

RAPPORT DE PROJET

BAPTISTE CORMORANT
MATTHIEU CORREIA
JUSTINE GUEULET

EPITA
Lyon

Table des matières

1 Plan de soutenance	4
2 Synopsis	5
3 Origines du projet	6
4 Membres du groupe	7
4.1 Justine GUEULET	7
4.2 Matthieu CORREIA	8
4.3 Baptiste CORMORANT	9
5 Cahier des charges	10
5.1 Ébauche initiale	10
5.1.1 Concept	10
5.1.2 Sites	10
5.1.3 Mécaniques	11
5.1.4 Graphismes	12
5.2 Version finale	13
5.2.1 Concept	13
5.2.2 Site	13
5.2.3 Mécaniques	14
5.2.4 Graphismes	15
5.3 Défis	16
5.4 Outils & Technologies	16
5.5 Besoins financiers	16
6 Développement	17
6.1 Justine	17
6.1.1 Map 3D	17
6.1.2 Menu : Ancienne version	19
6.1.3 Menu : Version actuelle	20
6.1.4 Mini laboratoire	21
6.1.5 Snaps : Probuilder & ProGrid	22
6.1.6 Map laboratoire	22
6.1.7 Salle du Réacteur principal	24
6.1.8 Salle du Réacteur immergé	25
6.1.9 Salle principale	26
6.1.10 Passage secret	27
6.1.11 Lounge	28
6.1.12 Balancement des armes & HeadBob	29
6.1.13 Ambiance sonore	30
6.1.14 Ambiance visuelle	31

6.2	Matthieu	32
6.2.1	Avatar 3ème personne	32
6.2.2	Gestion des entrées utilisateur	33
6.2.3	Réglages caméras	34
6.2.4	Animations de déplacement avatar	35
6.2.5	IA zombies	36
6.2.6	Système de tir	37
6.2.7	Drone animations	38
6.2.8	Déplacements drone	39
6.2.9	Addressables	40
6.2.10	Installateur InnoSetup	40
6.2.11	Ragdolls & Death Animations	41
6.2.12	Object Pooling & Spawner	42
6.2.13	Multijoueur	43
6.2.14	IA tourelle	44
6.2.15	Avatar 1ère personne	45
6.2.16	Système de collecte	46
6.2.17	Outil SceneBreaker	47
6.3	Baptiste	48
6.3.1	Site	48
6.3.2	Sectorisation	52
6.3.3	Jaquette	53
7	Ressenti	54
7.1	Justine	54
7.2	Matthieu	55
7.3	Baptiste	56
8	Annexes	57
8.1	Schémas de conception	57
8.2	Aperçus	59
8.2.1	Avant changement de concept	59
8.2.2	Après changement de concept	61
8.3	Sources	68

1 Plan de soutenance

Remise des éléments du projet :

- Clé USB & jaquette
- Manuel d'installation et d'utilisation

Présentation :

- Contexte
- Historique des soutenances
- Membres du groupe

Démonstration du jeu :

- Installation
- Gameplay
- Désinstallation

Site Web

Conclusion

2 Synopsis

"... J'ai froid ... Il fait sombre ... où est-ce que je suis ..."

Dans une cuve sombre et étroite, je me réveille au bruit des alarmes qui retentissent. Leurs lumières rouges qui tournent et se reflètent sur les surfaces me glacent le sang. J'hésite à m'aventurer hors du seul endroit qui me paraisse familier au milieu de ce chaos, mais l'asphyxie est proche et m'y oblige.

Ranimé en territoire inconnu, le protagoniste est complètement désorienté et térrifié. En s'extirpant de la cuve qui le contient, il est abasourdi d'apercevoir un corps sans vie et ensanglanté gisant à ses pieds. En tenu de protection contre les radiations, cette personne porte une arme qui semble avoir été utilisée, mais utilisée contre quoi au juste ?

Le protagoniste s'en saisi alors, car elle lui confère un infime sentiment de protection dans cet abîme apocalyptique, et ce malgré le fait qu'elle n'ait pas eu l'air très utile pour celui à ses pieds.

Des bruits s'approchent progressivement, des grognements, quelque chose frotte les parois métalliques. Je dois sortir de là ou ce sera la fin...

3 Origines du projet

Depuis des mois nous l'attendions avec impatience, après avoir subi l'AFIT nous étions heureux de pouvoir enfin nous lancer sur un projet plus libre où nous imposerions nos propres limites et objectifs, le projet S2.

Nous avions tous déjà fait des projets liés à l'informatique, que soit en NSI ou en Terminale et cela avait été une grande source d'enrichissement personnel, ce qui explique notre joie de retrouver ce sentiment d'accomplissement à l'issue de ce projet.

Tout a commencé lorsque nous prenions le bus ensemble pour rentrer des cours. Au fil de la conversation, le sujet s'est porté sur ce projet et si certains d'entre nous avaient déjà des groupes ou des idées de projet. Originellement, trois d'entre nous voulions tous faire sauf un jeu, du simulateur 3D de navigation spatiale au sein du système solaire en fonction de la puissance des vaisseaux et des points d'applications des poussées jusqu'à une application de création de notes (mémos) et partage de fichiers intégré.

Malgré toutes ces idées hétérogènes, notre choix s'est porté sur un jeu 3D de survie contrairement à ce que nous avions prévu, et ce parce qu'un des membres du groupes paraissait très motivé pour le mener à bien. Cependant, nous avons eu la désagréable surprise au cours du projet de remarquer que l'inverse s'est produit et que malheureusement la personne qui nous avait convaincus ne s'est pas investie et a désormais officiellement quitté le projet.

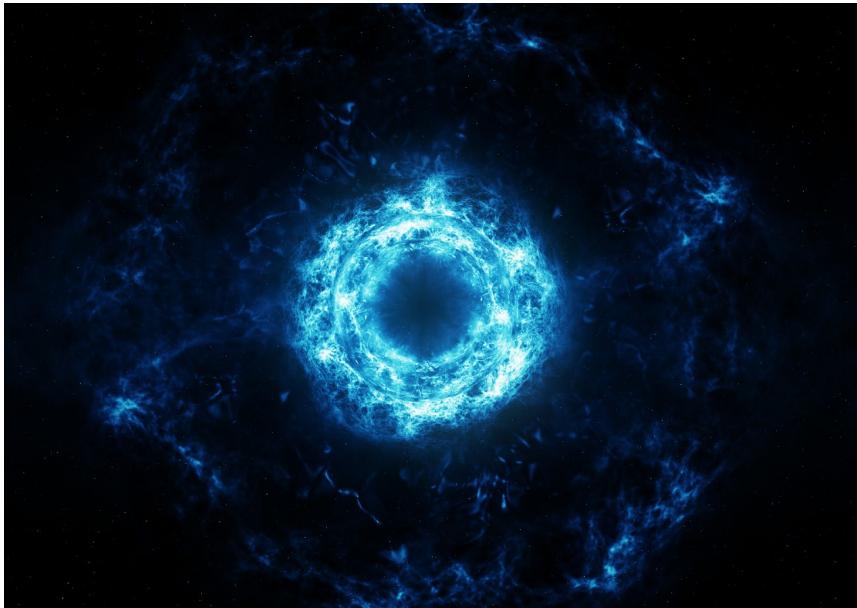
4 Membres du groupe

4.1 Justine GUEULET



J'ai toujours grandement apprécié tout ce qui touche à l'art. Depuis mon plus jeune âge, je dessine et je peins. Ma mère peut d'ailleurs vous dire que j'ai commencé à dessiner avant même de parler. Le fait de pratiquer l'art permet de me détendre et de me concentrer et c'est ainsi que l'art est très rapidement devenu mon passe-temps préféré. Que ce soit l'art culinaire, le maquillage, la couture, la peinture, la conception de bijoux, modélisation 3D ou dessin, j'ai presque tout essayé dans ce domaine. Pendant plusieurs mois, ce qui m'a le plus intéressé, c'était la customisation de vêtements et chaussures. J'ai même failli commencer un business, mais le temps me manquait. Mais ce n'est pas mon seul passe-temps, j'ai passé beaucoup de temps à regarder des documentaires et à lire les actualités. Ces derniers étaient principalement à propos de politique ou de science, ce qui a développé chez moi une certaine passion pour la science (notamment l'espace) et la politique. J'ai d'ailleurs participé au club de débat MUN (Model United Nations) pendant 4 ans. Le fait d'avoir vécu mon enfance au Qatar m'a aidé à faire ressortir ma créativité ainsi que d'avoir une pensée cosmopolite.

4.2 Matthieu CORREIA



Dès mon plus jeune âge, je me suis intéressé à la science et aux découvertes. Lorsque je rentrais de l'école je regardais des documentaires sur la chaîne *RMC Découverte* où passaient de nombreux thèmes passant par la faune et la flore aux grandes guerres mondiales mais également des sujets plus poussés comme le nucléaire, la physique quantique et ses condensats de Bose-Einstein par exemple ainsi que les voyages spatiaux supraluminiques grâce au théorique vaisseau à distortion d'Alcubierre. La science me fascine car elle rend possible l'impossible avec assez d'effort et de patience. Une vidéo que j'ai vu il y a quelques années nommée *Timelapse of the futur* m'a profondément impacté car je me suis rendu compte de l'infini dans laquelle nous nous trouvons, mais que même elle possède une fin où même l'inexistence cessera d'exister. Cette considération est principalement ce qui m'a poussé dans mes études d'ingénieur car je veux participer aux avancées dans l'espoir de permettre à l'humanité d'échapper à ce destin funeste qui équivaudrait à n'avoir jamais existé. Pour nuancer cette pensée, une série nommée *The Expanse* qui relate de ce que la vie humaine pourrait devenir d'ici quelques centenaires après la démocratisation du voyage spatial me redonne de l'espoir et me rapporte à un idéal accessible.

4.3 Baptiste CORMORANT



Je peux commencer par dire que je suis passionné par la nouvelle technologie ce qui paraît logique vu l'école où je suis, mais je vais préciser un peu. J'ai fait de nombreux projets de développement web, des sites (en particulier des sites vitrine) et des bots, cela avait de très nombreuses utilités, mais dans la plupart des cas, c'était pour découvrir de nouvelles technologies sur le web comme tree.js (permet de mettre des modèles 3 d sur un site), react.js (application/site), svelte.js (comme react, mais en plus optimisé, mais une plus petite communauté), express.js (serveur web), avec aussi l'utilisation de base de données telle que SQLite ou Mongodb. La prochaine étape va être de découvrir des frameworks css. Je fais aussi du réseau, avec l'utilisation de proxmox qui permet de séparer sa machine en plusieurs petites machines, c'est-à-dire crée des machines virtuelles (c'est un système d'exploitation.), ou encore plesk pour gérer une multitude de sites web sur le même VPS et sur la même IP ou encore pterodactyl qui permet grâce à docker de lancer des processus dans des conteneurs et d'avoir un panel propre. Ce que je compte faire prochainement et m'intéresser à l'iot, avec par exemple l'utilisation de capteur de température pour avoir rapidement la température autour de la maison (et activer chauffage ou clim en conséquence) ou encore l'utilisation de capteur de mouvement pour activer ou éteindre les lumières automatiquement, plein de petits projets qui seront très sympas à faire dans le futur. En-dehors de l'informatique, je suis aussi un passionné d'histoire et de science, histoire qui englobe les différentes mythologies (nordique, grec, mésopotamienne...) et les histoires de différents pays, c'est quelque chose qui me passionnait au lycée/collège. Pour la science, c'est surtout que ça permet de comprendre le monde qui nous entoure, j'aime bien comprendre ce qui m'entoure et par la même occasion utiliser la science pour répondre à ces questions. Comme marché le processeur ou comment les plantes poussent et se nourrissent. Pour conclure, j'aime comprendre et voir le monde qui m'entoure.

Comment s'est passé l'année à EPITA, comme tout le monde, on a des hauts et des bas. Je n'ai pas perdu ma motivation puisque je savais que c'est ce que je voulais faire mais à des moments elle en prenait un coup dans le cas d'une mauvaise note ou juste de questions sur mon avenir, je sais que je n'ai pas forcément besoin d'un diplôme d'ingénieur pour apprendre, mais je le fais à EPITA puisque je sais qu'en sortant j'aurais des connaissances et un diplôme pour le garantir même si ce n'est pas ma motivation principale, ce qui ai bien aussi c'est de découvrir de nouvelles personnes et découvrir de nouveaux horizons et grandir par la même occasion. Mais je reste bien motivé pour aller au bout parce que je sais que ça en vaut la peine.

5 Cahier des charges

5.1 Ébauche initiale

5.1.1 Concept

Notre projet sera un jeu vidéo où le joueur sera un des rescapés d'une calamité environnementale. Equinox apparaît comme un jeu d'exploration libre (Open World) où le joueur est libre d'explorer la carte à sa guise. Immérité dans une ambiance post-apocalyptique, le joueur est tenu en haleine par son environnement hostile.

Dans un monde dystopique où les humains ont tenté de résoudre la crise environnementale qui les auraient mené à leur fin, des scientifiques ayant découvert une nouvelle matière (le Neutronium) tentent de la maîtriser afin de créer un nouveau type de source d'énergie propre. Leurs recherches furent sans succès et l'humanité périclita à cause des températures glaciales engendrées par les bouleversements climatiques. Le protagoniste se réveille, amnésique, entouré d'un brouillard dans des montagnes enneigées et partira en quête de réponses sur les causes de cette situation.

Le protagoniste aura pour but de comprendre la situation, savoir où il est et ce qu'il s'est passé. Il devra survivre dans un environnement traumatisant et extrême qui le poussera dans ses retranchements aussi bien physiquement que mentalement. Il devra faire attention à ses ressources et progresser dans ses découvertes pour répondre à ses questions et comprendre les circonstances qui ont menées à la périclitation de l'humanité.

5.1.2 Sites

Le projet aura deux sites à sa disposition, un wikia secondaire permettant aux joueurs de pouvoir accéder facilement et rapidement à des spécificités sur le jeu et où seront enregistrés chaque détail des mises à jour et actualisés régulièrement, et un site principal contenant les détails techniques et légaux ainsi que le lien de téléchargement.

5.1.3 Mécaniques

Le jeu contiendra différents mécaniques qui permettront d'agrémenter le scénario et l'ambiance tels que :

L'Avatar : Protagoniste que nous contrôlerons à l'aide de la souris et du clavier, c'est à travers lui que nous explorerons l'environnement et intérragirons. Il sera caractérisé par différentes données telles que la vie, la satiété, l'hydratation et la batterie. Si le taux de satiété et/ou d'hydratation atteint zéro, il commencera à perdre continuellement de sa vie jusqu'à ce qu'il soit stabilisé, de plus il mourra si sa vie atteint zéro.

PNJ : Les PNJs sont des survivants non-joueurs que l'on peut rencontrer sur la map. Leurs interactions sont sources d'informations pour découvrir le scénario et de matériel tel que des armes ou encore des provisions.

Ennemis : Pendant sa progression dans l'environnement, le joueur fera face à l'adversité qu'offrent les ennemis sous forme de zombies, de mutants ou encore d'égarés (entité spectrale).

Objets : Afin de garantir sa survie, le protagoniste se devra de porter une attention constante sur ses systèmes de survie susmentionnés et établir une planification adaptée. Afin de les garder stabilisés, il pourra faire usage de consommables qu'il trouvera dans son environnement tels que des provisions ainsi que de l'eau pour satisfaire sa satiété et son niveau d'hydratation.

5.1.4 Graphismes

Afin d'établir l'ambiance désirée pour ce jeu (à savoir une tension constante et pesante), il est nécessaire de spécialiser certains décors pour nos besoins. Nous allons donc centraliser les modèles que nous trouverons en accord avec le jeu sur certains sites tels que l'Unity Asset Store ou encore Sketchfab, puis les modifierons en respectant les conditions légales des licences attribuées afin qu'ils s'assimilent les uns aux autres.

Le jeu se déroulera dans une zone montagneuse enneigée comportant quelques villages de survivants. Il arborera une ambiance dystopique soutenue par une luminosité naturelle rouge sang émise par une lune fragmentée visible dans le ciel qui sera la seule source de lumière naturelle du jeu.

