

Rapport de projet



Enclave

Louis Plaire

`louis.plaire@epita.fr`

Sacha Skopan

`sacha.skopan@epita.fr`

Martin De Dorlodot

`martin-de-dorlodot@epita.fr`

Angel Pasquin

`angel.pasquin@epita.fr`

Ethan Bordas

`ethan.bordas@epita.fr`

Aura Corp.
EPITA

Table des matières

1	Introduction	3
2	Origines	4
2.1	Origines et nature du projet	4
2.2	L'entreprise et l'équipe	6
3	Présentation chronologique et individuelle des fonctionnalités	10
3.1	Angel Pasquin	10
3.1.1	Inspirations artistiques et ambiance visuelle	10
3.1.2	Le personnage principal	12
3.1.3	Le bestiaire	13
3.1.4	Environnement	14
3.1.5	L'animation en sprite sheet	15
3.1.6	Sound design et musiques originales	16
3.1.7	Interface et HUD	17
3.1.8	Police d'écriture	17
3.1.9	Jaquette et logos	18
3.2	Louis Plaire	19
3.2.1	L'intelligence artificielle	19
3.2.2	Le multijoueur	21
3.2.3	Direction générale du projet	22
3.3	Ethan Bordas	24
3.3.1	Les débuts dans le projet	24
3.3.2	Les menus	24
3.3.3	Site web	26
3.4	Martin de-Dorlodot	27
3.5	Sacha Skopan	36
3.5.1	Level design	36
3.5.2	Histoire	40
3.5.3	Capacités	42
4	Conclusion	47

1 Introduction

Le développement d'Enclave touche désormais à sa fin, le projet Enclave a continué à évoluer de manière significative depuis notre dernier rapport, tant sur le plan technique que créatif et visuel. Plusieurs ajouts et améliorations ont permis de renforcer l'expérience de jeu.

Premièrement, l'une des avancées les plus importantes est la mise en place d'un mode multijoueur en réseau et plus seulement sur le même ordinateur. Les joueurs peuvent maintenant découvrir l'univers d'Enclave à 2.

Niveau contenu, l'intégralité des armes, ennemis et Boss prévus ont été implémentés pour enrichir la variété de combats. Visuellement, de nouvelles textures ont été ajoutées pour offrir plus de choix au niveau du level design. De plus, de nouvelles musiques originales ont été ajoutées au jeu pour enrichir l'ambiance sonore et générale de notre jeu.

Nous avons également amélioré l'expérience utilisateur, les menus ont été retravaillés pour plus de clarté et de cohérence avec l'environnement de notre jeu.

Aujourd'hui, nous pouvons dire que presque tout ce que nous avons imaginé a été concrètement réalisé. Enclave est plus qu'un prototype : c'est un jeu abouti, jouable, cohérent, et fidèle à l'univers que nous avons voulu construire.

2 Origines

2.1 Origines et nature du projet

Enclave n'est pas un jeu qui reprend les codes traditionnels des grandes entreprises qui ont tendance à produire des jeux dits triple-A, des jeux à budget conséquent, souvent critiqués pour leur manque d'âme, de personnalité, de créativité...

Enclave, par opposition, est un jeu dans lequel la créativité et l'inspiration jouent un rôle clef puisque notre objectif avec Enclave était de développer un jeu auquel nous aimerions nous même jouer plutôt qu'un jeu fait pour plaire au plus grand nombre. L'univers dans lequel Enclave se déroule est le fruit des envies et inspirations de chacun des membres d'Aura Corp. Tous ceux qui ont travaillé sur le projet ont eu l'occasion d'y faire passer leur message, de contribuer en apportant une part de leur personnalité au projet que ce soit à travers de simples détails visuels présents dans quelques niveaux ou encore à travers un ennemi remarquable aux mécaniques de jeu uniques et complexes.

Ce jeu est l'aboutissement d'un long travail passionné et dévoué dont l'identité vient plutôt de ceux qui l'ont conçu, car chacun de ceux qui ont contribué à ce projet y a mis du cœur.

Enclave est un Rogue-lite, c'est-à-dire un jeu reprenant certaines des mécaniques de jeu propres au Rogue-like. En d'autres termes, c'est un genre qui associe mort permanente du joueur et perte de progression lorsque celle-ci survient, sans pour autant qu'il y ait de génération procédurale d'environnements. Pour la réaliser Enclave nous avons puisé notre inspiration dans les œuvres qui nous ont marqué.

Dead Space, un jeu d'horreur se déroulant dans l'espace, pour l'ambiance.

Rogue, un jeu de donjon à mort permanente, pour le Gameplay.

The escapists pour le pixel art.

Le but du jeu est de survivre à une horde de créatures extraterrestres s'étant introduites dans un vaisseau de transport reconverti en arène par des mercenaires.

Notre héros y est retenu captif et a pour objectif d'atteindre le trentième niveau de ce vaisseau dans le but de s'échapper et retourner chez lui, sur la planète à laquelle il a été arrachée : Marvin-IV.

2.2 L'entreprise et l'équipe

Nous sommes Aura Corporation, une petite entreprise de développement de jeux vidéo basée à Paris. Nous considérons le jeu vidéo comme un art et aimons y investir notre créativité.

Notre collectif a décidé de se nommer ainsi pour différentes raisons. L'Aura, c'est l'atmosphère qui entoure ou semble entourer un être ou une chose. Ce terme est à connotation méliorative, il nous évoque l'art, la créativité, nous aimons donner une aura passionnelle et artistique à nos jeux. Corporation est une satire des entreprises à jeux dits triple-A mentionnées précédemment que l'on pourrait qualifier de corporation.

L'objectif de nos jeux est d'apporter aux joueurs des épreuves de plus en plus difficiles, qui apportent un défi. Nous visons donc un public composé majoritairement de personnes capables d'investir un bon nombre d'heures dans nos jeux, capables de persévérer et de ne jamais abandonner malgré la difficulté.

Nos jeux partagent également une identité visuelle commune. Nous développons souvent des jeux en deux dimensions avec vue en plongée. Un des thèmes récurrents et originaux de nos jeux est ce qu'on appelle un "Level Gimmick".

C'est une mécanique unique qui s'applique à un ou plusieurs niveaux, qui peut changer la manière dont certains objets fonctionnent et obliger le joueur à s'adapter, à changer sa manière de penser.

Je m'appelle Louis Plaire. Je suis le chef de ce projet et le président directeur général de Aura Corporation. Je veille au bon déroulement des divers travaux que nous menons au sein de l'entreprise et suis la glue qui la maintient soudée. Originaire du Val de Marne, j'ai passé une grande partie de ma vie en Normandie où j'ai appris les rudiments de la programmation. Après mes études d'ingénieur en informatique à l'EPITA, j'ai travaillé pour l'agence spatiale Européenne pendant 5 ans et j'ai pu y développer mes compétences en gestion de projet. Le manque de créativité que demandait mon métier à l'ESA m'a poussé par la suite à me

réorienter dans le monde du jeu vidéo indépendant, j'y ai travaillé les 3 années suivantes. J'ai notamment pu y développer "Free Bob", ma plus grande réussite. Au bout de ces années de développement en solitaire, j'ai gagné en ambition et ai décidé qu'il était temps de voir plus grand. En 2037 j'ai alors décidé de rassembler une équipe de gens passionnés pour fonder Aura Corp, l'entreprise qui laisse libre cours à l'imagination de ses membres !

Je m'appelle Sacha Skopan, je suis responsable du level design, du système d'événements ainsi que la création de l'univers d'Enclave dans l'entreprise Aura Corp. Originaire de Saint-Germain-en-Laye où j'ai passé une partie de ma scolarité et où j'ai appris les bases de la programmation ainsi que les bases de création d'histoire. J'ai par ailleurs des compétences en langues étrangères (anglais et allemand) acquises en partie lors de mes voyages à l'étranger. En effet, j'ai passé quatre ans au Japon dans le cadre d'une expatriation. J'ai été scolarisé au Lycée Français International de Tokyo. Après mes études à L'EPITA, j'ai été approché par Alpine qui m'a proposé un poste en tant que responsable des systèmes embarqués. Grâce à mes connaissances poussées en Formule 1 et en informatique, j'ai pu remodeler la philosophie d'Alpine. Ce qui nous a permis d'être champion du monde de 2032 à 2036. Suite 'à cela, je voulais m'orienter vers un secteur moins compétitif. En 2037 un ancien confrère de ma promotion d'EPITA m'a contacté pour un poste dans sa toute nouvelle entreprise, pour faire un jeu vidéo en tant que responsable du level design. La perspective d'une opportunité nouvelle m'a poussé à accepter.

Je me prénomme Ethan Bordas. J'ai fait mes études dans le 92 à Issy-Les-Moulineaux. De prime abord, on me décrit comme une personne empathique et à l'écoute des autres, favorisant la cohésion et la bonne entente d'équipe. Je suis une personne dotée d'une forte capacité d'analyse et de résolution de problèmes. A l'âge de 15 ans j'ai su combler les failles du site internet d'une célèbre marque. Lors des mes études à l'EPITA, j'ai consolidé mes aptitudes en programmation et en développement web. J'ai notamment perfectionné des langages informatiques comme le C-sharp, le python, le shell ou encore le html et le css. Alors que je cherchais un métier où je pourrais m'épanouir sans problème tout en faisant ce que j'aime : le jeu vidéo, j'ai été approché par Aura Corp., une entreprise de renom qui m'apporta exactement le cadre de travail que je cherchais. J'y suis maintenant responsable de la gestion de notre site web.

Je m'appelle Angel Pasquin, et je suis directeur artistique chez Aura Corporation. J'y ai élaboré les visuels, les musiques et les sons du jeu Enclave.

Originaire de Saint-Germain-en-Laye, c'est là que j'ai passé une grande partie de ma scolarité. D'es mon plus jeune âge, l'art, la musique et l'informatique m'ont captivé. Je passais des journées entières à dessiner, jouer de la guitare, et surtout à créer des jeux. Cette capacité à imaginer des histoires et des univers est rapidement devenue une passion dévorante. En 2023, j'ai décidé de poursuivre mes rêves d'enfant en rejoignant une école d'ingénieurs à Cergy, qui proposait un double cursus Ingénieur/Manager du numérique. Là, j'ai acquis les bases de l'informatique et de la programmation. Mais, face à de nombreux obstacles et insatisfactions, j'ai pris la décision de changer de cap pour intégrer l'EPITA, une école d'ingénieurs dont les enseignements me paraissaient plus en accord avec mes valeurs et répondaient à mes attentes. Après mes études, j'ai travaillé pendant près de cinq ans chez Ubisoft, une des plus grandes entreprises de jeux vidéo. En tant que game designer, j'ai pu affiner mon expertise et gagner de l'expérience. Cependant, cette entreprise est devenue pour moi "un miroir aux alouettes", j'ai ressenti le besoin de quelque chose de plus authentique, plus humain.

En 2037, j'ai fait un choix qui a changé ma vie, j'ai accepté l'invitation de Monsieur Louis Plaire, président d'Aura Corp. Une personne d'une rare bienveillance. Dès mon arrivée, je me suis senti accueilli, entouré de personnes en qui j'avais enfin confiance. Cette expérience n'a pas seulement transformé ma carrière. Elle a changé ma vie, redonnant un sens à ma passion.

