

Rapport de projet

Lurking Home

LRMH CORP



Louis POURRAT
Hugo BLANCHET
Martin BRUCHE
Rayane TIBAR



Juin 2025

Table des matières

1	Introduction	2
1.1	Présentation du projet	2
1.2	Présentation de l'entreprise	2
1.3	Présentation des membres	2
1.4	Répartition des tâches	4
2	Avancements du projet	4
2.1	Design du jeu	4
2.1.1	Logo	4
2.1.2	Jaquette	7
2.1.3	Menu du jeu	11
2.1.4	Carte du jeu	12
2.2	Gameplay du jeu	16
2.2.1	Les objets et assets	16
2.2.2	La récupération d'objet	21
2.2.3	L'inventaire	22
2.2.4	Les missions	23
2.2.5	Les fragments d'âmes	30
2.2.6	L'artefact de la maison	31
2.2.7	UI du joueur	32
2.2.8	Animations	33
2.3	Multijoueur	37
2.3.1	Problèmes majeurs rencontrés	37
2.3.2	Son fonctionnement final	38
2.4	Intelligence Artificielle	41
2.4.1	Introduction	41
2.4.2	Soutenance 1	41
2.4.3	Soutenance 2	42
2.4.4	Soutenance 3	43
2.4.5	Difficultés	44
2.5	Son	45
2.5.1	L'importance du son dans un jeu d'horreur	45
2.6	Site Web	46
2.7	Conclusion	49

1 Introduction

1.1 Présentation du projet

Notre projet s'intitule Lurking Home, c'est un jeu d'horreur et d'escape où l'objectif principal est de s'échapper d'une maison mystérieuse. Les joueurs doivent accomplir des tâches dans chaque pièce pour obtenir des fragments d'âme. Une fois 5 fragments rassemblés, ils permettent de créer un artefact puissant, capable de libérer la maison de son emprise maléfique. Cependant, le temps est compté : l'entité qui hante la maison cherche à capturer les joueurs, et la maison elle-même semble prête à les engloutir à tout instant. Si les joueurs échouent à compléter l'artefact à temps, ils seront pris au piège et la maison les consumera jusqu'à ce qu'une nouvelle équipe tente sa chance. Ce jeu mêle exploration, accomplissements de tâches et tension constante, tout en plaçant les joueurs dans une course contre la montre pour leur survie. Il est important de savoir que certaine capture d'écran de notre jeu ont été faite avec la lumière de base afin de que certain détail soit plus visible.

1.2 Présentation de l'entreprise

LRMH Corp est une jeune entreprise dynamique fondée par une équipe de passionnés du numérique, de la technologie et du développement interactif. Bien que notre cœur d'activité ne soit pas spécifiquement axé sur le jeu vidéo, et encore moins sur le genre horrifique, nous avons choisi de relever un défi audacieux pour notre premier projet commun : créer un jeu d'horreur immersif et marquant. Ce projet, baptisé Lurking Home, incarne notre volonté de sortir de notre zone de confort, d'explorer des mécaniques de gameplay nouvelles, et de repousser les limites de notre créativité. Ce premier pas dans le monde du jeu vidéo nous permet non seulement de tester notre cohésion d'équipe, mais aussi de montrer notre capacité à concevoir des expériences interactives abouties, même dans des domaines que nous n'avions encore jamais explorés.

1.3 Présentation des membres

- Louis POURRAT (Chef de projet)

À la suite de mon année de terminale avec les spécialités Maths et NSI, j'ai décidé d'aller à l'EPITA et j'en suis arrivé à réaliser un projet de jeu vidéo en groupe pour le second semestre. La programmation est un domaine que j'affectionne particulièrement grâce à son aspect créatif qui permet, à partir de rien, de créer des logiciels, des applications ou même des jeux. Lors de mon année de terminale, par groupe de trois, nous avons réalisé un petit projet de jeu durant lequel nous avons recréé le jeu vidéo Pac-Man. Ce projet m'a permis de développer mes compétences en Python, d'apprendre à travailler en groupe, mais surtout de comprendre que la rédaction d'un cahier des charges est indispensable pour un projet. Au sein de l'équipe, je vais pouvoir apporter mes compétences en développement web, qui sont plutôt solides grâce à mes pratiques personnelles. De plus, j'ai une assez bonne connaissance des concepts globaux de la programmation, grâce à mon expérience personnelle mais également scolaire, ce qui va me permettre d'être efficace pour le projet.

- Rayane TIBAR

Je m'appelle Rayane TIBAR, 18 ans, né au Maroc et je suis passionné pour la technologie dès mon plus jeune âge. À 14 ans, j'ai construit mon premier ordinateur, apprenant la patience et l'ingéniosité nécessaires pour assembler un système complet. À 16 ans, j'ai réalisé un projet en Python inspiré des mosaïques marocaines, combinant récursivité et géométrie pour créer des motifs numériques reflétant mon héritage culturel. Arrivé en France à 18 ans pour étudier à l'EPITA, une école d'ingénieurs en informatique j'étais prêt à pousser cette passion pour l'informatique encore plus loin.

- Hugo BLANCHET

Je m'appelle Hugo BLANCHET, j'ai 17 ans et je suis étudiant à l'EPITA. J'ai rejoint l'EPITA pour perfectionner mes compétences en informatique et devenir ingénieur dans ce domaine. Au lycée, dans le cadre de la spécialité NSI, j'ai réalisé un projet de jeu vidéo en Python. Avec mon camarade Louis, nous avons recréé notre version du jeu Pac-Man. Ce projet m'a permis d'acquérir de l'expérience en gestion de projet et en programmation. Je pense que cette expérience acquise grâce à ce projet me sera utile pour notre projet Lurking Home, car elle m'a permis de mieux comprendre comment visualiser et gérer le travail d'équipe sur un projet commun. Elle m'a aussi donné les bases de la programmation d'un jeu vidéo et un aperçu de ce que représente la création d'un jeu vidéo. J'ai hâte de réaliser un nouveau jeu vidéo qui, je l'espère, surpassera tout ce que j'ai pu accomplir jusqu'à présent.

- Martin BRUCHE

Depuis mon adolescence, l'informatique m'a toujours fasciné, en particulier le développement de jeux vidéo. Cette passion m'a conduit à poursuivre des études à l'EPITA, où je continue de développer mes compétences. Au lycée, j'ai travaillé sur des projets concrets comme la création d'un site internet sur les sneakers en groupe, ce qui m'a permis d'apprendre la conception web et la gestion d'équipe. J'ai également acquis de l'expérience en vente en ligne sur Vinted, où j'ai développé des compétences en communication et en gestion de clients. Mon parcours s'est enrichi grâce à un stage d'un mois chez W3COM, qui m'a permis de renforcer mes compétences techniques, et par une expérience de sept semaines en grande surface, où j'ai développé des compétences en gestion et en service client.

1.4 Répartition des tâches

Tâches	Louis	Rayane	Hugo	Martin
Intelligence artificielle			Responsable	
Décors et joueurs design		Responsable		Suppléant
Site web	Suppléant			Responsable
Multijoueur	Responsable			
Logo et jacket		Responsable		Suppléant
Bandé sonore		Responsable	Suppléant	
Gameplay	Responsable		Suppléant	
Histoire et DA		Suppléant	Responsable	
Conception des missions	Suppléant			Responsable



2 Avancements du projet

2.1 Design du jeu

2.1.1 Logo

Créer un logo pour représenter LRMH Corp n'était pas simplement une question d'esthétique. Il fallait refléter nos valeurs, notre ambition et notre spécialisation dans l'univers horrifique. Après de nombreuses esquisses, nous avons opté pour une typographie forte, anguleuse et sobre, qui évoque à la fois la robustesse et une certaine tension. Nous avons dû iterer plusieurs fois, en prenant en compte la lisibilité, la symbolique et la manière dont le logo s'intégrerait dans différents supports numériques et papier. Ce processus nous a fait prendre conscience de l'importance de l'image de marque dans l'industrie vidéoludique. L'identité visuelle est ce que le joueur retiendra en premier. Nous avons aussi testé la visibilité du logo sur différents fonds, clairs et sombres, et adapté sa résolution pour des affichages HD comme mobiles ou PC. Chaque version a été rigoureusement testée.



Pour la création du logo de notre jeu, nous avons choisi d'opérer un changement important dans la direction artistique. En effet, notre ancien logo, composé principalement de bleu et de noir, évoquait une atmosphère sombre et oppressante, mais ne correspondait plus totalement à l'identité visuelle vers laquelle nous nous dirigeions.

Au fil du développement, nous avons en effet décidé d'orienter notre direction artistique autour du rouge et du noir. Le rouge, couleur forte et percutante, est traditionnellement associé à la terreur, au sang et à la violence, des thèmes très présents dans notre univers horrifique. Le noir, quant à lui, évoque les ténèbres, l'inconnu, le danger latent, autant d'éléments que nous cherchons à transmettre aux joueurs dès les premiers instants.

Ce changement de palette nous a donc conduits à repenser totalement notre logo. Le nouveau logo utilise une combinaison de rouge profond et de noir, créant un contraste visuel fort et saisissant, qui attire l'œil tout en renforçant l'ambiance pesante et inquiétante du jeu. Ce contraste est également un bon reflet du gameplay, où les moments calmes peuvent rapidement basculer dans l'horreur et le chaos. Nous avons également accordé une attention particulière à la typographie utilisée. La police choisie est épaisse et marquée par des empattements, ce qui renforce l'idée de stabilité et de force, tout en ajoutant

une touche légèrement gothique qui colle parfaitement à l'univers de notre manoir. Cette typographie rappelle également celle utilisée dans le logo de notre entreprise fictive, créant ainsi une cohérence graphique entre notre identité de studio et le jeu lui-même.

Ce logo, bien plus qu'un simple visuel, constitue donc une porte d'entrée vers l'univers du jeu, et participe à ancrer les joueurs dans l'ambiance bien particulière que nous avons voulu créer. Il s'agit d'un élément central de notre communication visuelle, utilisé aussi bien sur notre site internet, nos jaquettes, que dans les menus du jeu.