Au centre de la map se trouvera une montagne principale qui surplombe les environs et servira de support pour le laboratoire. À son pied se trouvera un grand lac autour duquel nous trouverons un bunker qui fera office de base au joueur. Sur le toit de ce bunker se trouvera une tourelle automatique qu'il sera possible d'activer avec suffisamment d'énergie et protègera le repaire des ennemis environnants. Plus loin nous trouverons quelques villages épars et éparpillés à travers la map ainsi que des forêts de bois morts et des plaines enneigées.

5.2 Version finale

5.2.1 Concept

Notre projet sera un jeu vidéo où le joueur se réveille seul et isolé dans un environnement hostile et inconnu. Equinox apparaît comme un jeu de survie en environnement clos où le joueur se devra d'explorer le laboratoire dans lequel il se trouve afin de survivre à travers les zombies qui peuplent les lieux et tenter de trouver un chemin vers la surface pour s'extraire de ce cauchemar. Immergé dans une ambiance sombre et pesante, le joueur est régulièrement incité à se déplacer et à rester en mouvement, ce qui permet de le tenir constamment aux aguets car il explore des territoires inconnus.

Dans un monde dystopique où les humains ont tenté de résoudre la crise environnementale qui les auraient mené à leur fin, des scientifiques ayant découvert une nouvelle matière (le Neutronium) tentent de la maîtriser afin de créer une nouvelle énergie propre. Un incident dans l'isolement des expériences a mené à une massive fuite neutroactive (radiations de Neutronium) qui a mené à une fracture dimensionnelle engendrant une mutations chez les scientifiques et le protagoniste. Malgré tout, leurs destins ont été différents, alors que les scientifiques devinrent des mutants aggressifs, le protagoniste se retrouva dans une superposition dimensionnelle qui le fait intéragir de façon incomplète avec la réalité. Comme on peut l'apercevoir uniquement en multijoueur lorsque l'on regarde un autre joueur, celui-ci n'apparaît pas sous forme humaine mais bien comme une entité spectrale incomplète.

5.2.2 Site

Le projet a à sa disposition un site web unique où l'on peut visualiser l'avancement du projet, la présentation des membres ainsi que les aides à l'installation et à l'utilisation du jeu. On y retrouve également le lien permettant le téléchargement du jeu.

5.2.3 Mécaniques

Le jeu contiendra différents mécaniques qui permettront d'agrémenter le scénario et l'ambiance tels que :

L'Avatar : Protagoniste que nous contrôlerons à l'aide de la souris et du clavier, c'est à travers lui que nous explorerons l'environnement et intérragirons. Il sera caractérisé par différentes données telles que la vie et le nombre de munitions. Il pourra s'équiper d'armes trouvées sur la map mais pourra tirer uniquement s'il lui reste des munitions, munitions qu'il trouvera également sur la map. Si sa vie tombe à zéro, il meurt et le jeu se fini.

Objets : Afin de garantir sa survie, le protagoniste devra se déplacer avec un maximum de munitions qu'il trouvera sur la map afin de se défendre des zombies en les utilisant pour ces armes, armes qu'il trouvera principalement près de corps de scientifiques morts qui auront essayé de se défendre avec. Afin de poursuivre son exploration vers la liberté, il devra récupérer des badges d'accès sur la map qui lui permettront l'accès à de nouvelles zones du laboratoire.

Ennemis : Pendant sa progression dans l'environnement, le joueur fera face à l'adversité qu'offrent les ennemis sous forme de différents zombies étant anciennement des scientifiques travaillant dans ce laboratoire.

Tourelles : Les tourelles sont des structures offensives positionnées principalement devant les réacteurs. Reliées au système électrique central, elles peuvent continuellement tirer des décharges de plasma sur les zombies. Si le joueur se retrouve à court de munitions, il peut attirer les zombies qui le poursuivent jusqu'à portée d'une tourelle afin qu'elle les annihile pour lui.

5.2.4 Graphismes

Afin d'établir l'ambiance désirée pour ce jeu (à savoir une tension constante et pesante), il est nécessaire de spécialiser certains décors pour nos besoins. Nous allons donc centraliser les modèles que nous trouverons en accord avec le jeu sur certains sites tels que l'Unity Asset Store ou encore Sketchfab, puis les modifierons en respectant les conditions légales des licences attribuées afin qu'ils s'assimilent les uns aux autres.

Le jeu se déroulera dans un laboratoire souterrain. Il arborera une ambiance horrifique soutenue par une luminosité sombre et partielle émise par de rares sources de lumières telles que des lumières de secours ou des néons.

La map sera constituée de nombreuses salles, parmi lesquelles :

- **SpawnRoom** : Salle où le joueur apparaît pour sortir de sa cuve et là où il aperçoit le cadavre et son arme.
- **MainRoom** : Grande salle qui connecte deux étages par le biais d'un ascenseur.
- **GlobeRoom** : Salle qui contient deux tourelles en fonctionnement.
- **OrbeRoom** : Salle contenant un réacteur à neutronium complet, accessible uniquement depuis la GlobeRoom, c'est donc censé être un endroit relativement sécurisé sachant qu'il est gardé par deux tourelles.
- **ComputerRoom** : Salle exiguë offrant une visibilité médiocre, donc très dangereuse pour repérer les possibles ennemis qui se tapissent dans l'ombre.
- **GeneratorRoom** : Immense salle d'expérimentation sur les réacteurs à neutronium en cours d'amélioration.
- **UnderwaterReactorRoom** : Salle de création des réacteurs à neutronium, avec une structure métallique plongée dans une cuve.

5.3 Défis

Pour la conception de ce projet, le premier défi va être d'appréhender le fonctionnement d'Unity. Ce jeu étant notre premier projet sur ce logiciel, il va nous falloir en comprendre le fonctionnement et les subtilités, que ce soit placer des objets sur la map et les faire interagir correctement entre eux, importer les ressources que nous désirons ou encore synchroniser notre travail élaboré sur différents ordinateurs. L'implémentation du multijoueur semble également s'avérer complexe car il va falloir tenter d'en prévoir les implications avec les comportements standards, et ce dans le but d'avoir à réécrire le moins de code possible et que le tout s'ordonne rapidement. Enfin, la tâche qui paraît la plus complexe est celle du développement de l'IA des zombies. Il va falloir de nombreuses recherches afin d'en comprendre le fonctionnement, est-ce qu'on la crée à l'aide de réseaux de neurones comme on l'entend si souvent ou cela se passe-t-il autrement? Il va également falloir que l'IA soit malléable, à savoir que l'on puisse rapidement l'adapter en cas de problèmes à contourner ou encore de changement dans le contexte du jeu.

5.4 Outils & Technologies

Afin de mener à bien ce projet, nous aurons besoin de plusieurs technologies. L'une d'entre elle est évidemment l'application Unity qui nous servira de moteur graphique, environnement de développement, gestionnaire de ressources et compilateur de projet. Il nous faudra également tirer parti de l'API Mirror afin d'implémenter le multijoueur. Nous utiliserons également Github afin d'avoir des sauvegardes de notre travail en lieu sûr en cas de problèmes.

5.5 Besoins financiers

Nous allons utiliser les versions d'essais de certains Assets mentionnés en annexes, ce qui implique que nous aurions besoin de payer la licence en vue d'une commercialisation du jeu. Cependant, ces frais approcheraient la centaine d'euros et seraient vite rentabilisés par le prix d'achat du jeu.

6 Développement

6.1 Justine

6.1.1 Map 3D

Au commencement du projet, le jeu devait se dérouler dans un milieu montagneux et enneigé. N'ayant jamais travaillé sur Unity, il fallut que j'apprenne tout de zéro : comment placer un terrain, les textures, etc. J'avais fait quelques croquis aussi pour avoir les idées plus claires. Je commençais à faire la map à la main, car je ne connaissais aucune autre option. Je n'avais pas non plus installé d'assets me permettant d'avoir accès à plus d'outils tels que des brushes (motifs de textures). La map ressemblait à tout sauf à des montagnes : le mont principal était tout lisse, même s'il y avait du relief, on aurait dit que la map était plate. Après plusieurs essais, j'ai de nouveau recherché plus de tutoriels et j'ai trouvé différents assets de textures et de brushes. J'ai donc recommencé la map avec ces assets et cette dernière ressemblait beaucoup plus à un environnement enneigé avec pleins de montagnes. Le mont principal paraissait immense (ce qui était l'effet escompté) et tout semblait assez réaliste. À la suite de quelques problèmes de sauvegarde, j'ai essayé de placer le joueur sur la map pour avoir un aperçu de ce que cela pourrait donner en résultat final. La map était complètement hors d'échelle : le personnage était minuscule comparé au reste, on avait l'impression d'être une fourmi. Les montagnes étaient superbes de loin, or de près elles avaient des textures étranges qui étaient tout sauf réalistes. J'ai réessayé plusieurs fois de tout recommencer pour tenter d'améliorer l'ensemble or rien ne fonctionnait. Sachant qu'il n'y avait pas encore de couleurs et de flore, j'en avais rajouté en espérant que cela rende mieux. Mais tout était à nouveau hors d'échelle. Les arbres étaient tellement grands par rapport au joueur qu'ils paraissaient pixelisés. J'ai dû à nouveau repartir de zéro, mais cette fois j'ai bien fait attention à ne pas les faire trop grands ou disproportionnés. La map était enfin à la taille du joueur, que ce soit les arbres, l'herbe ou les montagnes. Avec cette étape de faite, il fallut maintenant rajouter quelques détails. En faisant quelques recherches sur le Unity Asset Store, je suis tombé sur différents assets

permettant de créer des rivières. Après en avoir importé un sur Unity, j'ai tenté de le mettre en place, or la rivière était toute rose ainsi que les textures de l'eau de l'asset. Ce problème était causé par le fait qu'on avait choisi de faire la map en URP (Universal Render Pipeline) et non en Built-In. J'avais vu avec Matthieu quelques tutoriels qui recommandaient l'URP car les projets étaient de meilleures qualités et que c'était plus simple de rajouter des effets grâce au Post-Processing. Le problème avec ce dernier, c'est qu'il faut des assets fait spécialement pour l'URP, sinon les textures et autres ne sont pas reconnues et deviennent inexistantes (représenté par la couleur rose vif sur Unity). Le projet passa donc de URP à Built-in pour des raisons de simplicité. Sauf qu'une fois en Built-In, les assets utilisés pour les arbres, les textures et autres étaient devenus transparents ; il fallut ainsi tout refaire à nouveau. Un autre problème conséquent était la taille de la map : on voulait une map très grande, or celle que j'avais était encore assez petite. Tout faire à la main prenait beaucoup de temps, sans oublier les crashes constant de Unity. C'est en faisant des recherches que j'ai découvert les Procedural World. Les Procedural Worlds génèrent des map de très grande taille assez rapidement (on peut les utiliser avec des assets ou avec du code). Après quelques essais, la map étaient très grandes, possédaient des monts ou chaines de montagnes assez réalistes, or quand on regardait de plus près la qualité était médiocre. Les textures paraissaient pixelisées ou irrégulières. De plus, il m'était impossible de placer différents Procedural Worlds et les relier. Il existe des assets permettant de relier différents terrains et des Procedural Worlds de manière harmonieuse, or ils coûtent assez cher. Ce qui est compliqué avec ces derniers, c'est qu'ils ne sont pas des terrains. Tous les outils permettant de rajouter du relief sont complètement inutiles. Ceci m'empêchait de surélever le mont principal et donc de le rendre plus imposant. Une map de très grande taille avec des forêts, de la neige, du brouillard, etc. nécessite de bonnes performances, mais mon PC portable ne possédait même pas de cartes graphiques. Il ne supportait pas du tout la taille de la map, ce qui empirait les crashes de Unity. Unity prenait aussi beaucoup de temps pour s'ouvrir (des fois 10 minutes, des fois 20 minutes) et cela avait beaucoup détérioré mon envie d'avancer sur le projet.

6.1.2 Menu : Ancienne version

Avant le changement de scénario, le menu était basé sur une image 2D représentant une montagne enneigée. Ce menu m'avait pris du temps à faire, car je n'avais jamais travaillé en 2D sur Unity, donc toute notion de Canvas m'était inconnue. Je ne savais même pas implémenter d'images personnalisées. C'est en faisant beaucoup de recherches que j'ai découvert les Sprites. Cette notion restait un peu floue, mais j'ai fini par prendre le coup de main. Savoir comment intégrer ses propres images sur Unity, c'est pratique, mais quand on manque d'inspiration, c'est tout de suite beaucoup plus compliqué. J'ai essayé de regarder sur des applications comme Pinterest des exemples de menus, mais je voulais garder une touche assez personnelle sur le menu. Je souhaitais que le menu représente notre jeu et mette en valeur l'ambiance de ce dernier. Après quelques essais en vain, je m'étais souvenu des croquis que j'avais dessiné, dont un où l'on voit en gros plan le mont principal. Cette image était parfaite pour le menu, car non seulement elle reste assez sobre, elle représente parfaitement l'environnement du jeu. À la suite de l'image de fond choisie, j'ai rajouté des détails tels qu'un cadre et des petits textes (la date, "searching for signal...", etc.). Pour animer le menu un peu et qu'il paraisse plus professionnel, j'avais décidé d'ajouter des animations. Comme le jeu prenait place dans un environnement montagnard et glacial, cela signifiait qu'il y avait peu de réseau, parce que les montagnes coupaient le joueur du reste du monde. L'antenne du laboratoire était du coup censée émettre un grésillement qui permettait de conduire le joueur jusqu'à ce dernier pour trouver du matériel et autre. Pour représenter ceci dans le menu, j'avais pensé à mettre une photo animée ressemblant à des bugs ou à des problèmes techniques (un peu comme ce qu'affichaient les télés des années 70-80 quand il n'y avait plus de réseau). Cette idée n'a pas porté ses fruits, car non seulement c'était compliqué de trouver l'effet escompté en GIF, mais il me restait encore à comprendre comment ajouter une GIF dans Unity. Pour importer les GIF que j'avais trouvé dans Unity, il fallut donc que je fasse des recherches sur différentes méthodes. Celle que j'avais trouvée nécessitait de découper mon GIF en ce qu'on appelle

un “Sprite Sheet”, soit les différentes frames composants notre GIF sont séparées et mises les unes à la suite des autres. J'avais utilisé le site Ezgif.com pour accomplir cette tâche. Une fois cela terminé, nous pouvons importer notre Sprite Sheet sur Unity pour ensuite faire en sorte qu'il s'anime. Pour recomposer notre image animée à partir du Sprite Sheet, on a besoin d'utiliser le 2D Sprite Editor. Mais après avoir essayé de lancer cet éditeur, j'ai eu un message d'erreur me disant que je ne possédais pas ce dernier. J'avais tout simplement dû redémarrer Unity et le problème était réglé. Pour ce qui est des boutons “PLAY”, “QUIT” et “OPTIONS”, j'avais supprimé sans faire attention l'EventSystem ce qui empêchait les boutons d'être fonctionnels. Une fois ce petit problème réglé, j'avais fait en sorte que quand on cliquait sur le bouton “OPTIONS”, ça nous ouvrait un Pop Up avec différentes options telles que le volume, la résolution, etc. Pour faire cela, j'avais créé un script qui détourne la réception des clics par le joueur et la renvoie vers une fonction définie dans le script. Ce script, à l'aide d'un booléen, peut déterminer s'il faut afficher ou cacher le menu des options. Pour rajouter une petite touche de professionnalisme, j'avais envie de mettre en place un Splash Screen. Sur Unity, quand on va dans les réglages du projet, il y a une option nous permettant d'implémenter un Splash Screen. Donc j'en avais mis un avec le logo d'Equinox.

6.1.3 Menu : Version actuelle

Depuis le changement de scénario, peu de choses ont changé sur le menu. En effet, les boutons restent les mêmes, ils ont juste été repositionnés. Or le cadre, les animations et l'image de fond ont été remplacés par une caméra donnant directement sur une des salles du jeu. Vu que c'est une caméra et non une image fixe, je n'ai pas eu à rajouter d'animations. J'ai aussi rajouté un champ de texte pour l'adresse IP afin de jouer en multijoueur.

