

# Rapport 3 - Shattered Mind

DEVONKHAIN

Noémie DA CUNHA LOBO, Chloé BAUER, Séléna CONSTANS

15 janvier 2025

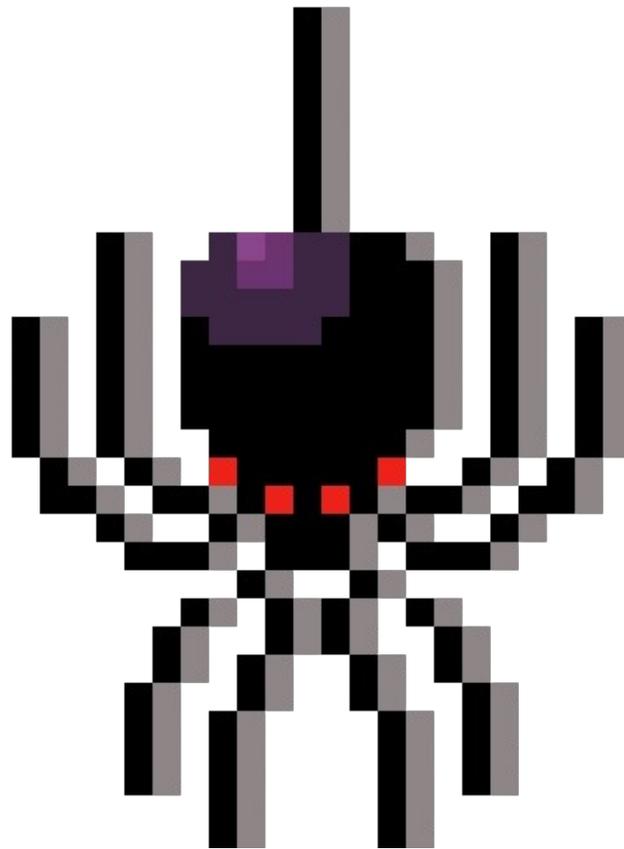


Figure 1 – Logo du jeu

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Le jeu "Shattered Mind" . . . . .	4
1.2	L'équipe DEVONKHAIN . . . . .	4
1.3	Ressentis des membres de l'équipe . . . . .	6
1.3.1	Noémie DA CUHNA LOBO (directrice technique) . . .	6
1.3.2	Chloé BAUER (cheffe de projet et Direction Artistique)	6
1.3.3	Séléna CONSTANS (Responsable de la Narration et Énigmes) . . . . .	7
<b>2</b>	<b>Organisation et gestion du projet</b>	<b>8</b>
2.1	Répartition des rôles . . . . .	8
2.2	Méthodologie de travail . . . . .	9
<b>3</b>	<b>Rendu final du projet "Shattered Mind"</b>	<b>10</b>
3.1	Expérience de jeu . . . . .	10
3.2	Environnement et niveaux . . . . .	10
3.2.1	Le Grenier . . . . .	10
3.2.2	Le Premier Étage . . . . .	11
3.2.3	Le Rez-de-chaussée . . . . .	13
3.3	Énigmes et mécanismes . . . . .	18
3.4	Éléments techniques et artistiques . . . . .	24
3.4.1	Site web . . . . .	24
3.4.2	Menu principal . . . . .	25
3.4.3	Implémentation du Multijoueur . . . . .	27
3.4.4	Image de chargement . . . . .	28
3.4.5	Animations des personnages et ennemi . . . . .	28
3.4.6	Cinématiques . . . . .	29
<b>4</b>	<b>Bilan technique détaillé</b>	<b>30</b>
4.1	Menu principal . . . . .	30
4.2	Gestion de la scène et du personnage . . . . .	31
4.3	Architecture technique et implémentation . . . . .	31
4.3.1	Network Manager . . . . .	31
4.3.2	Game Manager . . . . .	32
4.3.3	Gestion de la session . . . . .	33
4.3.4	Personnages et Ennemi . . . . .	34

4.3.5	Intelligence artificielle . . . . .	36
4.4	Gestion des erreurs et solutions apportées . . . . .	37
4.4.1	Réalisation des cinématiques . . . . .	37
<b>5</b>	<b>Chronologie des avancées du groupe</b>	<b>38</b>
5.1	Novembre . . . . .	38
5.2	Décembre . . . . .	40
5.3	Janvier . . . . .	40
5.4	Mars . . . . .	41
5.5	Avril — Construction narrative et premières cinématiques . . . . .	41
5.6	Mai — Derniers ajustements et assemblage du jeu . . . . .	41
<b>6</b>	<b>Remerciements</b>	<b>42</b>
<b>7</b>	<b>Conclusion</b>	<b>42</b>
<b>8</b>	<b>Annexes</b>	<b>44</b>
8.1	Galerie d’animations . . . . .	46
8.2	Réseaux Sociaux . . . . .	49
8.3	Diagramme de Gantt . . . . .	49
8.4	Références et crédits . . . . .	50

# 1 Introduction

## 1.1 Le jeu "Shattered Mind"

Le jeu que nous proposons se nomme Shattered Mind, il sera un mélange d'horreur et d'énigmes avec une progression en étages dans lequel deux protagonistes doivent tout faire pour sortir d'un endroit hanté et ainsi terminer le jeu.

Nos parcours respectifs ont de nombreux points communs comme la passion du jeu, une soif de connaissance quant à la programmation et la création d'un jeu qui plaira au maximum de personnes. Cependant, nos parcours ont aussi chacun leurs propres différences ce qui ajoute grandement à la créativité au sein de notre entreprise et nos différences d'expérience nous permettent de trouver des solutions plus facilement pour permettre au projet de progresser au mieux.

## 1.2 L'équipe DEVONKHAIN

Nos parcours individuels ont de nombreux points communs comme la passion du jeu, une soif de connaissance quant à la programmation et la création d'un jeu qui plaira au maximum de personnes. Cependant, nos parcours ont aussi chacun leurs propres différences ce qui ajoute grandement à la créativité au sein de notre entreprise et nos différences d'expérience nous permettent de trouver des solutions plus facilement pour permettre au projet de progresser au mieux.

Le logo initial de Shattered Mind représentait une paire de crocs de loup, symbole de danger et de bestialité. Cependant, en avançant dans le développement, nous avons ressenti le besoin de mieux aligner notre identité visuelle avec la direction artistique du jeu.

Nous avons donc décidé de le remplacer par une araignée en pixel art. Ce choix s'inscrit davantage dans notre palette de couleurs — noir, violet et rouge — évoquant le mystique, la peur, et le sang, des thèmes centraux du jeu. L'araignée, en plus d'être un symbole fort de tension et de fragilité, est un insecte que nous apprécions toutes les trois. Ce nouveau logo représente ainsi à la fois l'univers du jeu et notre vision commune.

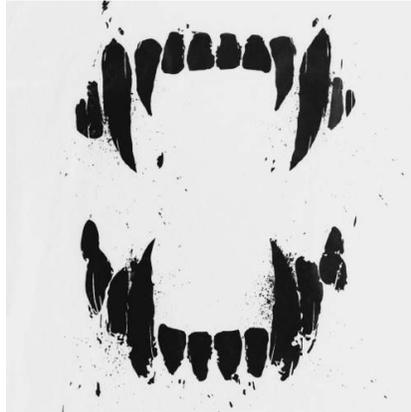


FIGURE 1 – Ancien logo de Shattered Mind

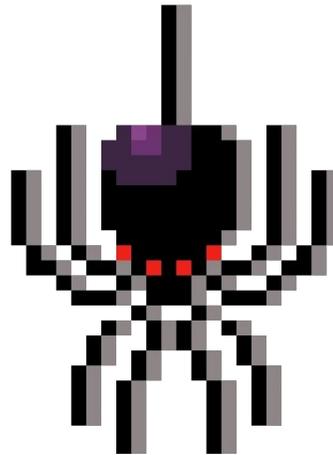


FIGURE 2 – Nouveau logo de Shattered Mind

## **1.3 Ressentis des membres de l'équipe**

### **1.3.1 Noémie DA CUHNA LOBO (directrice technique)**

Ce projet a été à la fois une épreuve et une belle aventure pour moi. Dès le départ, j'étais stressée face à l'ampleur des tâches à accomplir et à la pression que je me mettais pour bien faire. J'ai dû apprendre à gérer ce stress, à rester patiente, surtout lorsque les choses ne fonctionnaient pas comme prévu ou prenaient plus de temps que je l'espérais.