2.1.2 Jaquette

La jaquette avant du jeu est la premiere interaction visuelle entre le joueur et l'univers de 'Lurking Home'. Nous avons longuement reflechi a l'ambiance a transmettre : mystere, solitude, danger. Le choix d'un manoir victorien, dans la brume rouge, et la silhouette d'un homme de dos, est venu naturellement. Cela suscite l'intrigue sans tout devoiler. Nous avons egalement experimente differentes palettes de couleurs et compositions pour obtenir ce rendu final, qui a ete valide apres plusieurs retours d'utilisateurs tests. Nous avons utilise des techniques de composition numerique et un il attentif aux details pour que chaque element (ombres, textures, effets de lumiere) cree une atmosphere inquietante mais coherente. Le tout a ete conçu pour intriguer le joueur sans lui reveler trop d'elements narratifs.

LURKING HOME



16[®]

www.pegi.info

La conception du dos de la jaquette fut tout aussi importante. Il fallait donner envie tout en posant les bases du scenario, sans spoiler. Le texte utilise est volontairement immersif et rythme, jouant sur la peur de l'inconnu et l'ambiance macabre. Nous avons aussi integre les mentions legales, le PEGI 16, les plateformes compatibles et un code-barres pour une version physique. Tout cela a ete fait dans le respect des normes, ce qui nous a confrontes a la realite des contraintes commerciales du jeu video. Nous avons aussi etudie des jaquettes de jeux d'horreur cultes pour nous inspirer de leur structure narrative. Le placement des textes, captures d'écran et mentions legales respecte les normes tout en gardant un cachet artistique unique.

**RENTREZ CHEZ VOUS...
ET AFFRONTEZ LA TERREUR QUI
SE TAPIT DANS L'OBSCURITÉ
DE VOTRE DEMEURE FAMILIALE
DÉCOUVREZ LES SECRETS OUBLIES
DE LA MAISON RONGÉE
PAR LA BRUME.**



Game



LURKING HOME

© LRHM Corp. Tous droits réservés.

2.1.3 Menu du jeu

Pour le menu du jeu, nous avons décidé de partir sur un design assez simple avec le nom du jeu affiché en gros et en rouge pour respecter notre direction artistique d'origine, qui s'articule principalement autour des couleurs rouge et noir. Le fond noir profond crée une atmosphère sombre et inquiétante, parfaitement en accord avec l'ambiance mystérieuse et oppressante de Lurking Home. On distingue en arrière-plan la silhouette d'une maison ancienne, légèrement floue et enveloppée d'ombres, qui donne tout de suite le ton et plonge le joueur dans l'univers hanté du jeu dès l'écran d'accueil. Ce choix visuel épuré permet de focaliser l'attention sur le titre tout en renforçant l'identité visuelle du jeu.

Les éléments du menu, tels que les boutons « Hoster une partie », « Rejoindre une partie » et « Quitter » ainsi que le champ vide afin d'insérer l'adresse IP de l'hôte sont disposés de manière intuitive et minimaliste. Cette simplicité dans la mise en page évite la surcharge visuelle et favorise une navigation fluide, tout en laissant transparaître une ambiance sombre et mystérieuse, qui prépare le joueur à l'expérience immersive qui l'attend. Par ailleurs, des effets subtils comme un léger scintillement rouge sur le titre ou une animation douce des ombres autour de la maison viennent apporter une touche dynamique sans distraire, renforçant l'atmosphère inquiétante dès le premier contact avec le jeu.

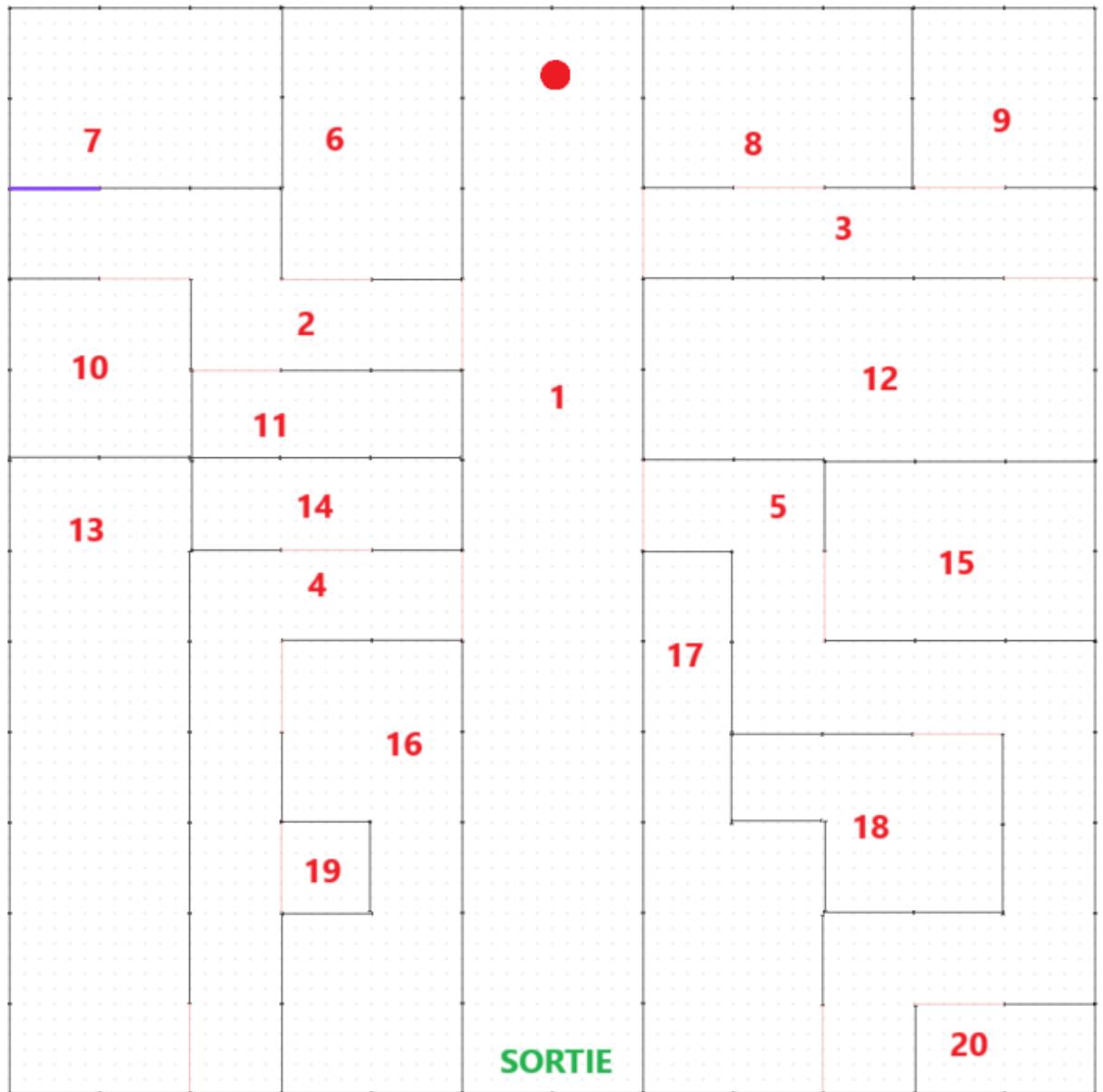
Pour le fonctionnement rapide du menu, un joueur doit cliquer sur hoster une partie afin de créer une partie et les autres joueurs doivent rentrer l'adresse IP du joueur hôte de la partie dans le champ vide prévu à cet effet et cliquer sur rejoindre la partie afin de rejoindre la partie créée par l'hôte.



2.1.4 Carte du jeu

La carte de notre jeu a subi de nombreux changements au cours du développement.

La première version était une grande salle de 120 par 120 unités, contenant 20 pièces. Cependant, nous nous sommes rapidement rendu compte que cette carte était trop grande et difficile à exploiter correctement en jeu. Nous avons donc décidé de la réduire à une dimension de 100 par 100 unités, avec seulement 7 pièces, mais ces dernières sont désormais plus grandes que les pièces initiales.



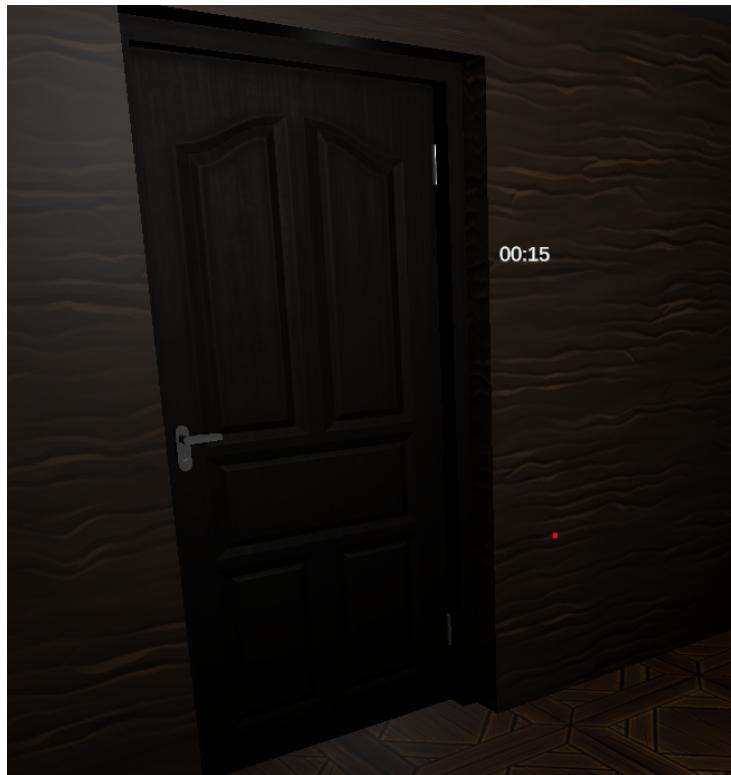


Nous avons également fait en sorte d'ajouter plus d'issues par pièce, afin d'éviter que les joueurs ne se retrouvent bloqués par l'IA dans une salle.

Ensuite, nous avons implémenté différentes portes grâce à des assets trouvés sur le Unity Asset Store, afin de fermer toutes les ouvertures de notre manoir.

Suite à ces modifications, nous avons réimplémenté le squelette de la carte dans Unity, en ajoutant tous les murs nécessaires. Nous avons ensuite défini l'utilisation de chaque pièce en répartissant les différents éléments du décor : on retrouve ainsi une cuisine, une salle de bain, un salon, etc.

La maison a été meublée avec de nombreux assets, tels que des bibliothèques, des tables, et bien d'autres objets pour renforcer l'immersion. Nous avons également ajouté des textures pour le sol, les murs et le plafond. Nous avons opté pour des textures de bois ancien, afin de coller à l'ambiance vieillotte et inquiétante du manoir.