6.1.4 Mini laboratoire

Dans le but de trouver des ressources nécessaires au joueur, ce dernier devait se rendre dans un petit laboratoire clandestin caché au sommet du mont principal. Pour rester dans une ambiance sombre, j'avais donc fait des recherches pour trouver de l'inspiration (principalement à l'aide de Pinterest). J'avais aussi fait des croquis du laboratoire pour ne pas m'y perdre. Le laboratoire devait être sur trois étages et était composé de huit salles. Parmi ces salles, on pouvait retrouver à l'étage 0 une entrée, une salle principale qui permettait d'accéder aux autres étages. À l'étage 1 il y avait une salle de recherche avec des ordinateurs, ainsi qu'une salle d'expérimentations. Et pour finir à l'étage -1 il y avait une salle de quarantaine ainsi qu'une salle de stockage et la pièce contenant la bombe à neutronique. Ce laboratoire clandestin était censé avoir une ambiance sombre et mystérieuse, avec des nuances de bleu canard et de rouge sanglant. Ces deux couleurs permettent de créer une ambiance stressante pour le joueur et garde ce dernier sur ses gardes. Plus la couleur rouge vif était présente et prenait le dessus sur le bleu, plus cela signifiait que l'on se rapprochait de zombies. Le bleu canard est souvent utilisé dans des endroits futuristes, il représente une couleur avant-gardiste et intimidante. Cette couleur était parfaite pour représenter ce laboratoire clandestin futuriste. Pour avoir une atmosphère chaotique, il fallut que le laboratoire soit en bazar. Des bureaux de travers, du verre brisé, des feuilles éparpillées sur le sol, des objets renversés... ; tout devait donner une impression que l'apocalypse avait débutée ici. J'avais longuement procrastiné la construction du laboratoire, car j'avais principalement travaillé sur une map en extérieur, je ne voyais pas comment je pouvais construire une map en intérieur. Je n'arrivais pas non plus, malgré des heures de recherches, à trouver un tutoriel là-dessus. Or, par chance, j'avais réussi à trouver un tutoriel sur Snaps. Avant le changement de scénario, j'avais seulement complété l'entrée. Il y avait des câbles partout, un petit générateur ainsi qu'un grand écran avec diverses données et une grande porte épaisse. Il n'y avait pas encore de lumière rouge, mais l'ambiance restait assez pesante avec la présence du bleu sarcelle.

6.1.5 Snaps : Probuilder & ProGrid

J'ai mentionné Snaps dans la partie précédente, or je ne l'ai pas expliqué. Snaps représente deux fonctions de Unity rassemblée : ProGrid et ProBuilder. Ces deux derniers permettent de rendre la création de map intérieur plus simple. En effet, ProGrid permet de faire en sorte que des objets s'agrippent à la map, ce qui assure une certaine précision et qualité de travail. ProBuilder permet de modéliser de simples objets en 3D tels que des cubes, escaliers ou autres. Pour la construction de la map, je me suis principalement focalisée sur ProGrid car je possédais déjà des assets high-poly. Snaps m'a beaucoup aidé dans mes débuts puisque je n'arrivais pas tout le temps à bien aligner les murs ou les objets n'étaient pas toujours bien placés, etc. Or, suite à quelques problèmes mentionnés plus tard, j'ai cessé d'utiliser Snaps.

6.1.6 Map laboratoire

En commençant l'entrée du laboratoire, je m'étais rendu compte de la simplicité des maps en intérieur, surtout à l'aide de Snaps. Comme la map extérieure était trop compliquée à gérer et que cela prenait énormément de temps, tandis que celle en intérieur était bien plus simple et rapide à réaliser. Sachant qu'il fallait pour la deuxième soutenance un jeu qui fonctionne du début jusqu'à la fin, il ne me restait pas assez de temps pour finir le milieu montagnard enneigé. C'est pour cela qu'on a décidé de changer de scénario, et donc de rester uniquement sur le laboratoire.

Par rapport à l'idée que j'avais pour l'ancien modèle du laboratoire, que très peu de choses ont changé : l'ambiance reste la même avec le mariage entre le rouge sanglant et le bleu sarcelle ; l'entrée devient une simple salle de stockage/un cul-de-sac ; la taille du laboratoire augmente et de ce fait de nouvelles salles et un étage se sont ajoutés. Lors de la conception de cette nouvelle map, plusieurs problèmes sont intervenus (que ce soit au niveau de l'inspiration ou de la construction).

Sachant que le laboratoire a changé, il fallut assigner à chaque étage une certaines fonctions pour que cela se rapproche le plus possible d'un vrai laboratoire. C'est pour cela que l'étage -1 est composé de salles d'expérimentation et de recherche (ainsi qu'une salle de réacteur). À l'étage 0, principalement utilisé comme étage pour des robots ou du stockage, on retrouve la salle principale avec laquelle on accède aux étages 1 et -1, ainsi que des salles de stockage et de fabrication de robots. L'étage 1 est assez petit, il ne contient qu'une salle de serveurs et un énorme ascenseur conduisant à l'étage 2 (à cet étage, il y a aussi un petit passage secret).

Pour finir, l'étage 2 est dédié à la relaxation des employés/scientifiques ainsi que des vestiaires et toilettes. Dans certains étages, il y a des signes avec de divers secteurs parmi lesquels on retrouve : le secteur 45 qui représente la salle principale, le secteur 27 qui est composé des salles de réacteur et pour finir les secteurs non autorisés (d'autres peuvent être rajoutés entre-temps et seront décrits à la soutenance finale). Quelques petits problèmes ont été causés par Snaps et c'est pour cela que je ne l'utilise plus. En utilisant ce dernier, les murs ne sont pas forcément rattachés au sol ou au plafond, ce qui détruit l'ambiance sombre du laboratoire. Snaps ne m'a pas été très utile lors de salle en diagonale non plus, car les différentes dalles ne veulent pas se relier. Dès que j'avais ces problèmes, il ne me suffisait que d'enlever l'option ProGrid et je plaçais manuellement les objets qui posaient un problème, mais cela me faisait perdre un peu de temps. Pour finir, le nouveau laboratoire possède en tout quatre étages et une quinzaine de salles.

6.1.7 Salle du Réacteur principal

La salle du réacteur a été une des seules salles où je ne me suis pas inspirée de images préexistantes. En effet, dans d'autres salles, j'avais de trop grandes attentes, car je pensais réussir à recréer ce que j'avais vu en ligne. Je voulais que cette salle soit complètement unique et imposante. Sa taille lui permet d'avoir des plateformes à trois différentes hauteurs, de manière à mieux étudier l'orbe en construction. Je voulais que la première chose que l'on voit en entrant soit cet orbe en suspension, entre les deux générateurs d'énergie. Pour montrer la quantité d'énergie requise pour charger l'orbe, il y a des petits faisceaux de lumières ainsi que des éclairs entre les deux générateurs. Je désirais que cette pièce donne l'impression d'être un "bazar organisé", c'est-à-dire que cet orbre fait l'objet de recherches en cours. C'est pour cela que les tableaux ont des calculs et annotations un peu dans tous les sens, les câbles par terre sont emmêlés, etc. Les ordinateurs sont tous tournés vers les générateurs et l'orbe vu que les scientifiques l'étudiaient avant que l'apocalypse se produise. J'ai aussi rajouté sur l'orbe un shader qui lui donne un effet instable, ainsi qu'un audio de chargement magnétique sur les générateurs. Si on écoute bien le bruit, on peut entendre qu'au début tout se passe bien, or à la fin les générateurs se déchargent d'un coup, comme s'ils manquaient de puissance. Avec cet effet ainsi que le manque d'organisation dans la salle, on peut en déduire en tant que joueur que l'orbe régnant au centre ne sera jamais stable. Pour rajouter un peu de vie en plus de l'orbe qui tourne sur elle-même, j'ai mis quelques animations (fumée, etc) ainsi que des tuyaux de refroidissement au sol et au plafond. Ces derniers devaient être juste bleu canard, or j'ai découvert qu'en mettant le shader de l'eau sur ces tuyaux, on a l'impression qu'ils contiennent réellement un liquide de couleur tape-à-l'œil. Je souhaitais vraiment que cette salle marque le joueur, mais ne soit pas qu'une simple décoration sans lien avec l'histoire.

6.1.8 Salle du Réacteur immergé

L'inspiration de cette salle m'est venue en regardant une vidéo d'un réacteur nucléaire émettant des radiations de Tcherenkov. Ces réacteurs sont immergés sous l'eau et produisent une lumière bleue canard éblouissante. C'est donc ce que j'ai essayé de reproduire dans cette salle. Quand on entre, la première chose que l'on remarque est un réacteur produisant une lumière quasi aveuglante dans une cuve d'eau. J'ai fait en sorte que l'on puisse regarder le réacteur derrière une baie vitrée assez épaisse, ou depuis des plateformes. Si l'on regarde autour, on peut remarquer plusieurs capsules énergétiques ainsi que des tuyaux dans la cuve d'eau. Ces tuyaux sont une grande source d'énergie pour le reste du laboratoire et vont alors dans d'autres salles.

J'ai varié les dalles au niveau du sol pour donner une impression que la salle est assez remplie, or si on regarde d'un peu plus près, il n'y a que très peu de meubles. Je voulais qu'on se concentre principalement sur le réacteur et non sur son entourage. Mon but dans cette salle est d'attiser la curiosité du joueur et de le faire monter sur la plateforme pour qu'il admire le réacteur et remarque certains détails tels que les tuyaux. Quand le joueur fixe le réacteur, on ne voit que la couleur bleu canard qui nous fait presque oublier qu'on est sur le point de se faire attaquer par des zombies. Une fois que le joueur souhaite quitter la salle et qu'il se tourne vers la porte, il verra quelques reflets bleus sur les murs, or la couleur rouge prend le dessus. Je souhaitais donner au joueur une sensation d'angoisse. Cela pourra peut-être donner à certains l'envie de rester indéfiniment dans cette salle, or elle reste un cul-de-sac avec une seule issue possible...

6.1.9 Salle principale

La salle principale est une des salles les plus importantes du laboratoire. Elle est assez spacieuse et est sur trois étages. Au début, j'avais comme idée pour cette salle de la faire plate. Je ne souhaitais pas avoir la vue sur l'étage d'en dessous, et je comptais la faire beaucoup plus massive. Or j'avais peur que le joueur se perdre dedans et qu'il ne passe trop de temps dedans. C'est pour cela que sa taille est plus modeste qu'avant, or elle est imposante. C'est une des seules salles où l'on voit différents étages, ce qui donne une impression au joueur de grandeur. Elle est le point d'accès entre l'étage -1, l'étage 0 et l'étage 1. Elle met en place l'ambiance et montre un petit aperçu de ce que l'on peut trouver sur les différents étages. En effet, quand on est à l'étage -1 et qu'on regarde autour, on peut apercevoir des pods, tandis qu'à l'étage 0 on remarque plus des robots ainsi que des ordinateurs, et pour finir, l'étage 1 est rempli d'ordinateurs. La palette de couleur reste la même dans cette salle que dans le reste du laboratoire, or il y a quelques petites touches de rouges qui rappellent l'ambiance glauque du jeu. Les murs sont détaillés, avec des morceaux métalliques ou des ventilations ; de même pour le sol où l'on peut voir à travers certaines dalles. Elle est un peu en bazar, ce qui garde l'aspect scientifique et chaotique qu'on pourrait avoir d'un laboratoire clandestin. Or, une fois meublée, je trouvais que la salle manquait de vie et qu'elle paraissait un peu lisse, sans relief. J'ai donc ajouté des animations (d'électricité, de fumée...), des Points Lights à différents endroits et des Spotlights. Pour donner du volume à la salle, j'ai rajouté un petit chemin suspendu (voir Passage secret) et j'ai mis des Spotlights juste en dessous. Cela permet non seulement à certains détails d'être plus visibles, mais rajoute aussi de la perspective. Vu que la salle est sur trois étages, on peut entendre l'alarme d'urgence du laboratoire résonner, ce qui glace le sang.

6.1.10 Passage secret

Ce que j'adore quand je joue à des jeux vidéos, ce sont des salles secrètes ou des petits passages cachés. Cela donne au joueur une envie de fouiller et de découvrir le reste du laboratoire. C'est pour cela que j'ai implémenté un passage secret dans le jeu. Lorsque l'on arrive dans la salle des serveurs au premier étage, si l'on sort du chemin principal, on peut remarquer deux petits tunnels verticaux. Ces tunnels nous mènent vers une salle avec des extincteurs, au cas où il y aurait un problème dans la salle des serveurs. Si l'on fouille un peu, on peut apercevoir un mur abîmé qui nous amène dans une salle assez tranquille. Cette pièce (qui a été aménagée par quelques scientifiques un peu flemmards voulant regarder la télévision) est assez sale. Il y a des déchets par terre, le canapé est vieux, des chaises de bureaux volées ont atterri ici par "coïncidences"... Je souhaitais créer un petit espace détente, créé par des employés qui ont découvert ce petit endroit par accident. La palette de couleur dans cette petite salle est assez chaleureuse, il y a plus de couleurs chaudes que de froides. Cet endroit est censé inspirer confiance avec les couleurs jaunes et rouges, or la couleur bleue reste omniprésente. Le joueur se sent plus détendu, mais reste quand même un peu sur ses gardes à cause de la présence de la couleur bleue. J'ai rencontré un problème en travaillant sur cette salle : on ne pouvait pas en sortir.

En effet, dans le jeu l'option d'escalader ou de grimper est inexistante. J'ai réglé cela en faisant un passage qui permet au joueur de retourner dans la salle des serveurs. Lorsque l'on emprunte ce passage, on commence par monter des escaliers et par atterrir de nouveau dans un endroit assez détendu et tranquille. On continue en passant par une salle de travail et de recherche tout en bazar avec une certaine quantité d'ordinateurs. En poursuivant le chemin, on voit en étant dans un petit couloir suspendu une grande salle (de la même taille que celle du réacteur) remplie d'ordinateurs. Cette pièce est dédiée à la recherche, avec un îlot central possédant quatre grands écrans et quelques terminaux. Ces deux salles sont inaccessibles au joueur, mais permet à celui-ci de voir que le

laboratoire est gigantesque. Elles sont remplies de cadavres et de zombies errants. En étant toujours dans ce petit couloir suspendu, on arrive dans la salle principale que l'on peut observer de haut. Cela nous permet de voir les différents étages auxquels on peut accéder au travers de cette pièce. Pour finir, on se retrouve de nouveau dans la salle des serveurs. Ce passage n'est pas forcément utile par rapport au synopsis du jeu, or elle donne au joueur l'impression d'être une fourmi dans un bac à sable.

6.1.11 Lounge

Le lounge représente tout l'étage 2. Ce dernier est dédié à la détente des employés et scientifiques, et possède donc des petits cafés, des coins détentes avec des canapés et fauteuils confortables, etc. L'ambiance dans cet endroit est complètement différente par rapport aux autres étages. La palette de couleur est beaucoup plus claire et chaleureuse : les murs sont blancs, les lumières émettent une couleur jaune assez agréable aux yeux, les distributeurs de boissons font même apparaître un petit arc-en-ciel sur les murs avec leurs couleurs variées. Les restaurants et cafés disponibles vendent des plats appétissants, l'étage possède aussi de la verdure ainsi que des fenêtres avec une vue sur la ville... Tout est censé paraître idyllique et reposant. C'est pour cela que c'est la première salle à travers tout le laboratoire à avoir des fenêtres, des plantes et de la lumière naturelle. Je voulais qu'une fois dans cette salle, on oublie en quelque sorte le reste du laboratoire. Plus de murs métalliques foncés ou des conteneurs encombrants. C'est aussi à cet étage que l'on retrouve les vestiaires et toilettes pour hommes et femmes, permettant aux employés de mettre leurs combinaisons de travail. Or cet aspect assez paradisiaque est teinté de la couleur rouge au sol et sur les murs... Cet étage est rempli de cadavres. Ce mélange entre endroit chaleureux et cadavres gisants aux pieds du joueur est censé donner une impression psychédélique, on se croirait dans une utopie cauchemar. C'est aussi dans cette salle que l'on trouve un balcon. Le seul balcon du jeu qui représente l'endroit le plus important. C'est en allant sur ce balcon que le jeu prend fin.

