

# Rapport de soutenance



LRMH Corp

*Epita*

**Date :** 13 mars 2025

**Membres :** Pourrat Louis  
Bruche Martin  
Tibar Rayanne  
Blanchet Hugo

Année 2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Présentation du projet . . . . .	2
1.2	Rappels sur les tâches individuelles et communes . . . . .	2
<b>2</b>	<b>Avancements du projet</b>	<b>2</b>
2.1	Gameplay . . . . .	2
2.1.1	Système d'inventaire . . . . .	2
2.1.2	Récupération d'objets . . . . .	2
2.2	UI du joueur . . . . .	3
2.2.1	Barre d'action . . . . .	3
2.3	Animations . . . . .	4
2.3.1	Déplacement du joueur . . . . .	4
2.3.2	Récupération d'un objet . . . . .	6
2.4	Multijoueur . . . . .	7
2.4.1	Synchronisation récupération item . . . . .	7
2.4.2	Synchronisation du UI des joueurs . . . . .	8
2.4.3	Synchronisation animations . . . . .	8
2.5	Intelligence Artificielle . . . . .	8
2.5.1	Rappel . . . . .	8
2.5.2	Les états . . . . .	9
2.5.3	Les transitions . . . . .	10
2.5.4	Les difficultés . . . . .	10
2.6	Mission . . . . .	11
2.6.1	Début, Difficultés et Solutions . . . . .	11
2.6.2	Création du Niveau . . . . .	12
2.7	Carte . . . . .	13
2.8	Assets du jeu . . . . .	14
2.9	Site web . . . . .	16
2.9.1	Introduction . . . . .	16
2.9.2	Processus de Conception et Choix Esthétiques . . . . .	16
2.9.3	Défis Techniques et Solutions Apportées . . . . .	17
2.9.4	Fonctionnalités et Optimisations en Cours . . . . .	18
2.10	Jaquette . . . . .	18
<b>3</b>	<b>Conclusion</b>	<b>18</b>
3.1	Rappels sur les avancées . . . . .	18
3.2	Les choses à faire pour la prochaine fois . . . . .	19
3.2.1	Gameplay . . . . .	19
3.2.2	Multijoueur . . . . .	19
3.2.3	IA . . . . .	19
3.2.4	Mission . . . . .	19
3.2.5	Site Web . . . . .	19

# 1 Introduction

## 1.1 Présentation du projet

Notre projet s'intitule Lurking Home. C'est un jeu d'horreur et d'échappée 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 ces 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, accomplissement de tâches et tension constante, tout en plaçant les joueurs dans une course contre la montre pour leur survie.

## 1.2 Rappels sur les tâches individuelles et communes

	Louis	Rayane	Hugo	Martin
Intelligence artificielle				
Décor et joueurs design				
Site web				
Multijoueur				
Logo et jacket				
Bande sonore				
Gameplay				
Histoire et DA				
Conception des missions				

	Responsable
	Suppléant

FIGURE 1 – tableau des taches

## 2 Avancements du projet

### 2.1 Gameplay

#### 2.1.1 Système d'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.

#### 2.1.2 Récupération d'objets

La partie la plus importante de l'avancement sur le gameplay est celle-ci : la récupération d'objet. Avant de parler de cette fonctionnalité, il est important de mentionner

que nous avons mis en place un système d'instance d'objets. Cela signifie que nous allons créer des instances pour chaque objet de notre jeu, et ces instances auront un nom, un visuel (correspondant au visuel 2D qui sera affiché dans la barre d'action du joueur), ainsi qu'une prefab de l'objet, comportant souvent tous les composants Unity nécessaires ainsi que les scripts que nous avons codés. Pour l'instant, nous avons seulement implémenté l'instance du livre afin de tester notre système de récupération d'items. L'avantage de ce système est que, par la suite, il suffira de créer une nouvelle instance de l'objet voulu, et notre système fonctionnera directement. Dès lors que le joueur ramasse un objet, c'est l'instance de cet objet qui sera ajoutée à son inventaire.

Ce 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.



FIGURE 2 – raycast

## 2.2 UI du joueur

(UI = User Interface = Interface Utilisateur)

Nous avons donc créé une UI pour le joueur, en utilisant des canvas pour afficher tout ce qui va apparaître sur l'écran du joueur lors de sa connexion. Pour l'instant, nous avons simplement implémenté un curseur temporaire, qui a surtout permis de tester la fonctionnalité de récupération d'items, ainsi qu'une barre d'action.

### 2.2.1 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. Comme nous n'avons que le livre comme instance d'objet pour l'instant, seul le visuel 2D du livre est affiché.