Enfin, nous avons ajouté différentes lumières et plongé le manoir dans l'obscurité, afin de réduire la visibilité des joueurs et augmenter le sentiment d'oppression, essentiel à l'ambiance d'un jeu d'horreur.



L'un des aspects les plus importants du développement de Lurking Home repose sur la construction de son espace principal : une maison à l'architecture réaliste, mystérieuse, mais marquée par des éléments perturbants. Cette map ne se limite pas à être un simple décor ; elle agit comme un personnage à part entière dans le récit, avec ses zones lumineuses, ses pièces froides, ses secrets enfouis et les traces d'un passé trouble. On a participé activement à la conception de plusieurs de ces espaces, en se concentrant particulièrement

sur l’agencement et l’habillage de trois pièces majeures : la chambre, la salle de bain et la bibliothèque. Chacune incarne une facette du jeu : l’intime, le choc, et l’étrange. Le travail a consisté à imaginer leur structure, sélectionner et disposer les éléments visuels clés, et faire en sorte que chaque pièce soit crédible, évocatrice et engageante.

La chambre, conçue comme point d’entrée dans l’univers du jeu, avait pour but d’installer un sentiment de familiarité, avant de semer subtilement le doute. Le mobilier choisi est classique : un lit ancien, une table de nuit, une armoire, une bibliothèque et un bureau. Mais l’usure visible du lit, l’ambiance poussiéreuse, et surtout, la présence intrigante d’un parchemin au sol viennent rapidement troubler cette impression de normalité. On a cherché à instaurer une tension discrète, en jouant sur les contrastes lumineux. Plusieurs sources d’éclairage ont été disposées de manière à créer des zones d’ombre et à diriger le regard du joueur. Des détails comme une bouteille abandonnée ou un tiroir entrouvert ont été ajoutés pour stimuler la curiosité. Chaque objet a été sélectionné pour sa charge narrative potentielle : peut-il être examiné ? déplacé ? révéler une information ? Cette pièce a également été pensée comme un point de départ, intégrant un spawn point, ce qui a nécessité une réflexion sur l’interface, l’ambiance sonore et la première interaction visuelle. Elle devait être simple, mais pleine de sens.

La salle de bain, quant à elle, adopte une approche radicalement différente. Ici, on vise le choc visuel. L’élément central est une baignoire tachée de sang, mise en scène comme un symbole fort, presque cinématographique. Le reste de la pièce reste volontairement épuré : toilettes sales, lavabos rouillés, miroir fissuré, peu d’éléments interactifs. Ce minimalisme permet de concentrer toute l’attention sur l’impact visuel du bain. L’éclairage a été conçu pour révéler progressivement la scène : la lumière ne tombe pas directement sur la baignoire, mais est filtrée pour en permettre une découverte partielle, renforçant ainsi la mise en tension. La salle de bain remplit aussi une fonction narrative essentielle : elle introduit un indice clé, qui servira de déclencheur à une autre séquence du jeu. Elle agit comme une transition vers une phase plus intense, plus surnaturelle. Cette pièce a permis d’expérimenter un level design plus sobre, mais centré sur l’émotion brute.

Enfin, la bibliothèque constitue sans doute l’espace le plus ambitieux de cette première phase de conception. Dès le départ, on a voulu créer un sanctuaire de savoir abandonné, un lieu empreint d’une histoire lourde. La pièce adopte une structure rectangulaire vaste, donnant une impression de profondeur labyrinthique. Chaque mur est couvert de bibliothèques, certaines en désordre, d’autres partiellement effondrées. Ce chaos apparent participe à la narration visuelle : tout semble indiquer une fuite ou une interruption soudaine. Le sol est ponctué de bancs de travail sur lesquels le joueur peut interagir avec des objets, lire des documents, ou manipuler des mécanismes. Certaines étagères ont été inclinées ou décalées volontairement pour casser la symétrie et induire un sentiment d’instabilité. L’un des défis majeurs a été d’éviter la surcharge visuelle : il a fallu ajuster la densité des objets, supprimer certains éléments redondants, et équilibrer les points d’interaction.

Cette pièce est destinée à contenir une énigme centrale (comme une séquence de livres à replacer, un passage secret, ou des indices sur une présence invisible), et à devenir un hub narratif permettant au joueur de mieux comprendre les mystères de la maison. Ce travail a renforcé la capacité à concevoir non seulement un espace esthétique, mais aussi une navigation mentale cohérente : vers où se dirige le regard ? Que peut-on toucher ? Où

peut-on se cacher ?

Dans l'ensemble, ces trois pièces ont constitué un terrain d'expérimentation riche pour développer des compétences en level design, narration environnementale, composition visuelle et maîtrise des assets. Chaque espace a été pensé avec soin, en tenant compte de la logique spatiale, de la lisibilité, du potentiel d'interaction, de l'ambiance sonore (simulée lors des tests) et du rythme de progression. Aucun n'est isolé : tous s'intègrent dans une vision d'ensemble. Chacun porte une ambiance propre, mais tous partagent une intention commune : plonger le joueur dans une maison où chaque pièce a une mémoire, un secret, une tension.

Pour la création de la carte dans son ensemble, nous avons voulu concevoir une maison véritablement immersive qui reflète l'essence même d'un jeu d'horreur. L'objectif principal était de plonger le joueur dans une atmosphère oppressante et angoissante, où chaque recoin et détail contribuent à renforcer la tension. Pour cela, nous avons opté pour un environnement sombre, avec un éclairage très limité et maîtrisé afin de jouer sur les ombres et les contrastes. La pénombre presque permanente force le joueur à avancer avec prudence, créant un sentiment d'insécurité et d'apprehension.

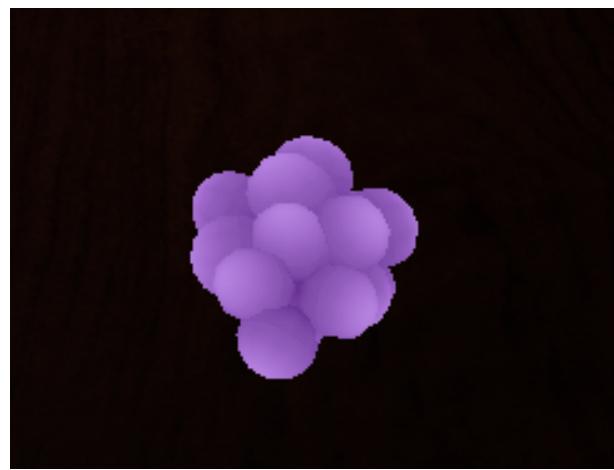
Nous avons également travaillé à rendre la maison crédible et vivante dans son état de délabrement, en intégrant des meubles renversés, des objets éparpillés au sol, des traces d'usure, et des détails visuels qui suggèrent une histoire trouble. Ces éléments sont essentiels pour susciter l'imagination et l'angoisse du joueur, l'incitant à se méfier de ce qui pourrait surgir à tout moment. Chaque pièce a été pensée pour offrir un cadre réaliste mais aussi pour favoriser les sensations d'isolement et de vulnérabilité, fondamentales dans un jeu d'horreur.

Cette approche a nécessité un travail minutieux sur le level design, en équilibrant suffisamment d'espace pour la liberté de mouvement tout en maintenant une sensation de confinement. La faible luminosité nous a aussi poussés à optimiser les sources de lumière ponctuelles, comme des lampes brisées ou des bougies vacillantes, qui deviennent des repères fragiles pour le joueur. L'ensemble vise à créer une expérience immersive où le décor joue un rôle actif dans la montée de la tension, amplifiant la peur et l'immersion tout au long du jeu.

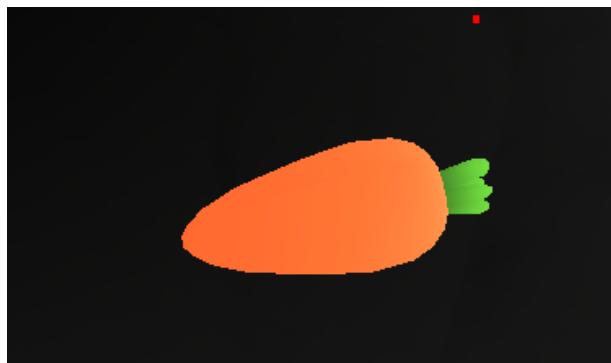
2.2 Gameplay du jeu

2.2.1 Les objets et assets

Nous avons mis un système d'instance d'objets, cela signifie que nous avons créer des instances pour chaque objet de notre jeu, et ces instances ont un nom, un visuel (correspondant au visuel 2D qui est affiché dans la barre d'action du joueur), ainsi qu'une prefab de l'objet, comportant tous les composants Unity nécessaires ainsi que les scripts que nous avons codés.









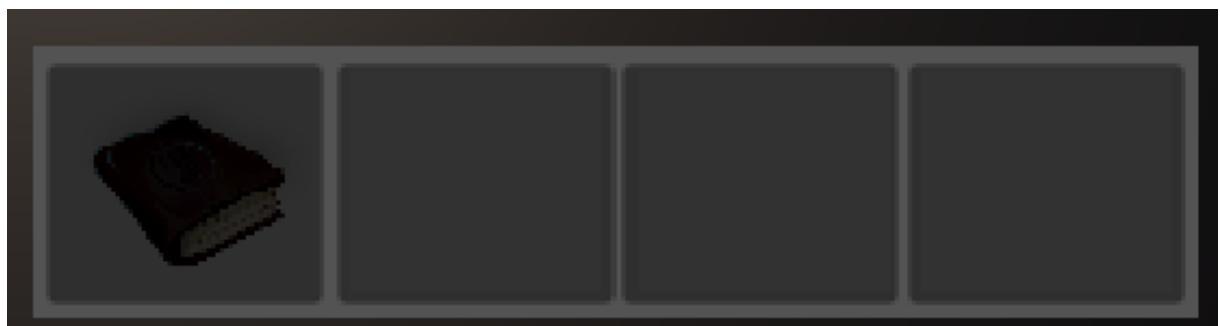
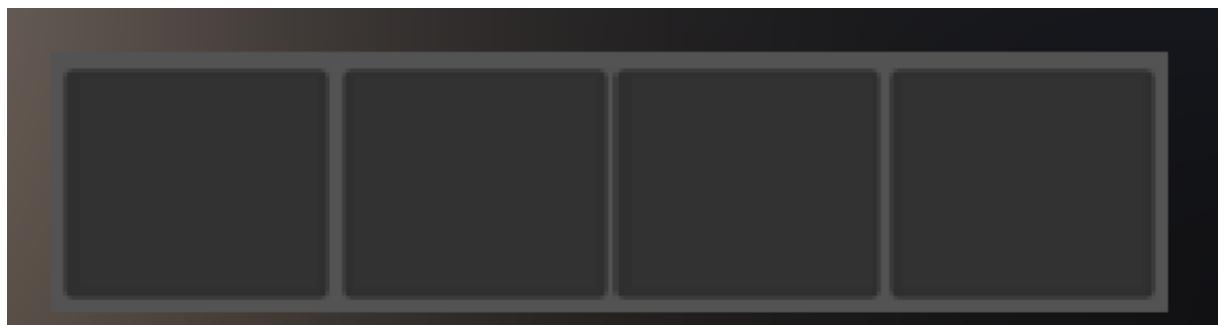
2.2.2 La récupération d'objet