6.1.12 Balancement des armes & HeadBob

Les armes et leur fonctionnement ne faisaient pas parties de mes tâches à compléter, or j'ai dû les faire après le départ d'un membre de l'équipe. Après avoir mis en place l'arme devant la caméra pour qu'on ait l'impression que le joueur la tienne, cette dernière paraissait statique par rapport au joueur. J'ai donc regardé plusieurs tutoriels qui donnaient quelques conseils sur comment donner de l'âme à un RPG. Parmi les nombreux conseils, deux sont revenus très souvent : le Head Bob et le Sway. Le Head Bob est très couramment utilisé dans les RPG ayant une vue en première personne, car il permet au joueur de se projeter. Pour faire cela, j'ai commencé par exposer les paramètres de vitesse et d'amplitude (avec SerializedField). Je vérifie que le joueur est bien sur le sol sinon le Head Bob ne sera pas activé. Pour que le Head Bob se calcule, il faut lire les touches du joueur (s'il court, s'il marche, s'il est fixe...). Si les valeurs d'entrée dépassent un certain seuil, on applique des modifications sur l'axe des ordonnées de la position de la caméra, tout en suivant un modèle sinusoïdal afin de simuler une oscillation verticale. Cela donne l'impression que le personnage marche et avance réellement au lieu de se déplacer de façon inhumaine.

Pour ce qui est du Sway (ou balancement de l'arme), cela donne l'impression que l'arme est assez lourde, ce qui fait très réaliste. Tout comme le Head Bob, il faut lire les touches du joueur et vérifier que leur valeur d'entrée dépasse une certaine limite. Si c'est le cas, il faut calculer une nouvelle position dépendante de la précédente tout en suivant une interpolation linéaire (Lerp) ou sphérique (Slerp). L'interpolation linéaire permet de changer la position tandis que la sphérique change la rotation. Pour continuer sur les armes, j'avais aussi vu la notion de Recoil (le fait que l'arme recule quand on tire), sauf que je l'avais passé à Baptiste afin de mieux répartir le travail entre nous.

6.1.13 Ambiance sonore

Sachant que le jeu se déroule dans un laboratoire, il a fallu trouver des effets sonores assez futuristes et effrayants. La première idée d'effets sonores que j'ai eue, c'était de mettre une alarme urgeant les scientifiques à évacuer le laboratoire. Il y a un autre système d'alarme devant une des salles du réacteur indiquant que le secteur du réacteur est fermé. Ces petites touches enrichissent le réalisme du jeu et ajoutent aussi à l'environnement du chaos et du stress. Cela permet au joueur de mieux s'immerger dans le jeu. Pour faire en sorte que l'atmosphère ne perde pas son côté futuriste, j'ai mis en place sur certains objets (tels que des ordinateurs, des terminaux...) des bruitages. J'ai aussi rajouté sur certaines animations des effets, comme sur l'animation de soudure où j'ai mis un bruit d'électricité. Certains de ces effets sonores n'étaient pas faciles à trouver et il a fallu que j'écoute des vingtaines de sons dans certains cas juste pour trouver celui qui irait le mieux. De plus, dans d'autre cas, j'ai dû faire des modifications à l'audio, par exemple en ajoutant quelques secondes de silence ou le mettre en "reverb" (effet d'écho). Pour que certaines salles paraissent plus vivantes, j'ai mis des petits effets sonores assez discrets : pour le passage secret et le lounge, sur les téléviseurs, j'ai mis un audio de personnes en train de discuter ou de débattre, sur les distributeurs de boissons, j'ai rajouté un bourdonnement... Pour ce qui est de l'orbe de neutronium, j'ai mis un son de trou noir pour qu'on s'aperçoive à quel point cette dernière est mystérieuse, mais puissante. Pour finir, le plus dur a été de choisir la musique de fond. Je ne voulais pas mettre de mélodies, de peur qu'on ne puisse plus entendre les autres bruitages par derrière. Je désirais qu'il reste assez discret pour que notre attention ne soit pas détournée du jeu, mais qu'il soit quand même présent. J'ai réussi à trouver un son de fond où l'on entend la ventilation ainsi que des tintements métalliques.

6.1.14 Ambiance visuelle

Pour accentuer l'ambiance sombre et pesante du laboratoire, la palette de couleur tourne principalement autour du noir et gris foncé. Or, cela n'était pas suffisant pour vraiment donner ce sentiment de stress au joueur. De plus, certaines scènes paraissaient un peu fades et manquaient de réalisme. J'ai alors essayé de rajouter quelques Points Lights et des animations (de la fumée...) pour rajouter un peu de vie, mais ce n'était toujours pas suffisant. Puis j'ai regardé plusieurs vidéos à propos de l'accentuation du réalisme d'un jeu sur Unity, et parmi les conseils, il y avait : ajouter des petits détails tout en restant sobre et le post-processing. J'ai rajouté des petits objets ou des détails qui semblaient avoir leur place (des boîtes contenant des capsules, des feuilles par terre, des tasses de café sur les bureaux...). Or, pour le Post Processing, je n'en avais jamais entendu parler et je n'avais pas la moindre idée du monde de comment cela fonctionnait. Après plusieurs tutoriels, j'ai compris que je devais rajouter une couche (ou un "layer") "Post Processing". Puis il fallut rajouter sur la caméra un "Post Processing Layer" et implémenter dans la scène un "Post Processing Volume" dans un GameObject vide. Par la suite avoir mis ce volume en Global, j'ai créé un nouveau profil de Post Processing (qui me sert à garder les réglages bien spécifiques de ce profil-là si jamais je crée un autre profil). À la suite de tout cela, on peut enfin ajouter des effets. J'ai choisi beaucoup d'effets tels que Vignette, qui assombrit les bords de l'écran, ou encore Bloom qui donne vie aux diverses lumières dans une scène. J'ai mis Grain aussi qui permet de rajouter un peu de texture, ainsi de l'Anti-Aliasing qui règle certains problèmes dû aux lumières ou à la qualité de certains assets. L'effet le plus impactant est l'Ambiant Occlusion, qui assombrit la scène et empêche le joueur de voir au loin. Ces effets rendent le jeu vivant et terrifiant. Sur certains objets, j'ai eu recours à des Shaders : l'eau dans la salle "Underwater Reactor", l'orbe et les éclairs dans la "Reactor Room". Le shader de l'eau m'a causé quelques problèmes, car j'ai découvert qu'on ne pouvait pas superposer plusieurs shaders. En effet, sur le générateur il devait y avoir des éclairs, or l'eau a pris le dessus. C'est pour cela que les éclairs ont été déplacés dans l'autre salle du réacteur.

6.2 Matthieu

6.2.1 Avatar 3ème personne

En accord avec l'ébauche initiale du cahier des charges, la map aurait été un environnement extérieur. Afin de tirer parti du grand volume qu'offre un espace ouvert, j'ai opté pour un avatar en 3ème personne, ce qui implique de pouvoir observer un avatar humanoïde en mouvement et d'avoir un point de vue majoritairement externe.

Étant à l'aube du projet, nous n'avions pas encore obtenu une modélisation spécifiques et j'ai dû opter pour le modèle fourni dans l'asset Third Person Character Controller fourni gratuitement par Unity. De plus, cet asset proposait également une base de code pour des mouvements et déplacements dont je me suis inspiré, on y retrouvait une implémentation basique des déplacement, des mouvements caméras et de l'influence gravitationnelle. J'ai dû ensuite modifier ces comportements pour amplifier le réalisme que l'on désirait obtenir, à savoir empêcher l'avatar de gravir des surfaces bien trop abruptes, adapter le saut à l'échelle du terrain, renforcer la détection du sol rendue obsolète par le terrain accidenté comportant nombre de petits trous qui n'étaient précédemment pas détectés et causaient de nombreux bugs, mais également adapter le code initial pour une transition vers le nouveau système d'entrées utilisateur natif d'Unity.



6.2.2 Gestion des entrées utilisateur

Introduit récemment dans Unity, le "New Input System" succède au système standard. Pour permettre de récupérer les interactions de l'utilisateur avec le jeu, il fallait précédemment remplir plusieurs conditions qu'il était difficile de synchroniser à travers le code. La première chose à connaître était le "KeyCode" de la touche du clavier dont on voulais inspecter l'état, il fallait ensuite tenter de faire correspondre son état actuel avec les différents états connus tels que 'enfoncé', 'appuyé' et 'relaché' par exemple, qui correspondent respectivement à l'appui sur une touche, son maintien enfoncé puis sa libération. Le concept est ainsi le même pour les interactions avec les différents types de périphériques comme un clavier, une souris ou encore une manette de console.

Cependant, la déficience majeure de ce système était de maintenir la synchronisation à travers le jeu en fonction de la situation. Par exemple, si l'on paramètre la touche 'E' pour contrôler l'accélération dans un véhicule et pour récupérer des objets hors d'un véhicule, il faut ajouter de nombreuses variables de contrôle permettant de savoir quelle action effectuer en fonction de la situation, variables qu'il faut donc garder à jour et sans qu'elles s'opposent les unes aux autres. Introduire ce nouveau système a donc permis de résoudre tous ces défauts et a également permis une abstraction bien utile qui est le changement de sets d'actions, à savoir regrouper les différentes actions correspondantes à un environnement spécifique et permettre d'intervertir rapidement ces différents contrôles si besoin est. Toutes ces actions étant désormais regroupées au même endroit dans ce que l'on appelle des "Action Maps" qui contiennent des "Input Actions", on peut désormais facilement surveiller les interactions depuis un composant accessible globalement nommé "Player Input" où l'on peut effectuer ces fameux changements d>Action Maps et récupérer les valeurs actuelles pour chaque interaction. L'avantage secondaire est également de pouvoir facilement et rapidement changer les associations entre touches et interactions auxquelles elles sont liées, et ce depuis une interface graphique intuitive.

6.2.3 Réglages caméras

Afin d'obtenir un rendu visuel et donc de permettre au joueur de se déplacer consciemment dans l'environnement et de s'immerger dans le jeu, il nous faut placer des caméras dans le jeu. Notre avatar étant un objet qui se déplace, il faut que le rendu visuel le suive constamment. Ce système peut être implémenté de plusieurs manières, soit de façon standard avec le composant natif "Camera", soit en utilisant le package Cinemachine. Les caméras standard se lient sur un affichage (un moniteur par exemple) et cet affichage ne peut alors rendre qu'un unique aperçu, ainsi il faut implémenter un système de changement de caméras qui va maladroitement activer ou désactiver certaines caméras si l'on choisit cette option. En plus d'être grossièrement bricolée, cette option est loin d'être optimisée car ils faut soit rechercher dans tous les objets du jeu spécifiquement la caméra que l'on veut modifier, soit réunir les références de toutes les caméras dans un même script et leur appliquer des attributs pour les différencier.

Après m'être rendu compte de tous ces défauts à gérer je me suis décidé à travailler avec le package Cinemachine qui me permet de placer autant de caméras que je le désire. Pour définir celle qui projettera son rendu sur le périphérique, je n'ai alors qu'à modifier un paramètre de priorité qui privilégie celle avec la plus haute valeur. Je tire aussi parti du chargement et déchargement de scène natif d'Unity qui permet de détruire la caméra associée à la scène modifiée et de possiblement basculer sur son affichage selon sa priorité. Pour faire fonctionner ce système, il suffit d'ajouter un composant 'Cinemachine Brain' sur l'unique caméra standard du jeu, puis de modifier le paramètre priorité des caméras cinemachine depuis le code en récupérant ces composants. Ce changement de caméras implicite pourrait également permettre d'implémenter facilement certaines cinématiques car ce basculement se fait de façon animé, avec un effet de glissement paramétrable entre la caméra anciennement prioritaire et la nouvelle.

6.2.4 Animations de déplacement avatar

Comme mentionné précédemment, je me suis servi de l'avatar fourni dans l'asset Third Person Character Controller. Étant un prototype servant de base, il est construit de façon à être malléable mais n'est pas spécialisé pour nos besoins, j'ai donc dû le modifier afin que le rendu visuel soit en accord avec le jeu. Premièrement, j'ai reparamétrisé les animations de déplacements afin qu'elle paraissent plus naturelles et réalistes. Pour ce faire, j'ai construit la base de ces animations sur certaines animations principales dont le 'sur place' (idle), la marche et la course. Ces animations se lient aux positions de certains membres afin de pouvoir être réutilisables pour tous les humanoïdes, impliquant que j'aurais également pu les utiliser sur les zombies ou encore des scientifiques (animation retargeting). Ayant ces animations à disposition, il a ensuite fallu leur appliquer un poids qui dépend de la vitesse de l'avatar définit par son script de contrôle. Ce poids permet d'influer sur la juxtaposition entre ces animations et permet d'avoir un mouvement fluide et naturel du passage de la marche au sprint et inversement par exemple.

Il a également fallu que je change le code lié aux déplacements car lorsque l'on se déplaçait vers l'avant, l'arrière, la gauche ou la droite les animations agissaient comme je le souhaitais, mais lorsque l'on se déplaçait diagonalement on pouvait alors observer un accroissement indésirable dans la vitesse de déplacement de l'avatar. Après avoir inutilement changé les paramètres des animations plusieurs fois, je me suis rendu compte que le problème résidait dans la gestion des entrées utilisateurs dans la fonction appliquant le mouvement. Après avoir acquis ces données, le traitement se servait de la magnitude du vecteur déplacement, causant l'augmentation de la valeur finale, mais cette magnitude devait être calculée après avoir normalisé ce vecteur afin de lisser les valeurs d'entrée et obtenir un mouvement fiable

6.2.5 IA zombies

Notre jeu se base sur une ambiance horrifique et pesante pour construire son intérêt et attirer les joueurs, il a donc besoin d'ennemis au comportement réaliste. C'est pourquoi j'ai effectué de nombreuses recherches pour premièrement comprendre le fonctionnement d'une IA ennemie sur Unity mais également comment la perfectionner au niveau des performances tout en la rendant la plus réaliste possible.

À l'heure actuelle, l'IA est complétée et gère toutes les situations auxquelles le joueur peut être confronté. Un ennemi peut repérer un joueur s'il se trouve à une certaine distance et un angle précis depuis la face avant d'un ennemi, permettant de simuler un champs de vision réaliste. Une fois détecté, le joueur va commencer à être poursuivi et ce jusqu'à ce qu'il soit à portée d'attaque où qu'il disparaîsse du champs de vision de l'ennemi et qu'un certain laps de temps simulant une mémoire soit écoulé. Si le joueur est atteint, le zombie va alors s'arrêter sur place et lancer une nouvelle animation d'attaque. Lorsque cette animation est lancée, les bras du zombie vont commencer à s'agiter pour frapper le joueur, et lorsque ses bras seront en position quasi-horizontale donc quand la portée de l'attaque sera maximale, l'animation va appeler une fonction de l'IA à ce moment précis qui va infliger des dégâts au système de santé du joueur.

Une autre partie plus poussée de l'IA consiste à conférer au zombie une sorte de sixième sens qui lui permet de détecter un joueur se trouvant hors de son champs de vision (son dos par exemple) s'il est assez proche, mais également permettre d'avertir d'autres zombies à proximité.

L'IA a été poussée dans son optimisation, à savoir paramétrier à volonté la fréquence de recherche (toutes les 200 millisecondes au lieu de toutes les frames par exemple) tout en gardant un aspect très réaliste pour le joueur.

6.2.6 Système de tir

Pendant sa progression, le joueur va devoir faire face à de nombreux ennemis. Or, comme mentionné précédemment, il se trouve dans une superposition dimensionnelle instable et ne possède pas de corps réel mais plutôt une forme spectrale. Il nous est donc impossible d'implémenter une mécanique de combat de mêlée pour le joueur dans laquelle il pourrait se servir de ses mains pour donner des coups par exemple. C'est pourquoi j'ai décidé de ne lui permettre que le combat à distance en se servant d'armes à feu.