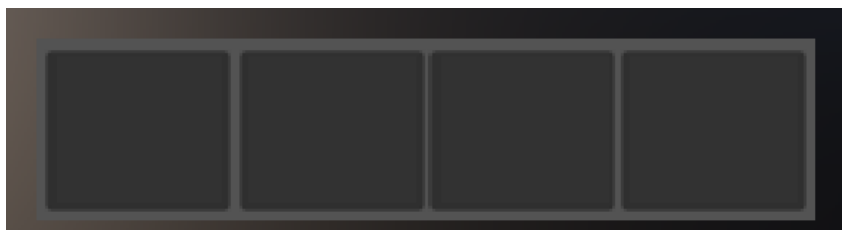


FIGURE 3 – Barre d'action

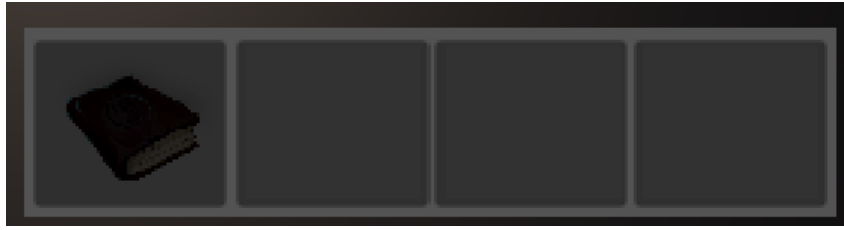


FIGURE 4 – Barre d'action avec objet

## 2.3 Animations

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

### 2.3.1 Déplacement 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.



FIGURE 5 – Asset de notre joueur



FIGURE 6 – Animation de marche

### 2.3.2 Récupération d'un objet

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.



FIGURE 7 – Animation de récupération d'objet

## 2.4 Multijoueur

Le multijoueur a été l'une des parties ayant posé le plus de problèmes. Nous avons donc principalement dû résoudre des problèmes de synchronisation liés aux différentes fonctionnalités que nous avons ajoutées.

### 2.4.1 Synchronisation récupération item

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.



### 2.4.2 Synchronisation du UI des joueurs

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, nousinstancions une UI unique.

### 2.4.3 Synchronisation animations

Les problèmes de synchronisation des animations entre les joueurs étaient similaires à ceux rencontrés lors de la récupération d'objets. Il a donc fallu que nous utilisions les principes de Command et de ClientRPC pour résoudre ces problèmes.

## 2.5 Intelligence Artificielle

### 2.5.1 Rappel

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 :

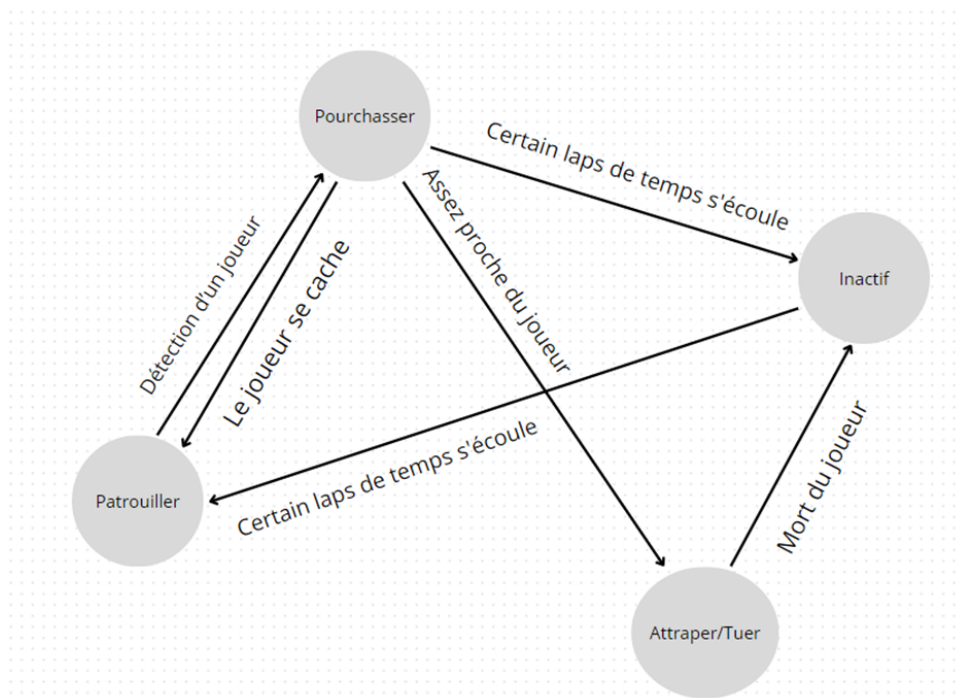


FIGURE 8 – Graphique FSM de l'IA

Pour commencer la base de notre IA, nous avons décidé de commencer par 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'A\* est important, car il est nécessaire pour l'état "pourchasser" ainsi que pour l'état "patrouiller".