Notre système de récupération d'item est basé sur un principe simple : le raycast. Le raycast est une méthode qui permet d'envoyer un rayon (une ligne invisible) dans une direction spécifique pour détecter des objets dans la scène. Nous avons un rayon invisible qui est envoyé en permanence devant le joueur, si ce rayon touche un objet qui est récupérable, et si le joueur appuie sur E, alors l'objet sera récupéré. De plus, l'objet sera détruit de la scène pour le joueur. Évidemment, une portée maximale de récupération est définie par un float ; le rayon n'a donc pas une taille excessivement grande. Dès lors que le joueur ramasse un objet, c'est l'instance de cet objet qui sera ajoutée à son inventaire.



2.2.3 L'inventaire

Notre inventaire n'est pas un inventaire classique comme dans les jeux de survie, par exemple, où on l'ouvre avec une touche. L'inventaire de notre jeu correspond directement à la barre d'action du joueur, qui est composée de 4 emplacements. Il est représenté par une liste, et n'est donc évidemment pas fixe ; il sera modifié chaque fois que le joueur récupère ou dépose un objet dans le jeu. De plus, cet inventaire est limité à 4 objets.



2.2.4 Les missions

La majorité du gameplay réside dans l'accomplissement des 5 missions réparties dans 5 pièces différentes de la maison. On va en expliquer 3.

Mission de la cuisine :

Le joueur doit d'abord lire l'inscription inscrite sur le mur, au-dessus des plaques de cuisson, qui indique qu'il faut prendre uniquement les ingrédients salés et éviter les ingrédients sucrés.



Dans cette pièce se trouvent six ingrédients : Quatre ingrédients salés, qui sont les bons à collecter pour terminer la mission ; Deux ingrédients sucrés, qui sont des pièges. Pour cette mission, nous avons dû implémenter un système qui détecte quand les quatre ingrédients salés sont portés par le joueur. Dans ce cas, le fragment d'âme apparaît et l'inventaire est réinitialisé. Nous avons également couplé toute cette logique avec Mirror, pour garantir la bonne synchronisation en multijoueur. Enfin, nous avons mis en place une vérification : si un ingrédient sucré est ramassé, alors l'inventaire du joueur est réinitialisé, tous les ingrédients déjà collectés sont remis à leur place, et un message s'affiche à l'écran pour avertir le joueur qu'il ne faut pas prendre d'ingrédients sucrés.



'5

Mission de la bibliothèque :

Dans cette pièce, trois livres maudits vont apparaître à trois endroits différents qui vont être choisi aléatoirement parmi 10 prédéfinies. Il faudra simplement les trouver et les ramasser, dès que les trois sont dans notre inventaire le fragment d'âme apparaîtra au milieu de la pièce.



Mission de la chambre et de la salle de bain :

L'objectif est de se rendre dans la salle de bain et d'y trouver un code inscrit sur le miroir.



Une fois ce code récupéré, le joueur doit retourner dans la chambre et allumer les bonnes bougies dans le bon ordre afin de révéler le fragment d'àme.

Si une mauvaise bougie est allumée, le fragment n'apparaîtra pas. Il faudra alors éteindre toutes les bougies avant de pouvoir retenter la séquence correcte.



Pour cette mission, nous avons dû gérer l'interaction entre le joueur et les bougies. Pour cela, nous avons utilisé un système de raycast, permettant au joueur de viser et d'allumer les bougies.

Nous avons également dû faire en sorte que l'état des bougies soit synchronisé pour tous les joueurs, ce qui nous a obligés à coupler nos scripts avec Mirror.

Concernant le spawn du fragment d'âme, nous avons réutilisé le même système que pour les missions précédentes : il s'agit de trouver le GameObject associé à la mission et de faire apparaître le fragment d'âme à cet endroit.

L'un des objectifs essentiels dans un jeu comme Lurking Home est de faire en sorte que chaque pièce ne soit pas un simple décor, mais un espace chargé de sens, d'actions et d'émotions. Dans cette optique, plusieurs missions ont été conçues autour des pièces modélisées, notamment la chambre, la salle de bain et la bibliothèque. Ces missions, bien que techniquement accessibles à ce stade du développement, sont pensées pour renforcer l'immersion, initier le joueur à différentes mécaniques, et approfondir la narration à travers des éléments d'interaction simples mais efficaces. Leur mise en place a été progressive dans Unity, en étroite collaboration avec d'autres membres de l'équipe, et avec plusieurs phases d'ajustements afin de répondre aux contraintes du projet.

La mission de la chambre a été la première à être conçue, en lien direct avec l'atmosphère étrange et intime que nous voulions instaurer dans cet espace clos. Dès les premiers instants, l'objectif était de plonger le joueur dans un sentiment de curiosité teintée d'inquiétude.

Au cœur de cette mission, un miroir ancien attire l'attention dans la salle de bain. En l'observant sous un certain angle, un message codé apparaît discrètement à sa surface. Ce message, volontairement ambigu, suggère un lien avec les bougies disposées dans la pièce. Il s'agit en réalité d'une énigme visuelle : les indices fournis par le miroir permettent de deviner l'ordre précis dans lequel les bougies doivent être allumées.

Cette séquence constitue une introduction subtile aux mécaniques d'interaction du jeu : elle pousse le joueur à explorer, à observer minutieusement son environnement et à faire preuve de déduction. Chaque bougie peut être activée via une interaction simple, mais seule la bonne combinaison permet de débloquer la suite. En cas d'erreur, les bougies s'éteignent automatiquement, obligeant à recommencer la séquence.

Une fois la bonne séquence exécutée, le fragment d'âme apparaît bien au milieu de la pièce. L'objectif de cette mission est de créer une progression fluide et gratifiante, à travers une énigme accessible mais bien intégrée dans l'environnement.

Malgré sa simplicité, cette mission joue un rôle fondamental : elle établit une règle implicite du jeu Lurking Home — ici, chaque élément a une signification, et tout détail peut révéler quelque chose de plus grand.

La mission de la bibliothèque est l'une des plus simples sur le plan mécanique, mais elle renforce l'ambiance mystérieuse du manoir tout en encourageant l'exploration. Le joueur doit y retrouver trois livres maudits dissimulés dans la pièce. Chacun est reconnaissable à son apparence singulière, marquée par une aura sombre ou des détails inquiétants dans leur design.

L'objectif est de pousser le joueur à fouiller chaque recoin, à observer attentivement les étagères, les meubles et les éléments de décor pour mettre la main sur ces ouvrages. Il n'y a pas d'énigme complexe ni d'ordre particulier à respecter : seule l'attention portée à l'environnement permet de progresser.

Les emplacements des livres ont été pensés pour stimuler la curiosité sans créer de frustration, en jouant sur la lumière, les ombres et la disposition du mobilier. Une fois les trois livres collectés, le fragment d'âme apparaît.

Cette mission, bien que simple, sert à renforcer la dimension exploratoire du jeu et à immerger davantage le joueur dans l'univers de Lurking Home, où chaque pièce semble porter le poids d'un secret ancien.

Ces différentes missions ont permis d'expérimenter concrètement dans Unity la transition entre un espace décoratif et un espace de jeu actif. Elles ont contribué à renforcer les compétences en triggers, interactions, feedbacks visuels et sonores, tout en s'intégrant à la logique globale du projet. Ce travail a également mis en évidence l'importance des détails dans le placement des objets, l'équilibre entre immersion et clarté, et l'impact direct des choix de design sur l'expérience du joueur. À terme, ces missions pourront être enrichies, reliées à d'autres événements narratifs, ou servir de prototype pour des mécaniques plus complexes. Dès à présent, elles posent les fondations solides d'un jeu centré sur la progression environnementale, la mémoire visuelle, et la tension narrative. De plus, nos missions ont été pensées pour être simples afin d'avoir un maximum de fonctionnalité au sein de notre jeu.

2.2.5 Les fragments d'âmes

Dans Lurking Home, les fragments d'âmes sont des éléments mystérieux et puissants que les joueurs doivent collecter pour progresser dans l'histoire et dévoiler les secrets du jeu. Chaque fragment représente une part d'énergie spirituelle dispersée dans l'environnement, souvent protégée par des énigmes ou des défis. Par exemple, lorsque le joueur réussit à réunir un certain nombre de livres anciens, un fragment d'âme apparaît au centre de la pièce, visible et accessible à tous les joueurs présents. En récupérant ce fragment, les joueurs renforcent leur connexion avec l'univers du jeu, débloquent de nouvelles capacités ou révélations, et avancent ensemble vers la résolution du mystère qui plane sur la maison hantée. Ces fragments sont donc au cœur de l'expérience coopérative, symbolisant à la fois le savoir et la puissance nécessaires pour affronter les ténèbres.



Pour ce qui est de l'implémentation, ils apparaissent simplement au milieu de la pièce où la mission vient d'être accomplie, le joueur a juste à passer dedans afin d'être récupéré et que le compteur des fragments augmente.