Pour fonctionner, ce système de tir doit récupérer le point central sur l'écran, une fois ce vecteur acquis, s'ensuit un Raycast qui part de la caméra du rendu actuel et se dirige vers ce point central en restant dans les limites d'une distance prédefinie. Ce Raycast va alors récupérer les informations du premier obstacle qu'il rencontrera. Je fais alors en sorte d'analyser les informations de cette cible afin de déterminer si c'est une cible valide. Si l'informations est nulle alors le tir est invalide, sinon il faut essayer de récupérer un composant HealthSystem depuis l'obstacle rencontré. Si l'on parvient à récupérer ce composant, cela signifie que l'obstacle rencontré est bien une cible qu'il est possible de blesser. Si ce composant existe, alors on peut instancier une giclée de sang au point d'impact puis diminuer la santé sur le HealthSystem récupéré, et éventuellement déclencher la mort de la cible si son système est corrompu. Une fois le tir effectué, il faut alerter toutes les IA ennemis de la présence du joueur dans un certain périmètre afin de simuler les ennemis détectant le son. Il m'a également fallu optimiser ce système, car lorsqu'on se trouve proche d'une horde de plusieurs centaines de zombies, ce système d'alerte est bien trop coûteux en temps et ne peut finir son traitement en une unique frame, faisant ainsi chuter drastiquement les FPS. Afin d'optimiser ce mécanisme, j'ai tiré parti des Coroutines, que l'on peut vulgariser par des Thread allégés qui permettent de simuler une exécution de code en parallèle. Désormais, ce système traite uniquement l'alerte de 25 zombies par frame (paramètre variable) et s'occupe du reste au fur et à mesure.

6.2.7 Drone animations

Lors de la conception du drone qui devait servir d'équipement de reconnaissance et de leurre lors de la première phase du jeu (avec le monde extérieur), j'ai rencontré de nombreux problèmes quant à l'animation du modèle de drone choisi sur Sketchfab. Ce modèle contient une animation unique contenant le décollage et le survol. Afin d'avoir un comportement réaliste, il me fallait pouvoir contrôller séparément ces 2 animations, avec lesquelles j'aurais pu créer l'animation d'atterissage en inversant celle pour le décollage. Or cette animation n'était pas dans formatée spécifiquement pour Unity et aucun des outils de découpage d'animations ne pouvait y faire. La première chose que j'ai tenté a été de copier les courbes d'animations qui m'intéressaient afin de les coller dans une nouvelle animation vierge, mais c'est là que mon calvaire a commencé. Ce modèle est moyennement complexe mais contient déjà un bon nombre d'objets indépendants de la structure principale, ainsi les courbes d'animations doivent se référer à chacun de ces objets afin de déterminer leur emplacement précis à un certain moment de ladite animation. Cet important nombre de données a été le point de chute, car lorsque j'essayais de copier l'animation de base, au moment de la coller sur la nouvelle je me suis aperçu en lançant la partie qui m'intéressait que les données étaient corrompues à certains endroits, ce qui créait des effets particulièrement indésirables comme des pâles qui sortent de leurs pivot ou qui vibrent en se déployant par exemple. Après m'être heurté à ce problème, j'ai tenté une autre option qui a été d'ouvrir le modèle du drone dans son logiciel de développement (à savoir Blender) afin de pouvoir recouper les animations avant l'export au format Unity. Malheureusement, avec cette méthode les textures et matérieux qui comptaient l'aspect visuel du drone se superposaient à certains endroits, impliquant sans doute que l'environnement de base de Blender manquait certaines librairies que la personne développant le drone avait dû utiliser. Après ces nombreux échecs et un temps conséquent de perdu, je me suis résolu à rectifier les données corrompues à la main, et bien que cela m'ait pris plusieurs heures et une intense concentration, j'y suis finalement parvenu, même si à l'heure actuelle ces animations ne sont plus utilisées.

6.2.8 Déplacements drone

Afin d'introduire ce concept de drone de reconnaissance, il m'a fallu implémenter des contrôles différents de ceux utilisés pour l'avatar. Tandis que l'avatar n'a besoin de connaître que la direction pour se déplacer puis son système de gravité s'occupera de gérer le dénivelé, le drone quant à lui doit inclure la hauteur et son altitude précédente dans son calcul de trajectoire.

Pour ce faire j'ai donc dû commencer par lui confectionner un ensemble d'interactions dans une Action Map afin de la différencier de celle de l'avatar, car si le drone est déployé et qu'on le contrôle ça n'a alors aucun sens que l'avatar se déplace également. Une fois fait, il m'a fallu coordonné le changement de contrôles de l'avatar vers le drone lors du déploiement puis inversement lors du rapatriement. J'ai me suis également servi du système des Cinemachine pour le transfert de la caméra de l'avatar à celle du drone qui est positionnée au niveau de son objectif pour plus de réalisme. Il a également fallu que j'impose des limites dans la rotation de l'objectif physique situé sous le drone et que j'implémente le mouvement de ce même objectif avec ses axes se déplaçant sur 2 objets différents servant de pivot. J'avais également mis au point un BatterySystem, qui comme son nom l'indique simule une batterie qui se vide avec le temps, ce composant étant attaché au drone et lorsqu'il est en échec cause la chute du drone. Afin de contraindre le drone pour qu'il ne soit pas un avantage trop massif pour le joueur, j'ai également implémenté un contrôle de la distance entre le drone et son propriétaire afin qu'il reste dans un périmètre raisonnable et simule l'intensité des ondes pour commander le drone. Malheureusement le drone est à ce jour relégué au rang de simple décoration car son envergure et concept ne lui permettent pas d'être un atout en intérieur, bien que son code soit toujours présent sur le github du projet.

6.2.9 Addressables

Lors de mon interminable et constante quête de l'optimisation, je me suis intéressé au package Addressables natif d'Unity. Précédemment lorsque le jeu se lançait, tous les objets référencés restaient en mémoire vive inutilement même s'ils ne servaient qu'une unique fois ou disons rarement. Cela peut causer des failles dans les performances ou tout du moins consommer la batterie de l'appareil plus rapidement. J'ai donc introduit ce système dans le jeu, il va permettre de construire une base de donnée des modèles que l'on souhaite et ne va les charger de façon asynchrone que lorsqu'ils sont nécessaires. Cependant, ces chargements peuvent devenir des défaut s'ils sont employés pour des objets fréquemment utilisés, ils seraient déchargés à chaque fois et cela aurait un coût grandissant sur la puissance de calcul. J'ai donc dû trouver un équilibre et ne l'utiliser que pour des ressources qui sont utilisées ponctuellement comme le sons de tir, précédemment pour le drone ou encore pour l'instanciation des zombies.

6.2.10 Installateur InnoSetup

Afin de permettre la distribution du jeu de façon plus organisée et pratique, nous utiliserons le logiciel InnoSetup. Cet installateur permet de regrouper nos différentes dépendances, qu'elles soient de simples fichiers ou des même des dossiers si besoin. Il contiendra également un lien direct vers le site web du jeu. Cet installateur va également nous permettre d'afficher l'icône de notre jeu dans la barre des tâches lorsque le jeu s'installera mais également de laisser l'utilisateur choisir où le jeu s'installera sur sa machine. Ce logiciel inclu un désinstallateur pour notre jeu afin d'en effacer toute trace.

6.2.11 Ragdolls & Death Animations

Après avoir majoritairement complété l'IA, j'ai commencé à me pencher sur la mort du zombie lorsque son HealthSystem est en échec. J'ai d'abord utilisé des animations venant de certains assets dont je m'étais déjà servi dont celui d'un des zombies par exemple, mais les animations de mort que je trouvais semblaient très artificielles. Cette sensation était probablement causée par le fait que ces animations faisaient que tous les zombies mourraient de la même manière et rendaient cela ennuyeux à la longue, ils chutaient tous vers la même direction et à la même vitesse, si l'on regardait le sol après en avoir tué quelques uns ils étaient tous orientés de la même façon.

Pour faire face à ce problème, je me suis renseigné sur les Ragdolls et leurs caractéristiques qui semblaient pouvoir palier à mon problème. Une fois les modèles de zombies équipés d'un Ragdoll, il tombaient bien plus naturellement sous l'effet de la gravité et de leur propre poids. Bien que ces Ragdolls aient apporté une amélioration visuelle significative, ils sont aussi venu avec un défaut majeur de performance. Les Colliders ajoutés au zombie permettant cet écroulement naturel pesaient lourd sur le moteur physique lorsqu'il commençait à y en avoir plusieurs dizaines, on observait alors une chute des FPS. Pour résoudre ce problème, j'ai créer un clone de chaque type de zombie qui allaient contenir la version avec Ragdoll tandis que le modèle original en serait privé. Les zombies parcourant l'environnement et interagissant avec le joueur sont les originaux et lorsqu'ils meurent, leur version avec Ragdoll est instanciée sur leur position avec la même orientation et l'original disparaît afin que le changement ne soit pas visible à l'œil nu. Dès lors que la version Ragdoll est instanciée et juste avant de disparaître, l'original transmet au Ragdoll le point et la force d'impact qui ont causé sa mort, et finalement le script Ragdoller applique une force sur la version Ragdoll avec ces données résultant en un effet de propulsion hérité de l'énergie cinétique du projectile. De cette façon, on obtient des animations de morts authentiques et réalistes qui n'impactent pas les performances.

6.2.12 Object Pooling & Spawner

Lors des tests de performances de l'IA, je me suis vite retrouvé submergé par le nombre de zombies à placer avant de pouvoir obtenir des résultats fiables, à savoir plusieurs centaines au minimum. Effectuer ces placements à la main est extrêmement chronophage et propice aux erreurs, en plus de mettre à l'épreuve la patience. C'est pourquoi j'ai pensé à implémenter un système de placement automatique des zombies, mais je me suis heurté à quelques problèmes de conception. L'IA des zombies utilise le Composant NavMeshAgent qui nécessite lui l'utilisation d'un NavMesh (zone de navigation des agents), or un NavMeshAgent a besoin de se trouver proche du NavMesh afin de fonctionner correctement. J'ai donc dû utiliser un NavMeshHit dans le code, qui me permet de récupérer le point le plus proche appartenant au NavMesh depuis une certaine position. En plus de pouvoir placer ces zombies, il me faut pouvoir les générer préalablement pour n'avoir qu'à en récupérer un existant pour le placer mais également avoir un système de pooling permettant de les recycler une fois qu'ils meurent et de n'avoir qu'à réinitialiser leurs paramètres avant de les faire réapparaître.

En plus de ce système que j'ai nommé Spooler (Spawner et Pooling), j'ai implémenter sur ce même script un système d'apparition par intervalles. Afin d'utiliser ce script, il nous faut d'abord créer des objets vides sur la map qui vont servir de zones d'apparitions, puis les renseigner dans l'éditeur sous forme de SpawnPoints. On peut alors pour chacun préciser le nombre initial de zombies et le rayon depuis lequel ils peuvent apparaître, ainsi qu'une possible récurrence dans cette apparition.

6.2.13 Multijoueur

Lors de la conception du multijoueur, j'ai dû réadapter plusieurs mécanismes dont notamment la détection du joueur ainsi que le système de tir, mais également décider d'un modèle d'architecture réseau sur lequel se baser afin de construire cette implémentation en utilisant l'API Mirror. Cette fonctionnalité qu'est le multijoueur est ici utilisée dans sa forme Hôte-Clients, où le premier joueur héberge sur sa machine le serveur qui va synchroniser la partie tandis que les machines des autres joueurs s'y connectant par la suite vont devoir relayer leurs actions au serveur. Étant donné que ces actions doivent être relayées, cela implique que le système de tir doit désormais passer par le serveur afin d'effectuer une action qui va être prise en compte sur toutes les machines, raison pour laquelle il m'a fallu isoler puis extraire ce mécanisme au sein du code de l'avatar. Suite à cela, j'ai dû ajouter des contrôles de nullité pour garantir la stabilité du système et qu'un zombie tué sur l'une des machines ne tente pas de remourrir depuis une autre et causant un bug. C'est d'ailleurs en implémentant le multijoueur que je me suis rendu compte qu'un effet indésirable de ricochet se produisait lorsque le joueur se trouvait proche d'un Collider et tirait dessus, lui faisant perdre de la vie mais surtout provoquant par intermittence un bug dans le transport de données.

Un soucis conceptuel s'est également posé après la remise en question du contexte du jeu et le passage en intérieur car nous abandonnions la perspective en 3ème personne pour une en 1ère personne, or la reformulation du scénario impliquait un corps invisible, ce qui pose problème lors d'un jeu de coopération. Afin de résoudre ce problème, j'ai tenté de représenter l'avatar sous forme spectrale avec des Shaders mais le résultat était assez grotesque et loin de ce que j'envisageais. Je me suis donc tourné vers les ParticleSystem d'Unity et ai joué sur le paramètre du bruit de perlin associé avec de fines particules en mouvement afin de symboliser une singularité gravitationnelle en accord avec le scénario.

6.2.14 IA tourelle



Lorsque je testais les performances des zombies, plusieurs centaines en étaient instanciés et je voulais observer le comportement des ragdolls lorsque des dizaines de zombies mourraient en même temps. C'est alors que j'ai eu l'idée de créer une deuxième IA pour une tourelle automatique qui détecte, cible physiquement le zombie puis tire, tout en ayant des effets spectaculaires. J'ai donc récupéré un modèle de tourelle, mais l'animation de celui-ci n'était pas polyvalente et même assez inutile. De plus, ce modèle avait été créé avec un système de coordonnées ambigu qui m'empêchait d'utiliser les fonctions classiques permettant de s'orienter vers une position comme 'LookAt' par exemple.

Délaissé par ces fonctions, je suis alors retombé dans la nostalgie de faire des schémas trigonométriques à la main afin d'en sortir un algorithme. Pour plus de réalisme, j'ai décidé que l'influence des parties mobiles de la tourelle seraient divisées en fonction des axes vertical et horizontal. Ainsi la rotation horizontal se fait sur le mât de la tourelle pour s'orienter vers le zombie, tandis que la rotation vertical est appliquée sur le canon de la tourelle. Afin de rendre cette tourelle plus polyvalente, j'ai exhibé ses paramètres de portée, de types de cibles, de timeout (intervalle de refroidissement entre chaque tir) ainsi que de quantité de dégâts par tir entre autres.

6.2.15 Avatar 1ère personne

Suite à certains problèmes au sein du groupe empêchant un rythme régulier et une progression nécessaire à la réalisation du concept initial, nous avons dû nous adapter et changer le scénario d'un monde extérieur vers un monde fermé qu'est le laboratoire. De ce fait, l'avatar en 3ème personne initialement prévu n'est plus adapté à cet espace fermé et j'ai dû le convertir à un avatar en 1ère personne. Pour ce qui est des interactions physique de base j'ai recyclé ce que j'avais implémenté pour l'avatar 3ème personne, à savoir la gravité, les déplacements et le saut. J'ai dû déplacer la caméra correspondante afin qu'elle soit placé plus ou moins au niveau des yeux supposés de l'avatar. Afin de lui offrir un rendu visuel permettant l'implémentation du multijoueur, j'ai commencé par chercher un Shader qui pourrait simuler une lentile gravitationnelle qui ferait référence à la superposition dimensionnelle mais aucun asset de qualité n'était disponible et je n'aurai pas eu le temps de le développer personnellement.

Après cette veine recherche, je me suis tourné vers la génération de particules auxquelles j'ai appliqué un bruit de perlin sur leurs déplacements, cet effet donne vie aux particules en leur conférant des mouvements imprédictibles et hypnotisants. De plus, la taille du joueur était précédemment ajuster pour correspondre à l'échelle des zombie, mais cette dernière était devenue invalide dû au changement de rendu visuel de l'avatar. Ce changement a donc nécessité un recalibrage des différents échelles appliquées aux différents objets inclus dans l'avatar car une modification de l'échelle d'un objet parent se répercute sur tous ses descendants.

