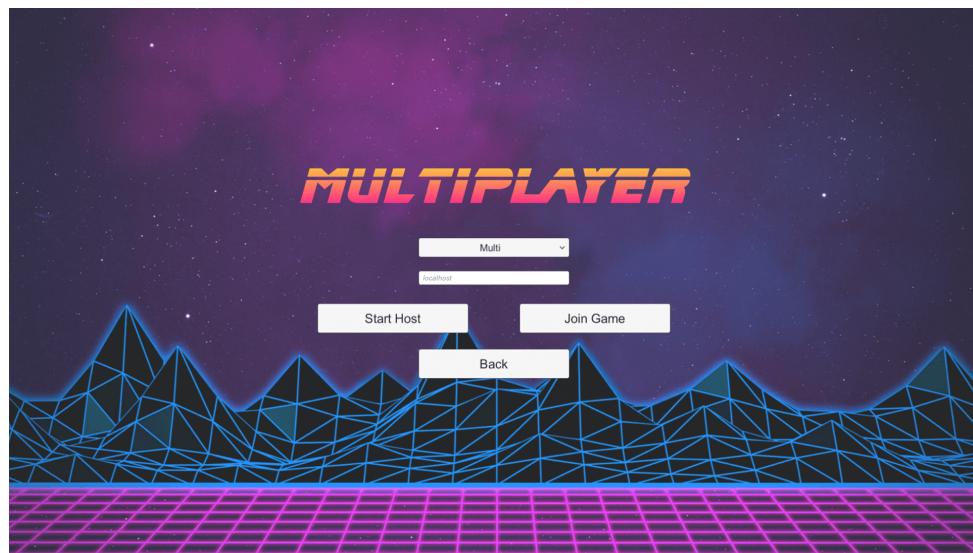
Le plus gros problème et qui est en général le problème avec toutes les interfaces est l'attribution d'actions aux boutons lors des changements de scènes. En effet lorsque l'on passait de la "scène" menu à la "scène" multi puis que l'on revenait au menu, tous les boutons n'étaient plus fonctionnel, car ils se font reset à chaque fois. Un des scripts a donc dû rétablir toutes les actions des boutons.

- **Login** : Pour le menu du Login, il a été crée en avec l'aide d'Hugo notamment pour les échanges avec le serveur web, mais il a été ajouté également un petit élément qui montre que la connexion est en cours, géré par son propre script afin d'animer cet image.



Menu login

- **Multijoueur** : Pour le menu multijoueur des éléments afin de récupérer le texte ainsi qu'un "menu" déroulant (pour faire le choix d'une carte il faut la sélectionner dans ce "menu"), en plus de remplacer l'hud de base, un script héritant du script multijoueur unity de base a été crée afin justement de faciliter les échanges avec les éléments afficher par le menu.



Menu multijoueur

## Menu Pause

Le menu pause permet de revenir au menu principal, pour le moment c'est un menu extrêmement simple et très peu d'éléments sont présents. Il est voué à

évoluer, mais pour le moment il s'agit d'un simple menu afin de revenir à l'écran principal.

Pour ce simple menu il a quand même fallu créer un script qui différencie notamment le mode de jeu, solo ou multijoueur, afin d'avoir différentes actions, par exemple en mode solo, on stoppe le temps. De plus tout comme le menu principal à chaque fois qu'un level est chargé on attribue les actions des boutons de ce menu. Par soucis de confort et pour montrer que le jeu est sur pause bien que les boutons soient visibles, un voile noir plus ou moins transparent fut ajouté.



Menu pause

### Interface ("HUD")

Il est nécessaire dans tout jeu de faire apparaître différents éléments sur son écran de jeu, cela permet notamment de se repérer. L'interface actuel est très simple puisque pour notre jeu il n'est pas nécessaire d'afficher beaucoup d'informations, on peut donc voir une barre de vie dans le coin en bas à gauche de l'écran. Mais il ne consiste pas que en cela, l'interface comprend également les animations lorsque par exemple on prend une "box".