Certaines étapes m'ont vraiment poussée dans mes retranchements. Il a fallu que je sorte de ma zone de confort, que je persévère même quand j'avais l'impression de ne pas avancer. Ce n'était pas toujours facile, mais chaque difficulté surmontée m'a rendue un peu plus confiante.

Travailler en petite équipe a aussi été une expérience enrichissante. J'ai appris à mieux communiquer, à écouter les idées des autres, à partager les miennes, et à faire confiance. Ce cadre plus intime a renforcé la cohésion, mais demandait aussi une implication constante de chacun.

Malgré les moments de doute, j'ai ressenti un vrai bonheur à chaque fois que j'arrivais à résoudre un problème ou à finaliser une partie du projet. Ces petits succès m'ont donné l'énergie de continuer et m'ont montré de quoi j'étais capable. Aujourd'hui, je ressors de cette expérience fière de ce que nous avons accompli ensemble et de ce que j'ai appris sur moi-même.

### **1.3.2 Chloé BAUER (cheffe de projet et Direction Artistique)**

Travailler sur *Shattered Mind* a été une expérience extrêmement riche, à la fois humaine et créative. J'ai ressenti une vraie joie à chaque moment où tout fonctionnait, où mes designs s'intégraient bien dans le jeu, et que l'équipe validait les éléments graphiques que j'avais imaginés. Voir mes dessins prendre vie dans Unity a été une immense source de motivation.

Ce projet m'a aussi beaucoup appris sur l'importance de l'esprit d'équipe. On a souvent dû se soutenir mutuellement, et il y a eu des moments plus durs, notamment quand une partie du travail se faisait à seulement deux personnes. Ces phases m'ont parfois mise sous pression, mais elles ont aussi renforcé ma volonté d'aller au bout.

Il y a également eu beaucoup de stress lié au temps : certaines choses, notamment dans Unity, prenaient bien plus de temps que ce que j'avais imaginé. La création et l'intégration des énigmes, par exemple, m'ont demandé énormément d'efforts et de patience. Ce décalage entre les attentes et la

réalité technique a parfois été frustrant, mais il m'a aussi permis de mieux comprendre la complexité du développement de jeu.

Avec du recul, je suis fière de ce qu'on a accompli ensemble. Ce projet m'a permis de gagner en autonomie, en rigueur, et surtout en confiance dans ma capacité à participer à la création d'un vrai jeu vidéo, du début à la fin.

### **1.3.3 Séléna CONSTANS (Responsable de la Narration et Énigmes)**

Pour moi, ce projet a été très intense émotionnellement. J'ai souvent ressenti de la peur et de l'inquiétude — peur de ne pas être à la hauteur, crainte que le jeu ne prenne pas la forme qu'on imaginait. Il y a eu des nuits très courtes, beaucoup de stress, et une pression constante à bien faire, à respecter les délais et à être en phase avec les attentes du groupe.

Malgré tout cela, j'ai aussi connu des moments de joie sincère. Notamment lorsque mes idées narratives prenaient forme, que le lore que j'avais imaginé se reflétait dans les détails du jeu, ou encore quand une énigme ou un dialogue fonctionnait parfaitement.

Ce projet m'a obligée à me dépasser, à croire en mes idées même quand le doute prenait le dessus. Et malgré les difficultés, je garde en tête que ce qu'on a créé ensemble a du sens, et ça, c'est une vraie satisfaction.

## 2 Organisation et gestion du projet

### 2.1 Répartition des rôles

Noémie DA CUNHA LOBO fait partie de l'équipe de développement. Elle a pris en charge l'implémentation des éléments techniques du jeu, notamment les fonctionnalités multijoueurs et l'intelligence artificielle (IA). Elle s'est également occupée de l'intégration des animations et du bon fonctionnement des interactions dans Unity, tout en assurant la maintenabilité du code.

Chloé BAUER occupe désormais le rôle de cheffe d'équipe. Elle est responsable de la coordination globale du projet, en plus de prendre en charge la direction artistique. Elle conçoit la majorité des éléments visuels du jeu, s'assure de leur intégration cohérente dans l'environnement numérique, et veille à l'harmonie visuelle entre tous les assets. Elle a également participé à la conception de plusieurs énigmes, en soutien à Séléna.

Tout design non accompagné par des crédits a été fait par Chloé BAUER. Sé-

léna CONSTANS s'est principalement consacrée à la conception des énigmes du jeu. Elle a également posé les bases du lore, servant de fil conducteur narratif à l'ensemble du projet. Ses idées ont permis de structurer les mécaniques de jeu autour de thématiques cohérentes, en collaboration avec Chloé.

## 2.2 Méthodologie de travail

Dans le cadre de ce projet, nous avons choisi de nous inspirer de la méthode Agile, et plus spécifiquement de celle de Kanban, qui correspondait le mieux aux habitudes et méthodes de travail de chacun.

Bien que nous ne suivions pas l'intégralité des recommandations proposées, nous avons fait le choix de conserver de nombreuses similitudes avec cette approche.

- Chaque membre de l'équipe dispose d'une liste de tâches à réaliser, sans être contraint par des périodes fixes comme dans un cadre de planification en "sprints" (tel qu'en Scrum). Dès qu'une tâche est terminée, une autre peut être commencée. Toutefois, pour éviter toute surcharge de travail, les tâches sont priorisées. Nous essayons de ne pas entreprendre plus de 2 à 3 tâches simultanément, en fonction de la charge de travail qu'elles impliquent. Cette priorisation permet de maintenir un équilibre dans la répartition des tâches et d'éviter une trop grande accumulation de travail pour un seul membre à un moment donné.
- Nous avons mis en place un système de gestion des dépendances entre les tâches. Lorsqu'une tâche dépend de l'achèvement d'une autre, les membres concernés se coordonnent afin de garantir que les tâches soient terminées dans les délais et que l'avancement de chaque membre ne soit pas retardé. Cette gestion anticipée des dépendances assure une fluidité dans l'exécution du projet et permet de respecter les échéances.
- Nos réunions sont organisées de manière flexible, en fonction de l'avancement des travaux et de la charge de travail de chacun. À l'approche des soutenances, nous augmentons la fréquence des réunions pour nous préparer au mieux. Celles-ci ont pour but de définir les priorités, en tenant compte des tâches en cours et à venir, des tâches dépendantes et de l'avancement global du projet. Nous nous adaptons également à l'évolution du projet, ce qui signifie que des réunions peuvent être demandées à tout moment par n'importe quel membre du groupe si nécessaire.



### 3.2.2 Le Premier Étage

Organisé autour d'un couloir central étroit et faiblement éclairé, ce niveau accueille plusieurs pièces majeures :

**La Salle de Bain** Une pièce vieillissante, marquée par la moisissure et l'humidité. Elle contient une énigme centrée sur la manipulation de tuyaux et le niveau de l'eau, demandant précision et observation.



FIGURE 4 – Vue de la Salle de Bain

**La Chambre Parentale** Spacieuse mais oppressive, elle dégage une impression de solitude figée. Les meubles imposants et l'ambiance figée y renforcent la tension. Des éléments de narration y sont dissimulés.



FIGURE 5 – Vue de la Chambre Parentale

**La Chambre d'Enfant** Dérangante par son apparente innocence, elle mêle mobilier coloré fané et ambiance lourde. Des objets clés liés aux énigmes ou au passé des habitants peuvent y être découverts.