Je m'appelle Martin De Dorlodot. Je suis le responsable du système d'objets (armes et accessoires), des stats, des buffs / debuffs ainsi que de l'équilibre de l'ensemble des capacités de l'univers d'Enclave. Je suis né en France à Neuilly-sur-Seine, et j'ai passé une grande partie de mon enfance à Courbevoie, avant de déménager à l'autre bout du globe, à Hong-Kong. Deux ans plus tard, avec ma famille, nous sommes partis vivre au Japon, où j'ai développé une grande passion pour les jeux vidéos et d'une manière générale, l'informatique. Après quatre ans et demi au Japon, je suis retourné en Europe, à Barcelone. J'y ai fait de nombreuses rencontres qui m'ont permis de m'épanouir. À l'âge de 17 ans, j'ai monté intégralement un ordinateur. Mon parcours international m'a permis d'élargir mes connaissances sur les cultures étrangères, et m'inspire beaucoup dans mes projets. Suite à mes études supérieures, j'ai intégré l'équipe de Cybersécurité de Black Rock, où je suis resté pendant 5 ans, puis j'ai travaillé dans les équipes de développements des jeux Roblox 2 et Fortnite 2 pendant 3 ans. En 2037, un de mes amis de l'EPITA m'a contacté avec un projet en tête : la création d'une entreprise et la réalisation de jeux du type Rogue-Lite et Rogue-Like. Et c'est

comme ça que je me suis retrouvé à travailler sur le projet Enclave, un Rogue-Lite créé par l'entreprise Aura Corp, en tant que responsable du système d'armes, d'objets et d'accessoires, ainsi que de l'équilibrage des objets et également en tant que Directeur Financier de l'entreprise.

3 Présentation chronologique et individuelle des fonctionnalités

3.1 Angel Pasquin

Concevoir Enclave n’a jamais été pour nous un simple projet ou juste un prototype de jeu. Dès le début, avec les premières idées, les premiers croquis et lignes de code, on a vu Enclave comme une création artistique à part entière. Chaque aspect visuel, sonore, interactif a donc été pensé pour servir un univers cohérent, vivant et immersif.

On a décidé qu’Enclave serait un jeu qui parle peu, voire pas du tout. Pas de dialogues : tout devait passer par l’environnement et les visuels... Le décor est le véritable narrateur du jeu. C’est l’univers lui-même qui raconte ce qui s’est passé. Dès les premières minutes, le joueur est plongé dans une sorte d’enclave fermée et étouffante.

3.1.1 Inspirations artistiques et ambiance visuelle

Les jeux

Dès les premiers jours de conception d’Enclave, nous avons cherché à définir un cadre visuel précis : oppressant, fermé, sale et cohérent. Pour y parvenir, nous avons puisé dans un large éventail d’influences, allant du cinéma de science-fiction horrifique aux jeux vidéo cultes et rétros. Ces inspirations ont non seulement guidé nos choix esthétiques, mais elles ont aussi modelé notre manière de raconter l’histoire d’Enclave.

ALIEN / Dead Space :

Le point de départ de notre réflexion artistique s’est naturellement tourné vers (Alien 1979), de Ridley Scott et Dead Space (2008). Alien est un film qui ne repose pas sur la terreur spectaculaire mais sur l’attente, le renfermement, et la suggestion, cela incarne tout ce que nous voulions créer. Dans Alien, ce n’est pas tant le monstre qui fait peur, c’est l’ambiance visuelle. Cela est quelque chose d’assez similaire dans Dead Space.

C’est exactement ce que nous avons voulu pour Enclave. Nos environnements sont faits de couloirs froids, et étroits, de décors abîmés, avec des taches de sang...

DOOM :

À cette horreur spatiale, nous avons voulu ajouter la violence graphique du Doom original (1993). Son pixel art, sa violence, ses monstres difformes nous ont influencés dans la construction de notre bestiaire. Là où Alien apportait le malaise, Doom nous a rappelé que le danger doit aussi avoir une forme. Nous avons utilisé des couleurs franches : rouges saturés pour la menace, verts acides pour l'infection, noirs/gris métalliques pour l'environnement. Nous avons volontairement évité les tons neutres ou apaisants. Rien dans Enclave ne devait sembler accueillant. Et comme dans Doom, nous avons placé l'action dans des environnements clos, segmentés, presque labyrinthiques. La station devient un labyrinthe sans repères clairs, ce qui oblige le joueur à observer et à mémoriser.

Le pixel art

Beaucoup associent encore le pixel art à une esthétique “rétro”, nostalgique ou simpliste. Avec moins de détails, chaque choix visuel prend plus de poids. La forme du monstre, la couleur d'un mur, la taille d'un couloir : tout devient signifiant. Le pixel art nous a également permis de sculpter l'ambiance à petite échelle. Là où un moteur 3D aurait pu donner une sensation de liberté, nous avons préféré la contrainte, la compacité. Cette réduction visuelle amplifie le sentiment de confinement, essentiel à notre récit. Le joueur n'est jamais à distance de l'horreur : il est enfermé avec elle.

Style Giger

Enfin, notre plus grande influence visuelle reste le travail de H.R. Giger. Son approche de la biomécanique, où la technologie semble vivante et la chair artificielle, a été une révélation. Nous avons cherché à intégrer cette esthétique dans nos créatures, mais aussi dans certaines textures murales ou objets interactifs.

En combinant les silences d'Alien/Dead Space, la brutalité de Doom, la structure du pixel art et la texture de Giger, nous avons cherché à forger un monde qui ne soit pas simplement effrayant mais profondément hostile et viscéral.

3.1.2 Le personnage principal

La première étape dans la création du jeu fut de trouver un personnage qui comme dans toute œuvre devienne quelque chose d'emblématique. Dans un jeu aussi silencieux et fermé qu'Enclave, où les dialogues sont absents, le personnage principal devient lui-même un vecteur de narration. Sa transformation au fil du jeu est un moyen pour nous de raconter une histoire sans mots. C'est pourquoi nous avons accordé une importance capitale à sa conception visuelle, et plus encore à son évolution à travers les armures.

Le joueur commence sa mission sans préparation. Il ne joue pas un héros mythique ni un super-soldat invincible. Il incarne un simple prisonnier, vêtu d'une combinaison orange, lâché dans un monde qui le dépasse. Ce point de départ, volontairement minimal, nous permet de poser un contraste fort avec la suite : plus le joueur progresse, plus il s'équipe, et plus son apparence change radicalement.

Les cinq armures :

Combinaison de prisonnier

Il s'agit de la tenue de base. Aucune protection, aucune défense. Le corps est visible. La couleur orange vif tranche avec l'environnement et souligne la vulnérabilité du joueur..

Spacesuit

Première amélioration tenue, plus complexe, visière transparente mais son corps reste lisible. On sent encore l'homme sous la protection.

Space Crusader Armor

Armure plus renfermée que la précédente, le visage n'est presque plus visible, la couleur rouge donne un côté plus agressif au personnage..

Star Hero suit

Armure emblématique d'Enclave, tenue plus militaire et renforcée, casque complètement fermé.

Enclave Suit

Silhouette plus massive, puissante Le joueur presque aussi monstrueux que ce qu'il affronte..

Symbolique et storytelling :

Au-delà de leur apparence, ces armures ont été pensées pour raconter quelque chose. Il ne s'agit pas simplement de montrer une montée en puissance, comme dans beaucoup de jeux vidéo classiques, mais de traduire un glissement de rôle : de témoin à acteur, de proie à prédateur, d'humain à machine.

3.1.3 Le bestiaire

L'univers d'Enclave est peuplé de créatures que nous avons conçues. Chacune d'entre elles est unique et ont leur propre histoire. Elles ne sont pas là pour décorer un niveau ou remplir un quota d'ennemis : elles racontent l'histoire de la station, par leur morphologie, leur comportement et leur lien avec l'environnement.

Nous avons développé 8 créatures distinctes, dont trois véritables boss, en prenant soin de leur donner une identité propre et une animation adaptée. Le joueur ne les rencontre pas toutes dès le début. Leur apparition suit une courbe pensée pour escalader la tension et la difficulté, jusqu'au dénouement final.

1. Prisonniers infectés ou dormeurs

Leur combinaison orange est encore visible, mais leur peau est nécrosée, leur membre déformé.

2. Le rempant

Monstre agressif et rapide, il ne possède pas de jambe mais se déplace à l'aide de ses bras puissants.

3. Le fort

C'est la force brute. Il se déplace à deux pattes, lourdement, mais avec précision.

4. L'Explosif

Une anomalie. Gonflé, pulsant, il approche lentement, et explose au contact. Il est le corps sacrificiel du bestiaire. Il ne survit pas à son attaque, mais sa simple présence est terrifiante.

Les boss :

5. L'Invocateur (Boss 1)

Le premier tournant du jeu. Il invoque, lentement, en levant les bras vers le ciel.

C'est une figure rituelle, un mal supérieur. Son design inspiré de Archville dans DOOM renforce la terreur dans Enclave.

6. La Reine Mère (Boss 2)

Il s'agit d'une masse informe, extrêmement rapide, elle est capable de rouler et de tout détruire sur son passage. Ces tentacules repoussants se déplacent en permanence, elle dégage quelque chose d'immonde.

7. Le Précurseur (Boss final)

Il est statique. Immobile, bras croisés, comme une statue. Il est la représentation du pouvoir total, d'un savoir interdit, d'une présence qui n'est ni vivante ni morte. Son design est très inspiré de Giger : symétrie parfaite, grand, forme humanoïde et biomécanique.

Chaque ennemi n'est pas seulement un obstacle. Il a une fonction narrative. Certains incarnent l'échec scientifique, d'autres la transformation physique, d'autres encore la fatalité. Le joueur apprend à les reconnaître, non pas seulement pour les éviter, mais pour comprendre ce qu'ils révèlent sur la station.

Le bestiaire agit comme un langage complémentaire. Il remplace les dialogues, Il dit ce qui s'est passé, ce qui va arriver, ce que le joueur risque de devenir.

3.1.4 Environnement

Dans Enclave, les textures ont été créées pour renforcer l'immersion le plus possible, on passe par des couloirs métalliques étroits, des salles de contrôle abandonnées, des laboratoires détruits, des cellules de confinement... Et parfois même des zones où la structure elle-même commence à muter, comme si la station était vivante ou contaminée de l'intérieur... Chaque décor très inspiré de ce que l'on peut trouver dans Alien où Dead Space est là pour renforcer le sentiment d'isolement et d'oppression.

L'idée était que le joueur puisse comprendre le monde sans qu'on lui parle : une salle pleine de sang, un laboratoire avec des corps. Ces éléments suffisent pour imaginer l'histoire et ressentir la tension. Le décor raconte ce que les personnages ne disent pas.

3.1.5 L'animation en sprite sheet

Lorsque nous avons entrepris de concevoir l'univers visuel d'Enclave, il ne suffisait pas de créer des personnages, des ennemis ou des environnements esthétiquement cohérents. Il fallait aussi les faire vivre, les animer de manière que leurs mouvements racontent quelque chose et donnent vie au jeu. C'est pourquoi nous avons opté pour une approche artisanale grâce à une application appelée Pixel studio. Toutes les animations du jeu ont été réalisées à la main frame par frame, sous forme de sprite sheets.

Dans un monde de plus en plus orienté vers l'animation vectorielle automatisée ou les modèles 3D, le choix du sprite sheet peut sembler étrange. Pour nous, il s'agissait au contraire d'un acte artistique volontaire. Chaque mouvement est un dessin. Le pixel art, pour être expressif, demande de la rigueur, de la logique, et une connaissance précise de la lisibilité du mouvement.

En optant pour ce système, nous avons pu créer des animations, où chaque créature se distingue immédiatement non seulement par son apparence, mais par sa manière de se déplacer. Cette différenciation est essentielle : elle permet au joueur de reconnaître instinctivement, dès les premières frames, le type de menace qu'il affronte, même sans interface explicite. Chaque animation en sprite sheet a été pensée dans un format réduit. Nous avons travaillé avec des résolutions basses, souvent entre 32x32 et 64x64 pixels pour les sprites d'ennemis standards, jusqu'à 128x128 pour certains boss. Cela nous obligeait à faire des choix radicaux : quelle posture privilégier ? quel mouvement est nécessaire ? comment rendre visible une intention en quatre ou six frames ?