Un unique script est au centre du fonctionnement de l'interface, il s'agit du script "Fade", il permet de réaliser un effet de fondue sur des éléments tel qu'une image ou encore du texte. Les animations de prise des "box" ainsi que l'apparition de l'indicateur de dégâts et l'écran de mort sont donc gérés par ce script. L'interface comprend aussi le "crosshair", qui permet de viser, où ici il s'agit d'une

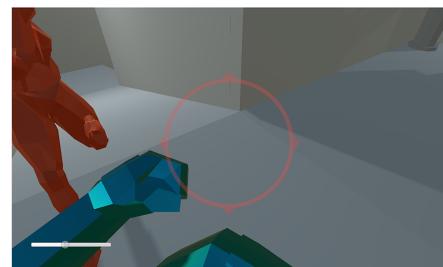
simple image que l'on afficher au centre de l'écran.

Ce script Fade se base maintenant sur ce qu'on appelle des "coroutines", et cela est très différent par rapport à la première méthode utilisé, mais cette simplification était nécessaire, et fût bien utile sur le long terme.

**Prise d'une box**



**Indicateur de dégats**



## 3.3 Pour la dernière soutenance

Ainsi chacune de nos réalisations personnelles nous ont permis une à une de nous rapprocher de notre but final : terminer notre jeu ; c'est-à-dire implémenter tout ce que nous voulions implémenter. Voici la liste des dernières réalisations qui seront faites pour la dernière soutenance et pour finaliser notre projet afin que l'expérience de jeu soit celle que nous avions imaginée dès départ.

### 3.3.1 Nouveau personnage

Un personnage important devrait voir le jour d'ici la dernière soutenance, le chef des *glitcher*. Il reprendra le modèle du joueur mais avec d'autres couleurs et aura une arme beaucoup plus imposante qu'un petit pistolet. En effet, il utilisera un fusil à pompe que Romain avait trouvé sur le site de partage de modèle 3d en même temps que le joueur puis modifié pour lui donner un aspect rétro. Nous attendions alors la bonne occasion pour utiliser cet *asset* et il se trouve qu'il colle parfaitement avec un chef agressif et imprévisible.

Ce personnage pourrait apparaître dans le dernier niveau solo, voire en multijoueur où il serait même incarnable. Il sera aussi certainement présent dans la bande-annonce du jeu.

### 3.3.2 Ajout de PNJ non hostile

Pour le moment nous n'avons ajouté que des PNJ qui, dès lors qu'ils ont repéré le joueur, foncent sur lui pour aussitôt lui tirer dessus afin de lui bloquer la progression dans le niveau. Ainsi, afin de renforcer l'immersion dans le jeu et le rendre beaucoup plus vivant nous voulons rajouter un type d'Human non hostile au joueur. Ces personnages n'auront aucune interaction directe avec le joueur et n'auront pour vocation uniquement de se déplacer librement dans le niveau afin de le rendre beaucoup plus vivant. La présence de la class Human permettra d'implémenter beaucoup plus rapidement ce type de PNJ. Il faut réaliser un algorithme qui leur permettra de se déplacer intelligemment dans le niveau et librement. En effet, il ne faudrais pas que le PNJ fonce dans le mur et continu de s'y cogner pendant le jeu, ça n'aurait aucun sens.

### 3.3.3 IA des gardes

Bien que globalement satisfait du comportement des gardes dans les niveaux actuels, nous souhaiterions améliorer le système de *path finding*. Nous allons notamment nous intéresser à la technologie NAVIGATION de *Unity*. Les possibilités pour rendre le comportement des PNJ plus naturel sont énormes, par exemple, nous pouvons définir des parcours à répéter lorsqu'ils n'ont pas encore détecté le joueur ou bien, leur faire manquer quelques tirs.