FIGURE 6 – Vue de la Chambre d’Enfant

### 3.2.3 Le Rez-de-chaussée

Introduit plus tard dans le développement, ce niveau ajoute trois nouvelles pièces explorables, chacune avec ses éléments de gameplay :

**La Cuisine** Sombre et désordonnée, elle évoque une époque révolue. Les éléments rouillés, les placards scellés et les détails visuels renforcent le malaise. Des énigmes peuvent s’y dissimuler entre les ustensiles et les notes abandonnées.



FIGURE 7 – Vue de la Cuisine

**Le Salon** Vaste et silencieux, il donne l'impression d'avoir été figé au moment d'un drame. La lumière filtrée, les portraits anciens et les canapés délavés contribuent à l'ambiance pesante. Le joueur est invité à y fouiller et à observer avec attention.



FIGURE 8 – Vue du Salon

**L'Entrée** Dernière pièce accessible du jeu, elle représente un point culminant de l'histoire. C'est ici que les deux personnages se rejoignent, et que la vérité sur le manoir se dévoile. L'architecture y est plus solennelle, presque cérémoniale, renforçant la gravité du moment.



FIGURE 9 – Vue de l'Entrée

**Le Couloir du 1er étage** Ce couloir relie toutes les pièces de l'étage et contribue à l'atmosphère oppressante. Son étroitesse, son côté très délabré participe à l'immersion et à la tension ressentie par les joueurs. Donc il y a 2 escaliers : celui de gauche permet de monter au grenier et celui de droite descendre au rez-de-chaussée. Ensuite il y a 3 portes, celle du côté droit mène à la salle de bain, celle en bas à la chambre parentale et la dernière à gauche chambre d'enfant.

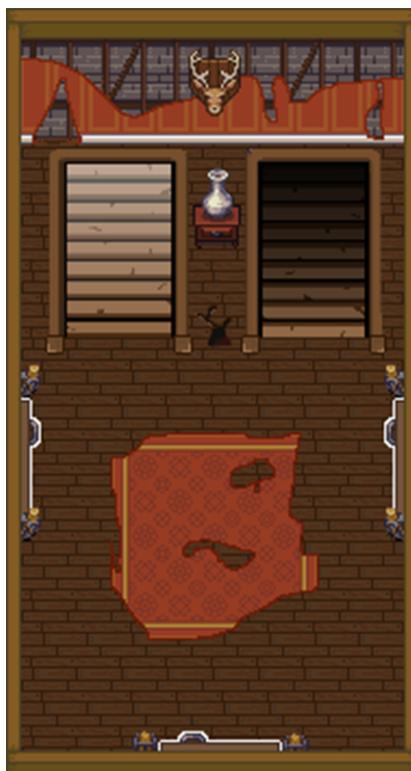


FIGURE 10 – Vue du Couloir du 1er étage

**Le Couloir du rez-de-chaussée** Il relie les différentes pièces et guide la progression des joueurs. Moins étroit que celui de l'étage mais tout aussi inquiétant, il est ponctué de détails narratifs et d'éléments d'ambiance renforçant l'idée d'un lieu ancien, vivant, mais corrompu.

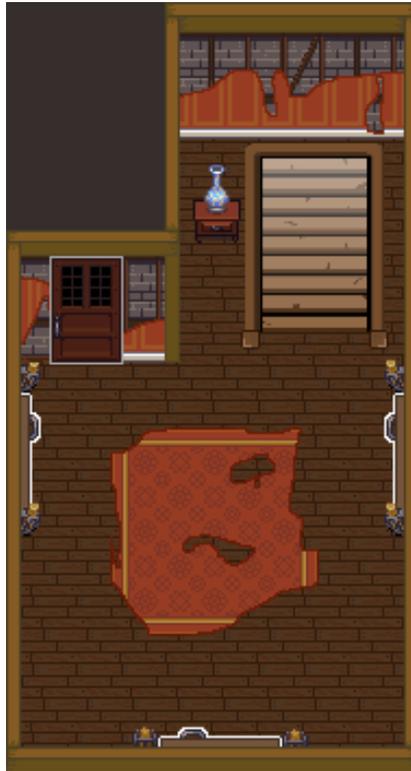


FIGURE 11 – Vue du Couloir du Rez-de-chaussée

### 3.3 Énigmes et mécanismes

Les énigmes sont au centre de Shattered Mind. Elles structurent la progression des joueurs, en sollicitant leur observation, leur logique, et surtout leur capacité à coopérer. Chacune d'elles repose sur une mécanique différente, conçue pour varier les rythmes de jeu et renforcer l'immersion dans l'univers.

**Horloge à régler** Les joueurs doivent ajuster les aiguilles d'une horloge à une heure précise. L'indication de l'heure correcte est disséminée dans une autre partie de la maison, ce qui oblige les joueurs à explorer séparément ou à se relayer pour tester. L'énigme se valide uniquement si l'heure exacte est maintenue pendant un court moment.

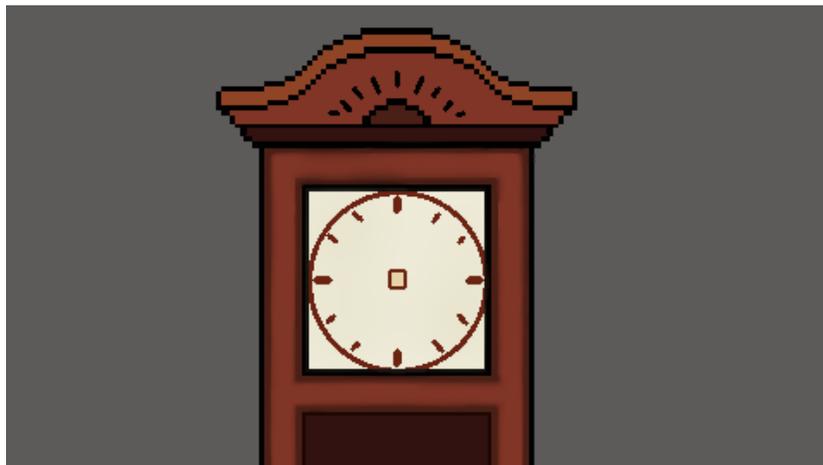


FIGURE 12 – énigme de l'Horloge

**Jeu d'échecs** L'objectif est de replacer correctement certaines pièces manquantes sur un échiquier. Les indices concernant la disposition correcte des pièces sont visibles ailleurs dans le jeu. Les joueurs doivent recouper les informations et reconstituer la bonne configuration à l'aide des éléments trouvés. La réussite dépend de leur capacité à observer, mémoriser et faire des liens.

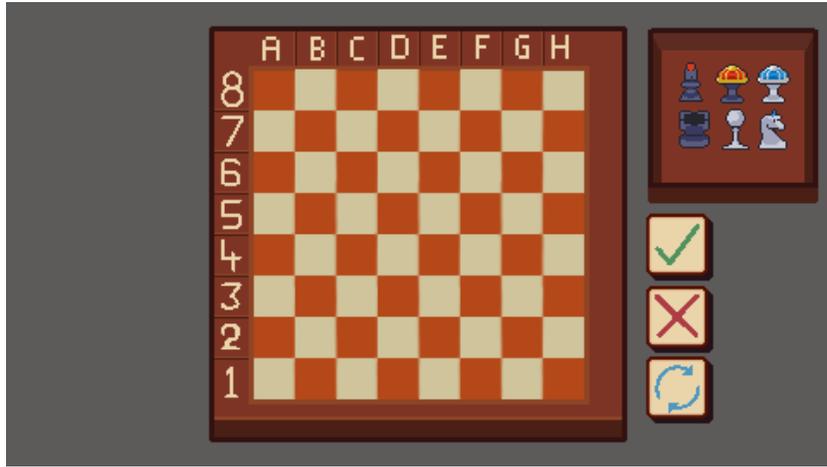


FIGURE 13 – énigme du Jeu d'échecs

**Tuyaux de salle de bain** Dans cette énigme, les joueurs doivent manipuler plusieurs tuyaux afin d'atteindre un niveau d'eau précis. En ajustant leur hauteur, l'eau monte ou descend, et il faut atteindre une valeur cible. La difficulté réside dans le maintien du bon niveau pendant quelques secondes, ce qui demande précision et coordination entre les joueurs.