Mais loin d'être un frein, cette contrainte a nourri notre créativité. Elle nous a poussés à condenser le geste, à lui donner un pouvoir symbolique.

En résumé, l'animation dans Enclave n'est pas un outil décoratif. C'est une part entière de la narration visuelle. En choisissant de tout animer à la main, pixel après pixel, nous avons voulu redonner au mouvement sa valeur expressive, émotionnelle, dramatique. Et dans un monde aussi silencieux que celui de notre jeu, chaque geste devient une parole.

3.1.6 Sound design et musiques originales

Dans un univers aussi visuellement dépouillé, fermé et hostile que celui d'Enclave, le son ne joue pas un rôle secondaire. Il est même l'un des piliers de l'expérience. Le sound design n'a pas été abordé comme un simple accompagnement des actions du joueur, mais comme un élément central de l'atmosphère, un tissu qui donne corps à l'environnement, un rythme et une immersion. Nous avons ainsi créé une bande-son (sur des logiciels tels que FL Studios et BandLab) et une ambiance sonore entièrement de nous-même. Composés, mixés et intégrés dans une seule et même logique artistique : plonger le joueur dans un univers auditif immersif, cohérent et angoissant.

Dans Enclave la musique est omniprésente. Pour rappeler en quelque sorte nos inspiration telle que DOOM. La musique occupe une grande place dans Enclave. Elle est constante, elle accentue l'action presque constante de notre jeu. Nous avons composé quatre morceaux originaux, chacun pensé pour une fonction dramatique spécifique.

Les trois musiques de combat :

Inspirées par les OST de Doom et Halo ces morceaux s'inscrivent dans le registre du métal industriel, mêlant guitares électroniques saturées, percussions, synthétiseur. Ces musiques ne cherchent pas à être "belles" : elles doivent traduire la brutalité, l'intensité, l'urgence. Chaque combat devient une transe rythmée.

Le thème final (crédits) :

À l'opposé, nous avons composé une pièce calme et introspective, directement inspirée de "Heavy Price Paid" dans le menu de Halo 2. Elle intervient à la fin du jeu, au moment des crédits, et marque une rupture radicale.

Le chaos est terminé. Cette musique agit comme une pause émotionnelle, une forme de calme après l'épreuve.

En conclusion, l'univers sonore d'Enclave n'a pas été pensé pour séduire. Il a été conçu pour plonger le joueur dans une sorte de rush constant. Pour enfermer

le joueur dans un espace acoustique puissant. Il ne cherche pas à distraire, mais à renforcer l'ambiance, à appuyer l'invisible.

3.1.7 Interface et HUD

Dans Enclave, l'interface n'est pas là pour en mettre plein la vue. Elle est là pour s'effacer, pour ne jamais briser l'immersion. Dès le départ, nous avons refusé les HUDs modernes et brillants. Ici, l'interface est brute, utilitaire, fonctionnelle. Elle fait partie intégrante de l'univers. Tout ce que le joueur voit à l'écran est sa barre de vie, c'est un design simpliste qui ne déconcentrera pas le joueur et permet d'avoir une meilleure vision sur ce qui se passe à l'écran.

Le design des barres de vie est très représentatif de ces personnages : pour le personnage principal, nous avons intégré un style futuriste/ militaire en métal, et la jauge couleur rouge sang pour rappeler l'humain. Les autres barres de vie sont attribuées au 3 Boss, leur design les représentent, les 3 barres de vie sont unique et ont un style organique qui reprend leur couleur : du feu pour l'invocateur, des tentacules pour la Reine mère, le fameux style Giger pour le précurseur.

Malgré son intégration immersive, le HUD reste lisible. En combat, on identifie immédiatement l'état du personnage. Les effets visuels sont calibrés pour ne jamais gêner la lecture : pas de flashes inutiles, pas de surinformation.

3.1.8 Police d'écriture

Dans le cadre de la création de l'identité visuelle d'Enclave, nous avons également conçu notre propre police d'écriture, utilisée notamment pour le logo du jeu et certains titres clés de l'interface. L'idée était simple : aucune typographie existante ne correspondait vraiment à l'ambiance que nous voulions créer. Il nous fallait une écriture unique.

Cette police a été dessinée pixel par pixel, dans le même esprit que le reste de notre direction artistique. Elle est rigide, géométrique, compacte. Chaque lettre a été pensée pour évoquer l'isolement, la fermeture, le danger.

Utiliser cette typographie dans le logo du jeu permet de donner tout de suite le

ton : froid, fermé, brutal. Mais au-delà du titre, nous avons également utilisé cette police dans certains écrans internes du jeu, ce qui renforce la cohérence globale de l'univers.

Créer notre propre police nous a permis de pousser plus loin encore le sentiment qu'Enclave est un monde fermé sur lui-même, avec son propre langage, ses propres codes, visuels comme textuels.

3.1.9 Jaquette et logos

Dans un jeu comme Enclave, tout commence bien avant que le joueur appuie sur "Démarrer". La jaquette est la première image qu'il voit. Elle doit résumer l'univers, l'ambiance, la promesse. Elle n'est pas un simple visuel accrocheur : elle est le condensé de tout ce que le jeu représente.

La jaquette :

Nous avons dessiné la jaquette à la main dans un style pixélisé mais plus détaillé que les visuels de notre jeu. Elle représente le personnage principal dans son armure, debout, figé, dans un environnement rouge sang et infesté.

Inspirations directes :

Halo 2 : posture solitaire, ambiance de guerre imminente. Dead Space : isolement, silence oppressant.

Doom (1993) : énergie brute, tension frontale, violence. Nous n'avons pas cherché à représenter une scène précise du jeu, mais à faire sentir ce qu'est Enclave : un jeu complexe où le joueur devra affronter des hordes de monstres pour s'en sortir, mais qui malgré la terreur ne recule devant rien pour sa survie.

Le logo Enclave :

Le logo du jeu a été conçu pour refléter la fermeture, la densité, le confinement. Typographie angulaire, légèrement pixelisée, les lettres blanches rappellent les étoiles dans l'espace, une écriture en lettres majuscules et épaisses qui donnent un côté oppressant au logo.

C'est une identité visuelle qui veut rester gravée, pas séduire.

Le logo d’Aura Corp :

Visible sur la jaquette (en bas à droite), le logo de notre entreprise : Aura Corp est circulaire, noir et blanc, orné de piques. Ces piques ont une signification : elles symbolisent la difficulté du jeu, l’hostilité de l’environnement de nos jeux.

Jaquette, typographies, symboles : tout a été conçu pour qu’Enclave ait une identité forte dès le premier regard.

On ne veut pas simplement attirer. On veut prévenir. Le monde dans lequel le joueur s’apprête à entrer n’est pas là pour l’accueillir. Il est là pour le tester.

3.2 Louis Plaire

3.2.1 L’intelligence artificielle

Dans un jeu de combat, il est bon d’avoir de nombreux adversaires possédant des mécaniques de jeu variées de façon à pousser le joueur à s’adapter à différentes situations. Enclave dispose de six IA différentes ayant chacune une difficulté et un comportement propre. Ces Intelligences artificielles nous les avons nommées : Le Dormeur, Le Fort, L’Explosif, L’Invocateur, La Reine et Le Précurseur.

Le Dormeur est la première intelligence artificielle que nous avons développé, elle est incarnée par deux ennemis différents le “Dormeur basique” et le “Rampant”. Le dormeur basique est un ennemi faible qui reste immobile jusqu’à ce que le joueur passe près de lui, ce qui le “réveille”, il se met alors à le pourchasser jusqu’à ce qu’il s’éloigne et soit hors de portée. Le “Rampant” fonctionne de la même manière mais il est plus puissant et sa portée est plus grande.

Le Fort quant à lui, fonctionne d’une toute autre manière. Alors que les ennemis utilisant l’IA du dormeur ne font qu’avancer en direction du joueur sans se soucier des obstacles sur leur chemin, Le Fort lui utilise un algorithme de PathFinding nommé A*. Cela lui permet de poursuivre le joueur efficacement tout en évitant les obstacles. De plus, celui-ci ne s’arrête jamais de poursuivre le joueur et le harcèle jusqu’à ce qu’il soit détruit.

L’Explosif est une IA similaire à celle du Dormeur, il court vers le joueur lorsque

celui-ci est proche, sans se soucier des obstacles. Il se démarque du Dormeur par sa vitesse, sa faible quantité de vie et le fait qu'au contact du joueur il explose, se donnant ainsi la mort et infligeant une grande quantité de dégâts au joueur avec qui il entre en collision.

L'Invocateur, La Reine et Le Précurseur sont un type d'ennemis tout fait différent. Là où il était possible de trouver plusieurs Dormeurs, Forts ou Explosifs par niveau, L'Invocateur, La Reine et Le Précurseur sont des ennemis uniques, qu'il n'est possible de trouver que dans un seul niveau et qu'il est plus difficile pour le joueur de vaincre. Nous qualifions ces ennemis de "bosses", un terme anglais signifiant "chefs" très utilisé pour qualifier ce genre d'entité de jeux vidéo. L'Invocateur possède un style de mouvement très particulier, Il avance vers le joueur jusqu'à une certaine distance mais lorsqu'il est en est trop près il s'en éloigne, ce qui pousse le joueur à le pourchasser sans relâche à travers le niveau. Occasionnellement, L'Invocateur invoquera une quantité variable de dormeurs autour de lui. Lorsqu'il les invoque il est immunisé aux dégâts du joueur qui doit alors se charger de défaire les Dormeurs pour rester en vie.

La Reine, elle, dispose d'un algorithme moins complexe, elle se contente de charger rapidement le joueur à intervalles de temps réguliers. Cela oblige le joueur à se mouvoir constamment à travers le niveau tout en observant attentivement la trajectoire de La Reine pour trouver le moment opportun auquel lui infliger des dégâts.

Enfin, Le Précurseur, c'est le dernier boss que le joueur doit vaincre. S'il y parvient, il déclenche alors la fin du jeu et termine Enclave. Le Précurseur, contrairement aux autres ennemis ne se déplace pas, il est immobile, dispose d'une grande quantité de points de vie et possède deux attaques différentes qu'il utilise sans relâche contre le joueur, simultanément ou non. Sa première attaque est une vague de projectile qu'il envoie à travers tout le niveau, il commence par en envoyer un premier à gauche, un second toujours vers la gauche mais plus en bas et continue ainsi jusqu'à envoyer un dernier projectile vers la droite pour former un demi cercle. Pour sa deuxième attaque il enregistre la position du joueur et y fait apparaître quelques secondes plus tard, un trou noir infligeant une grande quantité de dégâts. Il est facile pour le joueur d'esquiver une telle attaque puisqu'elle frappe sur sa position après un laps de temps à un endroit auquel le joueur se trouvait et pas à l'endroit auquel il se trouve. La seule manière pour le joueur d'être touché

par cette attaque est d'être immobile.

3.2.2 Le multijoueur

Le multijoueur est un aspect fondamental d'Enclave, il permet à nos joueurs de se retrouver, de coopérer et de triompher ensemble. Son développement fut le plus long et le plus fastidieux de toutes les fonctionnalités que nous avons implémenté en jeu.

Plutôt que de développer le multijoueur d'Enclave avec une solution native à Unity, notre moteur de jeu, comme NetCode nous avons fait le choix d'utiliser un outil extérieur développé par First Gear Games nommé Fish-Networking qui sera ici raccourci en FishNet. FishNet permet à n'importe quel ordinateur de se comporter comme hébergeur ou client d'un serveur de jeu et ce de façon très simple. Cette solution est idéale pour un jeu comme Enclave dont le multijoueur est local.

