

Rapport de projet Day&Night

Théo Castanie, Mathéo Lechalope, Aurèle Viale, Maxime Blinov



SOMMAIRE

Introduction et Contexte

Introduction	4
Reprise du cahier des charges	4

Théo Castanie

Concept et Scénario	8
Site web	9
Équilibrage	14
Monstres et Animations	15

Aurèle Viale

Scénario et Classes	19
Map	24
Musiques	26
Jaquette	27

Mathéo Lechalupé

Mouvements	29
Modèles et Particules	30
UI	37
Débogage	40

Maxime Blinov

Blender	42
Ressentis	48

Conclusion

Conclusion générale	51
---------------------	----

Introduction :

Voici le rapport de projet retracant tout le parcours et le travail donné pour le jeu vidéo que nous avons créé. Pour rappel, Day&Night est un FPS type horreur dans un thème mêlant science-fiction et espace. Le jeu inclut une partie PVP et une partie PVE où le joueur doit survivre face à tous ces ennemis. Ce jeu avait pour vocation de réunir toutes sortes de communautés à travers différents gameplays et expériences de jeu.

Au fil de l'année, nous avons pu apprendre et vivre toutes sortes de péripéties, que ce soit dans la découverte des logiciels utilisés ou bien dans les différentes erreurs que l'on a pu commettre au niveau technique. Cela nous a permis de développer différentes compétences dans différents panels de construction d'un élément numérique.

Dans la suite de ce rapport de projet, vous pourrez découvrir les différents aspects graphiques (que ce soit pour le jeu, pour le site web,...

), les différentes étapes de conception où sont passés les différents membres du groupe et surtout rentrer dans nos cheminement de pensée, nos hauts et nos bas qui ont mené au produit final. Vous verrez en premier une reprise du cahier des charges, puis une explication dans l'ordre chronologique de chaque membre du projet.

Reprise du cahier des charges :

Voici la planification que l'on s'était donné lors de la rédaction du cahier des charges :

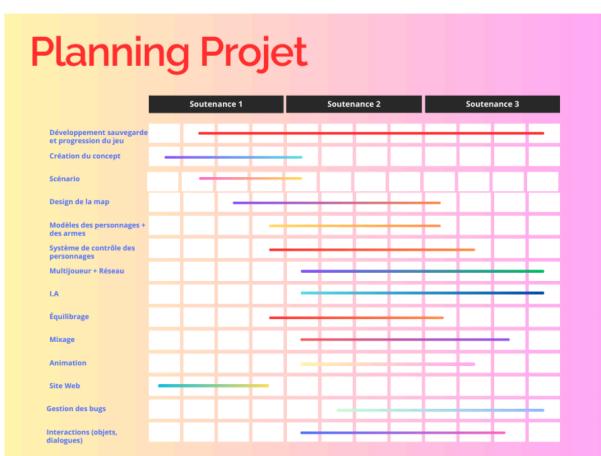
TABLE 2 – Planification du projet

Tâches	1ère soutenance	2e soutenance	3e soutenance
Création du concept	100%	100%	100%
Scénario	100%	100%	100%
Design de la map	50%	90%	100%
Modèles des personnages	25%	100%	100%
Modèles des armes	25%	100%	100%
Multijoueur	20%	90%	100%
I.A	25%	80%	100%
Équilibrage	10%	90%	100%
Mixage	25%	100%	100%
Animation	10%	100%	100%
Site Web	100%	100%	100%
DLC	0%	50%	100%
Jaquette	0%	50%	100%

Et voici la répartition des rôles, après le départ du 5e membre du groupe

Tâches	Répartition des tâches : deux personnes par tâche: (R)esponsable & (S)uppléant				
	Login1	Login2	Login3	Login4	Login5
Création du concept	R		S		
Scénario	S		R		
Design de la map		S			R
Modèles		S			R
Multijoueur		R	S		
I.A		S	R		
Équilibrage	R			S	
Mixage/ Son	S		R		
Animation		R		S	
Système de contrôle du personnage	S	R			
Réseau	S	R			
Gestion bugs	R			S	
Dev sauvegarde et progression du jeu		S	R		
Interaction (objet, dialogues)			S	R	

Légende
Resp
Suppl



Il est logique que certains éléments aient changé ou que d'autres aient été ajoutés/supprimés en cours de route. Toutefois, malgré les imprévus, nous avons su maintenir le cap et rester globalement fidèles aux délais que nous nous étions fixés en début d'année.

- **Le concept et le scénario** ont été les premiers éléments abordés par l'équipe. Nous avons rapidement convergé vers une idée de jeu solide, tout en laissant la porte ouverte à des ajustements. Même si des modifications ont été apportées au fil de l'année (notamment en lien avec la faisabilité technique et le temps disponible), cela ne nous a pas empêchés d'avancer efficacement vers les phases de développement.
- **La création de la map** a connu une progression constante, bien que non linéaire. Elle a demandé un gros investissement en termes de modélisation et d'optimisation, et chaque itération a permis d'améliorer la fluidité du gameplay et la cohérence de l'univers.
- **Les modèles 3D des personnages et des armes** ont été réalisés en parallèle, avec un soin particulier apporté à leur intégration dans Unity. Dès la première soutenance technique, des prototypes jouables étaient prêts, ce qui a permis de valider rapidement les directions artistiques et mécaniques prises.
- **Le multijoueur**, cœur technique du projet, a pu être mis en place dès les premières semaines de développement. Grâce à l'utilisation de Mirror, nous avons pu tester très tôt les interactions entre joueurs, corriger des bugs réseau et affiner les mécaniques de jeu en ligne.
- **L'IA et les animations** ont avancé en même temps. Bien qu'ayant rencontré des retards ponctuels dus à la complexité d'intégration et à des conflits de versions Unity, elles sont

aujourd’hui pleinement opérationnelles. Les ennemis possèdent des comportements cohérents et des transitions animées fluides, qui renforcent l’immersion.

- **L'équilibrage du jeu** a été plus court que prévu, principalement par manque de temps et à cause des priorités données à d'autres fonctionnalités. Néanmoins, les quelques ajustements réalisés ont eu un impact significatif sur le ressenti des joueurs et la dynamique des parties, rendant l'expérience plus agréable et moins frustrante.
- **Le mixage** a été l'un des volets les plus réactifs : chaque soutenance a été accompagnée de nouvelles pistes musicales, créées pour renforcer les ambiances jour/nuit. Cela a permis de poser une identité sonore forte, cohérente avec l'univers visuel et narratif.
- **Le site web** a été un outil essentiel de communication tout au long du projet. Mis en ligne très tôt, il a évolué avec le jeu, proposant de plus en plus de contenu et une présentation de plus en plus soignée. Il a servi de vitrine du projet, mais aussi de plateforme de téléchargement et de documentation, renforçant l'aspect professionnel de notre démarche.
- **Le DLC**, bien qu'envisagé dans un premier temps comme un ajout facultatif pour enrichir l'expérience, a finalement été abandonné. Nous avons préféré concentrer nos ressources sur la finition du jeu principal afin de garantir une base solide et cohérente plutôt que de s'éparpiller sur des contenus annexes.
- **La jaquette** du jeu, créée en toute fin de projet, a bénéficié d'un soin particulier. Elle constitue l'identité visuelle ultime du projet, celle qui attire l'œil et résume en un visuel les ambitions du jeu. Nous avons veillé à la rendre à la fois

esthétique, évocatrice et fidèle à l'univers de *Day&Night*.

Dans l'ensemble, chaque membre du groupe a su faire preuve de polyvalence et d'adaptabilité, en prenant en charge des tâches dépassant parfois leur rôle initial. Le départ du cinquième membre nous a contraints à revoir notre organisation, mais il nous a aussi soudés et forcés à trouver des solutions collectives. Cette répartition dynamique des responsabilités est sans doute l'un des facteurs-clés de la réussite de notre projet.

Théo :

Mes premières tâches attribuées ont été de créer le concept, le scénario (surtout dans la possibilité de créer un mode histoire), la gestion des bugs et l'équilibrage. Nous avons aussi voté pour élire un chef de projet et c'est moi qui ait été choisi, bien que nous ayons fourni des efforts pour que chaque membre puisse avoir ses propositions acceptées, cela s'est vu particulièrement dans le concept du jeu où chacun a apporté une contribution. J'ai tenu aussi à ce que chacun ait une certaine autonomie dans la façon de faire son travail, je m'occupais surtout d'imposer des deadlines, de corrélérer les parties de travail de chacun et des relations entre les différents membres et c'est une très bonne chose qu'elles aient été très agréables du début à la fin. Et je pense que les tâches que l'on m'a attribué sont un moyen d'avoir une supervision de tout ce qu'il se passe.

Avec Aurèle, on a commencé à discuter de comment nos idées pour le jeu et le scénario allaient concrètement être implémentés dans le jeu. Dans l'incertitude de si on allait pouvoir tout finir à temps, on s'est mis d'accord sur un principe de base : un FPS dans un évènement horrifique avec des modèles cartoons. Pour ce qui est du mode histoire, l'idée d'avoir différentes ressources à trouver

dans la map avec des monstres nous pourchassant a été évoquée mais n'a pas été considérée comme étant une priorité. Aurèle étant responsable de cette partie, il sera plus qualifié pour en parler.

Dans l'optique d'avoir une partie de développement, j'ai été chargé de gérer le site web (j'ai évidemment fait d'autres parties que je "développerais" plus tard). Mon premier avis sur faire un site web était que ça allait être quelque chose de rapide et facile à faire. Je suis allé du côté de Odoo, qui permet de créer des sites web de façon simple et avec très peu de code, avec des visuels très attractifs.

Un premier modèle était établi :



Mais directement plusieurs problèmes sont apparus, utiliser était un bon moyen d'avoir une base facilement configurable mais seul les premières pages étaient gratuites, pour avoir plus de fonctionnalités il fallait payer. De plus, cela ne correspondait absolument pas aux standards de programmation demandés. Il a fallu chercher un autre moyen. Et je suis tombé sur HTML5UP qui permet d'avoir de base une page principale et qui permet après de créer et relier d'autres pages tout en ayant de quoi coder. Le template permet aussi de faciliter l'agencement responsive (c'est-à-dire que depuis n'importe lequel des appareils, tout sera bien agencé)

Il a logiquement été décidé d'utiliser HTML, car il est connu et facilement implantable sur un hébergeur. CSS est utilisé pour transcrire le code sur un site. D'autres langages ayant un impact mineur ont été utilisés comme SCSS ou JavaScript, servant principalement à la configuration et ne servant pas au développement du site en lui-même.



Images ne représentant pas le produit fini mais pour montrer le responsive

Maintenant il fallait tout construire, car à part des Lorem Ipsum il n'y avait pas grand chose. De plus, le nombre de fichiers était assez faible.

	index		12/01/2025 14:39	Opera Web Document	11 Ko
	left-sidebar		11/01/2025 17:48	Opera Web Document	15 Ko
	README		11/01/2025 17:48	Document texte	2 Ko
	right-sidebar		11/01/2025 17:48	Opera Web Document	15 Ko
	LICENSE		11/01/2025 17:48	Document texte	17 Ko
	no-sidebar		11/01/2025 17:48	Opera Web Document	10 Ko
	assets		12/01/2025 14:45	Dossier de fichiers	
	images		12/01/2025 14:45	Dossier de fichiers	

Il fallait en premier imaginer quelles étaient les catégories essentielles et celles demandées. La page d'accueil (qui est accessoirement la plus importante car celle que le public voit en premier et qui les attire ou non). On a donc mis l'image qui devait de base être le logo, donnant un aspect visuel intéressant. Puis une légère introduction avec le logo, la chronologie synthétique, les membres du projet et les (faux) contacts.

```

<head>
    <title>Day&Night By Pixel&Nerve</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <link rel="stylesheet" href="assets/css/main.css" />
</head>
<body class="homepage is-preload">
    <div id="page-wrapper">

        <!-- Header -->
        <section id="header" class="wrapper" style="background-image: url('images/Design_jeu.jpg'); background-size: cover; background-position: center;">

            <!-- Logo -->
            <div id="logo" style="text-align: center; top: 3em;">
                <p style="color: #4aa606; font-size: 1.1em; margin-top: 6em;">
                    Un jeu pour tout type de communauté
                </p>
            </div>

            <!-- Nav -->
            <nav id="nav">
                <ul>

```

La partie la plus dure à ce niveau fut de bien centrer les images et que les couleurs soient bien adaptés, pour qu'elles rentrent plus dans l'ambiance du jeu. Le fichier permettant de changer ça fait des dizaines de milliers de lignes, c'était très dur de situer quelle partie changeait la couleur de quoi. Il y a ensuite eu la catégorie téléchargements où on peut retrouver le projet dans sa forme complète et allégée, les différents rapports de soutenances ainsi que le dossier de justification utile en méthodologie. C'était une partie spéciale car il n'y avait pas de page fixe comme les autres catégories mais en soit c'était beaucoup de répétition et de copie-coller de code car c'était juste le contenu du téléchargement qui changeait.

```

                </ul>
            </nav>
        </section>

        <!-- Main Content -->
        <section id="main" class="wrapper style2">
            <div class="title">Soutenance 1</div>
            <div class="container">
                <div class="row gtr-150">
                    <div class="col-12">
                        <!-- Contenu de la page Soutenance 1 -->
                        <h2>Télécharger le document de la soutenance 1</h2>
                        <p>Voici le fichier Word de la soutenance 1 que vous pouvez télécharger :</p>
                        <a href="Rapport-de-soutenance.docx" class="button">Télécharger la soutenance 1</a>
                    </div>
                </div>
            </div>
        </section>

    </div>

```

Ensuite, il y a la partie boutique permettant d'acheter (virtuellement) des objets spéciaux pour ensuite passer au paiement

Nom sur la carte:
jaipasmisdereturndansuneboucle

Numéro de carte:
1984 001 1999 2012

Date d'expiration:
03/26

CVV:
092

PAYER

Paiement réussi !