6.2.16 Système de collecte

Que ce soit pour récupérer différents armes présentes sur la map, collecter des munitions ou encore des badges d'accès, il faut pouvoir intéragir avec leur représentation physique sur la map grâce à une fonction de récupération. Cette fonction utilise OverlapSphere qui permet de récupérer tous les Colliders à une certaine distance définie en paramètre depuis la position de l'avatar et permet de spécifier une Layer à prioriser. J'ai donc profité de ce dernier paramètre et ai appliqué la Layer 'Collectable' sur les badges, caisses de munitions et armes à ramasser. De plus, j'ai distingué les différents objets récupérables en fonction de leur nature par un tag spécifique. Ainsi les armes possèdent le tag 'Weapon', les badges le tag 'Pass' et enfin les caisses de munitions le tag 'AmmoCrate'. La récupération d'un badge copie ses informations dans la liste de badges acquis du joueur puis détruit sa représentation visuelle. Pour ce qui est des caisses de munitions, elles contiennent une certaine quantité de munitions, et afin de rendre son fonctionnement plus réaliste j'ai imposé une limite au nombre de munitions que le joueur peut transporter (qui est actuellement de 50) et lorsque celui-ci récupère des munitions, uniquement le montant nécessaire est prélevé de cette caisse de munitions. Mais lorsque le joueur a récupérer toutes les munitions disponibles sur cet objet, la caisse de munitions disparaît afin que le joueur ne pense pas à un bug et n'ai pas le faux espoir de pouvoir se ravitailler indéfiniment à cet endroit. Pour ce qui est des armes, elles sont toutes prépositionnées sur le modèle de l'avatar afin d'obtenir une qualité maximale dans le placement, et lorsqu'une est 'collectée' sur la map, elle devient simplement visible sur l'avatar et se détruis sur la map. De plus, si le joueur possède déjà une arme et en ramasse une autre, c'est la nouvelle qui va être rendue comme arme équipée et une copie de l'ancienne va être instanciée sur la map à l'endroit où était placée la nouvelle. Pour ce changement d'armes, il m'a fallu faire bien attention à mettre la Layer 'Collectable' uniquement sur les copies placées au sol et à bien retirer la partie '(Clone)' ajoutée au nom de l'arme instanciée afin qu'elle puisse être récupérée plus tard car la fonction utilise un système de comparaison de noms afin de savoir quelle arme rendre visible depuis l'avatar.

6.2.17 Outil SceneBreaker

Lorsque j'ai vu que le développement du laboratoire prenait de l'ampleur, j'ai commencé à craindre pour les FPS et la surchauffe. Le laboratoire utilise de nombreux modèles différents, et chacun des matériaux de chacun de ces modèles ont besoin d'être rendu indépendamment des autres, et cela pour chaque frame si aucune optimisation n'est mise en place. Toutes les scènes chargées doivent alors être rendues pour chaque frame, ce qui m'a alors fait penser à segmenter le laboratoire en une multitude de scènes afin de ne garder actives que celles directement visibles pour le joueur et ainsi épargner une grande quantité de ressources. Je me suis donc renseigné sur les méthodes d'automatisation pour les modifications de scène et j'ai compris que l'on pouvait créer des menus accessibles directement dans la barre d'options Unity en plaçant un script à l'architecture spécifique dans un dossier nommé 'Editor'. Je me suis donc servi de cette notion afin de créer un partitionneur de scène, son but est de prendre un objet représentant un étage en entré, et de segmenter chacune des salles qu'il contient en plusieurs scènes, chacune portant le nom de la salle qu'elle contient. Afin d'activer le chargement de ces scènes, j'ai créé un script 'SceneTrigger' qui s'accroche sur un objet muni d'un Collider avec un volume calqué sur celui de la salle à charger, et lorsque le joueur entre dans le Collider passé en mode 'Trigger', alors la scène correspondante est chargée, et inversement lorsqu'il quitte ce Collider alors la scène est déchargée. Ce mécanisme permet de palier à certains défauts de l'Occlusion Culling comme le Bake ainsi que les calculs parfois coûteux permettant de savoir ce qui est à rendre de ce qui ne l'est pas. Ce mécanisme partitionneur de scène permet donc un grand gain de performance mais j'ai également fais en sorte qu'il soit réversible pour éviter d'avoir à effectuer de nombreuses sauvegardes avant de l'utiliser. J'ai également implémenté sur cet outil une fonctionnalité de fusion de modèles dont je pourrais éventuellement avoir besoin, mais également un système permettant de relier automatiquement les dépendances dans un dossier lorsque l'on doit exporter notre travail sur une autre machine par exemple, il suffit alors de définir le tag 'Dependify' sur les objets des scènes actives que l'on veut exporter.

6.3 Baptiste

6.3.1 Site

Tout d'abord, le site Equinox est hébergé sur dyjix et mis en ligne depuis la 1re soutenance.

Le site contient deux pages, avec chacune ces spécificités, et ses objectifs, nous reviendrons sur les différentes technologies utilisées pour le site, pourquoi ce choix ensuite nous parlerons de la conception du site, du choix des éléments dans chaque page et en dernier des différentes dates et des différentes versions.

- Technologie utilisée

J'ai eu le choix entre différentes technologies pour le site, j'ai hésité pour la nouvelle technologie svelte.js, qui est bien documenté, très légère et permet de simplifier la conception de sites sur certains aspects et c'est ce que j'ai utilisé, le problème étant que pour simplifier la compréhension avec les autres membres du groupe et aussi au vu de la communauté de svelte qui est plus petite que html-css-js (la version que tout le monde connaît) j'ai décidé de passer sur un site avec la forme basique du html-css-js, le développement a été pimenté de nombreuses erreurs, mais ayant de l'expérience dans le développement de site ça a été plus facile.

- Conception

Chaque page à son intérêt et un but précis. Pour commencer, dans la première page nous avons un aperçus sur le jeux, les informations sur le jeux, sur l'équipe et des informations de contact. Nous avons donc quatre rubriques :

- L'accueil, elle nous donne une illustration animé de l'ambiance du jeux, et de son thème et mets en avant le nom du projet Equinox, l'illustration animé donne aussi un aperçus sur ce qu'on aura dans le jeux, le plus important a ressentir c'est l'ambiance, qui est sombre. Ce qu'on voit est le spawn des joueurs donc un paysage qu'il vont voir dès le début de leur aventure

- A propos, Cette partie nous donne des informations importante sur le concept de base du jeux pour cerner au mieux le jeux et ses stratégie et son scénario, les objectifs du jeu. Si une personne qui ne connaît rien au jeux arrive sur le site, il aura un aperçus de l'ambiance avec l'accueil puis des informations sur le jeux dès les première minutes. Notre volonté est vraiment que le voyage sur le site soit concret et efficace, c'est pour ça aussi que le site est sur la longueur est pas seulement des pages pour chaque catégorie.

- Notre équipe, après en avoir appris plus sur le jeux et son ambiance, l'équipe derrière est aussi importante, si on connaît mieux l'équipe on comprendra certain choix artistique/choix des technologies. Cela aussi permet de comprendre quelle compétence l'équipe à pu mettre en commun.

- Information, cette partie, nous avons un lien vers la page de téléchargement et les liens de contact pour les personnes souhaitant parler avec l'équipe, cela aide aussi pour le coté commercial/marketing. Comme le site possède son propre hébergement, le site possède son propre nom de domaine et donc son propre système de mail avec des mail personnalisé pour chaque pôle

Ensuite, dans la seconde page, il y a un 2e aperçus du jeux qui montre un paysage qui fait un lien avec la fin du jeux, c'est dans la continuité, on arrive sur le site on vois un aperçus du début du jeux, on parcours et on arrive a la page téléchargement ou on a cette fois un aperçus de la fin, puisque arriver sur cette page marque la fin du but de faire connaitre le jeux grâce au site. Nous avons sur cette page ensuite des informations sur les dépendances et technologie utilisé à partir de ce moment, c'est du plus, ce n'est pas obligatoire, mais c'est destiné aux partenaires commerciaux ou au joueur qui souhaiterait en savoir plus et pour finir nous avons les rapports fait chaque semaine avec les différents ajouts. Nous avons donc 4 rubriques aussi.

- Accueil, comme dit l'illustration est ici pour faire une continuité avec la page d'accueil de la première page, et de marquer la fin de la mission du site pour la plupart des utilisateurs

- Jeux, nous avons ici les différents liens de téléchargement du jeu, de son client, du rapport. Tout est fait pour rendre cela très simple, vous avez la possibilité d'installer le jeu sans son client, juste le jeu de base.

-Technologie/dépendance, cette catégorie réservée pour les connaisseurs ou les partenaires commerciaux. Nous utilisons différentes technologies non incluses dans Unity (logiciel que nous utilisons pour le jeux) et dedans nous les listons ici, par exemple le logiciel utilisé pour les modèles 3 d ou encore la librairie Mirror pour le multijoueur, nous listons aussi les technologie utilisé dans l'asset store, ou d'unity.

- Rapport, comme dit, les rapports sont envoyé chaque semaine pour que les intéressés puissent suivre l'avancement du jeu en temps réel, c'est un de nos moyens de montrer que l'équipe est motivé et est toujours sur le projet, cela permet d'avoir surtout la progression du développement.

Pour conclure sur cette présentation du contenu et sa conception, nous pouvons reprendre le fait que, notre but était d'apporter un réel apport de connaissance sur le jeux quand la personne arrive sur le site, nous avons une réelle continuité dans les illustrations et des les différentes connaissances livrée. Notre but étant de pousser le joueur à télécharger notre jeux, il faut lui faire aimer et ça passe par l'apport des connaissances nécessaire pour voir le jeux sous le meilleure angle

- Date clé et version

rapide retour sur les différentes date clé et version associé, tout d'abord la V1 du site avec le tout premier fond d'écrans qui était en rapport avec les montages que nous devions parcourir, le fait étant que l'idée principale du jeux ne change pas mais nous avons changé de milieu pour le jeux pour des soucis de performance et de beauté, donc le site à eu une nouvelle beauté à partir de la 2e soutenance, ce qui est donc sa V2, mais il restait un problème c'est que le site ne représentait pas assez le jeux et certaine illustration ne coller plus assez, nous avons changer le fond de la première page pour avoir cette continuité dont nous parlions

Pour conclure sur le site, celui-ci a évolué avec le jeu, avec chacune de ses versions et son but est bon. Le site en lui-même est optimisé et hébergé et grâce à lui tout est fait pour rendre le joueur le plus content possible.

6.3.2 Sectorisation

Définition de la sectorisation : Organisation par secteurs, Nous aurons donc dans le jeux plusieurs secteurs, et des pièces nécessaire pour entrer et sortir.

- Changement

Lors de la 2e soutenance, je parlais d'un système de quête celui-ci n'est plus d'actualité puisqu'il ne correspondait plus à l'idée que l'on avait du jeux. Ici nous avons donc un nouveau système en partie présenté lors la 2e soutenance. La sectorisation qui reprend certain principe du système de quête mais en enlevant quelque fonctionnalité et en rajoutant d'autre.

- Comment ça marche ?

Le jeux est séparer par différents secteurs et pour passer une porte donc passer a un nouveau secteur il faut trouver et prendre un badge perdu dans le secteur pour accéder au suivant et donc ouvrir la porte, les badge peuvent être de partout, sur des bureaux, à coté de pc, par terre. Cela rajoute du réalisme puisque pour rappel il est seul dans le laboratoire et tout a été abandonné.

Le système est très simple, comme vous n'avez pas de main dans le jeux, lorsque vous vous approchez du badge et appuyer sur une certaine touche ça le récupère et le badge disparait (vue qu'il va dans vos poches) puis lorsque vous allez vers la porte, celle-ci va check qui vous possédiez bien le badge associé à la porte et que vous avez les badges des portes précédentes si oui elle s'ouvre sinon elle ne s'ouvre pas

6.3.3 Jaquette

Nous avons besoin d'un boitier de présentation pour le jeux et je me suis occupé d'en faire une réalité. Rappelons que le but du boitier et d'attirer le client/joueur et de donner un premier aperçu sur le jeux. Nous devons ainsi donner le plus d'information tout en étant clair et concis

Face arrière : Pour la face arrière, nous allons mettre les aperçu du jeux tel qu'un réacteur

Face avant : Nous aurons ici le logo de notre jeux ainsi que le nom du jeux

Face droite : Nous aurons les logo des logiciels que nous avons utilisé pour le jeux par exemple unity, blender

Face gauche : Nous aurons une phrase mystérieuse lié au synopsis du jeux « . . . J'ai froid . . . il fait sombre . . . où est ce que je suis . . . »

Pour conclure, nous avons donc juste en regardant la boite, toute les informations principale que nous devons retrouver, et qui va surement attirer de nombreuse personne

7 Ressenti

7.1 Justine

Avant ce projet, je n'avais jamais essayé de faire de jeux vidéo et donc Unity m'était tout sauf familier. J'avais un peu touché à la modélisation 3D et à la conception de plans de maisons, d'appartements, etc, et même si ce n'était que très peu d'expérience, je me suis tout de suite proposée pour tout le côté graphique du jeu. Travailler sur Equinox m'a permis de me découvrir une nouvelle passion : faire des jeux vidéo. Au tout début du projet, je n'avais même pas pour idée de me lancer sur un jeu. Or un ami m'a convaincu de faire un jeu RPG d'horreur. C'est comme ça que le projet Equinox est né. J'étais très peu motivée et enthousiaste, car je n'avais jamais ouvert Unity et je n'avais AUCUNES idées de comment ce logiciel s'utilisait. Dès que j'essayais d'avancer sur le projet, je ne comprenais pas comment une certaine notion marchait et je devais passer des fois des heures à regarder des tutoriels. Après m'être forcée à avancer un peu pour ne pas retarder tout le groupe, la map ressemblait enfin à quelque chose de potable. Sauf que j'ai tellement galéré entre tous les bugs et crash Unity, que ma motivation a chuté.

Ce n'est qu'une fois après avoir changé le scénario du jeu que ma motivation a très fortement augmenté. En effet, non seulement j'avais beaucoup moins de bugs, mais c'était aussi beaucoup plus divertissant et intéressant de faire un laboratoire que de mettre des arbres dehors. Il y a eu pas mal de problèmes d'entente au niveau du groupe, ce qui m'a un peu tendue. Travailler en groupe signifie qu'on découvre le vrai visage des gens. À part ces soucis-là, je suis vraiment heureuse d'avoir travaillé sur ce projet, et je sais désormais que ce ne sera pas le dernier jeu sur lequel je travaillerai.

7.2 Matthieu

Il y a quelque années je m'étais intéressé aux moteurs graphiques comme Unity ou encore Unreal Engine, mais le nombre important de concepts à appréhender avant d'avoir un quelconque rendu était assomant et m'avait découragé. Cette année, le projet S2 était l'occasion de retenter ma chance avec cette fois considérablement plus de connaissances en programmation mais également en recherche. Le plus compliqué a donc été de me lancer dans ce projet et de commencer à réunir des informations et développer mes habitudes sur Unity, mais une fois fait j'ai commencé à travailler bien plus efficacement et avec plus d'aisance.

La répartition du travail dans ce projet a été un grand point de débat, non parce que la différence de niveau imposait ce partage mais plutôt parce que certaines personnes n'ont pas forcément envie de s'améliorer et préfèrent se reposer sur les autres. De plus, ce projet m'a permis de bien me rendre compte de l'importance de la distinction des domaines privé et professionnel lors de la composition d'un groupe car certaines personnes que l'on croit connaître peuvent se révéler totalement différents lors d'un enjeu, en l'occurrence un enjeu scolaire ici, et n'hésite pas à s'approprier le travail des autres à leurs dépens.

Ce projet m'a donc permis d'en apprendre davantage sur la conception 3D mais aussi plus globalement sur l'informatique en me forçant à traquer les optimisations et viser la perfection sans s'arrêter au fonctionnel. Malheureusement ce périple captivant a été terni à cause du comportement inapproprié de certaines personnes mais n'en reste pas moins une bonne expérience que je serais heureux de réitérer dans de meilleures conditions.

7.3 Baptiste

Ce projet de S2 à été une bonne expérience, j'ai l'habitude sur mes projets perso de faire ça seul puisque sur les projets à plusieurs c'est toujours très compliqué de tout bien organisé et tout bien cadre, ici même s'il y a eu quelque soucis on as réussi à faire quelque chose à la fin et quelque chose de bien je trouve. Il a aussi été une bonne expérience puisque j'ai découvert unity que j'ai bien aimé et je compte l'utilisé plus et l'approfondir, j'ai aussi découvert blender pour la modélisation 3d et bonne surprise c'est aussi un logiciel que je compte approfondir.