Il nous a été aisé d'implémenter un système permettant à deux joueurs de se connecter à un même niveau et de le parcourir ensemble. Cependant, il fut bien plus difficile de rendre les intelligences artificielles que nous avons développées avant le multijoueur compatibles avec celui-ci.

En effet, les ennemis utilisent le système de simulation physique de Unity pour se déplacer. Ce système de simulation effectue de nombreux calculs complexes et à intervalles très courts, ce qui fonctionne très bien lorsque le joueur joue seul et que les calculs peuvent s'exécuter directement sur sa machine. En revanche, lorsque plusieurs joueurs sont connectés, il est difficile pour le serveur d'envoyer autant d'informations aussi régulièrement aux clients, ce qui peut provoquer des désynchronisations entre ceux-ci et perturber l'expérience de nos joueurs qui pourraient par exemple chacun voir un même ennemi à deux endroits différents sur leur ordinateur respectif.

Pour résoudre ces problèmes de synchronisation, il nous a fallu implémenter un système appelé "Client Side Prédiction" ou CSP. Le CSP est un système qui consiste à faire les calculs de simulations physiques de Unity sur le serveur et d'appliquer ensuite ces calculs à tous les clients. De cette manière, on s'assure que tous les joueurs voient bien la même chose sur leur écran, qu'ils sont synchronisés.

Les principales difficultés que nous a posé FishNet sont son manque de documentation et son nombre d'utilisateur trop faible. En effet, lorsqu'on utilise un outil en informatique il est bon d'avoir à disposition un manuel qui référence toutes ses fonctionnalités, une documentation. FishNet possède une documentation en ligne et quelques tutoriels, mais la plupart du temps, c'est uniquement pour les fonctionnalités les plus utilisées. Pour les fonctionnalités plus poussées ou moins utilisées, il n'y a en général pas de tutoriels et la documentation est souvent peu claire ou inexistante.

Il est aussi plus facile d'apprendre à se servir d'un outil lorsque bon nombre de personnes ont déjà essayé de l'utiliser avant puisque les utilisateurs aguerris peuvent aider les novices. FishNet ne dispose pas d'assez d'utilisateurs expérimentés qui puissent aider les débutants. Cela nous fit perdre beaucoup du temps puisqu'il nous fallut développer certaines fonctionnalités sans savoir comment nous y prendre. Bien que le développement du multijoueurs soit terminé, quelques problèmes subsistent sans que nous puissions les résoudre, notamment des problèmes de latence et de connexion. La meilleure solution à ces problèmes serait d'abandonner FishNet pour le multijoueur et de passer à NetCode, la solution native à notre moteur de jeu.

3.2.3 Direction générale du projet

Au cours du développement d'Enclave nous avons choisi d'implémenter ou non certaines fonctionnalités, nous aimerions revenir sur certains de ces choix à travers lesquels sont mises en exergue certaines des volontés de notre entreprise. Tout d'abord, il nous a été conseillé par le jury lors de la deuxième soutenance d'ajouter des barres de vie aux ennemis.

Après quelques recherches et mûre réflexion de la part de chacun des membres de l'entreprise, nous avons décidé de ne pas suivre les préconisations du jury. Enclave est un jeu dynamique, où le danger est partout.

Ainsi, il est parfois possible d'avoir plusieurs dizaines d'ennemis autour du joueur; une barre de vie à côté de chaque ennemi surchargerait l'écran du joueur qui n'y verrait alors plus guère. Il s'avère également que les joueurs de jeux vidéo en tous genre témoignent régulièrement d'un certain inconfort amené par un usage excessif d'éléments d'interface utilisateurs encombrants dans la plupart des jeux sortis par des entreprises de renom. Nous préférons laisser les joueurs d'Enclave découvrir par eux même la quantité de points de vie de chacun des ennemis.

Dans les derniers jours du développement d'Enclave, lorsque nous avons intensifié les tests du jeu, nous nous sommes rendus compte qu'il était facile de simplement esquiver les ennemis, éviter le combat et passer au prochain niveau sans aucun risque. Un réel problème pour nous puisque nous avions pour objectif de faire un jeu difficile dont seuls les plus aguerris pourraient venir à bout. C'est un problème pour lequel il y a plusieurs solutions. Nous avons par exemple pensé à fermer les portes d'une salle lorsque le joueur y rentre et les rouvrir lorsque tous les ennemis de la salle ont été tués. Cependant nous aurions perdu la liberté du joueur de se déplacer dans le niveau comme il le souhaite ce qui était hors de question. Nous en sommes finalement venus à trouver une solution plus originale. Il est de notre avis qu'un bon jeu vidéo se démarque par une mécanique unique qui force le joueur à le penser autrement que des jeux du même genre auxquels il aurait pu jouer le rendant ainsi moins dépendant de son expérience et plus à même de vivre le jeu comme une expérience nouvelle. C'est en apportant une telle mécanique que nous avons pallié au problème précédemment mentionné. Le joueur perd constamment des points de vie. Puisque les ennemis lâchent à leur mort de petits objets qui soignent le joueur, faire baisser la vie du joueur le pousse à s'arrêter les combattre pour regagner des points de vie. De plus, cette fonctionnalité se marie parfaitement avec l'univers du jeu. Le joueur est entouré de monstres et de parasites, son environnement lui est néfaste, naturellement, y passer du temps l'affaiblit.

Le personnage que contrôle le joueur fait partie des éléments centraux d'un jeu vidéo, il définit le cadre dans lequel le joueur pourra interagir avec son environnement. Dans Enclave, le joueur est un personnage situé au centre de l'écran qui peut se déplacer sur les deux dimensions du jeu. Nous avons pris la décision de rendre son contrôle précis, il se déplace vite et s'arrête instantanément au relâchement d'une touche. Cela permet au joueur d'avoir un contrôle absolu sur son personnage et son style de jeu, ce qui évite les frustrations ou le sentiment de perte de contrôle de ses actions.

Cette idée de maîtrise de son avatar nous a été inspiré par l'oeuvre de Maddy Thorson, "Celeste" un jeu de plateforme dans lequel on incarne un personnage dont les mouvements sont précis et qui est capable "dash", c'est à dire de se déplacer presque instantanément vers les endroits qui lui sont proches. Ce "dash" nous en avons un prototype dans Enclave aux premiers jours du développement. Nous l'avons finalement retiré puisque c'est une fonctionnalité que nous avons trouvé incohérente avec le jeu dans sa globalité.

3.3 Ethan Bordas

3.3.1 Les débuts dans le projet

Nos premiers pas dans la programmation se sont faits grâce à ce projet. Avant cela, notre expérience se limitait aux quelques travaux pratiques réalisés à EPITA. On percevait le code comme quelque chose de complexe, d’inaccessible. Mais en nous lançant dans ce projet, nous avons découvert Unity. Dès l’ouverture du logiciel, son interface, ses boutons, ses composants... tout nous a semblé confus. On ne savait pas par où commencer.

Heureusement, nous n’étions pas seuls. Nous avons passé beaucoup de temps à visionner des tutoriels sur YouTube pour assimiler les bases. Certains créateurs de contenu expliquent les choses de manière très claire, ce qui nous a énormément aidés. Et surtout, nous avons pu compter sur Louis, notre directeur de projet, qui maîtrisait déjà très bien Unity. Il a su nous guider et répondre à nos questions.

C’est également à ce moment-là que nous avons commencé à apprendre le C#. Au début, c’était un véritable casse-tête : des erreurs incompréhensibles, des accolades mal fermées, des messages d’erreur à répétition... Mais à force d’essais, de tests et de corrections, nous avons commencé à en saisir la logique. Nous avons compris que, par le code, nous pouvions déclencher des actions, interagir avec des objets et créer du comportement dans le jeu. Cette compréhension progressive nous a donné confiance pour la suite.

Nous nous souvenons encore de ce premier moment où une ligne de code a fonctionné comme prévu. Ce n’était peut-être pas grand-chose, mais pour nous, c’était énorme. Cette réussite a été un déclic, qui nous a permis d’appréhender plus sereinement les autres aspects du projet. Grâce aux tutoriels et aux conseils de Louis, nous avons réussi à accomplir des choses que nous pensions impossibles au départ, et que nous vous présentons dans ce cahier des charges.

3.3.2 Les menus

Après avoir compris les bases du langage C# et les fondements d’Unity, nous nous sommes rapidement confrontés à un autre défi : les menus. Ce qui pouvait sembler simple à première vue, quelques boutons sur un écran, s’est révélé bien plus complexe une fois dans l’éditeur. Ethan, en particulier, a été le premier à se

lancer dans cette aventure avec son script nommé MenuManager. Il pensait au départ qu'un menu, c'était juste une affaire d'interface. En réalité, chaque bouton, chaque interaction, chaque transition entre les scènes demandait une logique bien précise, et donc, beaucoup de code.

Dès les premiers essais, les erreurs se sont enchaînées. Une fonction mal nommée, un GameObject non assigné, un panel qui refusait de s'afficher... À ce moment-là, on a compris qu'Unity ne pardonne rien. Le moindre oubli peut bloquer le jeu entier. Et pourtant, quand le premier menu a enfin fonctionné, ce fut une petite victoire. Cliquer sur "Jouer" et voir la scène se charger correctement nous a procuré une vraie satisfaction. C'était la preuve que notre code commençait à prendre vie.

Le menu Game Over a été l'un des passages les plus difficiles. On voulait au départ un système propre et automatisé : un panel qui s'affiche à la mort du joueur, avec une animation, un fond, des boutons. Mais rien ne marchait comme prévu. Prefab mal instancié, canvas invisible, caméra mal placée... Après plusieurs tentatives frustrantes, on a fini par simplifier. Sur les conseils de Martin, on a opté pour un simple changement de scène. Et là, tout s'est mis à fonctionner. Parfois, les solutions les plus simples sont les plus efficaces.

Le système de mort, au final, repose sur deux scripts bien séparés. Dans PlayerCharacter, une fonction Die() vérifie si la santé tombe à zéro ; si c'est le cas, elle supprime les données d'armes et charge une scène appelée GameOverScene. De l'autre côté, le script Dead gère l'interface : il affiche le panel de fin de partie et propose les options de retour au menu ou de recommencer. Ce découpage nous a appris l'importance de la structuration du code et de la séparation des responsabilités. En parallèle, on a aussi travaillé sur d'autres éléments de l'interface : le menu pause, les crédits, l'écran de fin... On a appris à interagir avec les Canvas, à activer ou désactiver des panels, à gérer les événements des boutons. Chaque fonctionnalité nous a demandé du temps, des recherches, des tests... Mais à la fin, tout fonctionnait ensemble de manière cohérente.

C'est en grande partie grâce à cette étape que nous avons progressé en C#. On a appris à mieux nommer nos fonctions, à utiliser les SerializeField, à organiser notre logique de manière claire. Et surtout, on a compris que dans le développement,

la partie la plus difficile n'est pas d'avoir une idée, mais de la faire fonctionner correctement.

3.3.3 Site web

En parallèle du jeu, on a aussi développé un site web pour accompagner le projet, accessible à cette adresse <https://louisplaire.github.io/EnclaveWeb/index.html>. C'est Louis et Ethan qui ont pris en main cette partie. Ils ont codé l'ensemble en HTML et CSS, d'abord avec des éléments simples, puis en y ajoutant des effets de style, comme un fond défilant pour donner de la profondeur et du dynamisme à la page.