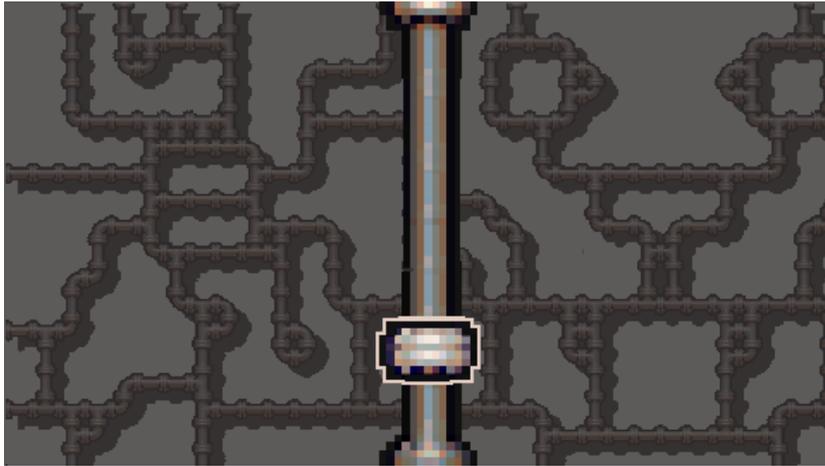


FIGURE 14 – énigme des tuyaux

```

1  using UnityEngine;
2  using UnityEngine.UI;
3  using TMPro;
4  using System.Collections;
5
6  public class BathroomPuzzle : MonoBehaviour
7  {
8      public Slider tempSlider;
9      public TextMeshProUGUI tempText;
10     public GameObject hiddenSymbol; // Image cachée
11
12     private float correctTemperature = 42.0f; // Température correcte
13     private float tolerance = 1.0f; // Marge d'erreur
14     private Coroutine revealCoroutine;
15
16     void Start()
17     {
18         hiddenSymbol.SetActive(false); // Cache le symbole au début
19     }
20
21     void Update()
22     {
23         tempText.text = "Température : " + tempSlider.value.ToString("F1") + "°C";
24         CheckTemperature();
25     }
26
27     void CheckTemperature()
28     {
29         if (Mathf.Abs(tempSlider.value - correctTemperature) < tolerance)
30         {
31             if (revealCoroutine == null) revealCoroutine = StartCoroutine(RevealSymbol());
32         }
33         else
34         {
35             if (revealCoroutine != null)
36             {
37                 StopCoroutine(revealCoroutine);
38                 revealCoroutine = null;
39                 hiddenSymbol.SetActive(false); // Cache le symbole si la température change
40             }
41         }
42     }
43
44     IEnumerator RevealSymbol()
45     {
46         yield return new WaitForSeconds(2f); // Attendre 2 secondes
47         hiddenSymbol.SetActive(true); // Afficher le symbole
48     }
49 }
50

```

FIGURE 15 – Code de l'énigme des tuyaux

**Papiers dispersés et le livre vert** Plusieurs papiers sont dissimulés dans les différentes pièces du manoir. Chaque papier contient une partie d'un indice. En les réunissant, les joueurs découvrent qu'ils doivent chercher un livre vert, dont la présence est suggérée mais non indiquée explicitement. Cette énigme repose sur la collecte, la lecture et l'interprétation progressive d'éléments éparpillés, et fait appel à l'exploration et à l'attention aux détails.



FIGURE 16 – Énigme du Livre Maudit

### **Codelock (avec image panel)**

Cette énigme repose sur la saisie d'un code numérique via un clavier interactif. L'un des joueurs doit entrer un code à quatre chiffres, pendant que l'autre observe ailleurs dans le manoir des chiffres qui clignotent brièvement à l'écran. Ces indices partiels nécessitent une bonne communication entre les deux pour identifier le bon code. Un système de vérification limite les essais et impose de recommencer en cas d'erreur, ce qui ajoute un peu de pression.



FIGURE 17 – Énigme du Codelock / Panel

```

public class Keypad : MonoBehaviour
{
    public TMP_InputField charHolder;
    public GameObject button1;
    public GameObject button2;
    public GameObject button3;
    public GameObject button4;
    public GameObject button5;
    public GameObject button6;
    public GameObject button7;
    public GameObject button8;
    public GameObject button9;
    public GameObject button0;
    public GameObject buttonclear;
    public GameObject buttonenter;
    public string password = "5679";
    public void B1(){
        charHolder.text = charHolder.text +"1";
    }
    public void B2(){
        charHolder.text = charHolder.text +"2";
    }
    public void B3(){
        charHolder.text = charHolder.text +"3";
    }
    public void B4(){
        charHolder.text = charHolder.text +"4";
    }
    public void B5(){
        charHolder.text = charHolder.text +"5";
    }
    public void B6(){
        charHolder.text = charHolder.text +"6";
    }
    public void B7(){
        charHolder.text = charHolder.text +"7";
    }
    public void B8(){
        charHolder.text = charHolder.text +"8";
    }
    public void B9(){
        charHolder.text = charHolder.text +"9";
    }
    public void B0(){
        charHolder.text = charHolder.text +"0";
    }
    public void ClearEvent(){
        charHolder.text = null;
    }
    public void enterEvent(){
        if (charHolder.text == password){
            charHolder.text = "Yes";
        }
        else {
            charHolder.text = "No";
        }
    }
}

```

FIGURE 18 – Code pour l'énigme du cadenas / Panel

## 3.4 Éléments techniques et artistiques

### 3.4.1 Site web

Dans le cadre de notre projet, nous avons développé un site internet accessible à l'adresse <https://shatteredmind.fr>. Ce site a pour objectif de centraliser toutes les informations essentielles autour du jeu Shattered Mind, de valoriser notre travail d'équipe, et de proposer une première immersion dans l'univers du jeu.

Le site s'ouvre sur une phrase d'accroche immersive, destinée à plonger immédiatement le visiteur dans l'ambiance inquiétante du manoir. Il présente ensuite un aperçu général du jeu : Shattered Mind est un jeu d'horreur en 2D, en pixel art, combinant exploration, résolution d'énigmes et coopération entre deux personnages — Yorick et Léon — piégés dans un lieu mystérieux et menaçant.

Nous y mettons en avant les éléments distinctifs de notre jeu : une direction artistique forte, un style graphique en pixel art, une atmosphère pesante, une narration fragmentée, et une expérience de jeu coopérative construite autour de la communication et de la réflexion.

Le site propose également une présentation de l'équipe de développement :

- **Chloé BAUER** — cheffe de projet et directrice artistique,
- **Noémie DA CUNHA LOBO** — développeuse principale et responsable technique,
- **Séléna CONSTANS** — conceptrice d'énigmes et initiatrice du lore.

Chaque membre y est brièvement présenté, soulignant son rôle dans le développement du jeu et ses compétences mises en œuvre tout au long du projet.

Enfin, le site contient des liens vers nos réseaux sociaux (Discord, Instagram et LinkedIn), permettant au public de suivre l'évolution du jeu, de découvrir des visuels exclusifs et, à terme, de rejoindre une communauté de joueurs. Nous avons conçu ce site non seulement comme un outil de présentation, mais aussi comme une vitrine professionnelle de notre travail collectif.

Dans une volonté de renforcer l'identité du projet et de présenter l'équipe de manière originale, nous avons également créé des personas pour représenter chaque membre du groupe.