Ce projet en lui-même va permettre de rendre les prochains projets à plusieurs beaucoup mieux, et je me suis rendu compte que malgré les grosses difficulté dans un groupe hétérogène en terme de niveau, c'est quand même une bonne chose puisque on peut faire de plus gros projet et on peut mieux faire ses parties, je prend l'exemple d'un projet personnelle que j'avais effectué ou je devais faire le site, le logiciel, les hébergé, gérer tout les soucis et gérer aussi les remonté utilisateur et c'est très compliqué de tout gérer, j'essaie toujours de tout automatisé pour rendre le travail plus simple, prendre des personnes pour le support mais même avec tout ça garder la motivation sur un projet qui te prend autant de temps n'est pas tenable, donc cela m'a permis de voir la nécessité de travailler en équipe

J'ai trouvé qu'a cause des soucis interne, le travail n'a pas bien été encadré et répartit, ce n'est pas normal qu'un personne se retrouve à faire le travail de deux personne, mais dans tout les cas cela aurait été très compliqué même sans départ ou conflit interne de bien répartir le travail puisque le groupe est hétérogène et chacun à ses prédisposition et c'est très compliqué de remonter le niveau de chacun en très peu de temps.

Pour conclure, je pense que ce projet à été une excellente expérience et m'a beaucoup appris.

8 Annexes

8.1 Schémas de conception

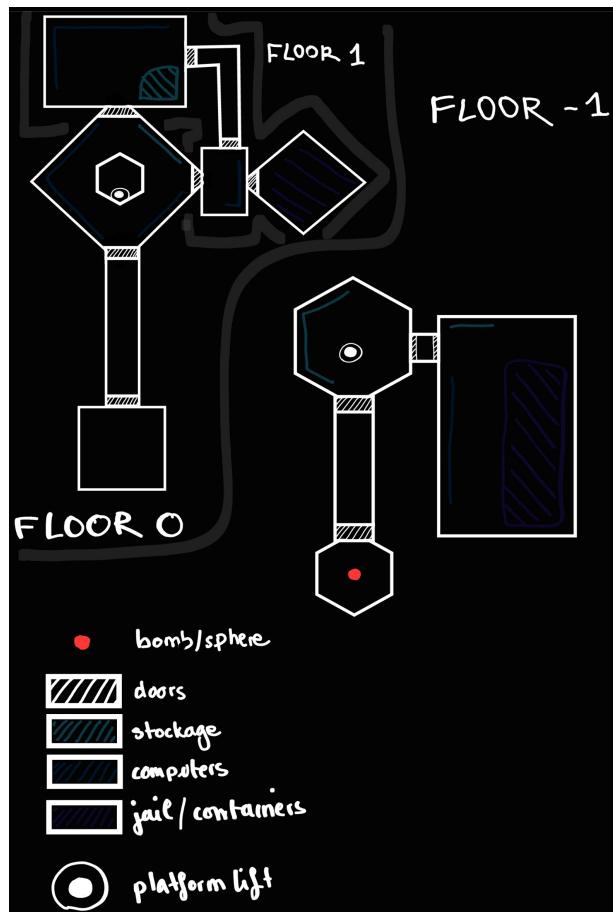


FIGURE 1 – Ancien schéma du labo

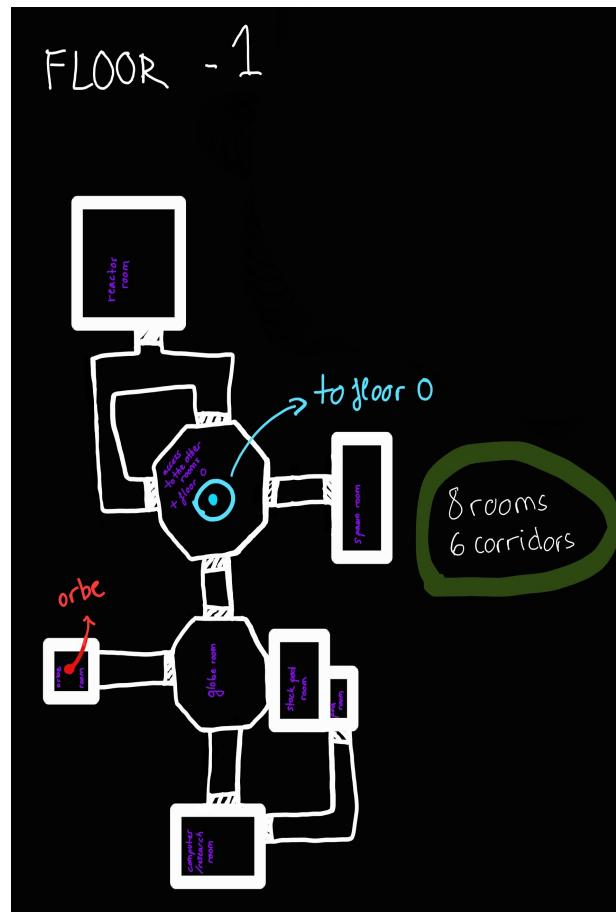


FIGURE 2 – Étage -1

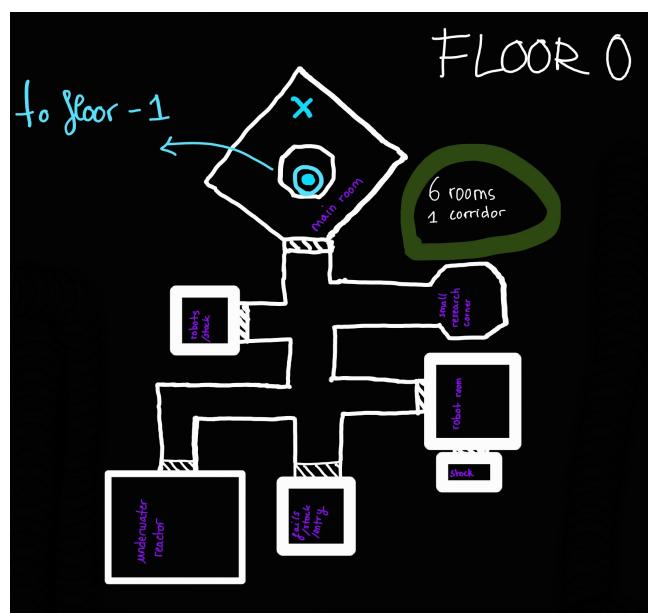


FIGURE 3 – Étage 0

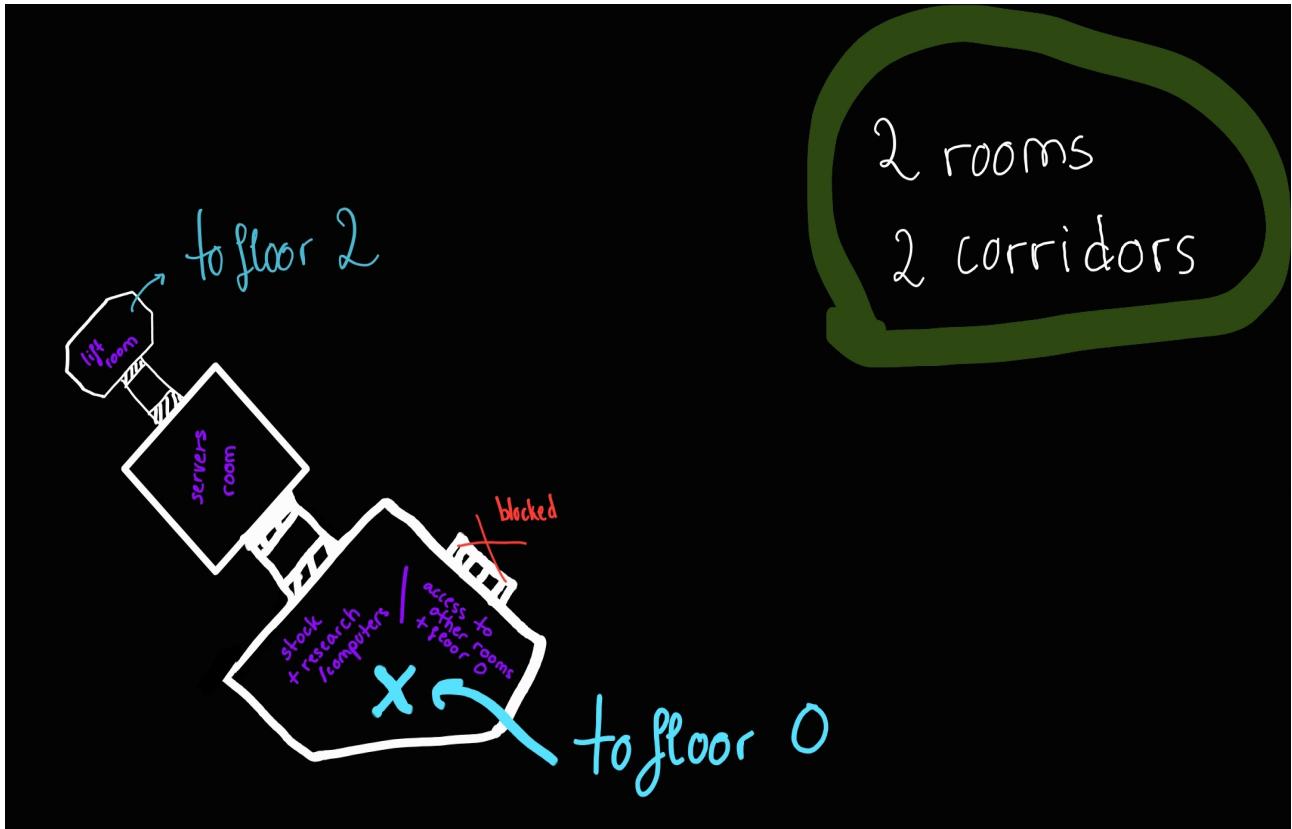


FIGURE 4 – Étage 1

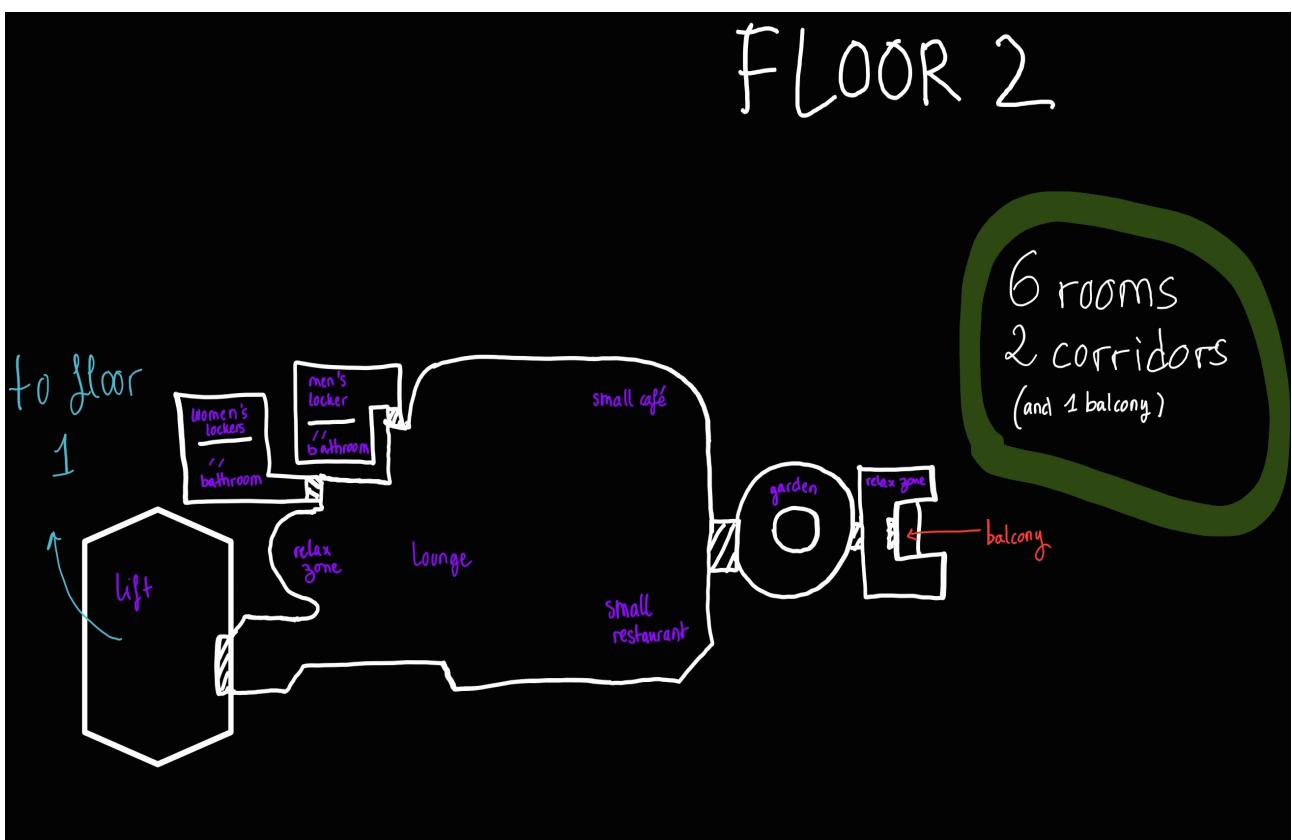


FIGURE 5 – Étage 2

8.2 Aperçus

8.2.1 Avant changement de concept



FIGURE 6 – Menu



FIGURE 7 – Drone au sol



FIGURE 8 – Drone en survol



FIGURE 9 – Map extérieure



FIGURE 10 – Map extérieure

8.2.2 Après changement de concept



FIGURE 11 – Spawn Room



FIGURE 12 – Main Room (below)