Le “paiement” est réussi que si chacune des zones de remplissage ont le contenu nécessaire.

La partie “Ressources” regroupe tous les supports que l’on a utilisés pour mener à bien le projet. Ici, pas de difficultés particulières à part quelques problèmes de syntaxe qui se supprimaient uniquement avec du code et non de la rédaction comme les points puces.

- Outil de modification du site : Note
- Langage utilisé : HTML
- Chaîne youtube : FormationVidéo
-

Maxime

- GitHub Repository
- Fichiers de configuration
- Blender
- Outil de partage : Google drive
-

Et enfin, la partie “À propos” qui regroupe un petit résumé du principe du jeu, ainsi que des potentiels crédits.

La principale difficulté était les catégories en haut où, en fonction de la page, les noms et le nombre de catégories n'étaient pas les mêmes



Et les pages n'étaient pas reliées en même. C'est là où j'ai passé le plus de temps : à essayer de bien tout intégrer. Et faire en sorte que cette partie soit "transparente" fut aussi compliqué à implémenter. Pour l'hébergeur, on a trouvé une solution simple et qui allait très bien avec les outils que l'on utilise : l'héberger sur GitHub. Grâce à Github Pages, il suffisait de rentrer les dossiers dans un

Par la suite, le départ d'Ayme, qui avait les plus grandes compétences de codage parmi nous, nous a forcé à revoir notre organisation et par conséquent à répartir les rôles laissés à l'abandon. Cela nous a aussi forcé à organiser plus souvent des réunions et à moi de prendre le temps de m'impliquer dans le travail de chacun. On en ressort plus fort mais c'était assez compliqué à ce moment-là. Notre ACDC aura aussi fait 2-3 réunions avec nous pour nous permettre de nous recentrer sur les priorités à traiter pour finir dans les temps le projet.

Au cours du début du 2^e semestre, je me suis aussi intéressé à l'équilibrage du jeu. Cette partie, souvent sous-estimée, est en réalité un très sérieuse, car elle demande de penser le gameplay comme un écosystème où chaque élément (armes, ennemis, capacités, rythme,...) est en interaction constante avec les autres. Il

ne s'agit pas simplement d'ajuster des chiffres, mais de construire une cohérence d'ensemble qui garantit une expérience fluide, juste et agréable pour tous les profils de joueurs. Un bon équilibrage repose sur une vision globale du jeu. Chaque mécanique, chaque système doit être évalué non seulement en lui-même, mais aussi en fonction de son impact sur le reste. Par exemple, si une arme est trop puissante, elle peut rendre caduques toutes les autres stratégies. Si un ennemi est trop faible ou trop présent, il peut déséquilibrer la tension du jeu. C'est un travail d'ajustement permanent, basé autant sur des tests que sur de l'intuition et du bon sens de joueur.

Pour notre jeu, les catégories à équilibrer étaient nombreuses : les dégâts des armes, la résistance des zombies, la durée des cycles jour/nuit, la vitesse de déplacement, la consommation du jetpack, la fréquence d'apparition des objets ou encore la répartition spatiale des ressources sur la carte. Toutes ces dimensions influencent directement la dynamique de jeu, les sensations de difficulté, et même le plaisir de jouer.

■ Dégâts/Tirs

:

■ Récompenses période nuit

■ Boutique post-nuit

■ Ressources nécessaires fu...

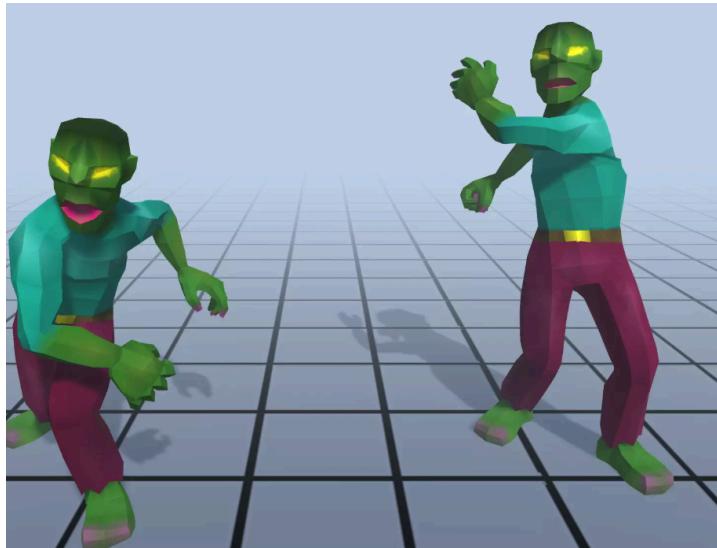
■ Temps période jour/nuit

Je dis bien “étaient” car beaucoup de ces catégories n'ont pas pu être traitées en profondeur, tout simplement parce que l'équilibrage vient en fin de chaîne : il ne peut être abordé efficacement que lorsque la majorité des éléments de gameplay sont déjà implémentés et fonctionnels. Or, dans notre cas, certains systèmes étaient encore en phase de finition ou en cours d'intégration, ce qui

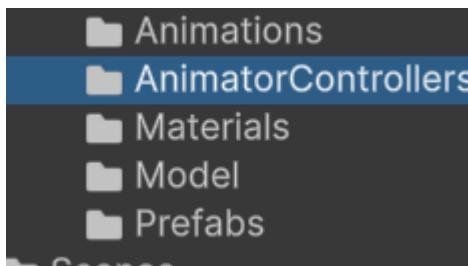
a limité notre marge de manœuvre. De plus, le gameplay volontairement simplifié que nous avions choisi, avec peu d'armes, des ennemis basiques, et des cycles relativement courts, a réduit le champ des variables à ajuster. Cela a rendu la phase de réflexion plus rapide que prévu, notamment sur des aspects comme les dégâts ou la durée des périodes. Cela dit, même avec ces contraintes, chaque ajustement a eu un impact significatif sur la jouabilité, et il a fallu faire preuve de rigueur pour que l'expérience reste équilibrée, notamment dans un contexte multijoueur où les écarts de puissance ou de rythme peuvent vite nuire à l'intérêt du jeu.

En conclusion à cette partie, cette tâche m'a permis de comprendre que l'équilibrage n'est pas une phase secondaire, mais un élément structurant du développement, à la croisée de la technique, du design et de l'expérience utilisateur. Même si le temps nous a manqué pour la pousser plus loin, elle m'a offert une nouvelle perspective sur le rôle fondamental de la cohérence dans la conception d'un jeu.

La dernière grande tâche auquel je me suis attardé était la création des zombies, donc des ennemis qui allaient attaquer le joueur. C'était donc pour moi une bonne occasion de me familiariser avec toutes les fonctionnalités de Unity. J'ai pu obtenir le modèle et les animations via l'Unity asset store

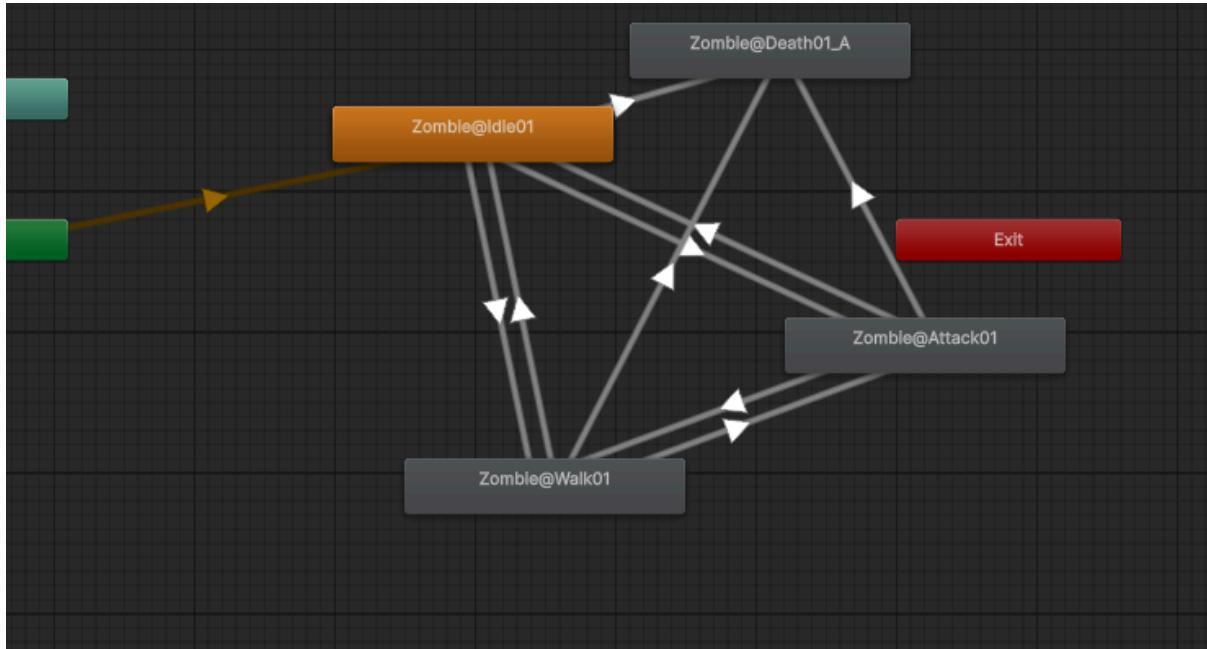


Ce qui m'a permis d'avoir de façon précise et organisée une base pour commencer à configurer le zombie



Bien que beaucoup d'éléments vitaux étaient présents dans les fichiers téléchargés, il a fallu créer de nouvelles instances tels que de nouveaux animators, des Nav Mesh Agent et d'autres pour qu'ils soient compatibles et plus facilement modulables avec le reste du projet.

Ce qu'il faut créer maintenant, ce sont ses conditions d'apparition, les interactions avec les différentes animations et avec le joueur. J'ai d'abord regroupé les différentes animations en un seul endroit pour que le zombie passe plus facilement aux différentes animations, pour que la transition soit plus fluide.



Une erreur qui m'aura fait perdre beaucoup de temps est à propos des lignes avec les flèches entre les animations car je ne savais pas qu'il fallait faire une condition de l'animation A à l'animation B puis de l'animation B à l'animation A (sauf pour l'animation de mort évidemment car il n'y a pas de résurrection).

Il a fallu ensuite créer l'IA. Pour cela, j'ai pu m'aider d'une vidéo qui créait les différentes méthodes nécessaires au bon fonctionnement du zombie, même si c'était à moi de l'adapter et de remplir le contenu de ces fonctions (on peut citer celles du type d'attaque et celle quand le zombie poursuivra un joueur).

```
private void ChasePlayer()
{
    agent.SetDestination(player.position);
    animator.SetBool("isWalking", true); //Active l'animation de marche
}
```

```
private void AttackPlayer()
{
    // Empêche l'ennemi de bouger pendant l'attaque
    agent.SetDestination(transform.position);

    transform.LookAt(player);

    // Ajout : arrête l'animation de marche
    animator.SetBool("isWalking", false);

    // Ajout : déclenche l'animation d'attaque
    animator.SetTrigger("Attack");

    if (!alreadyAttacked)
    {
        // Code de tir
        Rigidbody rb = Instantiate(projectile, transform.position, 0);
        rb.AddForce(transform.forward * 32f, ForceMode.Impulse);
        rb.AddForce(transform.up * 8f, ForceMode.Impulse);
    }
}
```

J'ai pu partager mon travail grâce aux repos GitHub que l'on a donné avec les autres membres du groupe.

Enfin, en tant que chef de projet, je me suis occupé de la gestion de la partie présentation et rendu concernant les rapports ou autres documents. Bien évidemment, chacun a contribué, je ne faisais que corriger et mettre en valeur les parties pour que chacune soit en accord avec les autres pour faire un mélange homogène et logique. Ce projet m'a permis d'améliorer mes compétences pour diriger un projet, entretenir les relations et savoir encaisser la pression qu'il peut y avoir pour être à tout moment quelqu'un à l'écoute de ses membres et respectable.

Aurèle :

L'univers du jeu repose sur un contexte de catastrophe spatiale, dans lequel un groupe de colons s'écrase sur une planète hostile et instable. Les infrastructures humaines — usines, bases techniques, avant-postes — sont à l'abandon, et seules quelques fusées de

secours, incomplètes, permettent d'envisager un retour vers l'espace. Chaque joueur incarne un survivant, dont l'objectif est de rassembler les pièces manquantes pour s'échapper... avant les autres.

L'ambition de cette narration minimalisté était de poser un cadre crédible, immédiatement compréhensible, tout en laissant suffisamment de liberté aux joueurs pour inventer leurs propres dynamiques et stratégies. Cette liberté narrative est renforcée par l'absence de dialogues ou de cinématiques longues : l'histoire est racontée par l'environnement, les mécaniques, et le comportement des autres joueurs.

L'élément central de notre gameplay est l'alternance **jour/nuit**, qui impose une rupture nette dans les mécaniques et les stratégies :

- **Le jour** : période active de récolte, d'exploration et d'affrontements entre joueurs. C'est le moment stratégique où les décisions comptent le plus : où aller ? Qui éviter ? Quelle ressource prioriser ?
- **La nuit** : apparition de créatures hostiles, obligeant les joueurs à coopérer temporairement. Le gameplay devient plus défensif, plus tendu, et le moindre bruit peut signifier la mort.