FIGURE 19 – Page de présentation des personas sur le site web

### 3.4.2 Menu principal

Le menu principal adopte une esthétique sombre, avec des boutons rouges, des ombres et un léger assombrissement de l'arrière-plan pour renforcer l'atmosphère inquiétante du jeu. En arrière-plan, on aperçoit le protagoniste dans une posture marquée par la peur, éclairé par une lumière rouge qui accentue cette tension. Cette image fait écho à un moment clé du jeu, suggérant une situation délicate et pleine de suspense, sans en dévoiler les détails. Les options "Jouer", "Options", "Quitter" et "Crédits" sont clairement disposées, et le menu intègre la gestion du multijoueur (hôte/client), simplifiant ainsi l'accès aux parties coopératives tout en conservant une cohérence visuelle et narrative. Ce dernier n'est pas notre 1er menu car avant lui nous avons plus opté pour un style complètement en pixel art en mettant l'accent sur la silhouette sur le pull du personnage principal, yorick, qui représente toujours le père, donc l'antagoniste.



FIGURE 20 – Menu principal du jeu



FIGURE 21 – Ancien menu principal

### 3.4.3 Implémentation du Multijoueur

Le système multijoueur de Shattered Mind est pleinement opérationnel et repose sur la bibliothèque Mirror. Il utilise un modèle client-serveur où le serveur (qui peut être hébergé par un joueur ou sur une machine dédiée) gère l'état global du jeu. Lorsqu'un joueur se connecte, le serveur lui transmet les informations nécessaires pour afficher la scène et interagir avec les autres. Chaque joueur contrôle un ou plusieurs objets (comme un personnage), et ses actions sont validées et répercutées par le serveur à tous les autres clients.

Pour optimiser les performances et minimiser la consommation de bande passante, le projet intègre des systèmes de **\*\*gestion d'intérêt (Interest Management)\*\***. Ces systèmes déterminent précisément quels objets doivent être visibles et synchronisés pour chaque joueur. Par exemple, un joueur ne reçoit que les informations sur les éléments proches de lui ou appartenant à la même équipe/partie, ce qui réduit considérablement les données échangées et améliore la fluidité du jeu.

La gestion des scènes et des objets réseau est automatisée. Que ce soit lors de l'arrivée ou du départ d'un joueur, ou de la création/destruction d'un objet, Mirror assure la synchronisation de ces changements auprès des clients concernés. Des scripts spécifiques permettent d'adapter le comportement du serveur pour divers modes de jeu ou configurations réseau, rendant le système multijoueur flexible et adaptable.

**Déplacement entre les cartes** Le déplacement entre les différentes cartes du jeu s'appuie sur le système de **\*\*scènes additives de Unity\*\***, combiné à la gestion réseau de Mirror. Chaque zone ou "map" du jeu correspond à une scène Unity distincte. Quand un joueur atteint une zone de transition (comme une porte), la nouvelle scène se charge de manière additive, permettant une transition fluide sans écran de chargement brutal et affichant simultanément plusieurs cartes si nécessaire.

Sur le plan réseau, Mirror synchronise le chargement et le déchargement de ces scènes entre les joueurs. Le serveur informe uniquement les clients concernés de charger la scène correspondante, assurant que chaque joueur voit et interagit seulement avec les objets et joueurs présents dans sa propre carte ou celles voisines. Cela optimise les performances en évitant d'envoyer des informations inutiles à l'ensemble des joueurs connectés au serveur.

La gestion des objets et des joueurs lors d'un changement de carte est également automatisée : les objets réseau sont déplacés d'une scène à l'autre ou

recr es si n cessaire, et leur  tat est synchronis . Ce m canisme garantit une exp rience de jeu coh rente pour tous les participants, m me s'ils  voluent dans des cartes diff rentes, et permet de concevoir des mondes connect s avec une architecture r seau efficace et  volutive.

#### 3.4.4 Image de chargement

Une image de chargement anim e, montrant les deux personnages courant c te   c te, symbolise la coop ration dans le jeu.



FIGURE 22 – gif de chargement

#### 3.4.5 Animations des personnages et ennemi

Les animations de marche de Yorick et L on sont compos es de 6 frames, un choix artistique destin    offrir une fluidit  visuelle coh rente avec le style du jeu. De m me, l'ennemi dispose d'une animation de d placement con ue pour s'int grer harmonieusement dans l'esth tique g n rale et renforcer l'atmosph re

### 3.4.6 Cinématiques

La première cinématique, qui montre Yorick et Léon discutant devant l'entrée du manoir, a été intégrée au jeu. Cette scène introduit l'histoire et instaure une ambiance de suspense. Par ailleurs, la cinématique de l'antagoniste sert à renforcer la tension narrative en laissant un certain doute et des questionnements sur ce qu'il advient de la fin du jeu.



FIGURE 23 – Cinématique de début



FIGURE 24 – Cinématique de fin

## 4 Bilan technique détaillé

### 4.1 Menu principal

L'intégralité du système multijoueur du jeu est gérée depuis le menu principal. Des objets essentiels comme le `NetworkManager` sont configurés dans la première scène et transférés d'une scène à l'autre. Le `NetworkManager` nécessite les scripts "`NetworkManager`" et "`Unity Transport`". Le `GameManager` est responsable de l'attribution des préfabs aux joueurs et de leur apparition dans la scène suivante, nécessitant le script `SC_GameManager` et le composant "`Network Object`". La gestion des sessions multijoueurs (héberger/rejoindre) est centralisée dans le menu, utilisant le package "`Multiplayer Widgets`" pour l'interface utilisateur.

## 4.2 Gestion de la scène et du personnage

Pour chaque personnage, les composants "Network Object", "Network Transform" (configuré sur Owner pour la synchronisation X et Y uniquement), et "Network Rigidbody 2D" sont essentiels pour la synchronisation réseau. Un hash unique pour chaque personnage est crucial. Le script `SC_OwnerNetworkAnimator` assure que l'animation est contrôlée par le client. Pour l'ennemi, les composants "Box Collider 2D" et "Rigidbody 2D" sont nécessaires pour les interactions physiques et les mouvements.

## 4.3 Architecture technique et implémentation

### 4.3.1 Network Manager

L'intégralité du système multijoueur du jeu est gérée depuis le menu principal. Des objets essentiels, tels que le `NetworkManager`, sont configurés directement dans la scène du menu principal, qui est la première scène à être chargée. Ces objets sont ensuite transférés d'une scène à l'autre tout au long du jeu, assurant ainsi la continuité et la stabilité de l'expérience multijoueur, quel que soit l'endroit où le joueur se trouve dans le jeu.

Pour sa mise en place, il est nécessaire de créer un objet vide dans Unity (`GameObject` → `Create Empty`) et de le nommer `NetworkManager`. Ensuite, il convient de lui associer deux scripts essentiels :

- Le script "**NetworkManager**" (fourni avec les packages importés) : Il gère l'ensemble des fonctionnalités liées au réseau, comme la connexion des joueurs, la synchronisation des objets et la gestion des scènes en multijoueur.
- Le script "**Unity Transport**" : Il permet de gérer la communication réseau en utilisant le protocole de transport d'Unity, assurant une transmission stable et rapide des données entre les clients et l'hôte du jeu.

Les réglages par défaut du script "Unity Transport" ne nécessitent pas de modification. Pour le `NetworkManager`, il est essentiel de s'assurer que la topologie du réseau est définie sur "Client-Server" dans les paramètres de `Network Topology`. Il n'est pas nécessaire de définir un `Player Prefab` par défaut, car deux préfabs ont été créés, un pour chaque joueur. Ces préfabs seront assignés dynamiquement lors de la connexion des joueurs, garantissant une gestion correcte des différentes instances de joueur.