Parmi les difficultés rencontrées, le choix de la police d'écriture a été l'un des points les plus surprenants. Il fallait quelque chose de lisible, mais qui reste cohérent avec l'identité visuelle du jeu. Notre équipe a testé de nombreuses polices avant de trouver celle qui convenait parfaitement. Ce travail, affiné grâce aux conseils de notre enseignante, nous a permis de comprendre à quel point le moindre détail visuel peut transformer l'expérience de l'utilisateur.

Au fil des semaines, de nouveaux éléments ont été ajoutés au site : Une présentation de l'équipe,

- Une frise chronologique retraçant l'évolution du projet,
- Une galerie d'images du jeu.

Voir le site évoluer en parallèle du développement de notre jeu nous a vraiment motivés. Chaque ajout le rendait plus professionnel, plus proche d'un vrai site de présentation de jeu. Ce travail sur le web nous a appris beaucoup : l'importance du design, de la hiérarchie visuelle, et de l'expérience utilisateur. Ce qu'on apprenait dans le jeu ou le site web nous servait dans l'autre. Par exemple, l'écran de Game Over a été conçu pour avoir un réel impact visuel, en s'inspirant des principes de design utilisés sur le site.

3.4 Martin de-Dorlodot

Enclave est un jeu où le combat est indispensable pour sortir vainqueur des nombreux conflits qui sont amenés au joueur. Afin de permettre au joueur de profiter d’une expérience de jeu où il a accès à un système de combat satisfaisant, qui ne rendrait le jeu ni trop simple, au point où l’on s’en lasse, ni trop dur, au point où il semble impossible, il a fallu développer un système d’armes et d’armure capable de répondre précisément à ces attentes. Pour réaliser ce système d’arme, nous avons tout d’abord vu les choses en grand : trois classes d’armes différentes, avec chacune leurs spécificités, leurs avantages, leurs inconvénients, et des armes spéciales qui seraient faites indépendamment des autres. Les trois classes d’armes que nous avons mises au point étaient : les armes à feu, les armes de mêlée et les rayons lasers.

Les armes à feu auraient l’avantage de la portée, pouvant engager des combats à distance, mais seraient moins utiles lorsqu’une horde d’ennemis encercle le joueur dans un espace confiné. Les armes de mêlée auraient une portée courte, mais efficace, permettant au joueur de se débarrasser de ses adversaires en deux ou trois coups au corps à corps, avec des dégâts de zone, lui donnant une grande capacité de “crowd control”, ou “contrôle de foule”, un terme utilisé pour décrire une arme ou un effet qui peut permettre de garder un ou plusieurs ennemis sous le contrôle du joueur, de manière à ne pas se faire déborder. Bien sûr, ces armes seraient désavantageuses ou présenteraient un désavantage lorsqu’un ennemi a la possibilité d’attaquer le joueur à distance.

Les rayons lasers auraient la capacité d’envoyer un rayon qui traverserait les ennemis, en causant de faibles dégâts, mais à plusieurs ennemis en même temps. Certains lasers seraient également dotés de la capacité d’infliger des “debuffs”, donc des effets négatifs sur les adversaires. Ces effets auraient donné au joueur une autre manière d’effectuer du “crowd control” sur ses adversaires. Pour ce qui est des armes spéciales, l’idée était de faire une arme qui pouvait utiliser une attaque de mêlée et de distance, comme une épée qui envoie des projectiles par exemple.

Lorsque nous avons créé ce système, nous avons commencé par faire les armes à feu, dans un premier script : l’objectif de ce script était de nous créer une base pour toutes les armes à feu à venir. Pour que le joueur puisse interagir avec l’arme, nous

lui avons donné un `BoxCollider2D` et un `RigidBody2D`, des composants de Unity qui permettent aux éléments d'interagir entre eux et d'être soumis à différentes forces. , de manière à ce que lorsque le joueur entre en collision avec l'arme, il équipe ladite arme. Ensuite, pour que l'arme tire un projectile, nous avons créé un projectile `Bullet`, qui est un `GameObject`, un élément que l'on peut placer dans une scène et auquel on peut ajouter différents composants qui nous permettront de modifier les propriétés de cet élément. que l'arme prendra en argument. Lorsque le joueur fait un clique gauche, l'arme va créer un clone de `Bullet`, et le projeter depuis son `"FirePoint"`, un composant qui détermine le point de départ des balles, pour qu'elles ne rentrent pas en collision avec le joueur, et aussi qu'elles sortent bien du canon de l'arme. à une vitesse donnée dans la direction de la souris. Lorsque le projectile rentre en collision avec un ennemi, il lui inflige des dégâts et peut également, dépendant de l'ennemi et de la vitesse du projectile, le repousser en arrière.

Ensuite nous avons créé les armes mêlées, mais c'est là où nous avons réalisé un problème : les différents types d'armes ne sont pas reliés entre eux par une classe parente, ce qui fait que le joueur ne pouvait pas équiper des armes mêlée s'il pouvait équiper des armes à feu et vice-versa. Pour régler ce problème, nous avons créé une classe qui a accès à l'ensemble des fonctions, méthodes et types de Unity, et nous avons fait en sorte que les deux classes d'armes existantes soient liées à cette classe.

Nous avons donc un moyen de faire en sorte que le joueur puisse équiper n'importe quel type d'arme, puisqu'elles héritent toutes de la classe `Weapon`. Pour ce qui est des armes de mêlée, leur fonctionnement est assez contre-intuitif, il faut l'admettre. Les armes mêlées ont eu deux types de fonctionnement. Le premier était de projeter un `Raycast2D` en direction de la souris tant que le joueur maintient le clic gauche de la souris. Un `Raycast2D` fonctionne de la manière suivante : on projette un rayon dans la scène sur une certaine distance. Une fois que ce rayon rentre en collision avec un objet, on a accès à son `GameObject`, le `gameObject` d'un élément étant l'ensemble des composants de cet élément, ce qui nous permet de le manipuler. Unity nous permet également de filtrer le type d'éléments avec lesquels on rentre en collision, en utilisant ce qu'on appelle un `Layer`. Nous avons donc créé un `layer` pour les ennemis de manière à ce que lorsque le `Raycast 2D` touche un ennemi, nous pouvons faire en sorte que cet ennemi prenne des dégâts. Etant donné que la première vraie arme de mêlée implémentée dans le jeu était une tronçonneuse, ce fonctionnement était parfait pour ce type d'armes. Mais cela

était moins cohérent quand nous avons implémenté des armes de mêlée à base de lames. Effectivement, pour ce type d'armes, il est mieux d'avoir un programme qui fait en sorte qu'il y ait une animation lorsque l'épée est utilisée. Donc nous avons mis au point un deuxième type de fonctionnement. On gardait toujours le premier de manière à ce que les tronçonneuses garde ce fonctionnement qu'il leur convenait, en vérifiant simplement via une nouvelle variable si l'arme était une tronçonneuse ou non. Pour le deuxième fonctionnement des armes mêlées, nous avons tout d'abord fait en sorte que l'arme soit immobile, et soit de taille réduite, comme posée sur la ceinture du personnage du joueur. Une fois que le joueur fait un clic gauche, l'arme se repositionne, elle pointe vers légèrement au-dessus de la souris, en reprenant sa taille originelle. Ensuite, elle effectue une animation vers le bas, une rotation d'environ 30°. Une fois la rotation effectuée, elle revient au niveau de la ceinture du personnage du joueur, et reprend sa taille réduite. Le Raycast 2D suit l'épée tout le long de l'animation.

Cela permet d'avoir une manière de représenter le coup d'épée de manière fidèle et fluide. En ce qui concerne les rayons lasers et les effets, ce sont les domaines dans lesquels nous avons rencontré le plus de problèmes. Il était évident que la meilleure manière de faire fonctionner les lasers était d'utiliser un Raycast, cependant cela s'est avéré plus difficile que prévu. En effet, il fallait que le laser puisse traverser les ennemis et qu'il s'arrête quand le collider rencontré est un obstacle. Il fallait également créer un visuel pour ce laser, ce que nous avons fait à partir d'un line renderer, mais cela aussi était bien plus dur que prévu.

Pour faire en sorte que le laser puisse traverser plusieurs ennemis, nous avons décidé d'utiliser un Raycast2DAll, une fonctionnalité de Unity qui va créer une liste de Raycast2D et qui va nous permettre de les manipuler. Similairement au Raycast 2D classique, le Raycast2DAll va d'abord envoyer un premier Raycast2D dans la scène dans la direction donnée. Lorsque ce laser va entrer en collision avec un élément, le Raycast2DAll stocke le Raycast2D dans la liste et en projette un nouveau, jusqu'à ce qu'il atteigne soit la distance finale ou bien qu'il atteigne un autre élément, ou dans ce cas il stockera à nouveau le Raycast2D, en projettera un autre, et ainsi de suite.

Comme avec le Raycast classique, on peut appliquer un filtre dépendant du layer de l'élément avec lesquels des différents Raycast2D entrent en collision. Dans le cas de nos Rayons lasers, les layers sont ceux des ennemis et des obstacles. On

projette les Raycast2D, et ensuite on parcourt la liste : si un élément est un ennemi, alors on lui inflige des dégâts, et on continue de parcourir la liste. Si par contre c'est un obstacle, on sort de la boucle.

Pour ce qui est du visuel, on a donc utilisé un LineRenderer de Unity, pour afficher le laser. Un LineRenderer est un composant de Unity qui trace une ligne entre différents points, sauf que dans notre cas, il ne s'arrêtait pas forcément sur un point. Il était difficile de savoir comment faire pour que le LineRenderer suive les Raycast2D. Une solution était de parcourir la liste du Raycast2DAll sur le LineRenderer aussi, afin de pouvoir tracer une ligne entre le FirePoint de l'arme et le dernier point de collision. Le premier problème était que si aucun Raycast2D n'avait de collision, le visuel du laser se dirigeait vers le centre de la scène. Étant donné que les levels seraient entourés de murs qui sont sur le Layer des obstacles, ce problème ne causerait pas de problèmes in-game, mais malheureusement pour nous ce n'était pas le seul problème.

Effectivement l'un des problèmes qui a rendu l'implémentation des lasers impossible dans la version finale du jeu est liée au multijoueur. Lorsque l'utilisateur serveur utilise, il n'y a aucun problème avec l'utilisation, le laser s'affiche bien sur le serveur et sur le client. Cependant, si le client essayait d'utiliser le laser, ni le visuel, ni le Raycast2DAll ne marchait. Il était complètement buggé du côté du client. Et si cela ne suffisait pas, l'atout principal des lasers, les effets étaient également buggés.

Les effets étaient censés être les grands avantages des lasers. Pour faire les effets, nous avons créé une classe et un Enum. Un Enum est un type de classe qui stock une liste de types. Cet Enum nous permettait de stocker tous les types d'effets, et la classe permettait de les faire fonctionner. Différents effets étaient prévus, tel qu'un effet infligeant des dégâts sur le temps, pour un nombre de dégâts fixes, croissants et un pourcentage des Points de Vie (ou PV) de l'ennemi, une réduction des dégâts que l'ennemi inflige, une réduction de sa défense, un ralentissement de sa vitesse.

Le premier effet testé était un effet de dégâts infligés sur le temps, et cet effet marchait bien. Il avait été testé en solo avec le laser et marchait comme nous le souhaitions. Nous les avons laissées de côté quelques temps, et lorsque nous avons

fait des modifications pour régler les différents problèmes de multijoueur, les effets ne marchaient plus.

Nous avons alors pris la décision de laisser les lasers et les effets en dehors de la version finale du jeu, car ils nous causaient trop de problèmes et trouver des solutions à ces problèmes prenait trop de temps.