### 2.5.2 Les états

Ensuite, nous avons voulu implémenter les différents états de notre IA. Pour ce faire, nous avons commencé par l'état "pourchasser" ainsi que "patrouiller", étant donné qu'ils étaient tous les deux liés à l'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. Pour l'état "patrouiller", en revanche, il y avait plus de choses à mettre en place. Il fallait d'abord comprendre et définir comment notre IA allait patrouiller. L'IA se déplace de manière aléatoire entre les salles déjà ouvertes par les joueurs, c'est-à-dire celles qui ont déjà été visitées. Pour ce faire, nous avons d'abord instancié une nouvelle grille nous permettant de délimiter les différentes pièces. Nous avons ensuite créé un nouveau script permettant de traquer les mouvements du joueur et d'ouvrir les pièces lorsqu'il y entre.

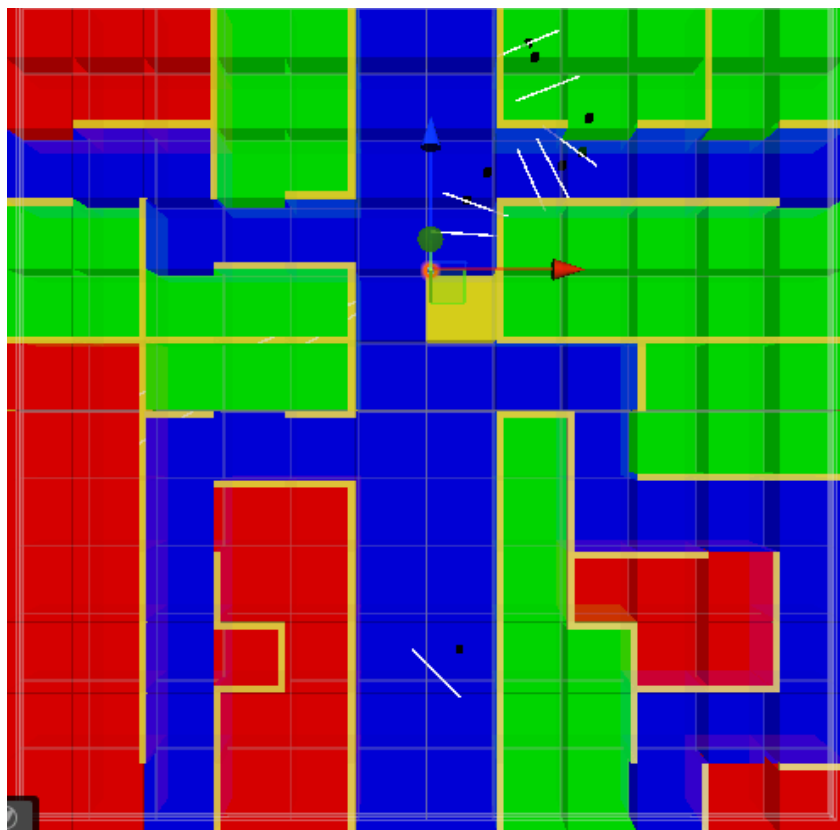


FIGURE 9 – Grille de notre carte

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` va alors se déplacer 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 instancier étaient l'état inactif ainsi que 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. Concernant l'état permettant d'attraper ou de tuer le joueur, nous n'avons pas eu le temps de l'instancier pour l'instant.

### **2.5.3 Les transitions**

Une fois ces trois états définis, il nous fallait maintenant mettre en place les différentes transitions. Nous avons seulement eu le temps de créer la transition entre l'état "pourchasser" et l'état "patrouiller". 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.

### **2.5.4 Les difficultés**

Durant les dernières semaines qui nous ont été données pour programmer, nous avons rencontré quelques problèmes qui nous ont fortement ralenti dans la création de cette IA. Le premier problème était l'absence d'une carte de jeu complète. Nous avions déjà les plans de cette carte avec les différentes pièces, mais elle n'était pas encore implémentée sur Unity. Or, sans celle-ci, il nous était impossible de définir quelles pièces étaient ouvertes ou non, rendant l'état "patrouiller" impossible à implémenter. Nous avons donc dû commencer par implémenter un squelette de notre carte sur Unity.