Cependant, il est nécessaire de placer les deux préfabs dans la Network Prefabs List. Cela permet au NetworkManager de reconnaître et de gérer correctement les instances des joueurs lorsqu'ils rejoignent ou hébergent une partie. En ajoutant ces préfabs à cette liste, le système peut automatiquement synchroniser les objets des joueurs à travers le réseau, garantissant ainsi leur apparition dans les scènes appropriées et leur gestion correcte en mode multijoueur.

```
public class SC_GameManager : NetworkBehaviour
{
    1 référence
    [SerializeField] private Transform Player1;
    1 référence
    [SerializeField] private Transform Player2;

    0 références
    public override void OnNetworkSpawn()
    {
        if (IsServer)
            NetworkManager.Singleton.SceneManager.OnLoadEventCompleted += SceneLoaded;
    }

    1 référence
    private void SceneLoaded(string sceneName, LoadSceneMode loadSceneMode, List<ulong> clientsCompleted, List<ulong> clientsTimeOut)
    {
        foreach (ulong clientId in NetworkManager.Singleton.ConnectedClientsIds)
        {
            Transform playerTransform = clientId == 0 ? Instantiate(Player1) : Instantiate(Player2);
            playerTransform.GetComponent<NetworkObject>().SpawnAsPlayerObject(clientId, true);
        }
    }
}
```

FIGURE 25 – Configuration du Game Manager dans Unity

### 4.3.2 Game Manager

Le GameManager est responsable de l'attribution des préfabs aux joueurs et du "spawn" (apparition) des joueurs dans la scène suivante.

Comme pour le NetworkManager, il faut créer un objet vide dans Unity (GameObject → Create Empty) et le nommer GameManager. Il est ensuite nécessaire de lui associer le script `**SC_GameManager**`, ainsi que le composant `**"Network Object"**. Ce dernier est indispensable pour que l'objet GameManager soit reconnu et synchronisé sur le réseau, permettant ainsi sa gestion et sa réplication à travers le système multijoueur.`

### 4.3.3 Gestion de la session

Pour la création et la gestion des sessions multijoueurs, nous avons décidé que l'ensemble du processus se déroulerait dans le menu principal. Ainsi, le bouton "Jouer" ne mène plus directement à la scène suivante, mais fait apparaître deux nouveaux boutons. Ces boutons permettent aux joueurs de choisir s'ils souhaitent héberger une partie en tant qu'hôte ou rejoindre une partie existante. Cette approche centralise la gestion des sessions et offre une meilleure expérience utilisateur en permettant aux joueurs de choisir facilement leur rôle dans la partie multijoueur avant de commencer.

Le package "Multiplayer Widgets" facilite l'importation directe des widgets nécessaires pour la gestion du multijoueur. Il permet notamment de créer le code permettant aux joueurs de rejoindre une partie, d'afficher un encadré où les joueurs peuvent entrer un code de session pour se connecter, ainsi que d'afficher une liste des joueurs actuellement présents dans la session. Ces widgets sont essentiels pour simplifier l'interface utilisateur et offrir une expérience de jeu multijoueur plus fluide et intuitive.

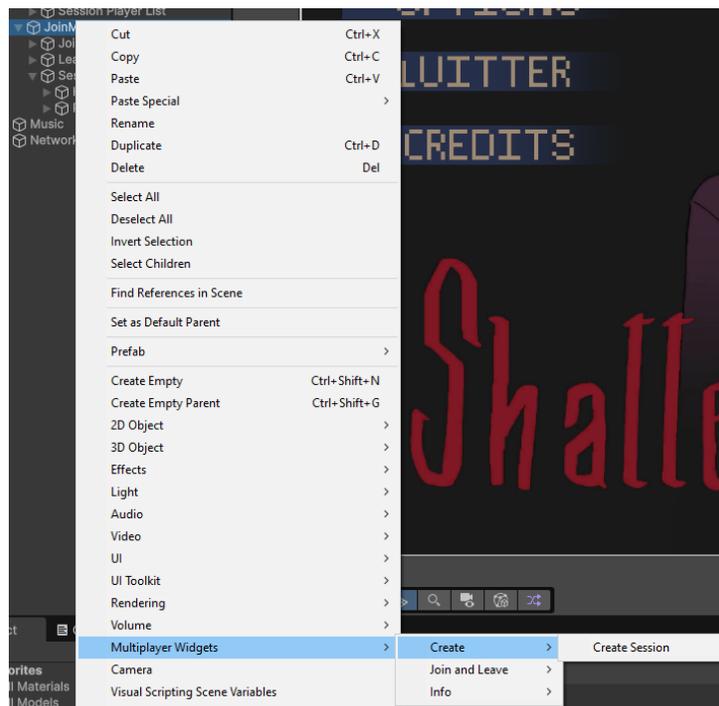


FIGURE 26 – Multiplayer Widgets



FIGURE 27 – Interface Hôte



FIGURE 28 – Interface Client

#### 4.3.4 Personnages et Ennemi

Pour chaque personnage, il est nécessaire d'ajouter les composants suivants : "Network Object", "Network Transform", et "Network Rigidbody

2D". Ces composants sont essentiels pour garantir la synchronisation correcte des objets à travers le réseau.

- **Network Object** : Permet de gérer l'état du joueur sur le réseau, garantissant que l'objet est bien synchronisé et reconnu par les autres joueurs.
- **Network Transform** : Permet de synchroniser la position, la rotation et l'échelle du personnage en temps réel entre les différentes instances du jeu.
- **Network Rigidbody 2D** : Est utilisé pour gérer la physique du personnage en 2D et s'assurer que les forces physiques appliquées sont correctement synchronisées.

Il est également crucial de vérifier que, dans le Network Object, le hash de chaque personnage soit unique. Cela garantit qu'il n'y a pas de conflits d'identification entre les instances de personnages, assurant ainsi une gestion précise des objets multijoueurs.

Pour le Network Transform, il est important de configurer le Authority Mode sur Owner, ce qui signifie que l'autorité sur la position de l'objet (dans ce cas, le personnage) sera donnée au joueur qui en est le propriétaire. Cela permet au joueur de contrôler directement la position de son personnage, tandis que les autres joueurs synchroniseront cette position à travers le réseau.

De plus, pour optimiser les performances et éviter une synchronisation inutile, seules les positions x et y doivent être synchronisées. Cela signifie que la position en z (pour les jeux en 2D) ou la rotation et l'échelle ne seront pas prises en compte pour la synchronisation réseau, réduisant ainsi la charge de synchronisation et assurant une expérience plus fluide en multijoueur.

En ajoutant le script **SC\_OwnerNetworkAnimator** à chaque personnage, celui-ci permet de gérer l'animation des personnages de manière réseau tout en utilisant le système de NetworkAnimator d'Unity. Ce script, qui hérite de NetworkAnimator, redéfinit la méthode OnIsServerAuthoritative(), en retournant false. Cela signifie que l'animation du personnage est contrôlée par le client, et non par le serveur, garantissant ainsi que l'hôte n'a pas l'autorité sur les animations des autres joueurs.

En d'autres termes, avec ce script, chaque client sera responsable de l'animation de son propre personnage, tandis que le serveur coordonne les autres aspects du gameplay. Cela permet une gestion plus fluide des animations en multijoueur, tout en évitant les conflits de synchronisation.

```

public class SC_OwnerNetworkAnimator : NetworkAnimator
{
    protected override bool OnIsServerAuthoritative()
    {
        return false;
    }
}

```

FIGURE 29 – Script OwnerNetworkAnimator

Dans les propriétés de l'ennemi, il est nécessaire d'ajouter les composants suivants :

- **Box Collider 2D** : Ce composant permet d'affecter une boîte de collision à l'ennemi, ce qui lui permet d'interagir correctement avec le monde du jeu et les autres objets (comme les obstacles ou les projectiles). La Box Collider 2D définit une zone de détection autour de l'ennemi pour gérer les collisions.
- **Rigidbody 2D** : Ce composant permet à l'ennemi de bénéficier des lois de la physique dans un environnement 2D. Le Rigidbody 2D permet à l'ennemi de se déplacer en fonction des forces physiques appliquées, comme la gravité ou d'autres forces. Il est également utilisé pour gérer la vitesse et la direction de l'ennemi lors de ses déplacements, ce qui est essentiel pour que les mouvements soient réalistes et cohérents avec l'algorithme de pathfinding.