Un autre concept que nous avons abandonné est celui d'une arme spéciale qui serait une épée qui enverrait des projectiles à chaque coup. Le problème est que cela prendrait trop de temps à configurer. Nous avons alors envisagé de faire autre chose : un autre style de mêlée qui serait du combat rapproché à très courte portée ou le personnage utiliserait un genre Cestus, un type de gants utilisé en combat. Cela nous permettrait d'implémenter au moins une nouvelle arme mêlée, et de faire une autre arme spéciale : un gant qui fait des coups au corps à corps avec un clic gauche, et qui envoie des projectiles avec un clic droit.

Cependant, il était évident que cela n'était pas une priorité comparée aux autres tâches qu'il nous restait à accomplir. Les armes spéciales sont donc devenues des armes qui utilisent les scripts basiques, mais ont une apparence et des stats qui les séparent du reste. Le Railgun, le Sniper Laser, le Minigun Laser et le Creuset sont des exemples de ces armes.

Nous avons tout de même créé plusieurs armes à feu et de mêlées, afin d'avoir un système d'arme de plus diverse possible. Parmi ces armes on retrouve : Un Minigun basique, un minigun se rapprochant du M61 Vulcan qui existe dans la vraie vie, mis au point par l'armée américaine, et un Minigun Laser extrêmement puissant. Un fusil d'assaut basique comme première arme, une SMG laser qui tire légèrement plus lentement mais dont les balles ont une vitesse supérieure, un pistolet laser, et une version améliorée appelée le "FunGun" qui tire plusieurs balles en même temps.

Deux fusils à pompe : un basique et un à énergie. Le pompe basique tire plus de balles en un tir mais fait moins de dégâts, et celui à énergie tire plus rapidement. Un fusil sniper moderne capable d'éliminer les ennemis des premiers niveaux en un ou deux coups et un Sniper Laser qui élimine les ennemis en 1 à 2 coups avec

une cadence de tir supérieure.

Un Railgun capable de faire des dégâts énormes mais avec un temps de rechargement long, ainsi qu'un fusil électrique appelé Electro-Rifle, qui a une cadence de tir et des dégâts raisonnables en mid-game. Une tronçonneuse qui possède des DPS élever un combat rapproché, et le Riot Control Baton, une arme inspirée de Star Wars.

Un sabre laser qui possède une zone d'attaque suffisante pour effectuer du crowd control, et une arme inspirée de Doom Eternal, le Creuset, capable de faire des dégâts élevés et de rapidement réduire les ennemis en miettes. (Voir annexe point 4, image 4, de haut en bas, de gauche à droite : Minigun, Fusil Automatique, Fusil Sniper Moderne, FunGun, Fusil A Pompe Classique, Minigun Laser, M61 Vulcan, Fusil Sniper Laser, Railgun, Electro-Rifle, Sabre Laser, Tronçonneuse, Creuset)

Dans Enclave, le joueur commence comme un humain qui vient de s'échapper d'une prison spatiale. Au fur et à mesure qu'il progresse dans les vestiges de la prison, il trouve plusieurs armes qui l'aident à surmonter les ennemis auxquels il doit faire face. Cependant, un arme ce n'est pas assez des fois, et c'est pour cela que le joueur a la possibilité de trouver des armures, qui lui permettront de survivre plus longtemps face aux terribles ennemis de ces ruines spatiales.

Pour réaliser ces armures, nous avons conçu une nouvelle classe qui nous permettrait de mettre les différentes armures dans une même classe. Les armures ont chacune leur propre script qui sont donc liés entre eux par la classe mentionnée au début du paragraphe. Chaque armure augmente la vie du joueur en la multipliant, il est donc essentiel de récupérer toutes les armures afin d'atteindre le maximum de points de vie. Les armures sont considérées comme équipées à partir du moment où le joueur entre en contact avec elles. Lorsque cela arrive, le script va remplacer les modèles de base et d'animation du joueur pour lui donner ceux de l'armure qu'il vient de récupérer. Étant donné que le joueur possède aussi une barre vie visible, il a fallu la modifier. Sa valeur maximale dépend donc de la vie maximale du joueur, et sa valeur actuelle dépend de la vie actuelle du joueur. Les points de vie maximum et les sprites se conservent lorsque le joueur passe d'une scène à l'autre, donc les autres composants tels que le slider s'adaptent de cette manière.

Les armures sont disponibles après certains niveaux, et elles permettent au joueur d'augmenter sa vie de manière à ce qu'il puisse mieux résister aux attaques des ennemis. Il y a au total quatre armures dans Enclave (Annexe, 4e point, image 1), chacune augmentant la vie du joueur. La première armure, la combinaison spatiale militaire, est disponible après avoir battu l'Invocateur au niveau 10. Cette armure permet au joueur d'avoir 150 Points de vie au maximum, et change également son apparence, passant d'une tenue de prisonnier à une combinaison d'astronautes.

La deuxième armure se trouve à la fin du niveau 15. Cette armure est la Space Crusader Armor. Cette armure ajoute 150 Points de vie aux PV maximum du joueur, pour un total de 300. Cette armure a une apparence gris-sombre et rouge, avec un masque sur les yeux de couleur brun-clair. La troisième armure est obtenue après avoir vaincu la Reine Mère. C'est le Star Hero Suit, et c'est l'armure que le personnage de l'affiche du jeu porte. Elle donne au joueur 100 PV additionnels pour un total de 400 PV.

La quatrième et dernière armure est l'Armure de l'Enclave. C'est une armure large, lourde, avec une visière plus large que les précédentes armures, qui renforce le personnage encore plus. Grâce à elle, le joueur peut atteindre un total de 800 PV.

Ce système donne au joueur plus de possibilités pour prendre des risques en combats rapprochés ou bien en utilisant des stratégies différentes ou il pourrait se permettre de perdre plus de PV. Il offre également un visuel différent, nouveau du personnage, ce qui donne au joueur la satisfaction de voir son personnage évoluer tout le long de son aventure.

Cependant, même la meilleure armure du jeu ne peut pas arrêter les parasites qui font constamment baisser la vie du joueur. Seule la mort du Précurseur peut empêcher le destin tragique qui attend le joueur.

La partie la plus technique de la réalisation du système d'arme a dû être

l'équilibrage des armes. Par exemple les miniguns : le premier était bien trop faible, le deuxième légèrement mais le dernier était beaucoup trop fort. Mais comment faire pour que lorsqu'on augmente les dégâts du premier minigun il ne devienne pas trop fort ? Il faut éviter qu'il soit plus valorisé que le deuxième car ce dernier est censé être plus puissant. Et quand on améliore le deuxième minigun, il ne faut pas qu'il soit plus puissant que le dernier, sinon on est dans un scénario où il y a 2 armes bien trop fortes pour leur environnement, qui doivent conserver leurs spécificités pour avoir quelque chose qui les sépare des autres armes, et qui ne peuvent pas être trop réduites en puissance car cela nous ferait retourner à la case départ.

De plus, ce genre de situation est très mauvais pour un jeu comme Enclave, car il devient un jeu de chance alors que le jeu ne se base absolument pas là-dessus. Il fallait alors trouver un compromis, pour ajuster parfaitement les stats des trois armes. Un autre problème qu'on a eu est que, suite à des tests, les dégâts de la tronçonneuse étaient quasi nuls, ce qui a fait d'elle la pire arme du jeu pendant plusieurs heures, à cause d'un oubli.

C'est en testant notre jeu que nous avons pu réaliser ces erreurs et résoudre ces problèmes. Ça peut être plus qu'une arme est trop puissante ou bien pas assez, c'est aussi possible que l'arme ne soit tout simplement pas satisfaisante à utiliser, qu'elle soit bizarre ou bien dérangeante, que ce soit visuellement ou bien dans sa sonorité.

Il est important d'avoir un retour sur les armes pour savoir ce qu'il faut changer, et comment il faut s'y prendre pour rééquilibrer le jeu, afin d'offrir aux joueurs la meilleure expérience de combat dans notre jeu. Si nous avons pris trop de temps, nous n'aurions pas pu tester nos systèmes, et on aurait eu un système de combat absolument pas équilibré, ou il est simple d'établir une "meta", une stratégie qui a pour but d'être la plus optimale possible, donc d'être la meilleure du jeu et de dominer les autres styles de gameplay.

Un autre exemple d'un aspect qui dépend de l'équilibrage du jeu serait le "Power Creep" : c'est ce qui arrive lorsque du nouveau contenu rend l'ancien contenu obsolète, pas seulement au sens que l'arme est meilleure qu'une autre, mais plus précisément sur le fait que deux armes pourraient être obtenues au même

stade du jeu, l'une étant l'arme la plus basique du jeu, et l'autre une nouvelle arme qui est sur un tout nouveau système. Si par exemple le joueur a un fusil dans les premiers niveaux, que les miniguns sont très forts et que les développeurs ajoutent un minigun dans les premiers niveaux, plus personne ne va utiliser le fusil basique puisque le minigun est meilleur.

3.5 Sacha Skopan

3.5.1 Level design

Comme prévu dans le cahier des charges, nous avons réussi à implémenter 30 niveaux. Nous avons tout d’abord fait des croquis des premiers niveaux pour nous décider de la taille de chaque niveau, de sa direction, du placement des objets et des ennemis. Nous avons grandement discuté sur ces croquis, de la façon dont seraient placés les niveaux, qu’il soit de haut en bas, de gauche à droite et vice-versa, de la forme, chacun avait sa propre idée pour les niveaux Enclave, un labyrinthe, un cercle, un rectangle, un triangle, un losange pour en citer quelques-uns.

Nous avons aussi débattu sur son fonctionnement, que fallait-il faire pour finir un niveau, quelles pourraient être les contraintes imposées au joueur pour rendre le jeu dynamique, relativement difficile, mais tout de même jouable. Il y avait un sujet sur lequel chaque membre du groupe voulait absolument ajouter sa touche personnelle, les Easter Egg (Un Easter egg (terme anglais pour “œuf de Pâques”) est, en informatique, dans le cinéma ou dans les jeux vidéo un clin d’œil renvoyant à une œuvre ou une fonction cachée au sein d’un programme (image animée, jeu, message électronique, etc.) accessible grâce à un mot-clé ou à une combinaison de touches ou de clics.). Au début, on devait avoir une salle vide sans ennemis qui représentait un bar dans le niveau 15, elle devait représenter un endroit de repos dans lequel le joueur pouvait s’arrêter et acheter des objets à un barman qui représente un professeur fort apprécié de ses élèves auquel nous aurions souhaité rendre hommage. Ces objets relèvent d’autres jeux vidéo, principalement en rapport avec *The Escapist*, *Dead Space* et *Doom*. Ces objets pouvaient seulement être obtenus si le joueur avait assez d’argent en ayant tué chaque ennemi de tous les niveaux précédents. Mais malheureusement, nous n’avons pas eu le temps de l’implémenter, donc nous avons uniquement des Easter Egg dans les armes. Toutes ces réflexions nous ont menés à une structure sur laquelle sont basés tous les niveaux d’Enclave.

Le premier niveau que nous avons créé était censé servir de tutoriel avec des indications pour l’utilisation des armes, le but du jeu et comment passer au prochain niveau. Pour rendre le jeu plus immersif, une cinématique était implémentée pour introduire le joueur à l’histoire du jeu. Malheureusement, avec l’implémentation du multijoueur, le niveau un a dû être annulé dû à la complexité de faire une cinématique avec la façon dont nous faisons apparaître le joueur. Par conséquent, les joueurs commencent à partir de la scène 2.

Pour la première soutenance, nous avons réussi à implémenter les 15 premiers niveaux. Les 15 premiers niveaux sont assez variés, mais la première salle, c'est-à-dire la salle d'apparition du joueur et la salle de fin, celle de l'ascenseur, est similaire à tous les niveaux. Chaque niveau possède : Un grid sur lequel nous avons placé des tilemap (petits carrés représentant une texture) qui servent de textures comme des couloirs, des murs et des sols.