Cette alternance aurait pu créer une dynamique cyclique de tension et de soulagement, et pousser le joueur à s'adapter constamment aux conditions extérieures. Elle favorise aussi des choix moraux ou

tactiques complexes : s'allier avec un rival, l'abandonner à son sort, ou le trahir au dernier moment.

Pour assurer la cohérence de l'univers, plusieurs éléments narratifs et de gameplay ont été soigneusement justifiés :

- Les pièces détachées sont les restes de l'ingénierie humaine abandonnée après le crash.
- Les monstres nocturnes sont issus de la faune mutée ou des effets d'expériences biologiques passées.
- Les zones industrielles, les dépôts, ou les laboratoires désaffectés constituent l'environnement principal du jeu, à la fois narratif et fonctionnel.

Le choix d'un style cartoon, malgré l'ambiance sombre et oppressante du jeu, vise à renforcer le contraste entre les mécaniques violentes et une apparence visuelle plus accessible, ce qui permet de ne pas sombrer dans un réalisme trop pesant.

Pour enrichir encore davantage la profondeur stratégique, j'ai conçu un système de classes jouables. Chaque joueur doit choisir un type de survivant parmi cinq profils distincts, chacun avec ses propres statistiques, compétences et bonus uniques. Cela permet de diversifier les styles de jeu et d'encourager des synergies — ou des oppositions — selon les situations.

- Le Combattant
- Force : 3/5

- Intelligence : 3/5
- Points de vie : 3/5
- Vitesse : 3/5
- Particularité : peut choisir un bonus de spécialisation parmi trois au choix :
 - *Ingénieur* (bonus de construction)
 - *Tireur* (bonus de visée/dégâts)
 - *Protecteur* (bonus de défense)

Classe équilibrée, idéale pour les débutants ou ceux qui veulent s'adapter à toute situation.

- L'Ingénieur
- Force : 2/5
- Intelligence : 5/5
- Points de vie : 3/5
- Vitesse : 2/5
- Bonus : capacité à transporter plus d'objets que les autres classes.

Il excelle dans la collecte de ressources et la réparation des fusées, mais souffre en combat direct.

- Le Tireur

- Force : 6/6 (*spécialisation hors-échelle pour marquer la supériorité à distance*)
- Intelligence : 3/5
- Points de vie : 1/5
- Vitesse : 2/5
- Bonus :
 - portée de tir augmentée
 - meilleure précision
 - capacité de zoom accrue lors de la visée
 - dégâts accrus à distance

Classe puissante mais très fragile, parfaite pour les joueurs patients et stratégiques.

- Le Protecteur
- Force : 3/5
- Intelligence : 2/5
- Points de vie : 5/5
- Vitesse : 2/5
- Bonus :
 - régénération passive des PV

- bouclier personnel ou transférable

Le tank du groupe : résistant, il excelle dans la défense et le soutien.

- L'Assassin
- Force : 5/5
- Intelligence : 1/5
- Points de vie : 1/5
- Vitesse : 5/5
- Bonus :
 - exécution instantanée si l'ennemi a <15% de ses PV
 - camouflage temporaire : invisibilité partielle + accélération de déplacement

Rapide et létal, c'est la classe la plus agressive mais aussi la plus vulnérable.

Ce système de classes a nécessité un travail de design asymétrique rigoureux, en veillant à ce qu'aucune classe ne soit objectivement supérieure. Chaque profil présente des forces et des faiblesses, créant une dynamique d'interaction où les joueurs doivent adapter leurs comportements non seulement à l'environnement, mais aussi à la composition des classes en présence. Cela contribue fortement à la rejouabilité et à la richesse

stratégique du jeu. Cependant, nous n'avons pas pu implémenter le système de classe à temps car trop compliqué par rapport au peu de temps que nous avions et comme dit précédemment, l'adaptation et l'équilibrage des classes avec le jeu demandait de nombreux tests compliqués à prendre en main avec le temps qu'il nous restait.

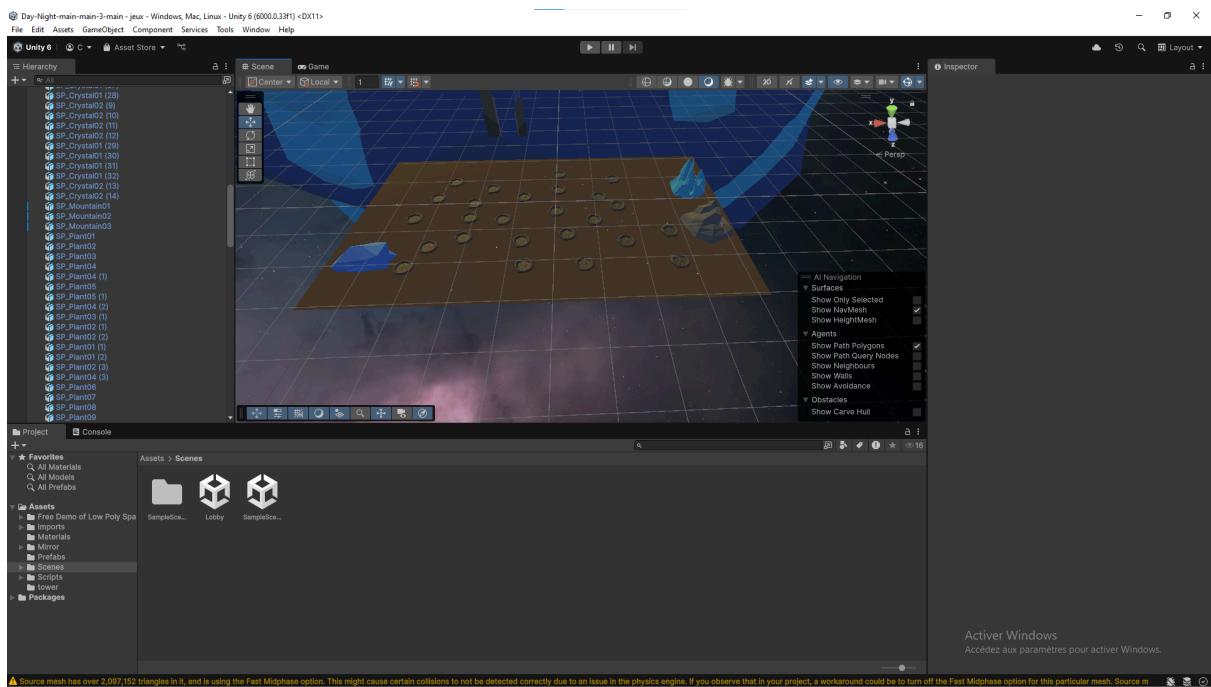
La carte offre un environnement vaste et varié, propice à l'exploration, aux affrontements et à la survie. Elle devait également refléter l'esthétique du jeu et soutenir les mécaniques de gameplay.

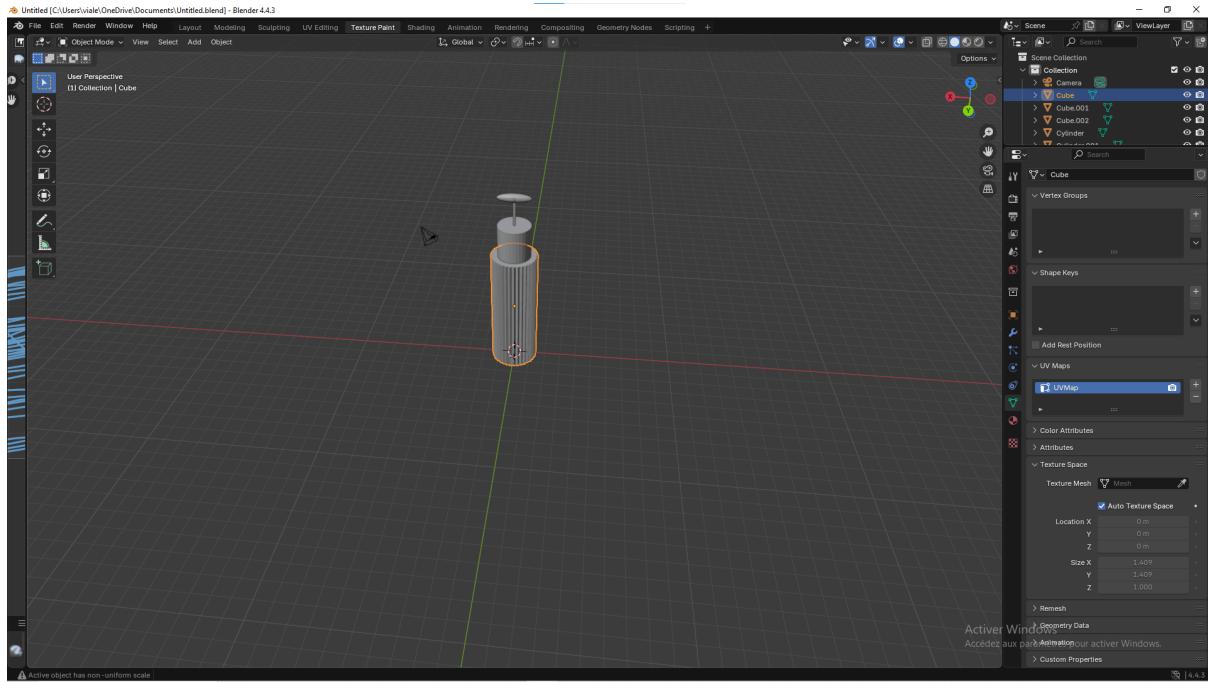
Nous avons utilisé Blender, un logiciel de modélisation 3D, pour créer la carte. Ce choix a nécessité une courbe d'apprentissage importante, l'auteur n'ayant aucune expérience préalable en modélisation 3D. Le processus a inclus :

- Modélisation : création de terrains, obstacles et objets interactifs.
- Texturisation : application de textures pour renforcer l'ambiance visuelle.
- Éclairage : mise en place de sources lumineuses pour simuler les cycles jour/nuit.

Plusieurs défis ont été rencontrés lors de la création de la carte :

- Optimisation : assurer des performances fluides tout en maintenant une qualité visuelle élevée.
- Cohérence esthétique : maintenir une direction artistique uniforme malgré la diversité des environnements.
- Adaptation aux mécaniques de gameplay : concevoir des zones favorisant la stratégie et l'interaction entre joueurs.





L'objectif était de créer deux ambiances musicales distinctes, correspondant aux phases jour et nuit :

- Jour : musique inspirée des films d'action, énergique et dynamique.
- Nuit : musique inspirée des films d'horreur, sombre et angoissante.

J'ai utilisé FL Studio, un logiciel de production musicale, pour composer les morceaux. Les étapes comprenaient :

- Création de boucles rythmiques : pour établir la base de chaque morceau.
- Arrangement : ajout d'instruments et de textures pour enrichir la composition.
- Mixage : ajustement des niveaux et de la spatialisation sonore.

Les compositions ont été bien accueillies, contribuant à l'atmosphère du jeu. Les transitions entre les ambiances jour et nuit sont fluides, renforçant l'immersion du joueur.

La jaquette devait refléter l'esthétique du jeu et attirer l'attention du joueur potentiel. Elle devait également être cohérente avec le style visuel du jeu.

J'ai utilisé Canva, une plateforme de design graphique en ligne, pour concevoir la jaquette. Le processus a inclus :

- Choix des éléments visuels : sélection d'images et d'illustrations représentant l'univers du jeu.
- Composition : agencement des éléments pour créer une image cohérente et attrayante.

- Typographie : sélection de polices et disposition du texte pour assurer la lisibilité.

La jaquette finale présente un squelette de dos, debout dans une usine abandonnée, faisant face à un ciel orange avec une lune orange. Ce visuel synthétise l'ambiance du jeu et suscite l'intérêt du spectateur.

En conclusion, ce projet m'a permis d'explorer un large éventail de compétences : écriture narrative, modélisation 3D, composition musicale, et graphisme. Chacune de ces dimensions m'a confronté à des défis différents, mais toutes m'ont permis d'apprendre énormément. J'ai découvert ce qu'implique la création d'un jeu multijoueur original, équilibré et immersif, et j'en ressors avec une meilleure compréhension des étapes de production, mais aussi de mes propres forces et axes de progression.