2.2.6 L'artefact de la maison

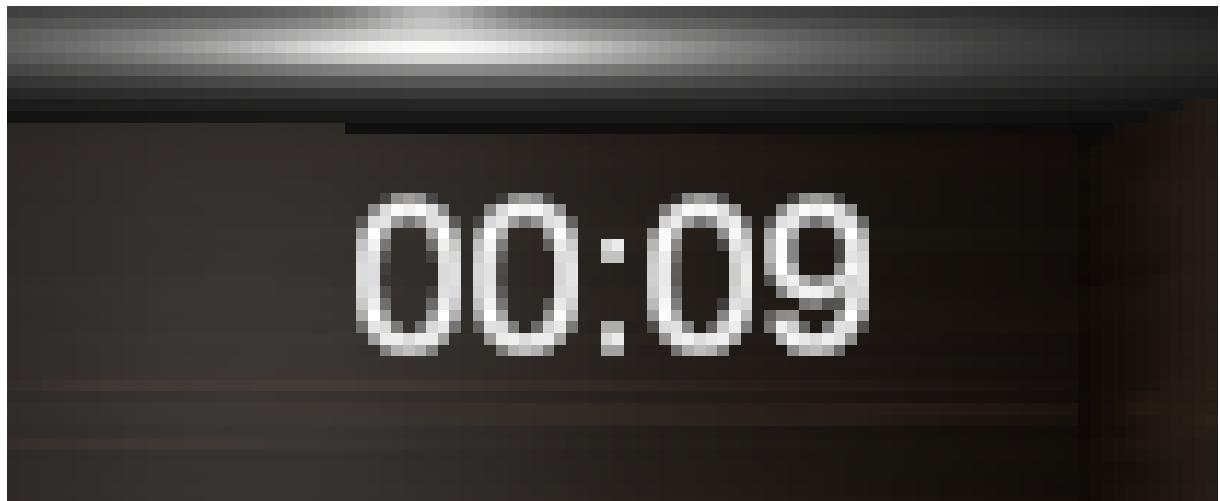
Comme rappelé dans l'introduction, l'objectif du jeu est d'obtenir un artefact suite à l'acquisition des fragments d'âmes, une fois les 5 fragments récupérés, l'artefact apparaîtra au milieu du grand couloir principal. C'est le même fonctionnement que pour les fragments, le joueur doit simplement passer dessus afin de l'obtenir et de pouvoir interagir avec la porte principale afin de s'échapper de la maison. Evidemment, si l'artefact n'est pas en possession d'un des joueurs aucun joueur ne peut s'échapper.



2.2.7 UI du joueur

Tout d'abord, chaque joueur possède sa propre barre d'action. Nous avons donc utilisé plusieurs canvas pour la barre d'action : un pour le fond et les autres pour représenter chaque emplacement. Ce sera donc le visuel 2D des instances de chaque objet qui s'affichera sur cette barre d'action lorsque le joueur récupérera l'objet en question.

Ensuite, le reste de l'UI est commun à tous les joueurs, ils ont donc en haut de leur écran un timer qui correspond simplement à la durée de la partie actuelle. Ils ont également le compteur de fragments d'âmes, initialisé à 0/5 et aussi un compteur pour l'artefact de la maison. Enfin, ils possèdent tous un curseur rouge afin de les aider à mieux viser surtout pour les missions lorsqu'ils doivent récupérer des livres ou allumer les bougies.

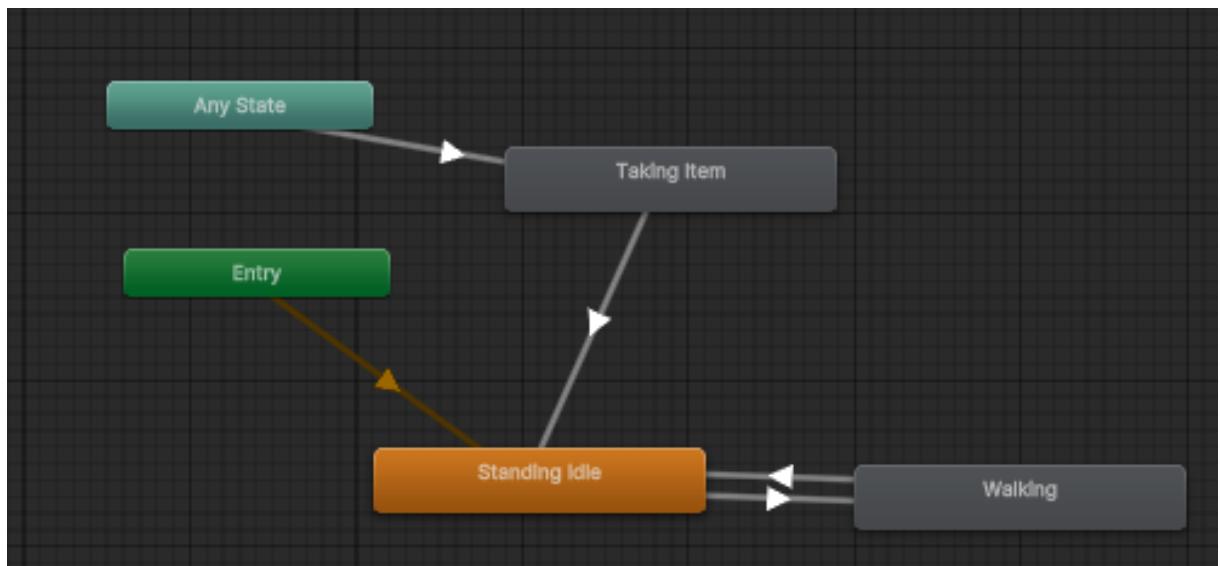


2.2.8 Animations

Pour gérer les animations de notre jeu, nous utilisons l'Animator de Unity. Nous avons trouvé nos animations sur le site Mixamo.



Tout d'abord pour les déplacements du joueur, nous avons utilisé deux animations pour le déplacement du joueur : une animation lorsqu'il ne bouge pas, que nous avons appelée "Standing Idle", et une animation de marche ("Walking").



Pour les paramétrages dans l'inspecteur de Unity, le plus important à savoir est que nous avons mis les deux animations en boucle afin qu'elles se répètent lorsque chacune est active. Nous avons donc lié l'animation "Standing Idle" à l'entrée qui correspond à l'animation de base que le joueur fait lorsqu'il n'effectue aucune action spéciale dans le

jeu. Ensuite, nous avons créé un booléen "Walk" ainsi que deux liaisons entre "Standing Idle" et "Walking". La condition pour que la liaison de "Standing Idle" vers "Walking" s'active est que "Walk" soit à "true", et respectivement à "false" pour la liaison retour. Nous avons un script de déplacement qui va mettre la variable "Walk" de l'Animator à "true" ou à "false" en fonction de si le joueur se déplace ou non.



Pour l'animation de la récupération d'un objet, nous avons utilisé un trigger qui permet simplement d'activer l'animation lorsque nécessaire, et le trigger se désactive une fois que l'animation est terminée. Le plus important pour cette animation a été de coordonner précisément le moment de la récupération de l'objet. En effet, l'objet n'est pas récupéré

dès le lancement de l'animation, mais plutôt lorsque l'animation atteint le moment où le joueur pose sa main sur l'objet.

Pour ce faire, nous avons utilisé la timeline de l'animation et avons placé un point précis, à l'image près, auquel nous avons associé notre méthode de récupération d'objet. Cela permet de lancer la méthode exactement à ce moment-là. Ce système rend notre récupération d'objet plus réaliste.



2.3 Multijoueur

2.3.1 Problèmes majeurs rencontrés

Tout d'abord, l'un des plus gros problèmes que nous avons rencontrés a été le fait que, lorsqu'un joueur ramassait un livre, le livre se détruisait correctement de la scène pour l'hôte, mais pas pour les clients du serveur. Pour résoudre ce problème, lorsqu'un joueur tente de récupérer un objet, nous vérifions si c'est l'hôte ou un client. Si c'est l'hôte, nous demandons directement la suppression côté serveur grâce à une méthode associée au NetworkServer. Cependant, si c'est un client, il envoie une commande au serveur pour détruire l'objet, et cette commande est marquée avec l'attribut Command afin qu'elle soit exécutée sur le serveur.

Ensuite, nous avons rencontré un autre problème : lorsque un client récupère un objet, il est ajouté à l'inventaire de l'hôte et non à celui du client. Pour résoudre ce problème, nous avons fait en sorte que le client envoie une commande au serveur. En plus de cela, cette commande possède l'attribut ClientRPC, ce qui permet au serveur d'exécuter la méthode sur le client spécifique et non sur l'hôte.

On avait également un problème concernant l'UI. La problématique résidait dans le fait qu'au début, lorsque plusieurs joueurs étaient connectés sur le même serveur, ils avaient tous la même UI. Par exemple, lorsque un joueur ramassait un livre, tous les autres joueurs voyaient le livre apparaître dans leur barre d'action.

En réalité, le problème venait du fait que nous n'avions pas mis en place une barre d'action unique pour chaque joueur. Nous avons donc résolu ce problème en créant une prefab de cette UI et, pour chaque joueur, dès leur connexion, nous instancions une UI unique.

Nous avons souhaité intégrer des effets visuels spécifiques pour renforcer l'immersion et l'atmosphère angoissante du gameplay. Parmi ces effets, la réduction de la vision du joueur — simulant une perte progressive de visibilité ou un champ de vision limité — devait jouer un rôle clé dans l'expérience de jeu. Cependant, la mise en place de ces effets s'est rapidement avérée beaucoup plus complexe que prévu. En multijoueur, la synchronisation des effets visuels entre les différentes instances du jeu est un vrai défi technique : certains joueurs pouvaient voir correctement ces réductions de vision tandis que d'autres ne les recevaient pas ou les voyaient de manière erratique. Ce comportement incohérent était souvent dû à des problèmes liés à la gestion des états visuels, à la latence réseau, et aux différences entre les clients.

Ces difficultés ont impacté non seulement la cohérence de l'expérience utilisateur mais aussi la stabilité générale du jeu, avec parfois des bugs qui pouvaient provoquer des crashes ou des ralentissements. Après plusieurs tentatives de correction et d'optimisation, nous avons constaté que ces effets, bien qu'intéressants d'un point de vue gameplay, ne pouvaient pas être déployés de manière fiable dans le temps imparti pour le projet. En parallèle, notre équipe disposait de ressources limitées et d'un calendrier serré, ce qui ne permettait pas d'investir davantage dans des fonctionnalités secondaires au détriment des fonctionnalités principales du jeu.

Par conséquent, nous avons fait le choix conscient de retirer ces effets de réduction de vision dans la version finale, afin de privilégier un jeu fonctionnel, stable et accessible à tous les joueurs, avec une expérience multijoueur fluide et sans bugs majeurs. Ce compromis a été motivé par la volonté de délivrer un produit fini de qualité plutôt que de risquer une expérience fragmentée ou dégradée, ce qui aurait pu nuire à la perception globale du jeu. Ce choix illustre bien la difficulté de concilier ambitions créatives et contraintes techniques dans le développement de jeux multijoueur, où la priorité reste souvent de garantir la jouabilité et la robustesse du système dans son ensemble.