Un GameObject (nom du type qu'à chaque objet dans une scène) nommé Shawn dans lequel il y a plusieurs composants edge collider 2D qui est un composant qui permet de dessiner, de créer des formes qui servent de collider. Ces composants permettent de créer les limites de chaque niveau pour garder les ennemis tout autant que le joueur dans l'enceinte des niveaux. d'un GameObject Enemy qui contient tous les ennemis du niveau, que ce soit des Forts ou des Dormeurs.

de A* qui gère le pathfinding des ennemis Forts D'un Canvas qui gère tout ce qui est affichage, que ce soit la barre de vie comme des indications. Un GameObject Blood qui regroupe toutes les textures qui ne sont pas du type tilemap, dans lequel se retrouvent tous les sprite représentant le sang, mais aussi tous les ordinateurs sur les murs du niveau et de la Camera

Seul le niveau 10 est complètement différent des autres, étant un niveau de boss. Il possède une grande salle avec deux plus petites salles qui permettront au joueur de se réfugier s'il manque de vie.

Pour la deuxième soutenance, nous avons réussi à implanter tous les niveaux manquants jusqu'au niveau 30. Donc les niveaux de 16 à 30. Les niveaux ont la même structure que les 15 précédents, sauf le niveau 30 qui est fait sur mesure pour le dernier boss, le Précurseur.

Nous sommes au courant que les edges collider 2D (composant de Unity) ont une grande complexité due au nombre de coins qu'ils peuvent contenir. Même si nous n'avons pas de problèmes de performance, nous sommes à même de trouver une alternative aux edges collider 2D. Pour ceci, nous avons remplacé les edges collider des niveaux 17, 18 et 19 par une multitude de box collider 2D (composant de Unity) qui font aussi le tour des niveaux et de son intérieur. Mais cette alter-

native est beaucoup plus coûteuse en termes de temps.

Après plusieurs essais nous n'avons pas vu de changement comparé aux edges collider 2D donc nous les avons laissées sur ces niveaux.

Suite aux playtests nous avons remarqué que certaines IA passent à travers les edges collider 2D particulièrement proches des portes des niveaux. Plus précisément, les Forts. Ils passent à travers les edges collider 2D une fois, puis ils sont bloqués après par les mêmes edges collider 2D et donc restent bloqués en dehors des niveaux.

Nous pensons que ceci est dû au fonctionnement des portes. Toutes les portes des niveaux possèdent un box collider 2D qui fait la taille de la porte, mais lorsque le joueur s'approche, la porte s'ouvre et le box collider 2D est désactivé. Le box collider 2D de la porte étant en contact avec l'edge collider 2D sur un court espace, le désactive, ce qui permet à l'IA de traverser les edges collider. Mais nous ne sommes pas sûrs, car ceci est dur à voir, parce que l'IA ne passe à travers les edges que sur certains tests, mais pas tous.

Ce phénomène n'a pas l'air de se produire dans les niveaux 17, 18 et 19, ceux qui contiennent des box collider 2D au lieu des edges collider 2D. Nous allons donc soit remplacer les edges collider 2D de tous les niveaux par des box collider 2D ou alors ajouter des box collider 2D sur les extrémités des portes pour éviter ce problème.

Avec l'implémentation du multijoueur, certains changements ont dû être faits sur chaque niveau. Donc, nous avons créé deux fichiers : MainPackage qui contient : A* pour le pathfinding

la Camera

NetworkManager (composant de FishNet qui gère notre multijoueur) qui gère le multijoueur et l'apparition des entités du jeu que ce soit le joueur ou les ennemis. PlayerManager (composant de FishNet qui gère notre multijoueur) qui s'occupe du joueur, Musica qui permet de lancer la musique a chaque niveaux le Canvas le Gameobject AnimationPlayer qui gère les animation du joueur Global Light 2D qui s'occupe de l'éclairage PlayerSpawn qui est un Gameobject qui sert de position

pour l'apparition du joueur.

Le numéro du niveau en question qui contient : Le gameobject grid dans lequel est la texture du niveau Le gameobject Spawn qui contient les edge collider 2D Le gameobject Blood qui contient toutes les taches de sang et toutes les textures d'ordinateur. Elevator button la texture qui représente le bouton d'ascenseur et qui a un script qui permet d'afficher le message "Appuyez sur E" lorsque le joueur se trouve proche de lui pour passer au prochain niveau.

Pour la soutenance 3, les 30 niveaux étant déjà intégrés, nous nous sommes consacrés à régler les bugs et à rajouter des textures. Pour chaque niveau, nous avons changé certaines textures, plus précisément celle des couloirs, car le responsable du Pixel Art nous a fourni des textures comprenant directement des ombres. Donc les niveaux sont maintenant plus réalistes avec des textures représentant des jonctions entre plusieurs couloirs, des coins avec une ombre qui s'incruste dans le mur.

Nous avons aussi implémenté des Gameobject comme : des caisses avec le GameObject Box qui est un prefab représentant une caisse. Ce GameObject possède un BoxCollider2D donc le joueur ne peut pas passer à travers. des chaises qui sont représentées par deux prefab TopChair, la chaise avec un dossier vers le haut vue de haut et LeftChair, la chaise avec un dossier vers la gauche vue de haut. Barrel qui représente un tonneau. Nous l'avons plus utilisé pour faire des piles de tonneaux pour rendre réaliste l'univers dans lequel se trouve le joueur.

Cuve qui est un prefab qui représente une cuve dans laquelle se trouve un autre GameObject cobaye. Ce prefab contient un script Cuve mouv qui permet au cobaye de se déplacer de haut en bas comme s'il était en train de flotter.

Pour trier les GameObject nous avons créé un dossier décor dans lequel se trouvent tous les nouveaux Gameobject ajoutés. Pour essayer de régler le problème des Strong One, nous avons bien ajouté des BoxCollider 2D autour des portes, mais après plusieurs playtest nous nous sommes rendu compte que le problème était toujours persistant. Nous avons deux hypothèses : Notre première idée est que dû à la taille du rigidbody2D (composant de Unity) du StrongOne et la taille

du edges collider 2D, le Strong One arrive a se bloquer entre deux coin du collider et passe à travers.

Notre deuxième idée était que simplement notre système de pathfinding était incompatible avec les edges collider 2D. Pour résoudre ce problème nous avons décidé de baisser la vitesse des Strong One et de diminuer leur nombre dans les niveaux pour diminuer la probabilité qu'ils passent à travers les murs et des edges colliders des niveaux.

3.5.2 Histoire

Nous avons très rapidement fini l'histoire dans laquelle prend place Enclave et le joueur.

NM ici est le nom du joueur.

En l'an 24 000, l'humanité prospère grâce à la conquête d'innombrables mondes à travers la voie lactée. Avec la création d'un nouvel empire appelé les Colonies Unies, chaque humain vivait dans la paix et le bonheur. Mais pour nourrir les innombrables citoyens de l'empire, il fallait terraformer des millions de mondes en mondes agricoles en éradiquant un grand nombre d'espèces extraterrestres. Dans le monde agricole connu sous le nom de Marvin IV, il existe depuis longtemps une histoire effrayante de mal, de malédiction et de malheur, souvent considérée comme l'incarnation de la vengeance. Cette histoire est connue sous le nom de ... ENCLAVE

ACTE I: Prélude

Marvin IV est un monde agricole comme les autres, où chaque habitant voit le sol qu'il travaille chaque jour comme une divinité, une entité qui lui rend les soins qu'on lui donne. Même si des tonnes et des tonnes de nutriments, de nourriture, de plantes médicinales étaient acheminées toutes les heures par des cargos gros comme des petites lunes, Marvin IV n'était pas touchée par la pollution comme d'autres mondes l'étaient. Ce qui semblait pour les citoyens un bon présage des dieux de la terre.

NM était un bon fermier de Marvin IV dont les journées se résumaient à l'enchaînement des mêmes événements : se lever tôt, labourer la terre, planter des graines, arroser les cultures et dîner avec ses amis. Mais pour NM, ce qui primait sur tout le reste, c'était de prier les dieux pour une bonne récolte, tous les soirs devant une statue de marbre représentant une entité lumineuse mi-humaine, mi-végétale. Même s'il avait la réputation d'être un homme pieux qui ne laissait rien passer, NM avait l'habitude de prier les dieux pour obtenir une bonne récolte tous les soirs devant une statue de marbre représentant une entité lumineuse mi-humaine, mi-végétale. Même s'il avait la réputation d'un individu dévot qui ne laisserait rien arriver à la terre qu'il vénérât, NM était plutôt tolérant et essayait toujours de voir le point de vue de l'autre. La veille de la première récolte de l'année, NM, comme tous les autres jours, se trouvait dans un bar pour célébrer la bénédiction de la terre avec laquelle lui et ses amis ont travaillé pendant tant d'années. Après avoir trop bu, NM s'est effondré et s'est évanoui.

Il/elle a rêvé qu'il/elle se trouvait dans un champ de maïs et qu'il/elle regardait des éclats de lumière qui ressemblaient à des nuages fantomatiques. Il/elle commençait à les suivre, mais à chaque pas, ils prenaient une forme de plus en plus humaine. NM avait du mal à garder les yeux ouverts quand soudain un grand éclair de lumière a jailli sur sa gauche. Lorsque NM a repris conscience, il/elle a eu un mal de tête fulgurant et en regardant autour de lui/elle, il/elle s'est rendu compte qu'il/elle était allongé(e) sur la route. A proximité, il y avait une robe déchirée en morceaux avec des traces de sang. Horrifié, NM s'est rendu compte qu'il y avait ce qui ressemblait à des organes humains sur le sol : la partie supérieure d'un cadavre de femme avec un collier portant le symbole d'un marteau. Il a alors vu une foule de personnes s'approcher de lui, des amis, des membres de sa famille et d'autres citoyens criant qu'il était un boucher, un meurtrier et un monstre. NM a vu qu'il était lui aussi couvert de sang. Il n'y avait aucun doute sur le fait qu'il était le coupable. Bien que Marvin IV ait été développé, de nombreux villages n'étaient pas soumis à la juridiction et à l'état de droit. NM a dû être chassé du village. Avec ce qu'il lui restait, il erra à travers les forêts et les champs, cherchant de l'aide là où il pouvait en trouver. Un jour, alors que NM était si affamé et si faible, il s'est effondré dans un champ de blé.

Lorsqu'il revint à lui, il se trouvait dans un bateau qui semblait en mauvais état. Bientôt, un homme s'approcha de NM et lui dit durement : "C'est donc toi

qui as ruiné ma vie, tu as pris ce qui m'était le plus précieux, tu es celui qui a assassiné ma fille. Je ferai en sorte que ta vie devienne un véritable enfer. Et tu sais ce que je viens de faire ? J'ai placé un petit dispositif dans ta tête et si tu ne fais pas ce que je te dis, tu ressentiras une petite douleur comme celle-ci" en disant cela, il cliqua sur un bouton rouge sur son bras et NM commença à ressentir une douleur insupportable qui parcourait toutes les veines de son corps. Après quelques secondes, l'homme dit : "Je suis Jordan et vous m'appellerez patron, je contrôle tout ici. J'ai aménagé cette épave pour qu'elle serve d'arène de jeu, nous parions sur le temps que des gens comme vous, des criminels et des voleurs de bas étage, pourront passer à survivre face à de vrais combattants. C'est ce que vous ferez, sinon vous aurez mal et si vous essayez de fuir, j'enverrai une petite escouade d'assassins entraînés pour tuer votre famille et vos amis. C'est clair ?" Jordan appuya sur le bouton rouge, s'amusant visiblement à regarder NM gémir de douleur. Au bout de quelques minutes, la douleur est trop forte et NM s'évanouit.