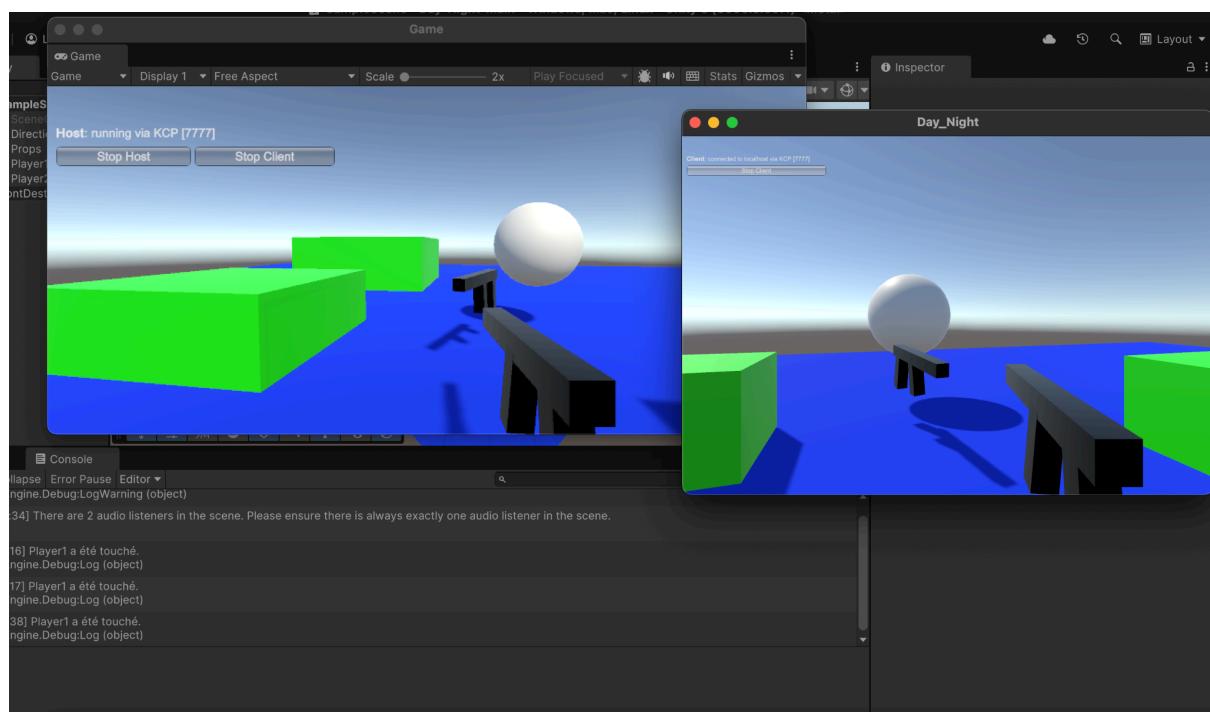
Mathéo :

Détaillons étapes par étapes : au début de ce projet, j'ai découvert Unity.

La prise en main à été assez difficile avec pas mal d'erreurs mais cela s'est amélioré au fil du temps. Je ne voudrais pas dire que je maîtrise Unity, ce qui n'est pas le cas, mais je peux dire sans difficultés que je sais me débrouiller avec.

Tout d'abord, j'ai décidé de lister les différentes tâches à réaliser pour avoir une base sur quoi construire notre jeu.

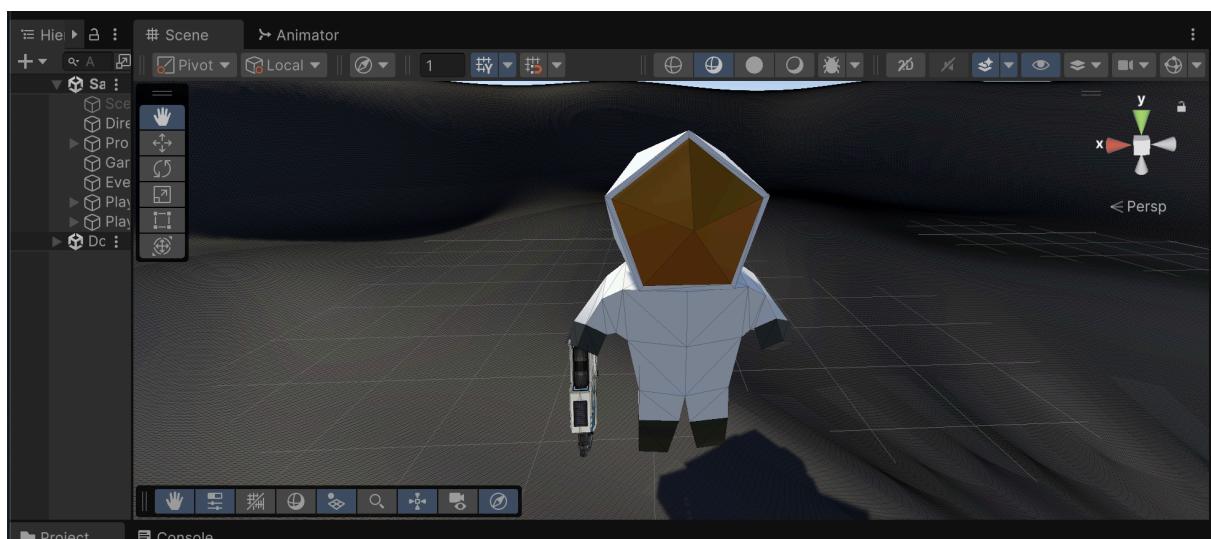
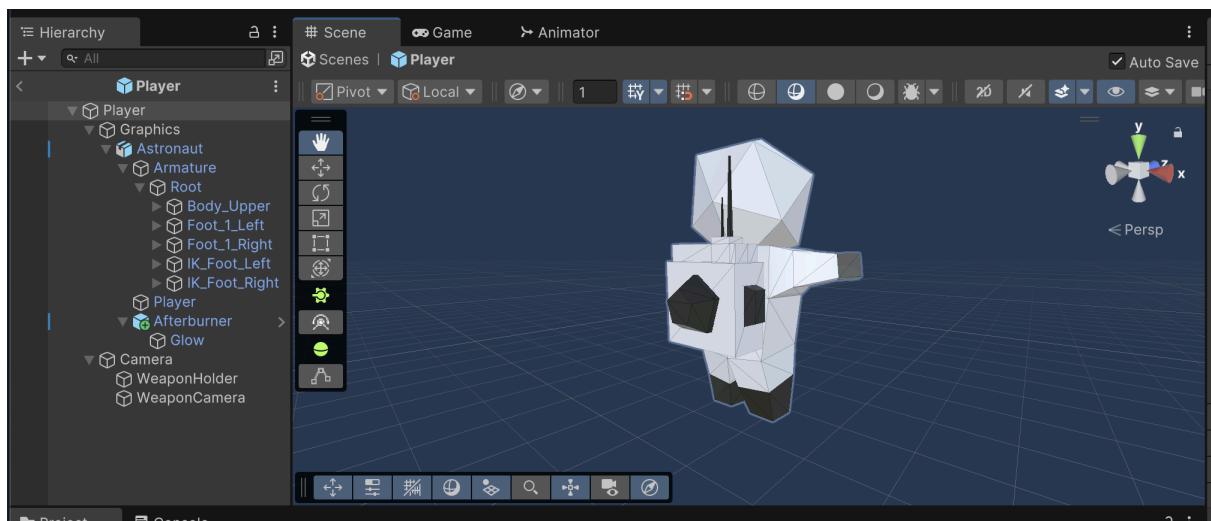
J'ai réalisé en premier lieu les mécaniques de mouvement avec seulement les objets de base présent dans Unity.



J'ai aussi créé une map pour tester toutes les mécaniques relatives aux joueurs, ainsi que la relation joueur-arme avec des objets de base comme des rectangles et une sphère.

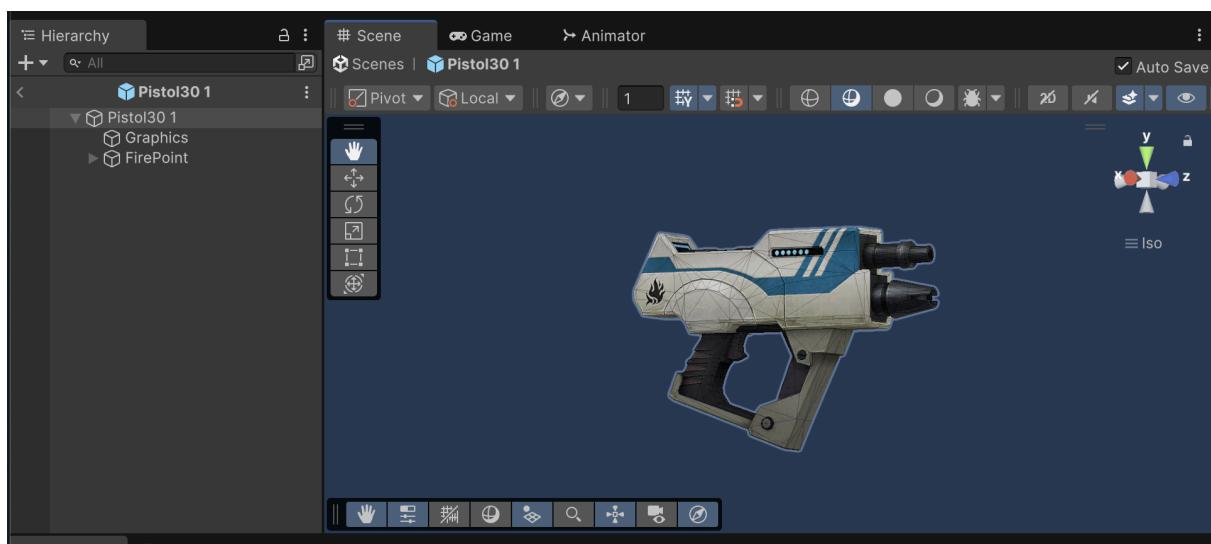
Ensuite j'ai passé plusieurs heures à créer les mouvements de base pour le joueur. Les premiers mouvements ont bien sûr été le contrôle du joueur (avancer, reculer, etc). Ensuite il y a eu le système de tir, qui était très rudimentaire au début. Les premiers échanges de tir ne pouvaient être vu que grâce à la console d'Unity. Sans balle ou effet, cela ressemblait à une bataille de regards entre les deux joueurs. Bien sûr, tous les mouvements ont été réalisés en utilisant Mirror, ce qui permettait d'implémenter directement le multijoueur dans notre projet.

Après avoir réalisé les fondements du jeu, il a fallu donner une forme à notre personnage. Après plusieurs discussions, nous avons décidé d'implémenter un modèle d'astronautes pour correspondre à nos idées initiales.

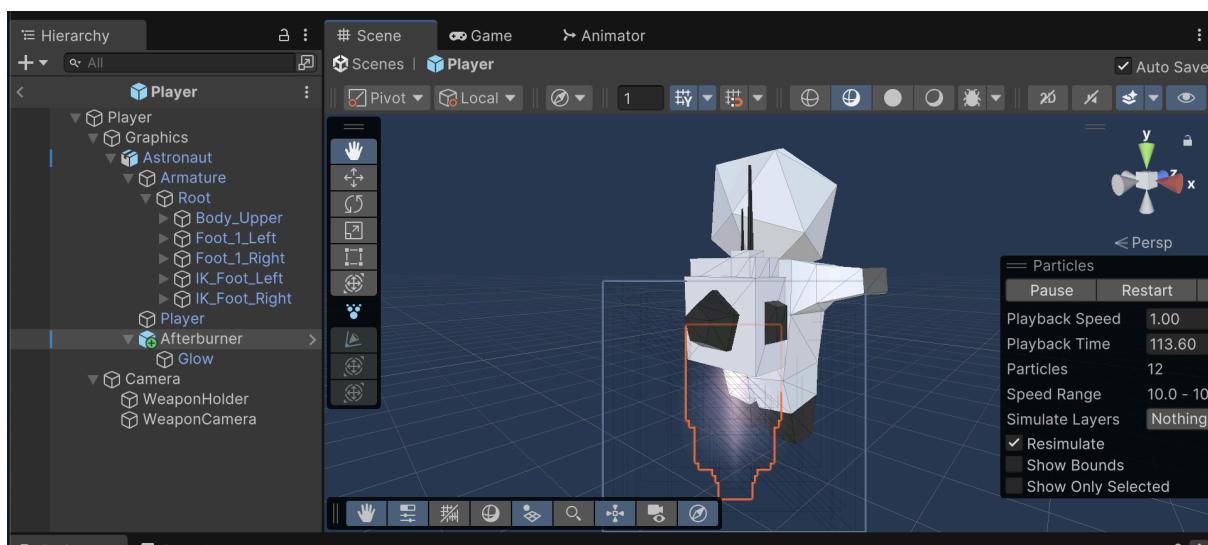
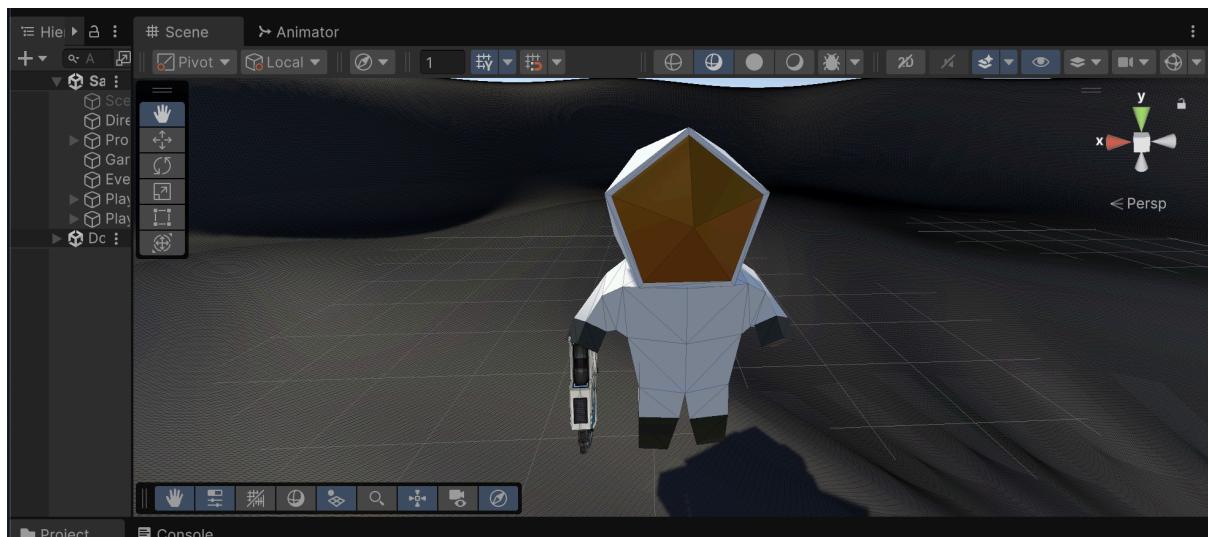


Ce modèle m'a aussi permis de rajouter une mécanique, qui est maintenant très importante dans notre jeu : le jetpack. Situé à l'arrière du modèle, j'ai pensé que cela ajouterait une mécanique plus intéressante que le saut tout simple. Le "thruster" comme je l'ai nommé dans le jeu permet à nos joueurs de se battre dans les airs, ce qui ajoute un autre gameplay que celui prévu de base.