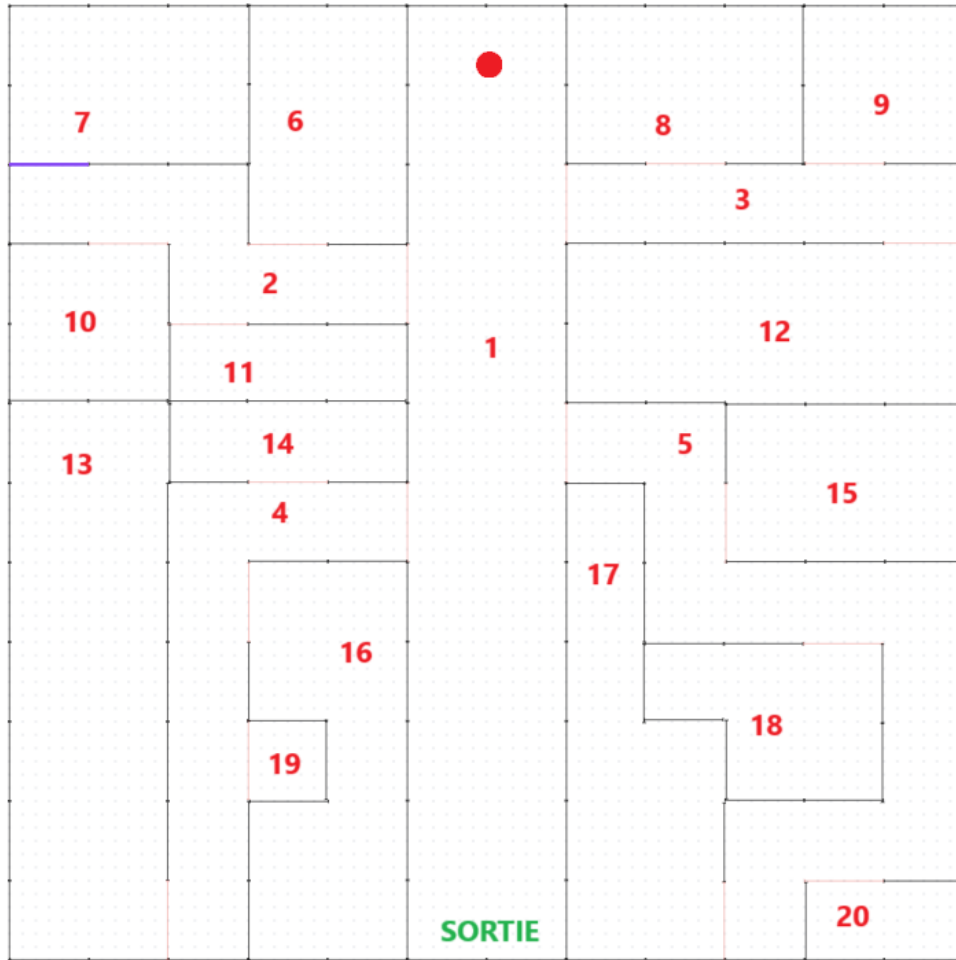


FIGURE 10 – Exemple de notre carte

Voici le plan de notre carte avec les différentes pièces numérotées. Le deuxième problème majeur a été l'intégration avec le multijoueur. En effet, nous avons essayé de fusionner nos avancées sur l'IA avec nos avancées sur le multijoueur. Or, cela a généré un très grand nombre d'erreurs que nous avons tenté de résoudre tant bien que mal, sans succès. Ces échecs nous ont fait perdre beaucoup de temps sur le développement de l'IA. Nous avons donc dû annuler et revenir en arrière afin de pouvoir progresser sur le développement de notre IA. Ces erreurs représenteront un défi majeur pour la suite du développement. Un exemple concret est que l'IA ne possède pas une seule cible, car plusieurs joueurs seront présents simultanément sur la carte. Il en va de même pour l'ouverture des pièces : plusieurs joueurs pouvant les ouvrir en même temps, nous allons devoir réadapter la plupart des scripts.

## 2.6 Mission

### 2.6.1 Début, Difficultés et Solutions

Dans cette deuxième phase du projet, l'un des objectifs principaux était de commencer l'implémentation concrète des missions dans Unity. Jusqu'à présent, nous avons surtout travaillé sur leur conception théorique, en réfléchissant aux mécaniques de jeu et à la manière dont elles pouvaient s'intégrer à l'univers du projet. Cette nouvelle étape devait nous permettre de transformer ces idées en éléments interactifs et jouables.

Cependant, plusieurs difficultés ont rapidement émergé, rendant cette mise en place plus complexe que prévu. Le principal problème a été un manque de temps. Ayant débuté trop tard l'implémentation, nous n'avons pas pu avancer suffisamment pour obtenir une version fonctionnelle des missions à temps pour la remise du rapport. Le développement d'une mission dans Unity demande de passer par plusieurs étapes : la création du décor, l'intégration des objets interactifs, la gestion des événements déclenchés par le joueur et, dans certains cas, la prise en compte des interactions avec l'intelligence artificielle. Ce processus prend du temps, surtout lorsqu'il s'agit des premières tentatives et qu'il faut encore s'adapter aux outils du moteur de jeu.