Ces deux composants sont cruciaux pour assurer un comportement correct et une interaction avec l'environnement, notamment en matière de collision et de mouvement dans un jeu en 2D.

#### 4.3.5 Intelligence artificielle

Pour gérer l'intelligence artificielle (IA) dans le jeu, nous utilisons le système A Pathfinding. Dans la scène de jeu, il est nécessaire de créer un objet vide que nous nommerons A. À cet objet, il faut ensuite associer le composant Pathfinder.

Ce composant permet de gérer les déplacements des entités contrôlées par l'IA en utilisant l'algorithme A, qui calcule le chemin optimal pour atteindre un objectif en tenant compte des obstacles et du terrain. Cette approche

garantit des déplacements fluides et intelligents pour les personnages non-joueurs (PNJ) dans le jeu.

Une fois le composant Pathfinder associé à l'objet A\*, il faut ouvrir les paramètres du composant et cliquer sur l'option "Graphs". Ensuite, sélectionnez "Grid Graph". Cela permettra de définir la grille de navigation utilisée par l'algorithme A\* pour calculer les chemins. Le Grid Graph divise la scène en une grille de cellules, où chaque cellule représente une unité de terrain que l'IA peut parcourir. Cela permet de créer une carte de navigation efficace et de gérer les déplacements de l'IA en fonction des obstacles présents dans l'environnement du jeu.

## 4.4 Gestion des erreurs et solutions apportées

Enfin, des **soucis d'intégration visuelle** avec Unity, comme des assets qui ne s'affichaient pas correctement ou des problèmes de rendu, ont été gérés par une collaboration étroite entre Chloé, Noémie et Séléna, s'assurant de la compatibilité des formats d'image, des réglages des shaders et de l'ordre de rendu des calques dans le moteur. La refonte du menu principal, bien que principalement artistique, a nécessité des ajustements techniques pour que les nouveaux éléments graphiques et la logique des boutons s'intègrent sans accroc.

Chaque erreur rencontrée a été une opportunité d'apprentissage, renforçant notre compréhension des systèmes techniques et notre capacité à travailler en équipe pour trouver des solutions efficaces.

### 4.4.1 Réalisation des cinématiques

Pour intégrer les cinématiques dans le jeu, nous avons utilisé une méthode simple et efficace en créant une scène dédiée dans Unity. Dans cette scène, nous avons placé un Canvas contenant une image statique représentant le visuel principal de la cinématique. Cette image a été redimensionnée pour s'adapter correctement à la résolution de l'écran, afin d'assurer un affichage cohérent et sans déformation.

Nous avons ensuite ajouté un TextMeshPro pour afficher les dialogues, ainsi qu'un GameObject vide (Empty Object) servant de contrôleur. Ce dernier contient un script permettant de faire défiler les répliques : à chaque clic de souris, le texte affiché change. Le TextMeshPro a été relié au champ

public du script via l'inspecteur, pour permettre au script d'interagir avec le texte à l'écran.

Ce système permet de créer une cinématique interactive et narrative, tout en gardant un contrôle simple sur le déroulement du dialogue. Grâce à cette structure, chaque cinématique peut être mise en place rapidement en changeant simplement l'image de fond et les lignes de texte.

## 5 Chronologie des avancées du groupe

### 5.1 Novembre

En novembre, Chloé a réalisé les premiers designs de Léon et Yorick, en s'appuyant sur une palette de couleurs trouvée sur Lospec. Elle a également conçu le design de l'ennemi. Ces premières ébauches ont posé les bases visuelles de nos protagonistes.



FIGURE 30 – Évolution du design de Yorick (ancien à gauche, nouveau à droite)



FIGURE 31 – Évolution du design de Léon (ancien à gauche, nouveau à droite)



FIGURE 32 – Palette de couleurs utilisée pour les designs initiaux. Source : Lospec, par Jake Stinnett.

Cette palette de couleurs, trouvée sur Lospec et créée par Jake Stinnett, a servi de base pour définir l'ambiance visuelle initiale de nos personnages, contribuant à l'atmosphère générale du jeu dès les premières phases de conception.

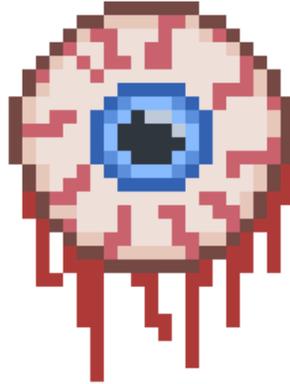


FIGURE 33 – Design de l’ennemi

Enfin, voici la conception de l’ennemi a été établie pour correspondre à l’ambiance et aux mécaniques du jeu.

## 5.2 Décembre

En décembre, le projet commence à prendre une vraie forme conceptuelle. Séléna élabore les premières ébauches du lore : elle imagine le passé du manoir, les éléments mystérieux liés à son ancien propriétaire, et les fondations d’un univers sombre et intrigant. Ces idées poseront les bases de la narration et serviront de fil conducteur à certaines énigmes et objets disséminés dans le jeu.

De son côté, Chloé crée le logo actuel du projet. Elle opte pour un style pixel art avec une araignée stylisée, symbole que l’équipe apprécie et qui s’aligne parfaitement avec l’univers visuel du jeu. Ce nouveau logo marque un tournant dans l’identité du projet, apportant une cohérence graphique plus forte et plus représentative de l’ambiance que l’on souhaite transmettre.

## 5.3 Janvier

En janvier, Noémie a développé le site internet du jeu et a créé la première pièce de démonstration pour illustrer les boîtes de collisions. Parallèlement, Chloé a conçu le design du menu principal avec ses boutons, et a également

contribué à la première pièce de démonstration. Noémie a mis en place l'animation de Yorick sur Unity, tandis que Chloé en a réalisé le dessin. Noémie a également continué le développement du site internet.

## 5.4 Mars

En mars, Chloé a dessiné la première partie de la map, incluant le grenier, le couloir, la chambre d'enfant, la chambre parentale et la salle de bain. Durant cette même période, Noémie a implémenté la fonctionnalité multijoueur et a débuté le développement de l'IA.

## 5.5 Avril — Construction narrative et premières cinématiques

Le mois d'avril marque un tournant dans l'équilibre entre les aspects techniques, artistiques et ludiques du projet.

Chloé entame la création des premières cinématiques du jeu. Ces séquences visuelles viennent renforcer le lien narratif entre les différentes parties de l'aventure et permettent d'ancrer davantage les joueurs dans l'univers du manoir. En parallèle, elle termine les dessins de la map, ce qui offre une vision complète et cohérente de l'architecture du jeu. Elle commence aussi la production graphique des premières énigmes, en concevant les éléments visuels nécessaires à leur intégration.

Pendant ce temps, Noémie poursuit le développement de l'intelligence artificielle. Elle améliore les comportements des ennemis, affine leur déplacement et renforce leur capacité à interagir avec l'environnement, ce qui intensifie la tension ressentie pendant l'exploration.

Séléna, quant à elle, finalise deux énigmes. Elle travaille leur logique, leur interaction, et les ajuste pour qu'elles soient à la fois accessibles et stimulantes pour les joueurs. Ces énigmes viennent enrichir le gameplay tout en respectant la cohérence narrative du jeu.

## 5.6 Mai — Derniers ajustements et assemblage du jeu

Mai représente le mois de l'aboutissement : chaque membre du groupe concentre ses efforts sur la finalisation du projet.

Chloé termine l'ensemble des cinématiques, qui rythment désormais le jeu de façon fluide, de l'introduction jusqu'à la conclusion. Elle met aussi

un point final à une nouvelle énigme complète, tout en finalisant les éléments graphiques manquants des puzzles déjà créés, garantissant ainsi une cohérence visuelle globale.

Noémie, de son côté, s'attaque à une tâche essentielle pour le rendu final : la connexion de toutes les pièces de la carte. Elle relie les scènes entre elles dans Unity, permettant aux joueurs de se déplacer de manière continue à travers le manoir sans rupture, et assurant ainsi une navigation logique et immersive dans l'environnement de jeu.