Après avoir reçu les différents modèles de la part de Maxime, j'ai pu implémenter les armes et commencé à réaliser des animations sur notre personnage.



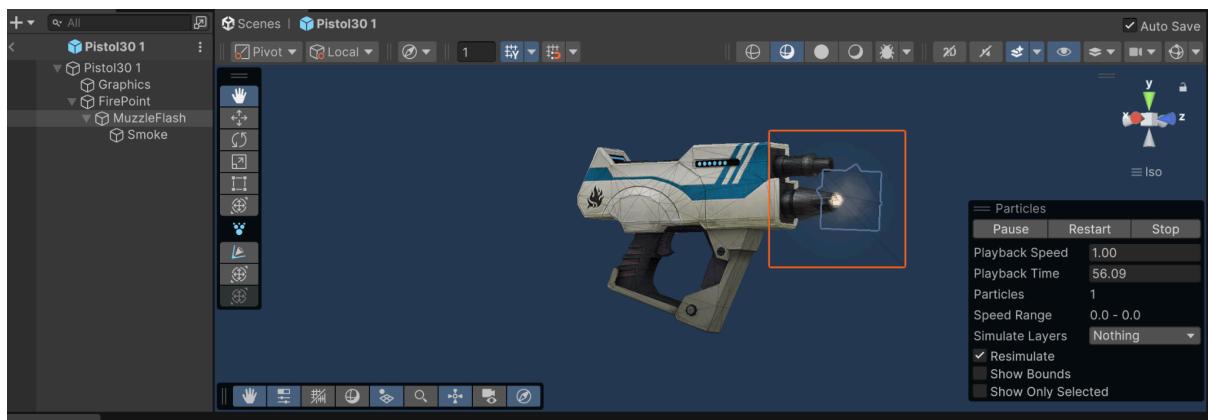
Les premières animations ont été les différents états du joueur : mobile, en mouvement ou dans les airs.



Ci-dessus vous pouvez observer, les différents systèmes de particules utilisés pour réaliser l'animation thruster.

Ensuite après le personnage il y'a eu les armes.





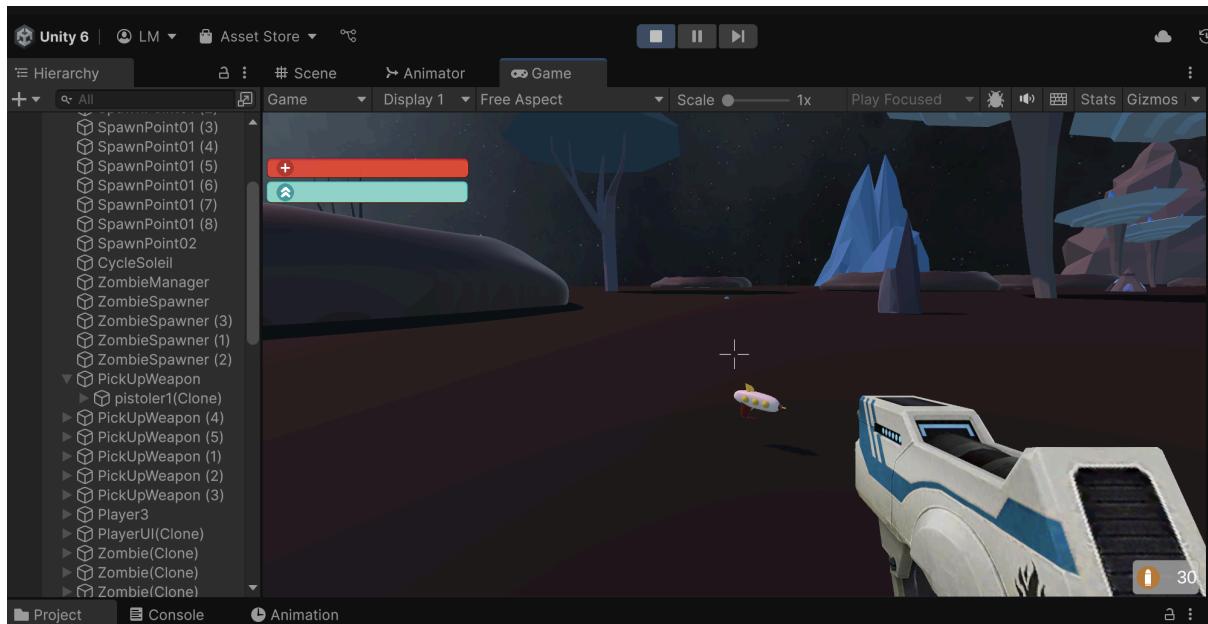
Sur les images ci-dessus, on peut voir les différents systèmes de particules de l'arme. Cela va du tir au refroidissement.

Il y a eu une dernière animation, représentant le rechargement de l'arme. N'étant pas très doué avec les animations, celle-ci reste très rudimentaire, en faisant juste pencher l'arme sur le côté pour recharger.

Après les animations, il a fallu réaliser un système de changements d'armes. Nous avons décidé de ne pas réaliser d'inventaires dans notre jeu pour éviter de s'éparpiller.

J'ai ainsi réalisé un script ainsi qu'un objet dans Unity, permettant de changer d'arme directement sur la map.

On peut changer d'arme en passant dessus ou encore recharger directement ces balles si c'est le même type d'arme que possédé.



J'ai placé les deux types d'armes proposées à six endroits différents de la carte.

Bon, après avoir réalisé les réelles bases de notre jeu, il a fallu commencer à créer un ennemi pour nos joueurs. Ils ont beau vouloir s'entretuer, il y a bien un moment où l'on veut changer de cible.

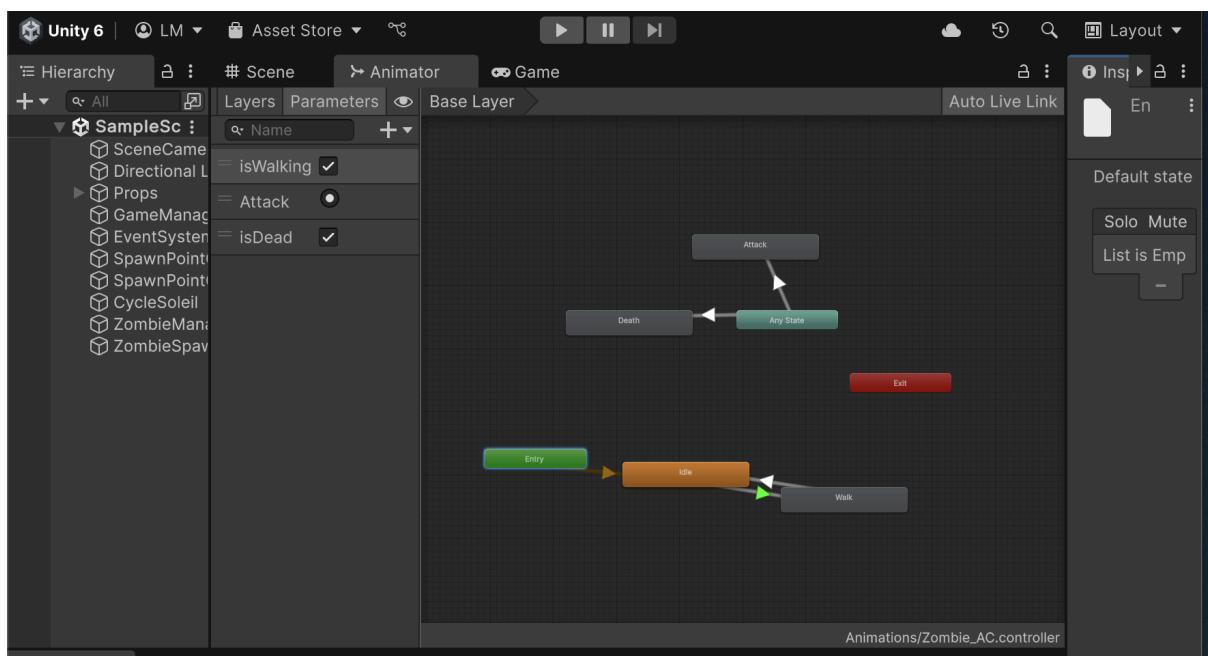
Théo à préparer un modèle de zombie avec un script permettant au zombie de tuer les joueurs.

Sauf que tout allait trop bien jusqu'ici, en implémentant le modèle et le script, je me suis rendu compte que Théo l'avait réalisé dans une version antérieure d'Unity. J'ai dû réadapter le code et le modèle pour qu'il convienne à la version du projet.

Mais les problèmes ne viennent jamais seul, le code du zombie ne fonctionnait pas avec Mirror. J'ai passé plusieurs jours à chercher une solution et après avoir réussi à la trouver, j'ai eu un problème avec le respawn de mes joueurs.

Les problèmes se sont enchaînés les uns après les autres, et j'ai eu une envie d'abandonner à certains moments mais malgré la fatigue, j'ai persévéré et j'ai réussi à résoudre tous les problèmes.

Voici le modèle de zombie qui m'a tant posé problème:



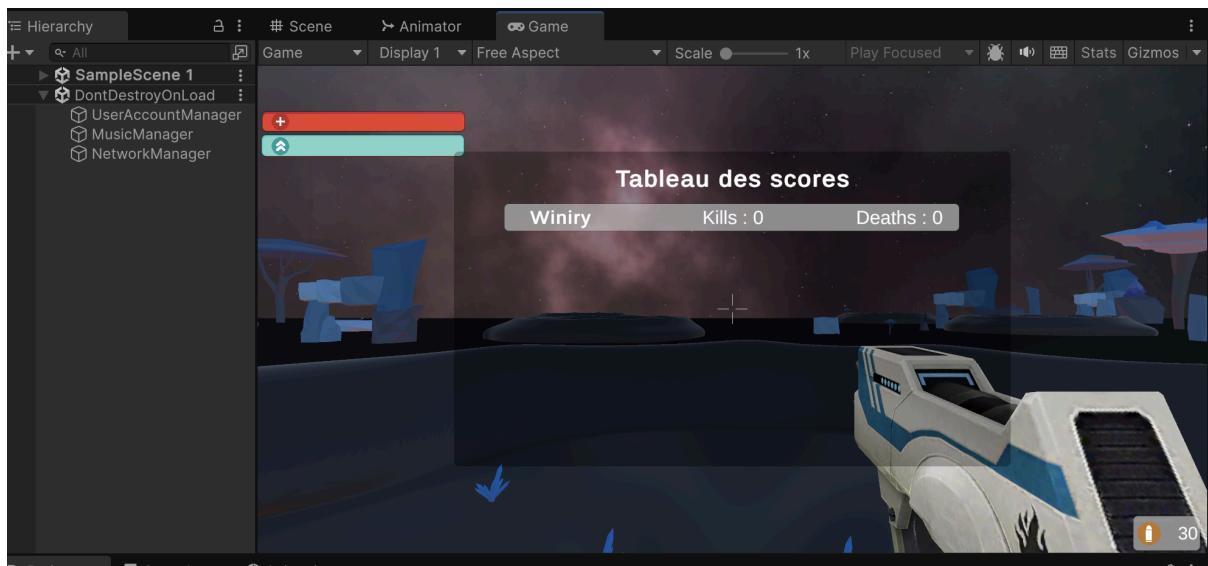
Et voici ces animations.

Ce modèle à plusieurs états :

- À l'arrêt, l'état Idle se déclenche, et le zombie se balance légèrement.
- En mouvement, l'état Walking se déclenche et le zombie lève les bras pour atteindre sa cible.
- L'attaque, l'état Attack se déclenche et le zombie donne une sorte de coup de poing au joueur.

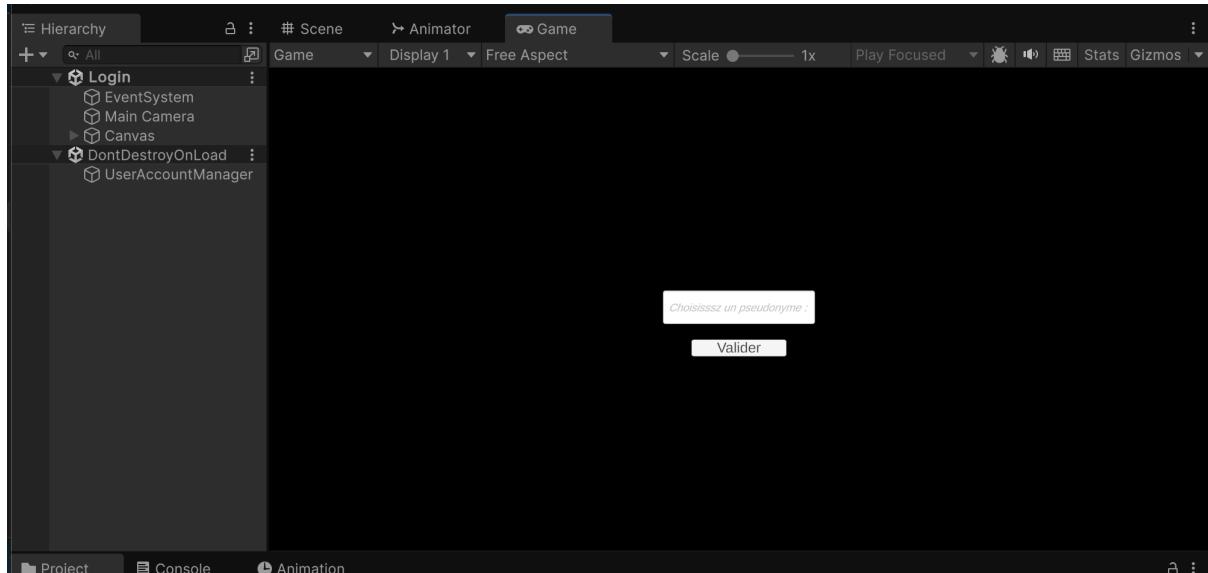
Après avoir implémenté le modèle du zombie, il a fallu créer le script de spawn des zombies ainsi que celui de dégâts. J'ai passé pas mal de temps à réaliser ces deux scripts, car j'ai dû différencier les dégâts des joueurs et des zombies et les gérer en même temps.