Un autre frein a été le manque de connaissances sur certains aspects d'Unity et de C#. Même si nous avons déjà quelques bases sur l'environnement de développement, certaines fonctionnalités comme la gestion des événements interactifs, les déclencheurs d'action et l'intégration d'éléments dynamiques nécessitent un certain apprentissage. Le travail dans Unity n'est pas seulement une question d'assembler des objets visuels : il faut aussi créer une logique qui permette au jeu de comprendre et de réagir aux actions du joueur. Ce processus nécessite l'écriture de scripts en C#, une compétence que nous sommes encore en train d'acquérir.

De plus, une désynchronisation avec le travail de l'IA a aussi été un facteur limitant. Les travaux ont été réalisés séparément, et il n'y a pas eu d'intégration directe entre nos avancées respectives. Or, pour certaines missions, notamment celles où le joueur doit éviter une entité hostile, il est essentiel que l'IA puisse interagir avec les éléments du décor et avec les actions du joueur. Pour avancer plus efficacement, il sera donc nécessaire d'unifier ces travaux et d'intégrer progressivement les missions au système global du projet.

Afin de combler ce retard et avancer plus rapidement, nous nous sommes concentrés dans un premier temps sur la création d'une pièce complète qui servira de support à l'une des missions principales du jeu. L'idée était de poser une base solide en réalisant un décor crédible et immersif avant de nous attaquer à l'intégration des mécaniques de gameplay. C'est ainsi que nous avons décidé de travailler sur la conception de la bibliothèque, un lieu clé du jeu, où se déroulera une mission axée sur l'exploration et la découverte d'indices.

### **2.6.2 Création du Niveau**

Avec la nécessité d'avancer rapidement et d'avoir un support concret sur lequel intégrer les futures mécaniques de jeu, nous avons entrepris la création d'un premier environnement entièrement conçu dans Unity : une bibliothèque mystérieuse et oppressante, qui servira de décor à l'une des missions centrales du projet.

L'un des premiers défis a été de trouver les ressources adaptées pour créer un espace réaliste et immersif sans passer trop de temps à modéliser chaque élément individuellement. Pour cela, nous avons exploré la boutique Unity, où nous avons trouvé un pack d'assets gratuits contenant des éléments essentiels, tels que des étagères remplies de livres, des meubles en bois anciens et des objets décoratifs, correspondant parfaitement à l'ambiance du jeu. L'utilisation de ces assets a été un véritable gain de temps, nous permettant de nous concentrer davantage sur l'agencement et l'optimisation de l'espace plutôt que sur la modélisation.

Cependant, l'utilisation d'assets préfabriqués ne signifie pas que tout s'intègre immédiatement sans ajustements. Il a fallu réorganiser les objets, ajuster leurs dimensions et positions pour éviter des problèmes de collision et assurer une navigation fluide du

joueur dans la pièce. Nous avons également dû ajuster les textures et l'éclairage pour que l'ensemble soit homogène et corresponde à l'identité visuelle du jeu.

Un autre aspect important dans la conception de cette bibliothèque a été la gestion de l'espace et de la circulation. Il fallait s'assurer que le joueur puisse se déplacer naturellement à travers la pièce tout en étant guidé vers des points d'intérêt spécifiques. Nous avons donc réfléchi à un agencement logique, plaçant des rayonnages de livres formant des couloirs étroits, renforçant le sentiment d'oppression et de claustrophobie, tout en laissant suffisamment d'espace pour que le joueur puisse explorer sans se sentir perdu.

Pour aider à structurer la pièce de manière cohérente, nous avons utilisé une IA pour calculer les dimensions optimales et vérifier que tout était bien proportionné. Une fois ces mesures obtenues, nous avons ensuite procédé à l'agencement manuel des éléments pour optimiser le rendu final.

Enfin, pour renforcer l'immersion et préparer le décor aux futures mécaniques de jeu, nous avons commencé à expérimenter avec l'éclairage et les effets visuels. Nous avons testé l'ajout de sources lumineuses tamisées pour créer des zones d'ombre et mettre en valeur certaines parties de la pièce. L'idée était de jouer avec la lumière pour orienter subtilement le joueur vers des éléments interactifs, tout en conservant une atmosphère angoissante et mystérieuse.

## 2.7 Carte

L'un des éléments fondamentaux du jeu est la map, qui définit l'environnement dans lequel les joueurs évolueront. Sa conception représente un défi majeur, car elle doit être à la fois immersive, cohérente et optimisée pour garantir une expérience de jeu fluide. Compte tenu de l'ampleur de cette tâche, nous avons décidé de partager le travail au sein du groupe afin d'accélérer la progression et d'assurer une meilleure répartition des responsabilités.

Nous avons divisé la map en plusieurs zones, chacune devant être modélisée avec soin pour respecter l'ambiance du jeu. La map mesure 120x120 unités et est composée de cases de 10x10 unités, permettant ainsi une gestion précise et détaillée des différents éléments qui la composent.