2.3.2 Son fonctionnement final

Le système multijoueur de notre jeu est construit autour de Mirror, une bibliothèque réseau open source spécialement conçue pour Unity qui facilite la création d'expériences multijoueurs robustes et évolutives. Mirror permet de gérer aussi bien des architectures peer-to-peer que client-serveur, mais pour notre projet, nous avons opté pour une architecture serveur-client classique. Concrètement, un joueur agit comme hôte en lançant la partie sur sa machine, ce qui fait office de serveur local, tandis que les autres participants se connectent à cette session en entrant l'adresse IP de l'hôte. Cette méthode garantit un contrôle centralisé sur la logique du jeu, permettant une meilleure synchronisation et sécurité des données.



Une fois connectés, tous les joueurs évoluent dans une session partagée où chaque élément important de l'environnement, chaque état de joueur, et chaque interaction est synchronisé en temps réel. Cette synchronisation est rendue possible grâce à la structure modulaire de Mirror, qui s'appuie sur deux composants fondamentaux : NetworkIdentity et NetworkBehaviour. Chaque objet de jeu devant être synchronisé possède un NetworkIdentity, qui lui attribue un identifiant unique sur le réseau et le lie à un NetworkBehaviour, un script permettant de définir les comportements synchronisés entre le serveur et les clients.

Cette architecture permet notamment la réPLICATION instantanée et cohérente des actions des joueurs. Par exemple, lorsqu'un joueur collecte un objet clé, comme un livre ancien disséminé dans la carte, cette action est d'abord envoyée au serveur via une commande ([Command]) que le client exécute. Le serveur valide l'action, met à jour la progression dans son état central, puis diffuse cette modification à tous les clients connectés grâce à des appels RPC ([ClientRpc]). Ces appels garantissent que le livre collecté disparaît simultanément de la scène pour tous les participants, évitant toute incohérence ou exploitation potentielle. Cette même mécanique est utilisée pour gérer l'apparition dynamique d'objets importants, tels que le fragment d'âme qui apparaît une fois que les trois livres sont réunis, rendant cet élément visible et interactif pour tous en temps réel.

La gestion des collisions et des interactions physiques entre joueurs et objets est elle aussi intégrée dans ce système via des NetworkColliders. Ces colliders détectent les interactions entre joueurs et fragments d'âme, et transmettent les événements au serveur pour validation. Ainsi, quand un joueur touche un fragment, le serveur prend en charge la mise à jour de l'état global du jeu, notamment l'interface utilisateur partagée qui affiche la progression collective, avant de désactiver l'objet fragment pour tout le monde, garantissant que l'événement est unique et synchronisé.

Au-delà de la simple synchronisation d'objets, Mirror facilite la gestion complexe des états de jeu multijoueur à travers un système efficace de commandes et de RPC qui forment le cœur des communications client-serveur. Ces mécanismes permettent au client d'émettre des requêtes spécifiques au serveur — par exemple, pour demander la collecte d'un objet ou le déclenchement d'un événement — tandis que le serveur, en garantissant la logique et la validation des actions, notifie en retour tous les clients des changements d'état via des appels RPC. Ce double canal de communication assure que tous les joueurs vivent une expérience cohérente, où chaque interaction est validée et partagée instantanément.

Enfin, Mirror intègre des fonctionnalités avancées pour gérer la robustesse des connexions, comme la reconnexion automatique en cas de déconnexion temporaire et la compensation de latence réseau, ce qui est crucial pour maintenir une expérience multijoueur fluide et stable. Ces outils permettent également de gérer la synchronisation d'états complexes même en cas de décalages ou interruptions, garantissant que la progression collective reste toujours à jour et cohérente. Dans un contexte de jeu d'horreur coopératif tel que *Lurking Home*, où l'immersion et la coopération sont essentielles, Mirror s'avère être un choix judicieux, offrant un cadre solide pour construire une expérience en ligne dynamique, immersive et sans faille.

Host une partie

Rejoindre une partie

2.4 Intelligence Artificielle

2.4.1 Introduction

Alors que nos joueurs se déplacent dans un manoir effrayant, notre IA, elle, représente un monstre terrifiant qui hante les murs du manoir et chasse les joueurs. Ce monstre se déplace en explorant aléatoirement les différentes pièces du manoir. Mais gare aux joueurs s'ils passent à proximité, car il se mettra alors à les pourchasser.

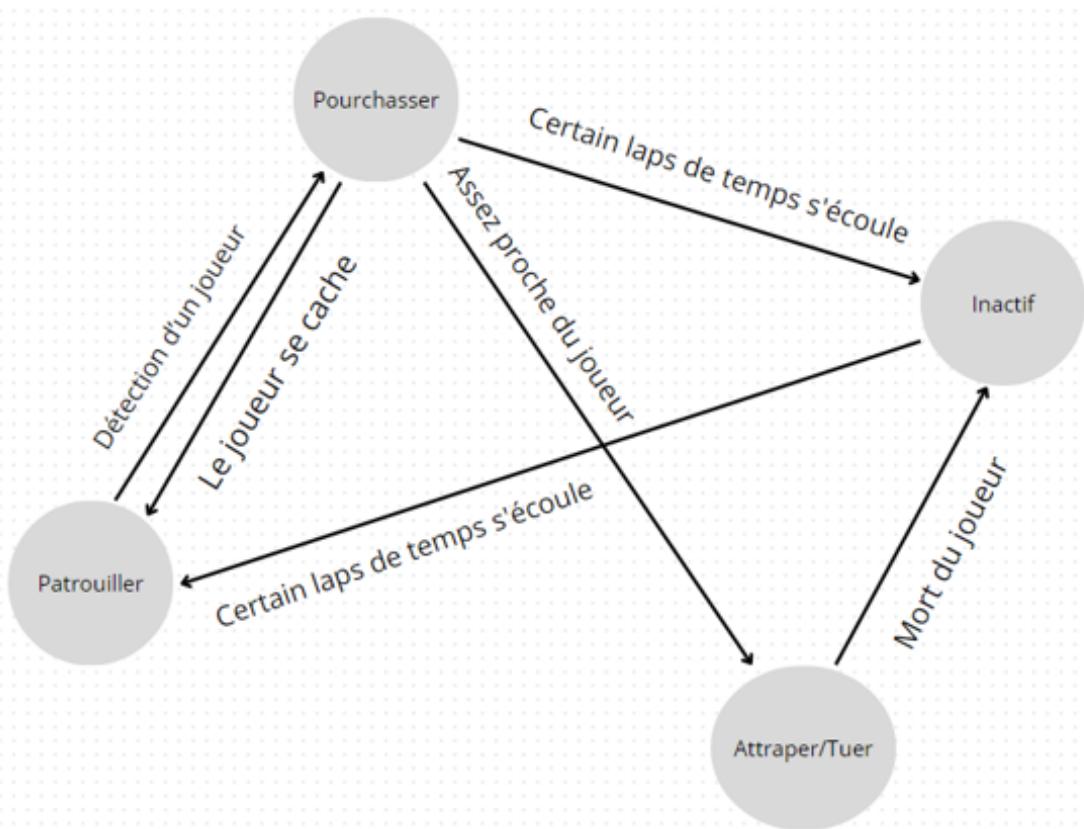
2.4.2 Soutenance 1

Pour rappel, l'IA de notre jeu est une entité qui pourchasse les joueurs dans la maison afin de les empêcher de s'évader. Pour définir l'IA de notre jeu, nous sommes partis sur un modèle appelé FSM (Finite State Machine), qui est un modèle reposant sur des états, des transitions et des entrées pour représenter les différentes actions de notre IA. Le graphe des actions de notre IA est le suivant :

Pour commencer la base de notre IA, nous avons décidé d'implémenter le pathfinding, ou algorithme de recherche de chemin en français. C'est un algorithme qui, comme son nom l'indique, sert à trouver un chemin (idéalement le plus court) entre un point A et un point B.

Nous avons choisi d'utiliser un type d'algorithme de pathfinding appelé A*. Cet algorithme est connu pour sa rapidité et son efficacité. L'algorithme A* est essentiel, car il est utilisé à la fois dans l'état « pourchasser » et dans l'état « patrouiller ».

Pour cette première soutenance, nous avons donc défini notre IA et implémenté un algorithme de pathfinding afin que notre IA puisse se déplacer.



2.4.3 Soutenance 2

Ensuite, nous avons voulu implémenter les différents états de notre IA. Pour ce faire, nous avons commencé par les états « pourchasser » et « patrouiller », étant donné qu'ils étaient tous deux liés à l'algorithme A*.

Pour l'état « pourchasser », il n'y avait pas grand-chose à faire : il suffisait simplement de définir la cible de notre IA comme étant le joueur.

En revanche, l'état « patrouiller » nécessitait davantage de mise en place. Il fallait d'abord comprendre et définir comment notre IA allait patrouiller. Celle-ci se déplace de manière aléatoire entre les salles déjà ouvertes par les joueurs, c'est-à-dire celles qui ont déjà été visitées.

Pour cela, nous avons d'abord instancié une nouvelle grille permettant de délimiter les différentes pièces. Nous avons ensuite créé un nouveau script permettant de suivre les mouvements du joueur et d'ouvrir les pièces lorsqu'il y entre.

Voici une image de notre carte. Les murs de la carte sont représentés en jaune, et on peut y voir les différentes cases de la grille avec quatre couleurs possibles :

- Le jaune représente la case dans laquelle se trouve le joueur.
- Les cases bleues représentent les couloirs de la map, qui sont donc des zones neutres.
- Les cases vertes et rouges représentent les différentes pièces de la carte :
- Les pièces vertes sont celles qui sont ouvertes.
- Les pièces rouges sont celles qui sont fermées.

Une fois ceci fait, nous avons créé un nouveau GameObject qui se place aléatoirement dans l'une des salles déjà ouvertes par le joueur. L'IA va alors se diriger vers ce GameObject, et si elle passe à proximité, cela signifie qu'elle est arrivée à destination. Le GameObject se déplace alors vers une nouvelle salle aléatoire.