Parlons ensuite des UI, ces textes affichés dans le jeu. J'ai oublié de le mentionner avant mais j'ai réalisé à tableaux de score sur le nombre de mort d'un joueur et le nombre de fois qu'il tue un autre joueur.

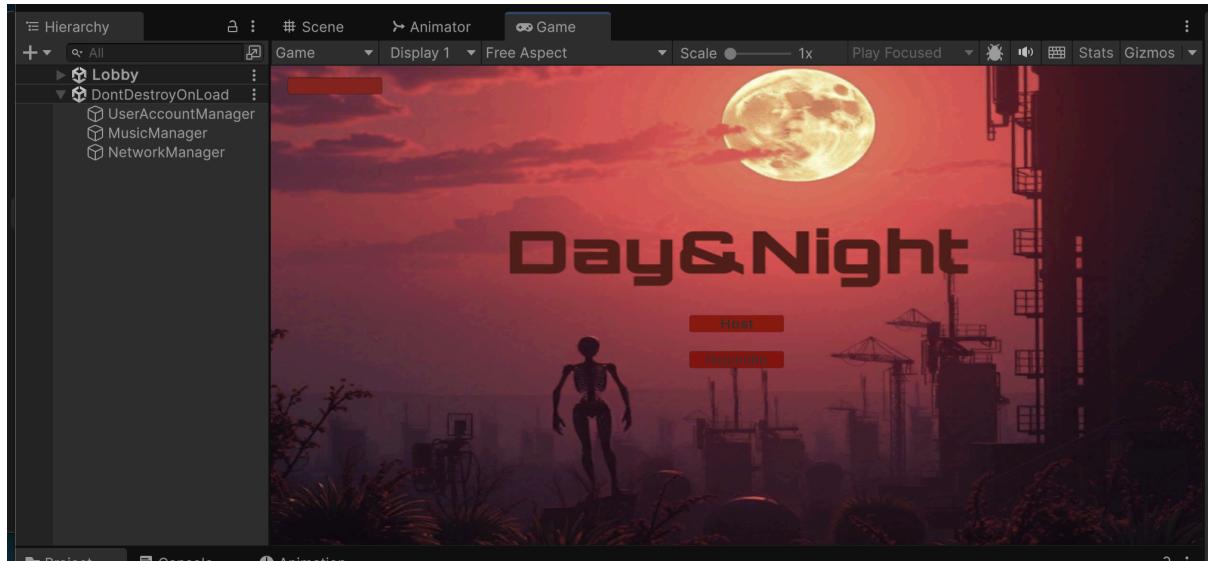


Ce qui nous amène à un autre ajout, très important, celui du pseudonyme.

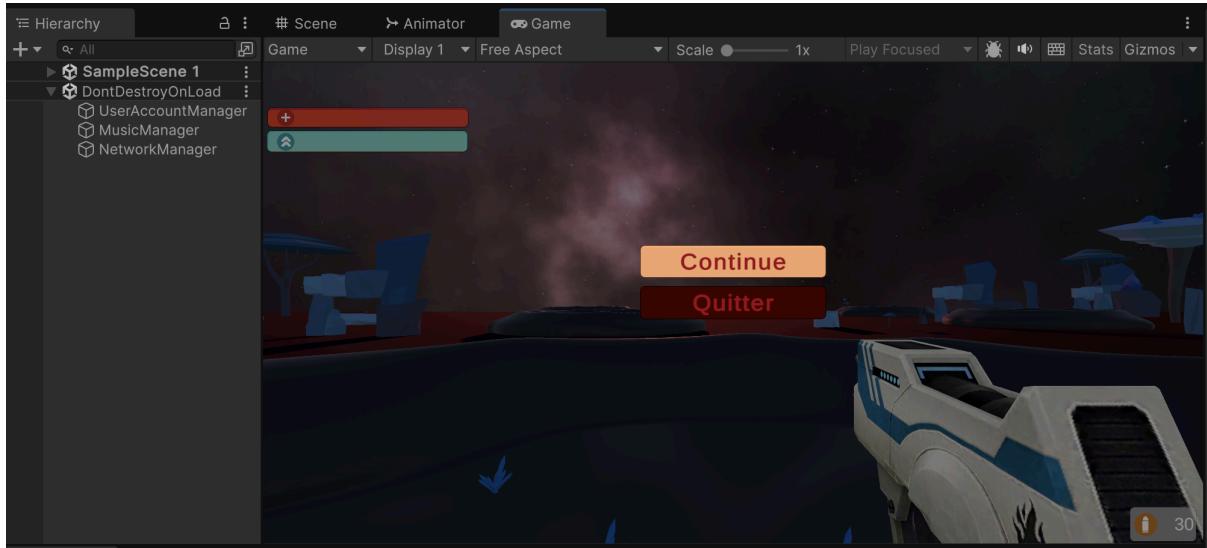
J'ai décidé de créer une nouvelle scène au tout début du jeu, qui permet au joueur de choisir leur pseudo.



Ensuite, une seconde page pour le menu cet fois-ci:



Un autre ajout à été le bouton quitter, pour pouvoir quitter la partie à tout moment.

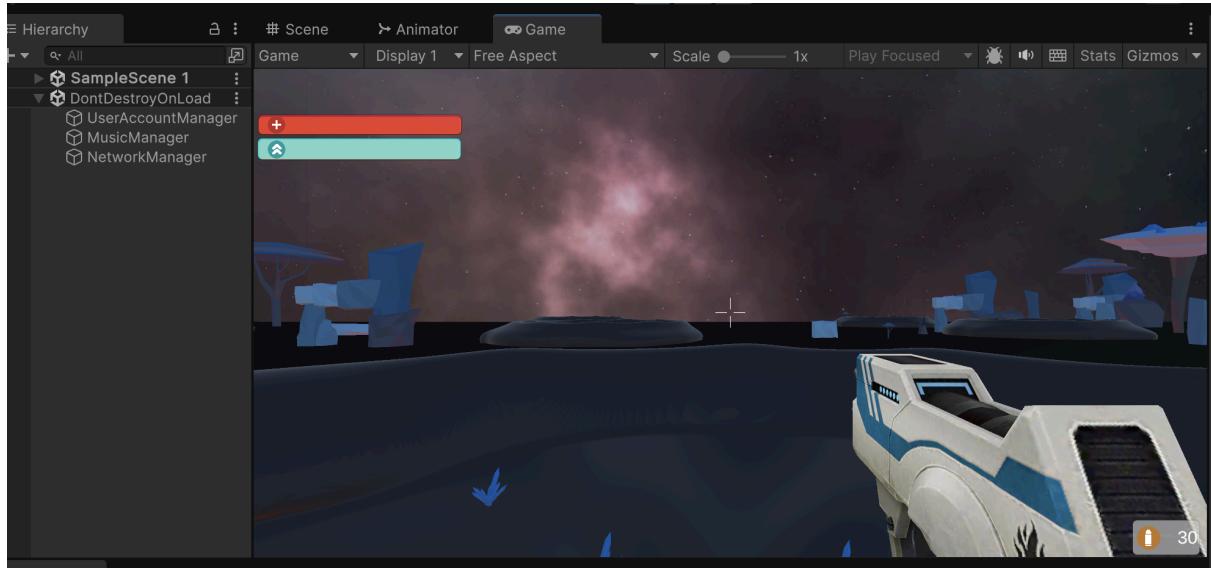


Les autres UI qui ont été implantés sont des classiques du jeu-vidéo :

- la barre de vie
- la jauge du thruster (d'endurance pour d'autres jeux)
- le nombre de munitions

Pour parler de ces UI, ils ont tous à peu près la même composition, avec différents textes et panel.

Pour le bon fonctionnement de ces UI, il a fallu créer un script qui permet de réduire la barre de vie et du thruster selon certains paramètres.



Le dernier ajout dans les UI à été l'ajout du pseudo et de la barre de vie de l'adversaire.

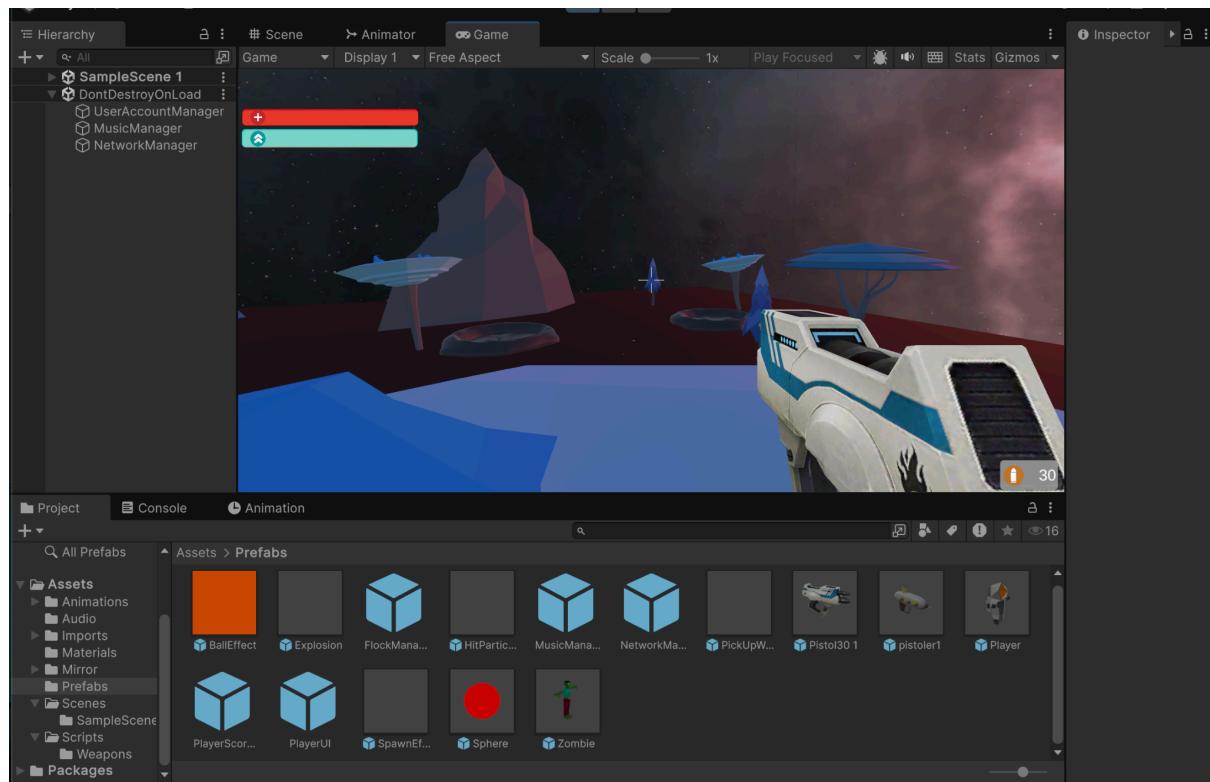


Vous remarquerez peut-être que le pseudo est à l'envers ...
Cela est dû à un script qui permet aux autres joueurs de voir la barre de vie et le pseudo du joueur de n'importe quel côté. Elle se tourne selon votre caméra. Comme je n'ai initié qu'un seul joueur, le nom et la barre de vie sont tournées vers sa caméra, ce qui fait le pourquoi du comment le pseudo est à l'envers.

La dernière partie à été le débogage de tous les problèmes cruciaux.

Le bug le plus critique se situait dans le système d'éliminations.