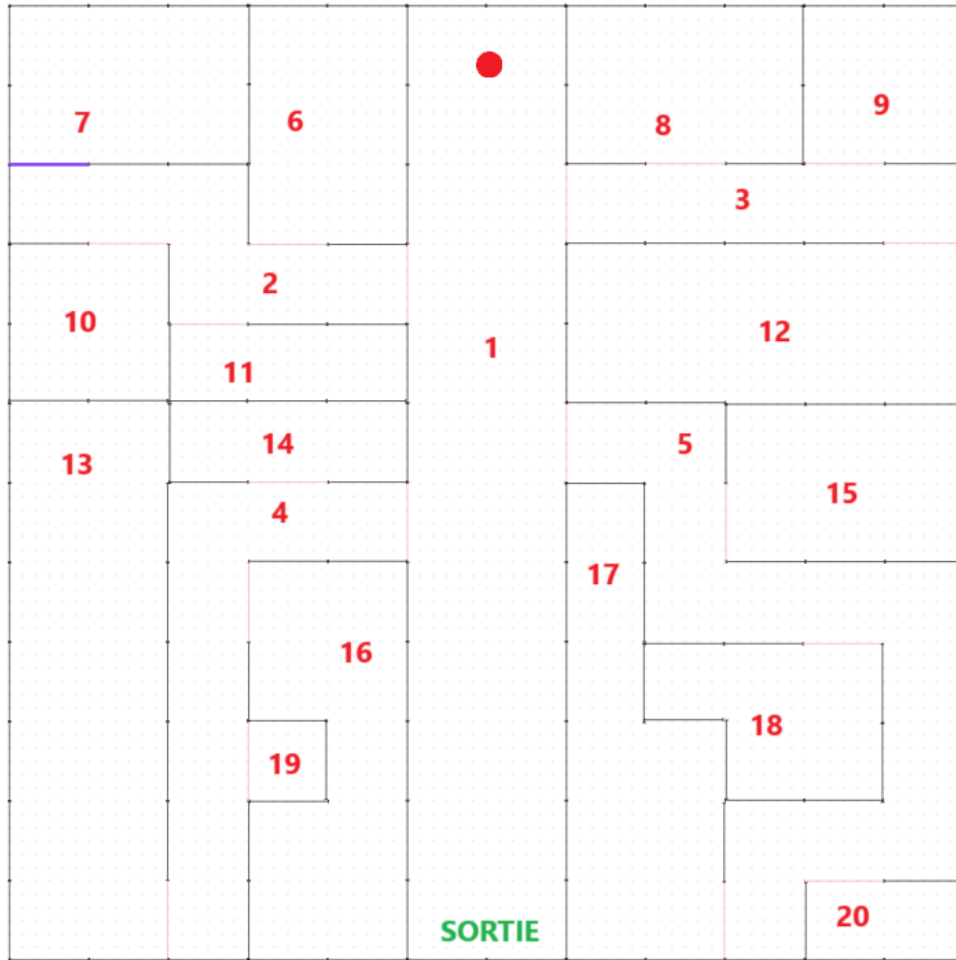


FIGURE 11 – Exemple de notre carte

Notre carte représente donc un manoir, et nous avons décidé de structurer les différentes pièces autour d'un couloir principal (la pièce numéro 1) et de plusieurs couloirs secondaires (les pièces numéros 2, 3, 4 et 5). Par la suite, nous avons délimité le reste de l'espace en 15 pièces distinctes, incluant les pièces classiques qu'on retrouve dans une maison, comme une chambre, une cuisine, ou encore une salle de bain.

La collaboration pour la création de la carte a été particulièrement bénéfique, car elle nous a permis d'échanger nos idées et d'apporter des améliorations en fonction des retours de chacun. Cependant, cette collaboration a également nécessité une coordination efficace pour éviter les doublons ou les erreurs de placement des éléments. Nous avons utilisé des outils de gestion de projet pour mieux nous organiser et suivre l'avancement des différentes parties de la map.

Bien que la carte ne soit pas encore totalement finalisée, une grande partie du travail a déjà été accomplie. Il reste encore quelques ajustements à faire, notamment au niveau des textures et des effets de lumière, mais nous sommes sur la bonne voie pour obtenir un résultat final satisfaisant.

## 2.8 Assets du jeu

Voici certains assets que nous avons déjà implémentés dans notre jeu :



FIGURE 12 – Asset livre



FIGURE 13 – Asset joueur



## 2.9 Site web

### 2.9.1 Introduction

Le développement du site web Lurking Home suit une approche progressive visant à offrir une vitrine immersive et esthétique pour le projet de jeu vidéo d'horreur. L'objectif principal est de concevoir un site à la fois simple, lisible et immersif, en accord avec l'ambiance sombre et mystérieuse du jeu. Dans cette phase du développement, l'accent a été mis sur la création et l'optimisation de la page d'accueil, en définissant une identité visuelle marquée et en améliorant l'expérience utilisateur.