FIGURE 13 – Research Room



FIGURE 14 – Pods Room

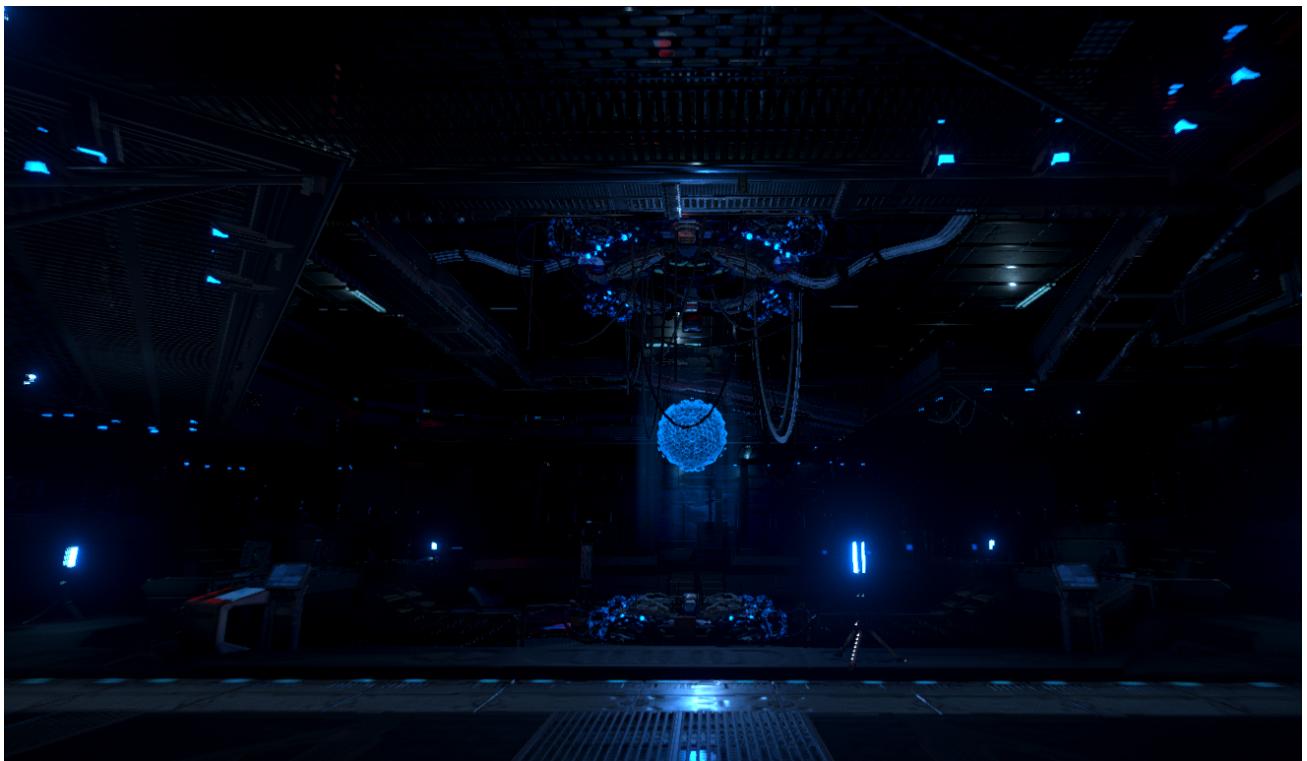


FIGURE 15 – Main Reactor Room

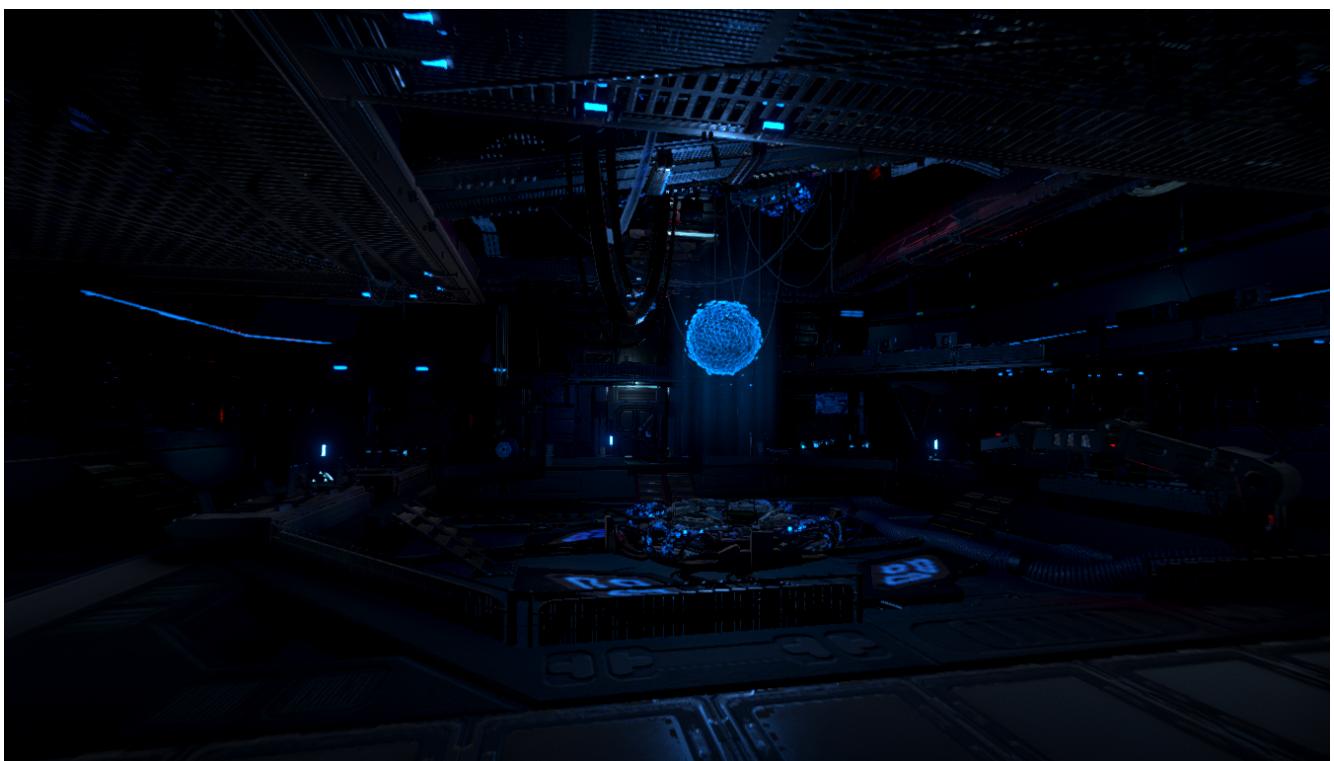


FIGURE 16 – Main Reactor Room

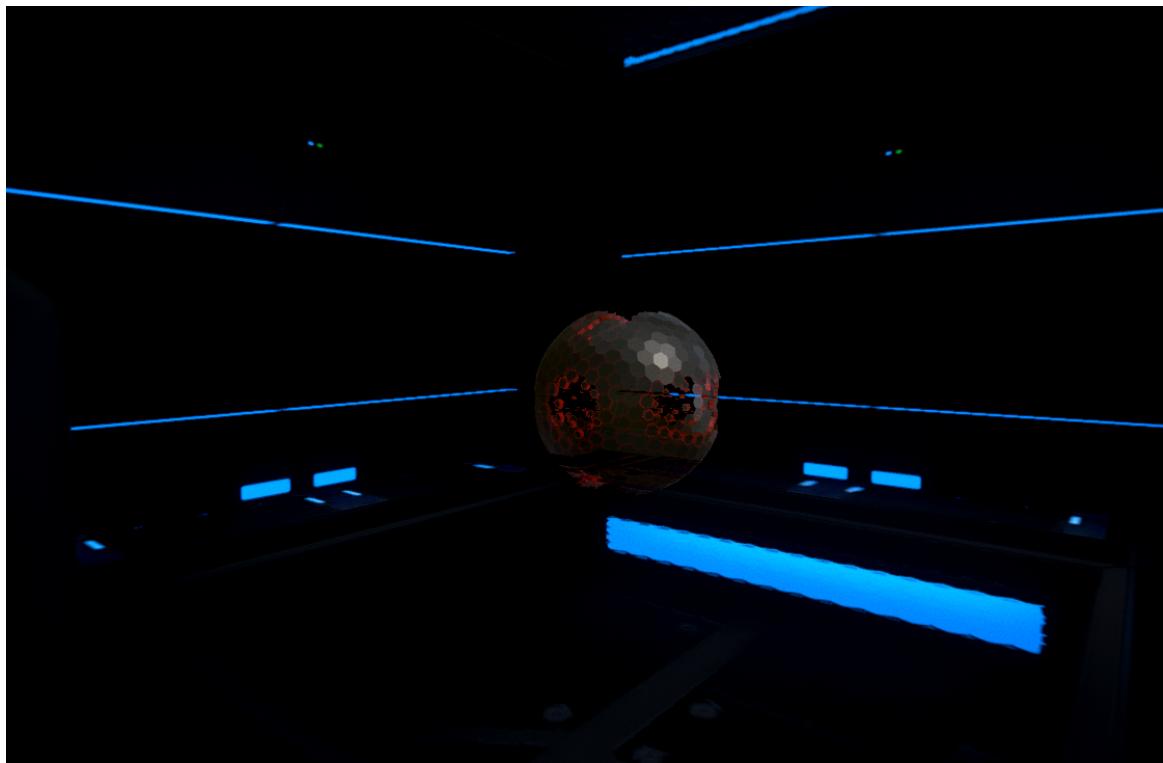


FIGURE 17 – Orbe Room



FIGURE 18 – Main Room



FIGURE 19 – Storage Room



FIGURE 20 – Submerged Reactor Room



FIGURE 21 – Upper Hidden Room



FIGURE 22 – Below Hidden Room



FIGURE 23 – Inaccessible Room 1



FIGURE 24 – Inaccessible Room 2

8.3 Sources

<https://www.youtube.com/watch?v=xppompv1DBg&t=455s> <https://www.youtube.com/watch?v=QajrabYTCJc&t=2s>
<https://assetstore.unity.com/packages/3d/environments/sci-fi/modular-sci-fi-corridor-142811>
<https://www.youtube.com/watch?v=KMO1Bfwd3SU> <https://www.youtube.com/watch?v=P49NWZW8ENs&start=8>
<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-construction-kit-modular-159280>
<https://sketchfab.com/3d-models/ecolog-green-f135d2904041407b93ad0d529900d902>
<https://www.youtube.com/watch?v=yZI0GFutK9U>
https://www.youtube.com/watch?v=v3OEL8FjTcY&list=OLAK5uy_mhufDs9bzBYaY9ZWLw7PXmvacQ6tOrI&index=5
<https://www.youtube.com/watch?v=-02e7q9FyY4>
<https://sketchfab.com/3d-models/tiling-material-crazy-wires-4869d9bd653b417fb93d79d47f643d96>
<https://www.youtube.com/watch?v=YOaYQrN1oYQ&t=292s>
<https://assetstore.unity.com/packages/3d/characters/humanoids/zombie-aa-74321>
<https://www.youtube.com/watch?v=Pzd8NhcRzVo&t=166s>
<https://sketchfab.com/3d-models/tripod-omni-directional-lamp-d7ee949704c24ddab792184fcfde0016>
<https://www.youtube.com/watch?v=WYjK5uHF5w>
<https://sketchfab.com/3d-models/low-poly-whiteboard-1cbf089ec4b741ad9d092d15db6d55ba>
<https://assetstore.unity.com/packages/3d/characters/ghoul-zombie-114531>
<https://sketchfab.com/3d-models/server-racking-system-6fe2cacf836b4aed96c650b286db5486>
<https://sketchfab.com/3d-models/electricity-wire-2d060429d6d946b0a4217dab04ee98df>
<https://www.youtube.com/watch?v=NeESf9aCZHQ> https://www.youtube.com/watch?v=u5rmP_kAJ2o&list=PLc74DR4WWHumNpmv-FEfB8KxqQyx4xEg-&index=5
<https://sketchfab.com/3d-models/ecolog-orange-69fb0b251ea44a998dfef4efb7ae4da5>
<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913>
<https://sketchfab.com/3d-models/dead-hazmat-female-2e879c914d6b4e15a05f8496977854fd>
<https://assetstore.unity.com/packages/vfx/shaders/shapes-fx-pack-156638>
<https://www.youtube.com/watch?v=UK57qdqlak&list=PLPV2KyIb3jR5PhGqsO7G4PsECAl-kPZ>
<https://assetstore.unity.com/packages/tools/particles-effects/led-light-blocks-195651>
<https://www.youtube.com/watch?v=64NblGkAabk&t=100s>
<https://assetstore.unity.com/packages/essentials/starter-assets-first-person-character-controller-196525>
<https://www.youtube.com/watch?v=nlcIz-czKyI>
<https://www.youtube.com/watch?v=eAoRvx7oh48&list=PLc74DR4WWHumNpmv-FEfB8KxqQyx4xEg-&index=2>
<https://sketchfab.com/3d-models/draft-sci-fi-nuclear-reactor-f5de4be4ca7948e998e01d9d40d6d28e>
<https://assetstore.unity.com/packages/3d/characters/humanoids/modern-zombie-free-58134>
<https://sketchfab.com/3d-models/scifi-terminal-ace1b257e2c24ec184662d69e4befca1>
<https://assetstore.unity.com/packages/3d/environments/3d-scifi-kit-starter-kit-92152>
https://www.youtube.com/watch?v=Dn_BUdAPg
<https://sketchfab.com/3d-models/wet-floor-sign-7673d504c2b34f7c94c72b443b3ca907>
<https://sketchfab.com/3d-models/drone-concept-500-followers-celebration-49a01e710d9545ceb13dfd8c9fe11fea>
<https://sketchfab.com/3d-models/bio-chamber-unit-free-asset-be85738730d7494a85450a0b9ce105c8>
<https://assetstore.unity.com/packages/3d/characters/humanoids/zombie-30232>
<https://sketchfab.com/3d-models/radial-symmetry-test-86e211e4f0a94a63aa731871ec537d97>
<https://www.youtube.com/watch?v=THnivyG0Mvo>
<https://assetstore.unity.com/packages/tools/modeling/mesh-combine-studio-2-101956>
<https://sketchfab.com/3d-models/chain-link-fence-low-poly-version-f7d34687536a45108fac3bf1a1055ee2>
<https://sketchfab.com/3d-models/accesspanel1-9c372a0728d1474e8caf5f2d9b846637>

<https://sketchfab.com/3d-models/crane-5eae572703f34fa0ae060bc30111c6a9>
<https://sketchfab.com/3d-models/warning-light-3df0b1fc0e0a445c8edcaddf3dd6ceb5>
<https://www.youtube.com/watch?v=4HpC-2iowE&t=301s>
<https://assetstore.unity.com/packages/3d/environments/sci-fi/bridges-3d-basic-pack-21302>
<https://sketchfab.com/3d-models/terminal-02-be1dfa8354af401c8e46910e24b11281>
https://www.youtube.com/watch?v=_c5IoF1op4E&pp=u gMICgJmchABGAE%3D
<https://sketchfab.com/3d-models/sci-fi-energy-capsule-20b23d4a703e4f7ebfb105b6b140b6fe>
https://www.youtube.com/watch?v=brWURZsA9CY&list=OLAK5uy_mU-YGXY2Sd4av1kr1MetUWozCEDu2r3Y&index=3
<https://www.youtube.com/watch?v=aOOQvWAPeuo&t=90s>
<https://www.youtube.com/watch?v=5MuA92xUJCA&t=258s>
<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-battery-pack-free-19738>
<https://sketchfab.com/3d-models/rotating-radar-7950456b9c134dfc9b2250098363a7ef>
<https://sketchfab.com/3d-models/sci-fi-crate-16ccda0f28d04b3d8b5b480a5f53df3c>
<https://sketchfab.com/3d-models/warning-signboard-5805213c2b54454fb22d8176c8741210>
<https://sketchfab.com/3d-models/flashing-light-mayak-c12-55m-133ca98cfdff458b92516c4239b06939>
<https://sketchfab.com/3d-models/accesspanel2-23d0f4583da74ed7b0ca34e0c13c3725>
<https://sketchfab.com/3d-models/scifi-platform-end-cd0218352d1f49578a799adadb21a5f7>
<https://www.youtube.com/watch?v=DInV-jHm9rk&t=131s https://www.youtube.com/watch?v=Htl7ysv10Qs&t=1s>
<https://www.youtube.com/watch?v=28JTTXqMvOU>
<https://sketchfab.com/3d-models/ecolog-yellow-4d7b730553c24418b7f4338d27e21ed6>
<https://assetstore.unity.com/packages/3d/modular-railing-set-17090>
<https://sketchfab.com/3d-models/warning-signs-set-2e8ef8f837c445578e3d08bdcd37ee3b>
https://www.youtube.com/watch?v=zc8ac_qUXQY&t=657s
<https://sketchfab.com/3d-models/scifi-water-tank-ad06ee30b8784201bb6509ec77edfcac>
<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-access-machine-162924>
<https://www.youtube.com/watch?v=U1nPbJZ1hlc&list=PLc74DR4WWHumNpmv-FEfB8KxqQyx4xEG-&index=3>
<https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-196526>
<https://sketchfab.com/3d-models/doom-3-vent-01-cadf506fc6504550b584ca63e3ef9841>
<https://sketchfab.com/3d-models/arstyl-wall-panel-liquid-nmc-63486e558fae41dd93224b2c1b031e2d>
<https://www.youtube.com/watch?v=U9QKH0k5i3c&list=PLc74DR4WWHumNpmv-FEfB8KxqQyx4xEG-&index=4>
https://www.youtube.com/watch?v=BLfNP4Sc_iA&t=207s
<https://sketchfab.com/3d-models/sci-fi-generator-32c5436e8e9c459eac7471dce7c3878f>
<https://www.youtube.com/watch?v=9tjYz6Ab0oc>
<https://sketchfab.com/3d-models/oxygen-station-3cf9b94a0da54efea0185417460ed8de>
<https://www.youtube.com/watch?v=uApYP7k3yJk&list=PLc74DR4WWHumNpmv-FEfB8KxqQyx4xEG-&index=6>
<https://sketchfab.com/3d-models/chain-barrier-a18364f9fc45450382f5dfcc269491d4>
<https://sketchfab.com/3d-models/ecolog-white-14bb5d65d5804d3abd0c4a1d4d36112b>
<https://sketchfab.com/3d-models/high-quality-scifi-generator-bf2e1a087adb4628a376ffbed17c41ff>
<https://www.youtube.com/watch?v=woCt1-uXEY&list=PLc74DR4WWHumNpmv-FEfB8KxqQyx4xEG-&index=1>
<https://www.youtube.com/watch?v=FEA1wTMJAR0&t=113s>
<https://sketchfab.com/3d-models/server-rack-c144d837358d446998c7b63cd1bdf825>
<https://assetstore.unity.com/packages/tools/particles-effects/lightning-bolt-effect-for-unity-59471>
<https://sketchfab.com/3d-models/railguntower-building004-4f473647f4154c6e9250f178daaeaa7c>
<https://assetstore.unity.com/packages/3d/props/guns/sci-fi-pistol-pbr-213679>
<https://sketchfab.com/3d-models/hologram-console-bfb8b481e98e4be38774b1d0204c192c>