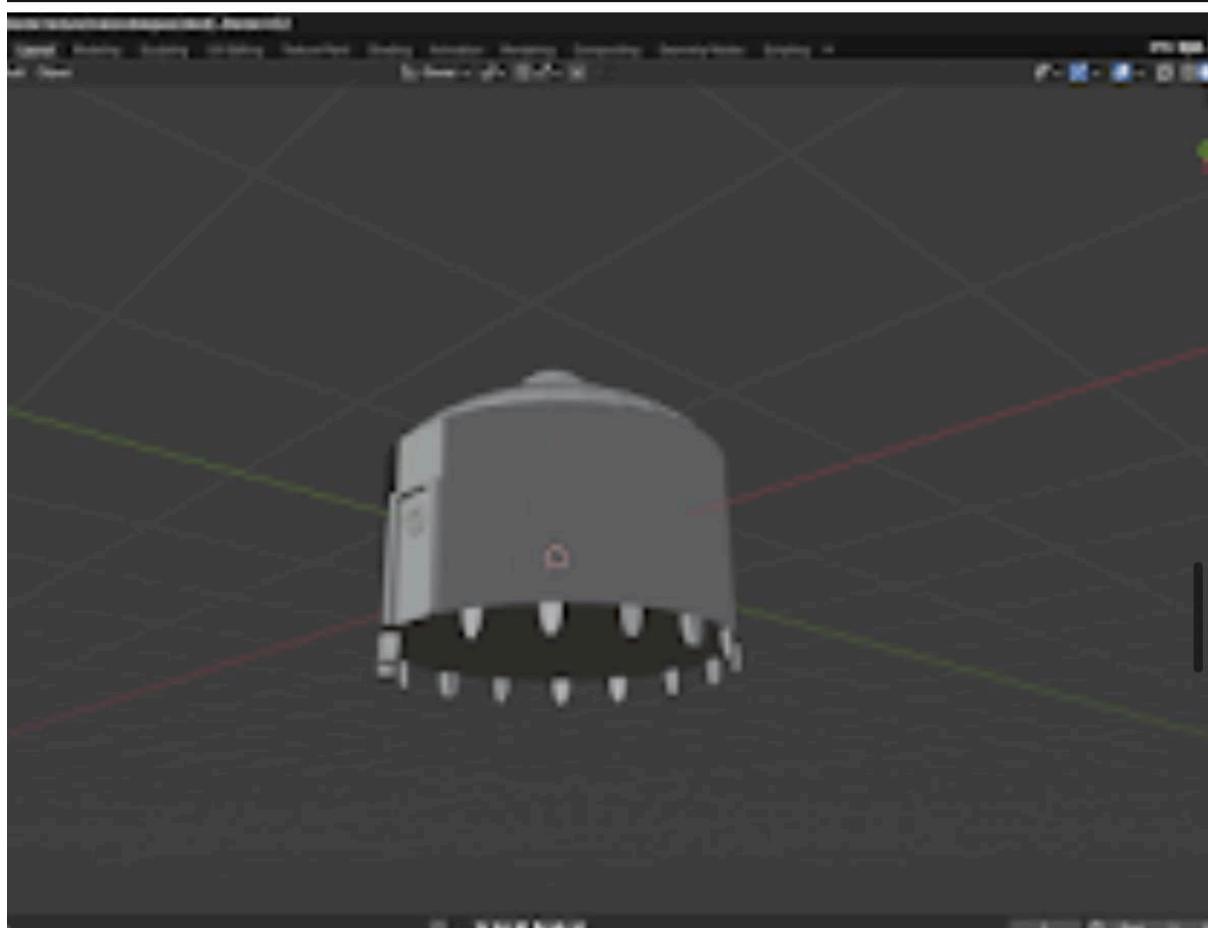
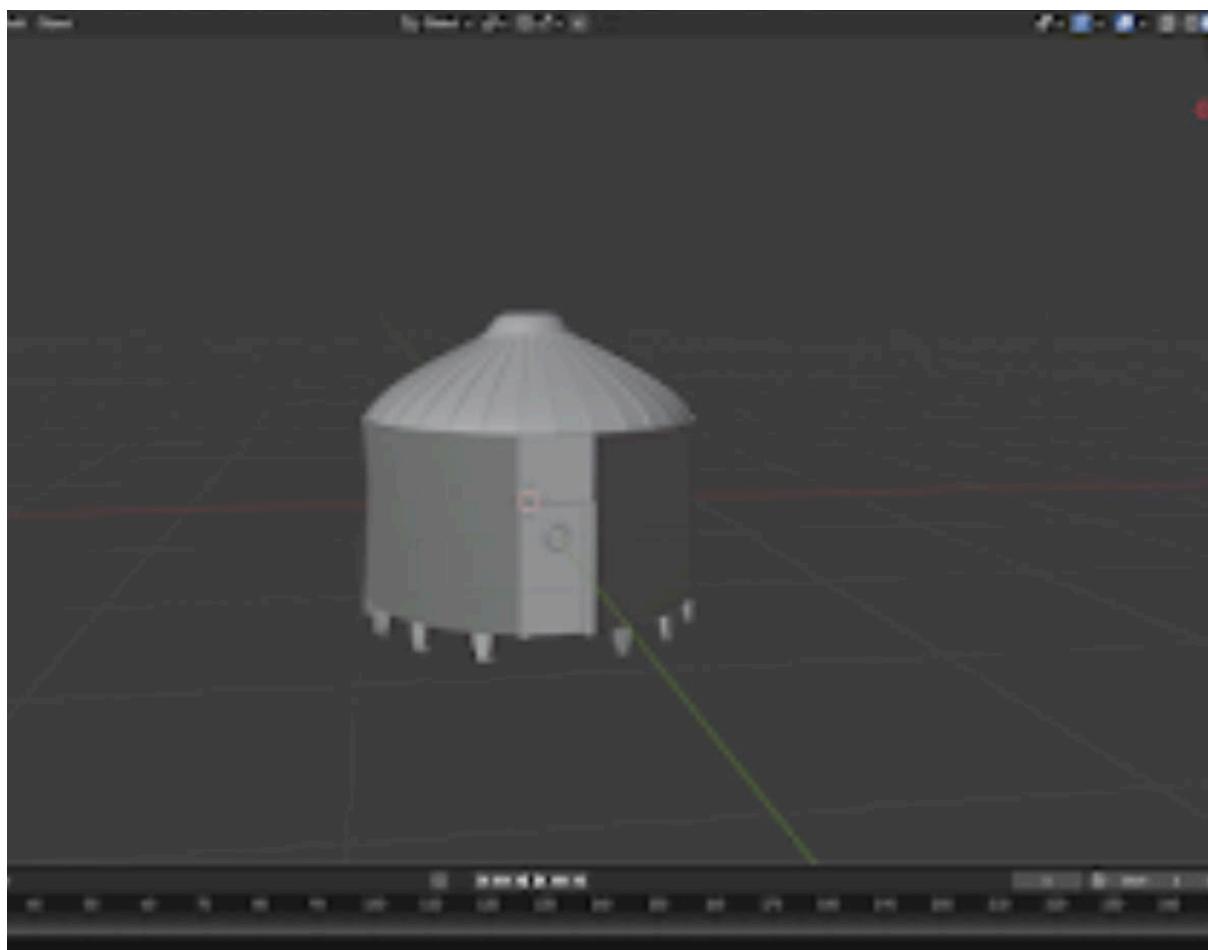
Pour donner un peu de contexte, quand un joueur rejoignait une partie, donc l'invité, et tuait des zombies, cela provoquait une déconnexion immédiate et bloquait le jeu. J'ai cherché de longues heures et fait de nombreux tests avant de réaliser que j'avais mal implanté une variable de Mirror. Il y avait un conflit entre les droits de l'host et de l'invité sur l'élimination des zombies. J'ai perdu énormément de temps pour résoudre ce bug et corriger les fonctions qui utilisait la variable buggé.



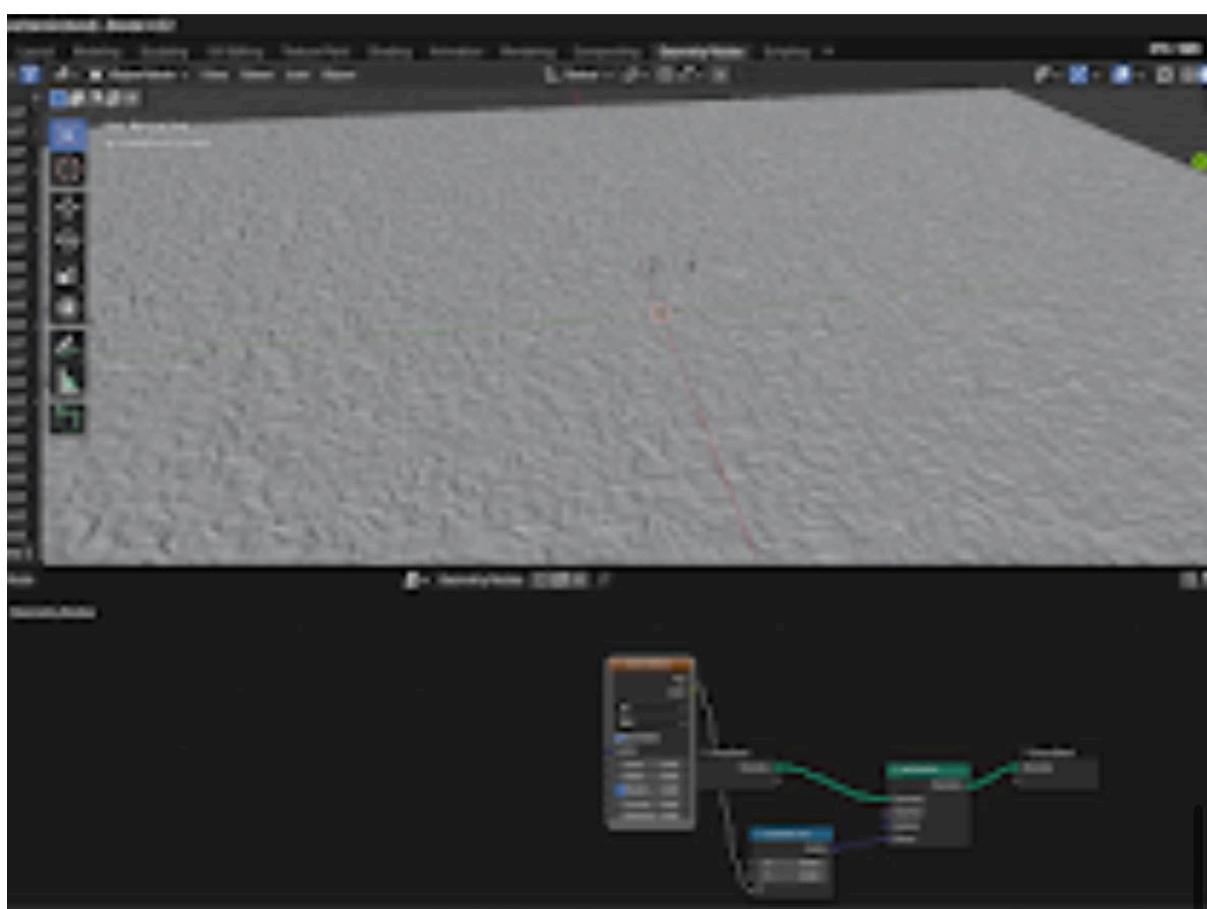
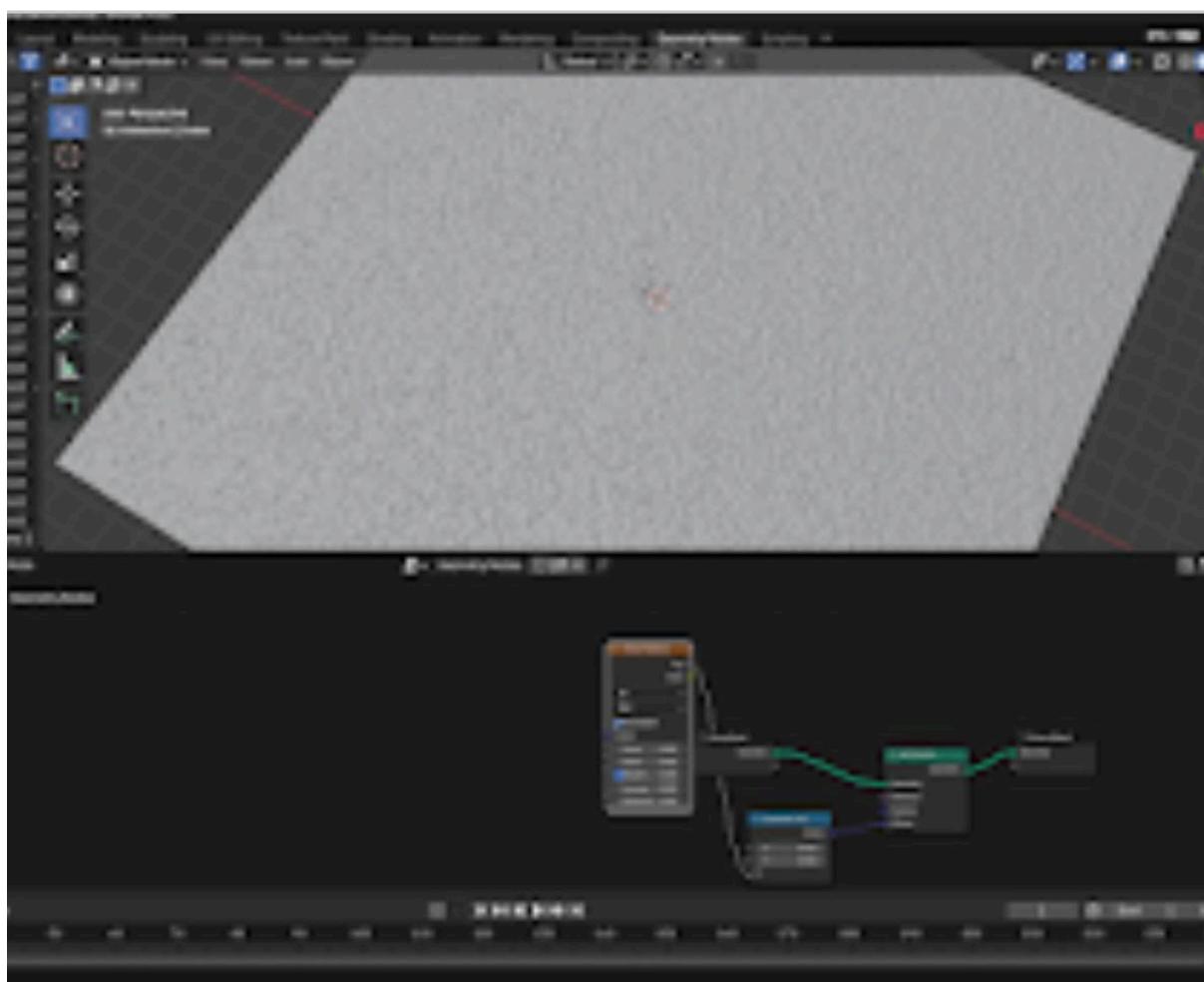
Ce projet m'a permis de découvrir Unity ainsi que d'approfondir mes connaissances en C#.