Le design du site repose sur une structure minimaliste, évitant une surcharge d'éléments tout en mettant en avant l'essentiel : le titre du jeu, une brève description et un appel à l'action via un bouton interactif. Le travail effectué jusqu'à présent s'est concentré sur le choix des polices, des couleurs, de la mise en page et des effets visuels pour garantir une première impression percutante aux visiteurs.

### 2.9.2 Processus de Conception et Choix Esthétiques

Dès le départ, nous avons une idée assez précise de ce que nous voulions pour la page d'accueil du site. Notre objectif était de créer une interface à la fois simple et percutante, plongeant immédiatement l'utilisateur dans l'univers du jeu.

Pour affiner cette vision, nous avons recherché des références en ligne, exploré différents designs de sites et analysé les tendances graphiques susceptibles de correspondre à l'ambiance sombre et immersive de Lurking Home.



FIGURE 14 – Visuel site web

Nous avons opté pour un design épuré et sombre, mettant en avant un fond noir intégral qui renforce l'aspect mystérieux et inquiétant du jeu. Les éléments principaux, comme le titre en rouge vif avec un effet de néon, ont été travaillés pour capter l'attention sans trop alourdir la mise en page. Ce choix de couleur donne une identité forte au site tout en rappelant les codes visuels des jeux d'horreur.

Le texte principal est affiché au centre de l'écran, accompagné d'une phrase d'accroche courte qui intrigue le visiteur et l'incite à explorer davantage. Juste en dessous, un bouton rouge "Explorer l'univers" invite à l'interaction et apporte une touche dynamique à

la page. Nous avons voulu que ce bouton soit visuellement impactant, tout en restant cohérent avec l'esthétique générale.

Pour assurer une navigation fluide, nous avons mis en place un menu fixe discret situé en haut de la page. Il permet un accès rapide aux différentes sections du site (Accueil, Équipe, Télécharger, Ressources), tout en s'intégrant parfaitement au style minimaliste du site. Au survol, les liens passent du blanc au rouge avec un léger effet de lumière diffuse, renforçant encore l'ambiance visuelle du projet.

Bien que ces choix puissent paraître simples, l'ensemble a nécessité plusieurs ajustements pour obtenir un rendu harmonieux. Il nous a fallu trouver un équilibre entre lisibilité et ambiance immersive, sans tomber dans un excès d'effets visuels qui aurait pu nuire à l'expérience utilisateur.



FIGURE 15 – Visuel site web

### 2.9.3 Défis Techniques et Solutions Apportées

Même si la conception de cette première version du site s'est appuyée sur des bases solides, plusieurs défis techniques ont rendu le processus plus complexe que prévu. L'un des principaux problèmes a été d'obtenir un rendu visuel équilibré. Parfois, certains éléments ne s'affichaient pas comme nous le souhaitions, ou l'effet escompté n'était pas aussi percutant que prévu.

Le premier obstacle concernait l'effet néon du titre principal. Nous voulions un rendu lumineux et légèrement flou, mais sans que le texte devienne trop agressif pour les yeux. Après plusieurs essais, nous avons affiné les ombres et les filtres CSS pour obtenir un effet lumineux subtil mais visible.

L'autre difficulté majeure concernait le positionnement des éléments. Le texte principal et le bouton d'exploration devaient être parfaitement centrés, mais selon la taille de l'écran, leur disposition pouvait être légèrement décalée. Grâce à Flexbox, nous avons pu nous assurer que la mise en page restait bien équilibrée et responsive sur différents écrans.

Pour résoudre ces problèmes, nous avons adopté une approche progressive :

- Recherche et visionnage de tutoriels pour comprendre les meilleures pratiques d'alignement et de gestion des effets CSS.

-Utilisation de l'IA pour obtenir des suggestions techniques lorsque nous étions bloqués sur certains détails.

Grâce à ces solutions, nous avons pu affiner le rendu visuel et garantir une présentation plus fluide et immersive.

#### 2.9.4 Fonctionnalités et Optimisations en Cours

À ce stade, la page d'accueil est fonctionnelle et bien intégrée dans l'univers du jeu. Le menu est opérationnel, les effets visuels sont en place, et le site offre une première impression immersive qui correspond aux attentes du projet. Toutefois, plusieurs éléments restent à optimiser et des ajouts sont encore à prévoir.

La structure du site doit être enrichie avec la création des autres pages (Équipe, Télécharger, Ressources), qui suivront le même style graphique afin de garantir une continuité visuelle cohérente.

Enfin, nous réfléchissons à l'intégration d'effets interactifs supplémentaires, comme des animations au défilement ou des transitions plus fluides, qui pourraient renforcer encore davantage la qualité du site. Cependant, la priorité reste le développement du jeu vidéo.