### 3.3.4 Ajout de niveaux en solo

Afin de finaliser le gameplay solo pour qu'il soit en accord avec l'histoire de notre jeu, nous allons implémenter deux nouveaux niveaux. Effectivement lors de la dernière soutenance, le joueur était prisonnier dans un complexe où l'on faisait des expériences sur lui. Je vous invite à jouer au jeu pour connaître la suite des aventures de Bluzz ! Ainsi le scénario sera fini et nous aurons un mode solo complètement fonctionnel. Evidemment les décors seront implémentés et des textures personnalisées apparaîtront !

### 3.3.5 Dernière carte multijoueur

Nous souhaitons mettre en place une dernière map multijoueur pour maximiser l'expérience de notre jeu en ligne. Actuellement deux map sont disponibles, et il nous paraît bien d'avoir le choix entre trois différentes, sachant que la première map réalisé était une sorte de prototype. De ce fait pour l'inspiration de cette dernière map, nous avons choisi la célèbre map Dôme, tout droit sortie de Call Of Duty Modern Warfare 3. Une map également célèbre pour ses combats rapprochés et dynamiques, digne héritière de Rust, colle parfaitement à l'orientation du multijoueur de The Fallen One.

### 3.3.6 Menu d'options

Un menu d'option sera créé, il permettra de modifier certains paramètres du jeu ainsi que l'interface elle-même. Par exemple on pourra afficher un compteur d'image par secondes (fps), ou encore changer la résolution à l'intérieur du jeu...

### 3.3.7 Graphisme menu et interface

Les graphismes vont changer, pour l'instant l'apparence unity de base est utilisée, mais la barre de vie ainsi que les boutons changeront. De plus un effet sera ajouté sur les boutons qui rendra le tout plus personnel et adapté au jeu.

### 3.3.8 Système de traduction des menus

Comme prévu dès le départ il y aura la possibilité de traduire les différents menus du jeu en anglais afin de faciliter la navigation pour de potentiels joueurs qui ne comprendrait pas le français. C'est une implémentation assez basique qui ne nécessite pas un algorithme très avancé. Le plus long sera sans doute l'écriture du fichier qui contiendra chaque traduction. Cette option sera disponible dans le menu décrit ci-dessus.

### 3.3.9 Sons d'ambiance

Afin de rendre le jeu plus vivant l'ajout de sons d'ambiances sera fait, afin que l'expérience de jeu soit moins terne et que l'univers soit légèrement plus immersif.

### 3.3.10 Bande-annonce

Quoi de plus attrayant qu'une belle vidéo pour présenter un jeu et potentiellement donner envie de le télécharger. Nous comptons donc faire une courte vidéo retraçant l'histoire de The Fallen One au travers des différents niveaux du jeu ainsi que montrer des scènes en multijoueur. Le montage vidéo sera donc une discipline de plus nécessaire à la réalisation du jeu. Ce qui prouve encore une fois l'étendue des compétences que nous avons dû amasser et utiliser tout au long de ce projet.

# Conclusion

Ce n'est pas sans verser une petite larme que nous concluons l'aventure The Fallen One. Ce jeu nous aura avant tout montré que le travail n'est pas forcément une corvée mais qu'il peut au contraire être source de satisfaction. En effet, voir le jeu se développer pas à pas et ressembler de plus en plus à ce que nous imaginions est une expérience très gratifiante. Cela nous a appris la valeur de la persévérance, lorsqu'il fallait passer plusieurs heures, seul devant son ordi, à résoudre un misérable *bug*, mais aussi celle du travail en équipe, car je pense qu'aucun membre du groupe n'aurait pu atteindre un tel résultat seul en si peu de temps. Nous espérons que ce projet vous semble aussi intéressant qu'il a été pour nous de le concevoir, le jeu est actuellement disponible en version bêta et bientôt en version définitive. Nous en sommes déjà satisfaits, et avons passé quelques heures à le tester et à simplement nous amuser dessus mais nous comptons encore l'améliorer jusqu'à la soutenance finale, voire même après...