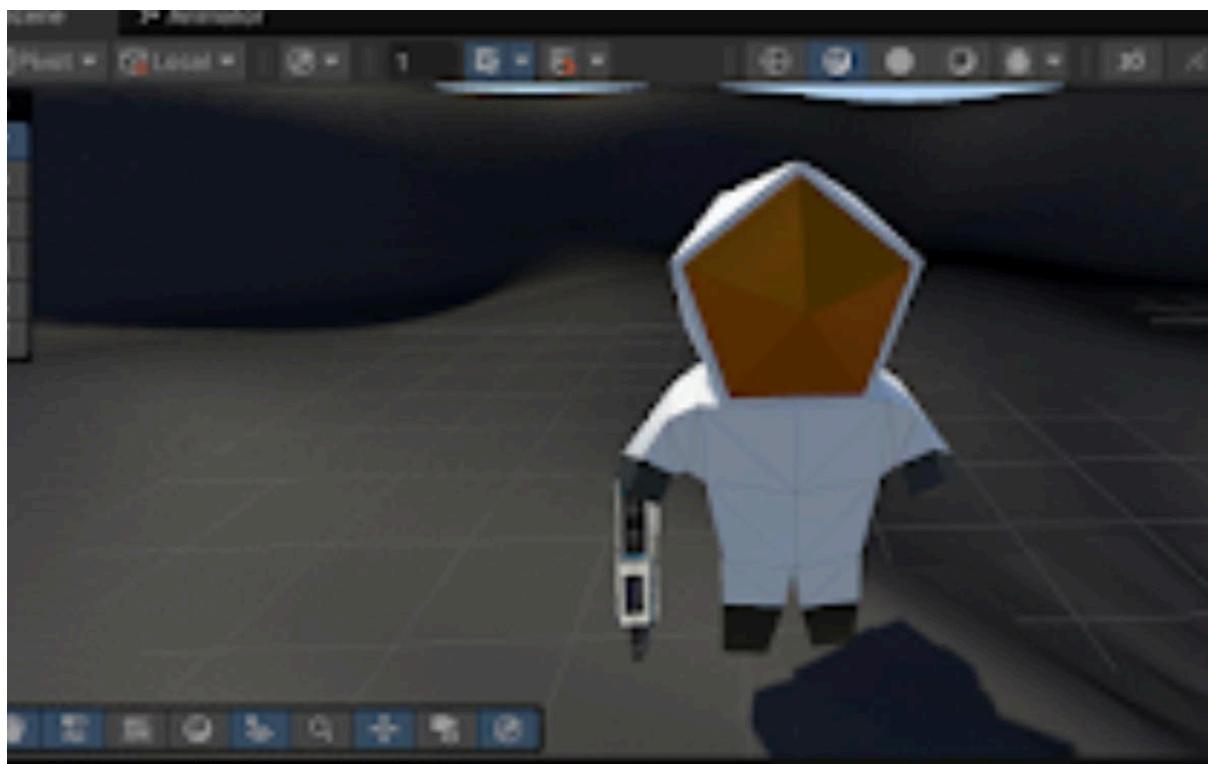
Il y a eu des hauts et des bas, des moments où j'ai voulu abandonner mais je me suis accroché et je suis fier du résultat.

Maxime :









I. Les difficultés pendant le projet

1. Problèmes de performances et limitations matérielles

Lors de la phase de production, l'équipe a rapidement été confrontée à des ralentissements majeurs sous Blender. Travailler sur des scènes de plus de 500 000 polygones provoquait des chutes de FPS (images par seconde) dans l'éditeur, rendant toute manipulation laborieuse. Par exemple, lors de la modélisation d'un tank détaillé pour le jeu, l'affichage en mode Sculpt générait des latences de plusieurs secondes à chaque coup de pinceau, ce qui a entraîné une perte de concentration et de productivité.

Pour pallier ces ralentissements, j'ai tenté plusieurs optimisations : réduction de la subdivision dynamiques, utilisation du "simplify" pour limiter la géométrie affichée en temps réel, et passage en mode "Bounding Box" pour certains objets lourds. Si ces astuces ont partiellement soulagé la situation, elles ont aussi entraîné une

perte de visualisation précise du modèle, compliquant le contrôle qualité en aval.

2. Bugs et instabilités logicielles

Au fil des versions de Blender utilisées (2.79, 2.80 puis 2.90), des bugs critiques se sont glissés dans le workflow :

- Corruption de fichiers .blend : deux fois, des sauvegardes automatiques n'ont pas suffi, et des scènes complètes – contenant plusieurs heures de travail – se sont retrouvées corrompues après un crash brutal du logiciel. La récupération partielle a nécessité l'intervention d'outils externes pour extraire des objets et du maillage, souvent de manière imparfaite (UV invalides, armatures désossées).
- Incompatibilité d'addons : certains scripts maison, conçus pour générer automatiquement des LOD (levels of detail), ne fonctionnaient plus après mise à jour vers 2.90. Les développeurs ont dû revoir toute la partie “export vers outil de pipeline” et réécrire des portions de code en Python, allongeant considérablement les délais.
- Problèmes d'export FBX : l'export FBX (format standard pour une intégration dans le moteur de jeu) générait parfois des inversions d'échelle (échelles en -1 sur l'axe X), brisant les normales et rendant le mesh invisible dans Unity/Unreal. J'ai décidé de mettre en place un protocole de vérification systématique : après chaque export, importer dans un projet de test du moteur, vérifier l'échelle et les UV, puis archiver la version validée.

3. Gestion des textures et shading

La création de matériaux PBR (Physically Based Rendering) dans Blender a aussi posé son lot de défis :

- Coordination des maps : la bonne configuration des textures (albedo, roughness, metalness, normal map) n'était pas

toujours intuitive. Certains artistes exportaient des fichiers TIFF en 16 bits, trop lourds pour le jeu, obligeant à des reconversions en PNG 8 bits, parfois avec perte de qualité visible.

- Cycles vs. Eevee : Blender propose deux moteurs de rendu intégrés. Si Eevee est plus rapide pour le viewport, il ne gère pas toujours les effets avancés (SSS, volumétrie) de la même manière que Cycles. L'équipe a tâtonné pour avoir un aperçu fidèle à ce que verraient les joueurs, sans pour autant sacrifier la réactivité de l'éditeur. En conséquence, de nombreux tests A/B ont été effectués : paramètres d'éclairage, baking des textures, vérification sur de multiples configurations GPU.

II. Coordination d'équipe pour importer les projets dans le jeu

1. Mise en place d'un pipeline standardisé

Sans pipeline clair, chaque export devenait un sujet de discussion :

- Choix du format : FBX pour l'animation, glTF 2.0 pour les assets statiques. Un document de spécifications précisait les versions supportées (FBX 7.4, glTF 2.0), les options à cocher ("Apply Transform", "Include UVs", "Batch Mode"), et la procédure d'export.
- Gestion des versions : les fichiers .blend et les exports destinés au moteur étaient stockés dans Git LFS. Chaque commit devait être accompagné d'un message structuré :

Lorsque j'ai commencé ce projet de création de modèles 3D pour un jeu vidéo avec Blender, je ne m'attendais pas à une telle intensité. Ce fut une aventure à la fois passionnante, stressante, enrichissante et pleine de rebondissements. Malgré les nombreuses difficultés, j'en garde un excellent souvenir, car ce projet m'a énormément appris — tant sur le plan technique que personnel.

Dès les premières réunions, j'ai compris que le projet ne se limiterait pas à une simple série de modélisations. Il s'agissait d'un véritable processus de production, avec des objectifs précis, des échéances régulières et une exigence professionnelle. Cette immersion dans un contexte quasi-professionnel m'a permis de me confronter aux réalités du travail en équipe, à la rigueur qu'implique une construction artistique, et à la nécessité de livrer des assets de qualité, optimisés et fonctionnels.

J'ai été impressionné par l'ampleur du projet et le niveau d'engagement attendu. Travailler sur un jeu vidéo, même à une échelle réduite, nécessite une attention constante aux détails, une capacité d'adaptation et une volonté de progresser chaque jour. Cela m'a forcé à sortir de ma zone de confort.

Je connaissais Blender de manière assez superficielle au départ — je savais modéliser quelques objets simples, utiliser les modificateurs de base et créer des matériaux sommaires. Très vite, j'ai dû apprendre à aller bien plus loin : UV mapping, baking, gestion des armatures, export propre vers un moteur de jeu... autant de notions qui étaient floues pour moi, et qui ont fini par devenir familières.

Ce qui m'a marqué, c'est la richesse du logiciel, mais aussi la complexité de son interface. Il m'a fallu du temps pour comprendre comment organiser efficacement mon espace de travail, comment utiliser les bons outils au bon moment, et surtout comment éviter les erreurs fréquentes (oubli d'appliquer l'échelle, normales inversées, UV mal déployées...).

L'apprentissage ne s'est pas fait en un jour. J'ai connu de nombreuses frustrations, notamment lorsqu'un fichier refusait de s'exporter correctement ou que mes textures apparaissaient déformées dans le moteur de jeu. Mais chaque échec a été l'occasion de comprendre ce que je faisais mal et d'améliorer ma méthodologie. J'ai appris à ne pas me décourager, à chercher des

solutions, à demander de l'aide — et à documenter ce que j'avais compris pour ne pas refaire les mêmes erreurs.

L'un des aspects les plus difficiles du projet a été la gestion du stress. Il y avait des deadlines à respecter, des retours réguliers de l'équipe, et parfois des changements de dernière minute à gérer. À plusieurs reprises, j'ai eu l'impression d'être débordé, notamment quand plusieurs tâches s'accumulaient et que je devais jongler entre modélisation, retopologie, UVs, et export.

J'ai compris qu'il était crucial de savoir prioriser : distinguer ce qui est urgent de ce qui peut attendre, estimer correctement le temps qu'une tâche allait me prendre, et anticiper les problèmes. Ce projet m'a permis de développer des réflexes de planification, que je n'avais pas auparavant. J'ai commencé à organiser mes plannings différemment : je divisais les tâches en blocs de travail plus courts, j'utilisais des checklists, et je me fixais des objectifs qui étaient réalistes.

Même si la pression était parfois pesante, j'ai fini par y trouver une forme de motivation. Le fait de devoir rendre un asset fini dans un délai serré me forçait à me concentrer, à donner le meilleur de moi-même. Et lorsque je réussissais à livrer un modèle validé du premier coup, la satisfaction était immense. Ce stress, que je percevais au début comme un frein, est devenu peu à peu un moteur d'efficacité.

Un autre aspect très enrichissant du projet a été le travail en équipe. J'ai dû apprendre à communiquer efficacement avec d'autres "artistes", avec chacun leurs forces et leurs faiblesses. Il ne s'agissait pas seulement de produire un bel objet en 3D, mais de créer un asset cohérent avec l'univers du jeu, adapté aux contraintes du moteur, et utilisable directement par les autres membres de l'équipe.

Au départ, il y avait parfois des malentendus : des modèles trop lourds, des fichiers mal nommés, des pivots mal placés... Cela

générait des frictions, car chaque erreur pouvait bloquer toute une chaîne de production. Mais avec le temps, on a mis en place des outils de communication efficaces : des tableaux partagés, un canal dédié sur Discord, des points réguliers pour valider l'avancement et corriger les erreurs.

Cette dimension collaborative m'a beaucoup plu. Elle m'a appris à être plus rigoureux, à documenter mon travail, à expliquer mes choix techniques, mais aussi à écouter les retours et à accepter la critique. Travailler dans un environnement où chaque membre dépend du travail de l'autre m'a fait prendre conscience de l'importance de la fiabilité et de la transparence dans un projet commun.

Même si le projet a été difficile par moments, je le considère aujourd'hui comme l'une des expériences les plus formatrices que j'ai vécues. Il m'a permis de :

- Renforcer mes compétences techniques sur Blender, en modélisation, texturing, rigging, export, et intégration dans un moteur de jeu.
- Développer mon autonomie : j'ai appris à me débrouiller face à des problèmes complexes, à chercher des solutions, à expérimenter.
- Mieux gérer mon temps et mon stress, en adoptant une méthode de travail plus organisée et plus professionnelle.
- Travailler en équipe, en intégrant les contraintes des autres, en communiquant clairement et en acceptant les ajustements nécessaires.

Au-delà des compétences techniques, ce projet m'a donné confiance en moi. Il m'a montré que je suis capable de m'investir pleinement dans une production exigeante, de tenir mes engagements, et de m'adapter à des contraintes réelles. Il m'a aussi donné envie de poursuivre dans cette voie, d'explorer

davantage la 3D, les moteurs de jeu, et peut-être un jour de participer à un projet encore plus ambitieux.

Conclusion :

Ce projet a été pour chacun de nous une aventure enrichissante et nous a permis de vivre toutes sortes de nouvelles expériences : même s' il peut y avoir des déceptions et des soucis, on a su rester concentré et motivé pour arriver au bout de notre quête. Les efforts de chacun n'auront pas été vains.

Enfin, ce projet nous a permis de mieux cerner nos envies, nos points forts, et les domaines dans lesquels nous souhaitons progresser à l'avenir. Et s'il ne devait rester qu'une seule chose de cette aventure, c'est cette envie de continuer de toujours créer. Même si toutes les fonctionnalités n'auront pas été finalisées, il n'est pas improbable que de notre côté nous voudrions continuer de créer un jeu abouti.