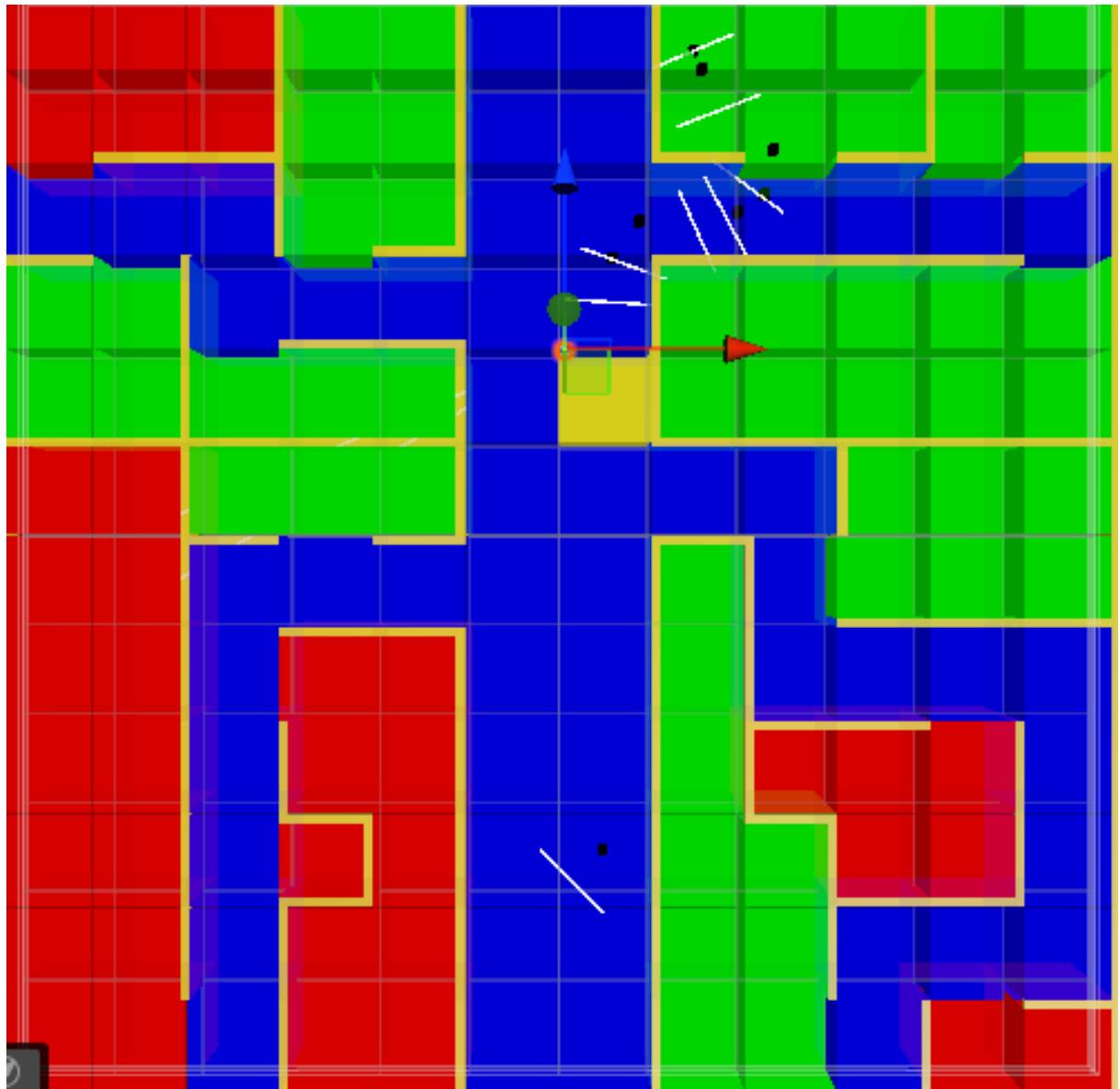
Ce système permet de simuler facilement le déplacement aléatoire de l'IA parmi les salles déjà ouvertes.

Les deux autres états qu'il restait à implémenter étaient :

- l'état inactif,
- et l'état permettant d'attraper ou de tuer le joueur.

Pour l'état inactif, il n'y avait pas grand-chose à faire : il suffisait simplement de bloquer les mouvements de l'IA et de la rendre invisible. Une fois ces trois états définis, il nous fallait mettre en place les différentes transitions. Nous avons seulement eu le temps de créer la transition entre l'état patrouiller et l'état pourchasser. Cette transition est assez simple : si le joueur passe à proximité de l'IA pendant qu'elle patrouille, la cible de l'IA change et passe du GameObject utilisé pour la patrouille directement au joueur.

Pour cette deuxième soutenance, nous avons donc instancié de nouveaux états pour notre IA, ainsi que de nouvelles transitions.



2.4.4 Soutenance 3

Enfin, nous avons dû relever notre plus grand défi jusque-là : faire en sorte que l'IA fonctionne en multijoueur.

En effet, jusqu'à présent, l'IA ne fonctionnait que dans une scène solo. Il nous fallait désormais l'adapter pour qu'elle fonctionne avec l'implémentation du multijoueur via Mirror. Pour cela, nous avons dû modifier tous nos scripts afin qu'ils prennent en compte la logique côté serveur. Nous avons également dû synchroniser cette IA pour qu'elle apparaisse correctement sur l'écran de tous les joueurs.

La deuxième amélioration que nous avons apportée est l'ajout d'un visuel pour notre monstre. Jusqu'ici, notre IA n'était rien de plus qu'une capsule. Nous avons donc cherché et intégré un asset trouvé sur l'Asset Store de Unity, afin de donner une apparence plus réaliste à notre entité. Nous sommes partis sur l'asset suivant car on le trouvait relativement effrayant et il correspondait aussi à l'ambiance du manoir qui possède de nombreux livres et de nombreuses bibliothèques. Enfin nous avons dû instancier le dernier état de notre ia à savoir le fait que quand le joueur est trop proche de l'IA celui-ci est automa-

tiquement mort. Pour ce faire nous avons utilisé des scripts très similaire à ceux utilisés pour la détection du Game object dans l'état patrouillée par exemple.



2.4.5 Difficultés

1ère soutenance Les difficultés rencontrées lors de cette première soutenance ont été principalement liées à la découverte complète de Unity. En effet, n'ayant aucune expérience préalable ni avec Unity, ni avec l'intelligence artificielle, nous avons dû tout apprendre depuis zéro, ce qui nous a pris beaucoup de temps.

Une seconde difficulté a été l'implémentation de l'algorithme A*, qui est relativement complexe. Sa mise en œuvre a demandé de nombreux efforts, tant en termes de compréhension que de programmation.

2ème soutenance Durant les semaines allouées à la programmation, nous avons rencontré plusieurs problèmes qui ont considérablement ralenti l'avancement du développement de l'IA.

Le premier problème était l'absence d'une carte de jeu complète. Bien que nous disposions déjà des plans avec les différentes pièces, la carte n'était pas encore implémentée dans Unity. Or, sans cette carte, il nous était impossible de déterminer quelles pièces étaient ouvertes ou non, ce qui rendait l'état patrouiller impossible à mettre en place. Nous avons donc dû commencer par implémenter un squelette de la carte dans Unity.

Le deuxième problème majeur a été l'intégration avec le multijoueur. En effet, nous avons tenté de fusionner nos avancées sur l'IA avec celles du multijoueur. Cela a entraîné un grand nombre d'erreurs que nous avons essayé de corriger, sans grand succès dans un premier temps. Ces échecs nous ont fait perdre beaucoup de temps sur le développement

de l'IA. Nous avons donc dû revenir en arrière pour pouvoir continuer à progresser.

Ces erreurs représentent un défi majeur pour la suite du développement. Par exemple, l'IA ne peut plus avoir une seule cible, puisque plusieurs joueurs seront présents simultanément sur la carte. De même, l'ouverture des pièces devra être repensée, car plusieurs joueurs peuvent les ouvrir en même temps. Il faudra donc réadapter la majorité des scripts existants.

3ème soutenance Les difficultés rencontrées lors de cette soutenance ont été encore plus nombreuses que lors des précédentes.

En effet, implémenter l'IA en multijoueur a généré de nombreuses erreurs, comme mentionné précédemment, et nous avons passé beaucoup de temps à toutes les corriger.

De plus, ayant pris du retard lors des précédentes étapes, il a été difficile de rattraper ce délai tout en poursuivant les ajouts nécessaires pour l'IA.

Une autre difficulté importante a été notre décision, après la deuxième soutenance, de réduire la taille de notre carte afin de la rendre plus compacte. Ce changement a naturellement impacté certaines fonctions, qu'il a fallu modifier ou adapter en conséquence.

2.5 Son

2.5.1 L'importance du son dans un jeu d'horreur

Dans un jeu d'horreur, le son joue un rôle fondamental dans la création d'une ambiance immersive et angoissante. Il ne s'agit pas seulement d'ajouter de la musique, mais de construire un univers sonore cohérent qui renforce la tension et maintient le joueur dans un état d'alerte constant.

Conscients de cet enjeu, nous avons passé beaucoup de temps à écouter différents assets sonores disponibles sur le Unity Asset Store. Après une sélection rigoureuse, nous avons retenu deux musiques qui nous semblaient particulièrement adaptées à l'ambiance que nous souhaitions créer.

La première musique est utilisée dans le menu principal du jeu. Il s'agit d'une mélodie oppressante et pesante, qui plonge immédiatement le joueur dans une atmosphère sombre et inquiétante, dès les premiers instants. Elle sert d'introduction à l'univers du jeu, posant les bases de l'angoisse à venir.

La seconde musique est jouée pendant la partie. Elle est plus discrète, afin de ne pas trop perturber l'attention du joueur, mais elle conserve une dimension inquiétante. Des bruits sourds et irréguliers, qui semblent provenir de l'intérieur des murs, contribuent à instaurer une ambiance pesante et menaçante, parfaite pour maintenir la tension.

En complément de ces musiques, nous avons intégré des effets sonores spécifiques pour renforcer l'immersion. Par exemple, au lancement du jeu, on peut entendre le cri d'un monstre terrifiant, ce qui permet de saisir le joueur dès les premières secondes. Ce cri peut également se faire entendre lorsque le monstre détecte la présence d'un joueur pendant la partie, ce qui accentue la peur et le stress dans ces moments critiques.

Pour résumer, nous sommes convaincus que le son fait toute la différence entre un bon jeu d'horreur et un jeu oubliable. C'est pourquoi nous avons accordé une attention particulière au choix des musiques et effets sonores, afin de créer une expérience audio

aussi immersive qu'effrayante.