### 2.10 Jaquette

En parallèle du développement du jeu, nous avons tenté de concevoir une jaquette qui représenterait fidèlement l'univers sombre et angoissant de Lurking Home. L'objectif était de créer un visuel impactant, capable d'attirer l'attention des joueurs potentiels tout en restant cohérent avec l'ambiance du jeu. Cependant, ce travail s'est révélé plus compliqué que prévu.

Nous nous sommes rapidement heurtés à plusieurs difficultés, notamment le choix des couleurs, la composition visuelle et l'intégration des différents éléments graphiques. L'un des principaux problèmes était d'obtenir un rendu professionnel et attrayant sans que l'image ne paraisse surchargée ou mal équilibrée. Nous avons expérimenté plusieurs designs et testé différentes approches, mais malgré nos efforts, le résultat final ne nous a pas totalement satisfaits.

Nous avons également rencontré des limitations techniques dans l'utilisation des logiciels de graphisme, ce qui nous a empêchés de peaufiner certains détails essentiels. Même si la jaquette n'a pas été réalisée à 100 %, cette tentative nous a permis de mieux comprendre les enjeux du design graphique et d'identifier les compétences nécessaires pour réussir ce type de projet. À l'avenir, il pourrait être judicieux de faire appel à un graphiste spécialisé pour garantir un rendu de qualité professionnelle.

## 3 Conclusion

### 3.1 Rappels sur les avancées

Les avancées sur notre projet sont, pour l'instant, les suivantes : l'implémentation d'un système d'inventaire ainsi que d'une barre d'objets, avec la possibilité de récupérer des objets et de les stocker dans l'inventaire. Nous avons également intégré un modèle de personnage avec des animations pour la marche, ainsi qu'une animation pour récupérer des objets. Toutes ces avancées ont été synchronisées en multijoueur.

Nous avons aussi progressé sur l'IA avec l'implémentation de nouveaux états pour celle-ci. Nous avons ensuite créé une scène qui nous permettra de gérer notre première mission. Enfin, nous avons amélioré notre site web et travaillé sur la jaquette de notre jeu.

## **3.2 Les choses à faire pour la prochaine fois**

### **3.2.1 Gameplay**

Pour la prochaine soutenance, il nous faudrait implémenter tous les nouveaux items, créer un système pour déposer les items, ainsi que finaliser toutes les interactions entre les joueurs et les items, mais aussi entre l'IA et les items. Par exemple, pouvoir allumer des bougies.

Il nous faudra aussi travailler sur le menu afin de pouvoir héberger une partie et rejoindre celle d'un autre joueur.

### **3.2.2 Multijoueur**

Pour la prochaine soutenance, il nous faudra synchroniser tous nos avancements, par exemple l'IA, mais aussi veiller au bon fonctionnement du P2P à plusieurs joueurs.

### **3.2.3 IA**

Pour la prochaine soutenance, l'objectif sera de terminer l'implémentation de tous les états, d'implémenter les différentes transitions, et de faire fonctionner les différents programmes avec le multijoueur.

### **3.2.4 Mission**

Avec cette première base mise en place, la prochaine étape sera de finaliser l'intégration des mécaniques interactives dans cette bibliothèque et de commencer à rendre la mission L'Énigme du Livre Interdit réellement jouable. Pour cela, les actions suivantes sont prévues :

- Intégrer des interactions de base, comme la possibilité pour le joueur de cliquer sur certains livres pour afficher un indice ou déclencher une action.

- Travailler sur la synchronisation avec l'IA, en coordonnant notre travail pour que l'entité hostile puisse réagir aux actions du joueur dans cet environnement.

- Apprendre et tester les triggers et événements dans Unity, notamment pour gérer des actions dynamiques comme des objets qui tombent des étagères pour distraire ou effrayer le joueur.

- Peaufiner les effets visuels et sonores, en intégrant des bruits d'ambiance et des variations lumineuses pour accentuer l'atmosphère d'angoisse.

### **3.2.5 Site Web**

La prochaine phase du développement consistera à compléter le site en ajoutant les pages associées au menu de navigation. Chaque page devra conserver l'identité graphique établie, tout en proposant un contenu interactif. La page Équipe servira à présenter les membres du projet, tandis que la section Télécharger offrira un accès aux fichiers du jeu lorsqu'ils seront disponibles. L'objectif final est d'obtenir une plateforme aboutie et

immersive, qui reflète pleinement l'ambiance du jeu Lurking Home et offre aux visiteurs une expérience captivante dès leur arrivée sur le site. Avec cette première version de la page d'accueil, une base solide a été posée pour la suite du développement. L'interface actuelle correspond aux attentes visuelles et immersives du projet, et les ajustements techniques réalisés ont permis d'affiner le rendu global.