3.5.3 Capacités

Le projet Enclave est un jeu qui mêle le fantastique et le futuriste, nous avons donc un grand champ de possibilités pour nos capacités.

Nous avons longuement débattu sur comment faire les compétences, comment les implémenter, comment les régler pour qu'elles ne soient pas trop puissantes, quel nom elle devait avoir, leur histoire pour qu'elle se mêle bien à l'univers que nous avons créé. Mais le plus dur a été d'être original. La majorité des membres du groupe ont déjà joué à des jeux vidéo qu'ils soient fantastiques, réaliste, futuriste, dramatique, horreur, survie, stratégie, mais chacun de ses jeux ont quelque chose en commun leur mécanique de jeu, chacun a réussi à faire sa place dans le monde du jeu grâce à l'ingénuité des développeurs de ces jeux vidéo, grâce à la liberté ou non qu'il laisse au joueur, grâce à l'univers qu'ils créaient, grâce au souvenir qu'il grave dans la mémoire du joueur. Et tout ceci est dû à l'originalité, au fait de se démarquer des autres. C'est pour cette raison que les capacités ont été plus dures sur le plan théorique que pratique.

Notre première capacité était dash qui permettait de faire un bon en direction de la souris, cette capacité était très efficace pour esquiver les ennemis et donc rendait le jeu dynamique, malheureusement après plusieurs playtest nous avons

remarqué que la capacité permettait aussi de passer à travers les murs et donc finir un niveau en seulement cinq secondes. Nous avons donc dû prendre des mesures et la retirer des capacités disponibles, même si on voulait la garder dans un endroit caché du jeu pour permettre au joueur plus expérimenté et ayant exploré tous les niveaux de s'amuser.

Nous nous sommes donc mis d'accord sur certaines capacités :

Zones:

Tier 1 Décharge (ralentit ou stop les ennemies autours pendant 3 secondes. Temps de rechargement 25 secondes)

Tier 3 Mind corruption (fait changer de camps a un l'ennemi pendant 10 secondes. Temps de rechargement 45 secondes)

Tier 2 Gravity Smash (Repousse tous les ennemis autour de vous. Temps de rechargement 30 secondes)

Buff:

Tier 1 Rage (augmente les dégâts de 25% de mêlée pendant 5 secondes. Temps de rechargement 40 secondes) Tier 1 Eagle Eye (augmente les dégâts de 50% de distance pendant 10 secondes. Temps de rechargement 40 secondes) Tier 2 Tough Body (réduit les dégâts reçus de 50% pendant 5 secondes. Temps de rechargement 90 secondes)

Tier 1 Retraite Hâtive (augmente la vitesse du joueur après avoir tué un ennemi.)

Tier 3 Humanity's Last Resort (Active lorsque le vie du joueur est en danger, Augmente tout type de dégâts de 75%, pendant 5 secondes. Temps de rechargement 90 secondes)

Chances:

Tier 2 Keen Eye (double le taux de chances d'avoir de la vie après avoir tué un ennemi.) Tier 3 Parasitic Mind (15% de chance de revivre après la mort. Temps de rechargement 120 secondes)

Maintenir (Doit maintenir le bouton de la capacité pendant sa durée sans prendre de dégâts): Tier 2 Blood Suction (dure 3 sec One shot l'ennemi. Temps de rechargement 45 secondes)

Chacune de ces capacités est le fruit de plusieurs heures de question, principalement pour trouver un nom qui rentre dans l'univers d'enclave, sans pour autant

être complètement irréaliste. Même si Enclave devait être un jeu comportant des mécaniques et des capacités originales. Nous voulons qu'il garde une âme de jeu vidéo. Nous avons donc pensé à s'inspirer de certains jeux vidéo comme la capacité Tough Body dont le nom provient du jeu Deepwoken. Retraite hâtive vient de la licence Diablo, étant un jeu avec un vu de haut, ayant une grande renommée pour sa difficulté et ses mécaniques de jeu, nous nous sommes inspirés pour les capacités. Nous voulions séparer les capacités en type :

Zones : des capacités qui ont un effet direct sur l'environnement autour du joueur, que ce soit des ennemis comme des GameObject Buff: des capacités qui augmentent le joueur comme sa vie, ses dégâts, sa vitesse.

Chance : des capacités passives qui s'activent toutes seules, mais qui ont un taux de probabilité faible en fonction de la capacité. Maintenir : des capacités qui ont une animation sur un ennemi, mais qui donc requiert l'appui d'une touche pendant plusieurs secondes sans interruption, par exemple pour la capacité Blood Suction le joueur devrait appuyer sur la touche H pendant 3 secondes sans bouger et sans qu'il reçoive des dégâts.

Malheureusement nous n'avons pu qu'implémenter quatre capacités : Tough Body, Parasitic Mind, Eagle Eye et Décharge. Les capacités de type maintenir ont été abandonnées dû à leur incompatibilité avec le multijoueur. Pour les autres capacités non implémentées nous avons une autre raison, ayant vue la puissance et l'utilité des capacités déjà implantées nous avons décidé de faire des capacités des jokers, des GameObject rares mais importants. Nous avons donc gardé seulement ces quatre ci car elles représentent le mieux Enclave:

Tough Body représente le fait de toujours trouver un moyen de survivre se renforcer et de persévérer

Décharge représente le fait de dépasser ses limites et surmonter l'adversité

Eagle Eye représente la stratégie, toujours trouver le meilleur moyen de survivre, trouver la meilleure combinaison avec armes, armure et capacité pour tuer l'ennemi.

Parasitic mind représente la corruption, une transformation en quelque chose que l'on ne reconnaît plus à force de la combattre, mais ceci prouve aussi la détermination et le fait de toujours redoubler d'efforts pour mieux faire.

Pour implémenter les capacités nous utilisons un component CapaManager

sur le joueur qui permet d'ajouter les composants adéquat à la capacité qu'il rencontre.

Au début le script des capacités était assez simple mais avec l'implémentation du multijoueur nous avons dû faire des changements drastiques dans la façon dont on incorpore les capacités au joueur. Ce fut simple mais long, nous avons dû faire une grande quantité de tests pour arriver à la bonne valeur, que ce soit pour ajouter ou pour retirer les capacités du joueur.

Système d'événement:

Le système d'événements aléatoires est une idée qui est censée apporter à Enclave un sentiment de renouveau à chaque partie, de toujours être surpris par ce que Enclave peut offrir que ce soit dans la difficulté ou dans la facilité. Notre système d'événements aléatoires permet de créer des interactions plus vivantes entre le joueur et les ennemis. Nous avons longuement délibéré sur la façon dont nous allions approcher ce défi, quel événement nous devions choisir, quel type ce devait être, quel moment, comment le déclencher, comment éviter des problèmes de collision ?

On avait pour idée au tout début de faire des événements aléatoires du type QTE (Un quick time event est, dans le jeu vidéo, un élément de gameplay de type action contextuelle généralement durant une scène cinématique où l'on suggère au joueur de faire une action dans un temps imparti.), avec soit des points à cliquer sur l'écran et si le joueur clique à côté il perd de la vie, ou des événements comme des questions de maths auxquelles il faut répondre rapidement et plein de petits mini-jeux que l'on voulait implémenter, mais avec notre temps limité nous n'avons pas pu les produire.

Après plusieurs prototype infructueux nous avons décidé sur trois types d'événements :

Le premier événement est un rampant qui apparaît devant le joueur
Le deuxième est du même type mais c'est un explosif qui apparaît devant le joueur
Le dernier est un fort

Au début, le dernier événement était censé être un marchand qui vend des ob-

jets au joueur en échange d'une monnaie qui aurait été obtenu en tuant les boss. Mais le multijoueur rendait compliqué l'affichage d'une interface avec laquelle seul un joueur peut interagir. Nous avons donc décidé de le remplacer par un fort. Notre première idée pour implémenter le système aléatoire était simple, nous avons choisi les salles qui s'occupent des événements, on y installe un rampant, un explosif et un fort. On les désactive et le script utilise un nombre aléatoire qui active un des ennemis. Mais avec le multijoueur nous avons dû modifier le système ce qui s'est avéré long mais fructueux. Nous avons donc maintenant dans chaque niveau une salle vide dans laquelle un ennemi peut apparaître devant le joueur.

4 Conclusion

Quand on regarde d'où on est partis au début et ce qu'est devenu Enclave aujourd'hui, on se rend compte qu'un vrai travail a été accompli. Le projet a évolué à tous les niveaux, et chacun d'entre nous a contribué à le faire grandir.

Au départ, on partait presque de zéro. Aujourd'hui, on a un jeu multijoueur fonctionnel et stable, grâce à notre chef de projet Louis Plaire, avec une IA dynamique qui s'adapte au joueur et le pousse à réfléchir à ses mouvements. Le système de santé, le fonctionnement du joueur, les armes... tout a été pensé pour rendre l'expérience plaisante pour l'utilisateur. Des choix de design comme la perte de vie constante ou la barre de vie visible ont renforcé l'immersion du jeu.

Côté ambiance, grâce au travail d'Angel, Enclave a une véritable identité. Au début, on n'avait que des sprites simples, mais maintenant, chaque élément visuel et sonore apporte un vrai plus pour le jeu. Les bruitages, les décors, les animations créent une atmosphère sombre, pesante et unique. Angel a réussi à donner au jeu une ambiance à la fois mystérieuse et terrifiante avec les moyens dont il disposait, ce qui donne un rendu vraiment magnifique.

Pour les niveaux, Sacha a assuré une bonne gestion dans le jeu. Il a imaginé et construit 30 niveaux complets avec une vraie précision du détail. Il a su s'adapter aux contraintes techniques du multijoueur, gérer les bugs de collision, équilibrer la difficulté dans les niveaux, et proposer des décors de plus en plus variés. Les niveaux qu'il a conçus ne sont pas de simples scènes de jeu: ils participent pleinement à l'immersion effrayante et captivante du jeu.

De son côté, Ethan a beaucoup progressé sur la structure et l'expérience utilisateur. Là où au début il avait du mal à faire fonctionner un simple script, il est maintenant capable de penser des systèmes entiers, comme un marchand ou un menu fluide. Il a aussi travaillé sur le site web du jeu, montrant que malgré ses difficultés sur Unity, il a su se démarquer par son travail comme le système d'échange ou même l'interface utilisateur.

Martin a été un élément important sur le gameplay. Il a travaillé sur les

mécaniques clés, le système d'armes malgré la complexité du Raycast2DAll, le système d'armure et les effets de caméra. Il a aussi participé à rendre le jeu plus vivant grâce à des petites idées simples mais efficaces, comme les bruitages de coups ou les effets d'impact qui renforcent le gameplay. Son implication a permis d'atteindre un jeu amusant et fluide.

Finalement, Enclave n'est plus un simple projet pour nous, il représente bien plus. C'est le jeu dont on rêvait de faire au début de l'année. Il reste encore des choses à améliorer pour atteindre la perfection, mais ce que l'on a accompli en une année est déjà énorme. Ce projet nous a fait progresser, individuellement comme collectivement, et nous sommes fiers du travail réalisé et des efforts fournis.

Annexe

Dessins des armes et barres de vie



Premier concept de l'invocateur



Premier concept de niveau



Premier concept du précurseur



Prototypes abandonnés :

Pnj 1



Prototype ennemi prisonnier infecté :



Prototype armure finale (variante)

Armure Enclave



Armure inspirée du jeu Fallout (concept oublié)



Design d'arme oublié



La jaquette





Les logos d'Enclave et d'Aura Corporation



Le Menu principal d'Enclave



Ecran de mort

PARTIE TERMINEE

RECOMMENCER

RECOMMENCER

Menu