2.6 Site Web

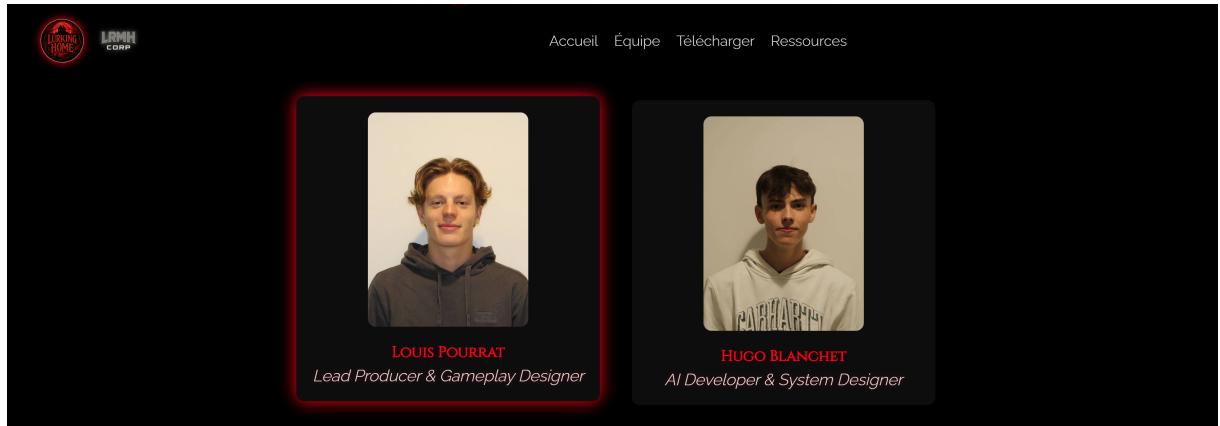
En parallèle du développement du jeu, on a assuré la conception, l'intégration et la gestion du site web officiel de Lurking Home. Ce site occupe une place importante dans la stratégie de communication du projet : il permet de présenter le jeu, de donner un aperçu de son univers, de valoriser l'équipe de développement et de fournir à terme un lien de téléchargement. Bien que le site n'ait pas connu d'évolution majeure depuis sa mise en ligne, il remplit pleinement sa fonction actuelle de vitrine du projet. Sa création a représenté un travail complet, mêlant réflexion esthétique, structuration du contenu, et maîtrise technique.

Dès le départ, plusieurs objectifs ont été définis pour ce site : simplicité, lisibilité, et surtout, immersion. Le jeu Lurking Home étant un titre à l'ambiance sombre, une charte visuelle a été choisie pour refléter cette identité : un fond noir profond, des titres rouges lumineux, et des textes blancs discrets. La typographie est fine et moderne, volontairement sobre pour laisser la place à l'ambiance. Chaque élément visuel a été sélectionné pour créer une continuité avec l'univers du jeu : le rouge symbolise le danger, l'inconnu, l'urgence ; le noir est omniprésent, il engloutit, il inquiète. Le site ne devait pas simplement "présenter" le jeu, il devait en donner un avant-goût.

Le site est structuré autour de quatre pages principales : Accueil, Équipe, Téléchargement, et Ressources. Chacune de ces pages a été conçue en HTML et stylisée avec du CSS pur, sans framework externe, dans un souci de légèreté et de maîtrise. La page d'accueil introduit immédiatement l'univers du jeu avec un titre central animé visuellement par un effet de lumière rouge, obtenu via text-shadow. En dessous, une phrase d'accroche renforce l'immersion, et un bouton bien visible propose d'"Explorer l'univers", point de départ vers les autres sections. L'enjeu ici était de capturer l'attention en quelques secondes : un design épuré, direct, immersif.



La page Équipe permet de valoriser les membres du projet. Leurs portraits ont été intégrés avec un style uniforme : fond noir, encadré discret, nom en rouge et rôle en blanc. Pour cela, le système Flexbox a été utilisé, ce qui a permis d'aligner les éléments dynamiquement en fonction de la taille de l'écran. Quelques petits soucis techniques ont été rencontrés, notamment sur l'alignement vertical des blocs sur mobile. Après test, il s'est avéré que la hauteur des images déséquilibrerait l'ensemble : ce problème a été corrigé en fixant des hauteurs minimales aux conteneurs, ce qui a permis d'harmoniser l'affichage quel que soit le support. Cette page, bien que sobre, rappelle que Lurking Home est un projet collectif, avec des compétences variées.



La page Téléchargement a été pensée comme une interface simple et directe, prête à accueillir les futures versions jouables du jeu. Deux boutons y ont été insérés : un pour la version Windows, un autre pour Mac. Ces boutons sont actuellement inactifs mais fonctionnels en structure : il ne restera plus qu'à insérer les liens de téléchargement une fois les builds prêts. Le style de ces boutons suit la même logique que l'ensemble du site : rouge vif, effet hover plus clair, avec une transition douce. Un court texte explicatif a également été intégré pour rassurer le visiteur sur la compatibilité des fichiers. Là aussi, un soin particulier a été apporté à la lisibilité du design sur toutes les tailles d'écran, en adaptant les marges, les tailles de police, et en testant les proportions sur plusieurs navigateurs.

TÉLÉCHARGER LE JEU

Bientôt disponible en téléchargement. Préparez-vous à entrer dans Lurking Home...

Version PC

Compatible avec Windows 10 et supérieur.

[TÉLÉCHARGER](#)

Version Mac

Compatible avec macOS 10.15 et supérieur.

[TÉLÉCHARGER](#)

Enfin, la page Ressources, encore en cours de développement, accueillera à terme des éléments téléchargeables tels que des affiches, des musiques, ou des documents narratifs. Elle a déjà été structurée en blocs verticaux, facilement dupliquables, avec une marge suffisante pour ajouter des médias ou des liens. Cette page est pensée comme une extension du lore du jeu, une façon d'immerger encore davantage les joueurs dans l'univers de Lurking Home, même en dehors du gameplay.

The screenshot shows a dark-themed website for 'Lurking Home Corp'. At the top, there's a navigation bar with the logo 'LRMH >' and links for Accueil, Équipe, Télécharger, and Ressources. Below the header, a large red-bordered box contains the title 'LE DÉVELOPPEMENT DE LURKING HOME'. Inside this box, a sub-section titled 'OCTOBRE 2024 – IDÉE DU PROJET' is visible. This section discusses the initial concepts of the psychological horror game, mentioning the team's focus on atmosphere and tension rather than action. It also highlights the creation of a dark, silent environment with mysterious sounds and shadows. Navigation arrows at the bottom right indicate the ability to scroll through more content.

Tout au long de cette création, un code propre, lisible et facilement modifiable a été produit. Chaque section des fichiers HTML et CSS a été commentée, ce qui permettra une reprise aisée en cas de refonte du site. Des éléments techniques essentiels comme un favicon, des balises de référencement (meta-title, meta-description), ont également été intégrés, et la validation du code a été vérifiée via les outils W3C.

Ce travail sur le site a permis d'approfondir les compétences en développement web, mais surtout, de réfléchir à la manière de transposer une atmosphère de jeu dans une interface web. Il ne s'agissait pas simplement de coder des pages, mais de créer un univers parallèle, statique mais expressif, qui accompagne le joueur même en dehors de la partie. Chaque élément – la couleur, le rythme, le silence graphique – participe à cette immersion.

2.7 Conclusion

En conclusion, beaucoup de travail a été accompli au cours de ces derniers mois. Nous avons tout d'abord repensé et reconstruit intégralement la carte de notre jeu, en réduisant sa taille pour la rendre plus cohérente et plus jouable. Nous y avons ensuite ajouté un grand nombre de meubles et d'éléments de décor afin de créer un environnement immersif et crédible. Ce travail sur l'environnement s'est également accompagné de l'ajout de bruitages d'ambiance et de musiques, renforçant l'atmosphère inquiétante que nous souhaitions transmettre.

Nous avons aussi développé plusieurs missions, comme une quête où il faut récupérer des livres, une autre où le joueur doit trouver les bons ingrédients dans la cuisine, ou encore une mission d'allumage de bougies dans un ordre précis. Chacune de ces missions nous a demandé de concevoir des mécaniques spécifiques et de les intégrer proprement au gameplay général.

Un des aspects les plus techniques a été le développement de l'intelligence artificielle. Nous avons implémenté des systèmes complexes de pathfinding, notamment l'algorithme A*, ainsi que des mécanismes de changement d'état pour rendre le comportement du monstre plus réaliste et imprévisible. Nous avons également dû relever le défi d'adapter cette IA au mode multijoueur, en utilisant Mirror, une solution réseau pour Unity. Ce passage au multijoueur nous a demandé de revoir en profondeur notre architecture de scripts pour bien prendre en compte les rôles du client et du serveur.

En dehors du développement du jeu lui-même, nous avons aussi appris à créer un site web pour présenter notre projet, ainsi qu'à gérer l'identité visuelle de notre jeu et de notre studio fictif. Cela inclut la création d'un logo, d'une jaquette, et d'autres éléments de communication. Ce projet nous a permis de découvrir et d'apprendre une multitude de choses. Nous avons appris à utiliser Unity, à comprendre des notions avancées en programmation et en game design, à rechercher et sélectionner des assets pertinents, à utiliser des concepts courants dans le développement de jeux comme les raycasts, les structures de données complexes (par exemple les tas), ou encore à comprendre le fonctionnement des connexions serveur/client pour le multijoueur.

Mais au-delà des compétences techniques, ce projet nous a aussi formés au travail d'équipe. Nous avons appris à gérer un projet sur le long terme, à répartir les tâches, à surmonter des désaccords, et à prendre en compte les idées de chacun. Pour beaucoup d'entre nous, il s'agissait de notre premier projet d'envergure, et ce fut une expérience très enrichissante sur les plans personnel et collectif.

Cela dit, le projet n'a pas été sans difficultés. La plus grande erreur que nous avons

commise a été de sous-estimer le temps nécessaire pour mettre en œuvre certaines idées, parfois trop ambitieuses ou trop complexes. Cela a eu pour conséquence de nous faire perdre un temps précieux, au détriment d'éléments plus essentiels. Le manque d'expérience dans la gestion de projet en groupe a aussi posé des problèmes, notamment au niveau de la communication et de la répartition des tâches, ce qui a parfois ralenti notre progression.

Malgré tout cela, nous sommes fiers du travail accompli et reconnaissants d'avoir pu participer à un projet aussi complet. Si nous devions refaire ce projet, nous anticiperions davantage certaines étapes et prendrions plus de temps pour peaufiner les détails et améliorer la finition du jeu.