Séléna, enfin, finalise trois énigmes supplémentaires, portant à cinq le total de puzzles jouables dans le jeu. Elle peaufine leur fonctionnement et leur intégration, tout en veillant à ce qu'elles renforcent le rythme global de l'expérience. Grâce à son travail, les joueurs sont confrontés à des défis variés, qui exigent observation, réflexion et coopération.

## 6 Remerciements

Nous tenons à exprimer notre sincère gratitude à toutes les personnes qui, de près ou de loin, ont contribué au succès du projet "Shattered Mind".

Nous remercions tout particulièrement [Nom/Organisme 1, ex : nos tuteurs de projet, nos professeurs] pour leurs conseils précieux et leur accompagnement tout au long du développement. Leur expertise et leurs retours constructifs ont été essentiels pour guider nos choix techniques et artistiques.

Nous adressons également nos remerciements à [Nom/Organisme 2, ex : nos camarades de promotion, nos familles] pour leur soutien et leur patience, qui ont été une source de motivation inestimable.

Enfin, un grand merci à la communauté des développeurs et aux ressources en ligne ([opengameart.org](http://opengameart.org), etc.) qui mettent à disposition des outils et des assets de qualité, nous permettant de donner vie à notre vision.

## 7 Conclusion

Le projet Shattered Mind a représenté bien plus qu'un simple exercice de développement : il a été une aventure humaine, technique et créative particulièrement marquante pour l'équipe DEVONKHAIN. À travers ce rapport, nous avons retracé les grandes étapes de cette réalisation, de l'organisation de nos tâches selon la méthode Kanban jusqu'à la livraison d'un jeu complet,

porté par une direction artistique forte, une narration travaillée et des fonctionnalités avancées comme le mode multijoueur et l'intelligence artificielle.

Tout au long du développement, nous avons été confronté-es à des défis complexes, tant sur le plan technique (réseau, physique, gestion de l'IA) que sur le plan organisationnel. Ces difficultés nous ont forcés à collaborer de manière rigoureuse, à nous entraider et à apprendre en continu. Chaque obstacle surmonté a renforcé notre cohésion et enrichi nos compétences, nous poussant à aller toujours plus loin dans la qualité du rendu et dans la maîtrise des outils de développement.

Sur le plan artistique, *Shattered Mind* s'est affirmé progressivement comme un univers à part entière, avec son ambiance sombre, ses choix visuels assumés et sa bande-son immersive. Le design du menu principal, les animations, les cinématiques, les énigmes et la structure narrative ont tous été pensés pour renforcer l'immersion et soutenir le message du jeu. L'histoire, développée autour de thématiques psychologiques et de coopération, propose une expérience intrigante et symbolique, qui se distingue des schémas classiques.

Au-delà des compétences acquises, ce projet nous a permis d'expérimenter le travail d'équipe dans un contexte proche de celui d'un studio indépendant. La communication, la gestion des priorités, le respect des délais et l'équilibre entre vision artistique et faisabilité technique ont été au cœur de notre démarche. Nous avons appris à défendre nos idées, à faire des compromis, et surtout à faire converger nos forces pour bâtir une œuvre cohérente.

En conclusion, *Shattered Mind* est le reflet de notre engagement, de notre créativité et de notre passion commune pour le jeu vidéo. Ce projet a été pour nous une étape formatrice, riche en apprentissages, et nous en ressortons avec une meilleure compréhension de ce qu'implique la création d'un jeu, de l'idée initiale à l'aboutissement final. C'est avec fierté que nous présentons aujourd'hui le fruit de ces mois de travail, et avec enthousiasme que nous envisageons la suite de notre parcours dans le monde du développement vidéoludique.

## 8 Annexes

### L'univers de *Shattered Mind*

L'univers de **Shattered Mind** puise ses racines dans l'histoire tragique de la famille **Anderton**, dont les membres ont marqué de leur présence l'imposant manoir de **Oakberry**, théâtre principal du jeu.

#### Personnages clés

- **Emil Anderton** : né à Londres en 1798, il est un industriel visionnaire et le premier docteur fictif à identifier les signes de la schizophrénie. Inspiré du réel Emil Kraepelin, il bâtit sa fortune sur l'industrie des allumettes. Ambitieux mais destructeur, il est victime d'une malédiction mystérieuse lancée par une ouvrière après des années d'exploitation abusive.
- **Penelope Anderton** : issue d'une famille aisée du nord du Royaume-Uni, elle incarne la résilience et la noblesse d'esprit. À travers son journal intime, elle témoigne du déclin progressif de son couple, de sa santé mentale et de leur foyer.
- **Eleanor Anderton** : fille unique du couple, née en 1835 à Phambansbrough. Elle grandit dans un manoir chargé d'histoire, témoin des conflits parentaux et de la lente descente aux enfers de son père.

#### Oakberry Manor

Situé au nord du Royaume-Uni, à proximité de la ville fictive de **Phambansbrough**, **Oakberry Manor** domine un paysage rural inspiré de la région de Grand Manchester et du Lanarkshire écossais. Construit en 1830, le manoir reflète l'esthétique victorienne tout en intégrant des jardins à la française, symbole de raffinement.

Longtemps symbole de réussite, le manoir est aujourd'hui laissé à l'abandon. Recouvert de lierre, ses pierres grises s'effacent dans un paysage aussi morne que mystérieux. Des rumeurs locales racontent qu'il serait hanté par Emil Anderton lui-même, devenu l'ombre d'un homme rongé par la culpabilité, l'alcoolisme et la folie.

## Chronologie de la famille Anderton (Extraits du journal de Penelope)

- **25 décembre 1829 – Londres** : Emil m’a promis de construire une maison à Phambansbrough, sur les hauteurs, avec vue sur toute la ville. J’hésite encore entre “Willow Leaf” et “Oakberry” pour le nom du manoir.
- **16 juillet 1835 – Phambansbrough** : Ma petite Eleanor est née il y a trois jours. Emil lui montre déjà chaque pièce du manoir avec fierté. Il semble redevenir lui-même, l’espace de quelques jours.
- **30 mai 1845 – Phambansbrough** : Depuis l’incendie de l’usine, Emil est devenu méconnaissable. Il boit, crie, frappe. Le manoir, autrefois source de bonheur, n’est plus que ruines et silence.
- **8 mars 1850 – Dernière entrée** : Si tu lis ces lignes, Eleanor, sache que j’ai fait tout ce que j’ai pu pour te protéger. Je t’aime, et j’espère qu’un jour, tu pourras quitter cet endroit. Penelope décédera le soir même. Les circonstances exactes restent floues.

## Note sur le Lore

Le lore détaillé de **Shattered Mind** n’est pas directement accessible in-game. Ces informations sont fournies à titre complémentaire pour enrichir l’expérience du joueur et approfondir l’histoire des personnages et des lieux.

## 8.1 Galerie d'animations



FIGURE 34 – Animation Léon

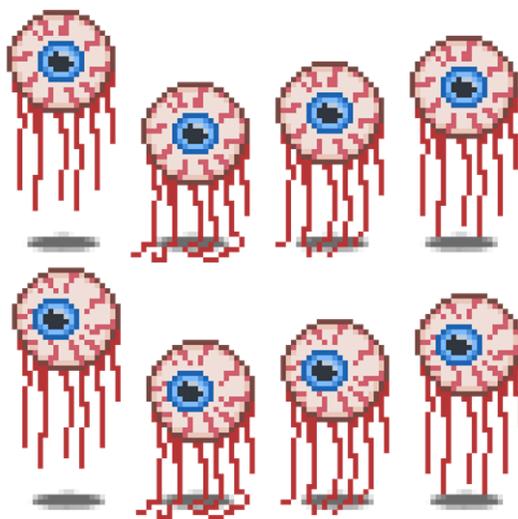


FIGURE 35 – Animation ennemi

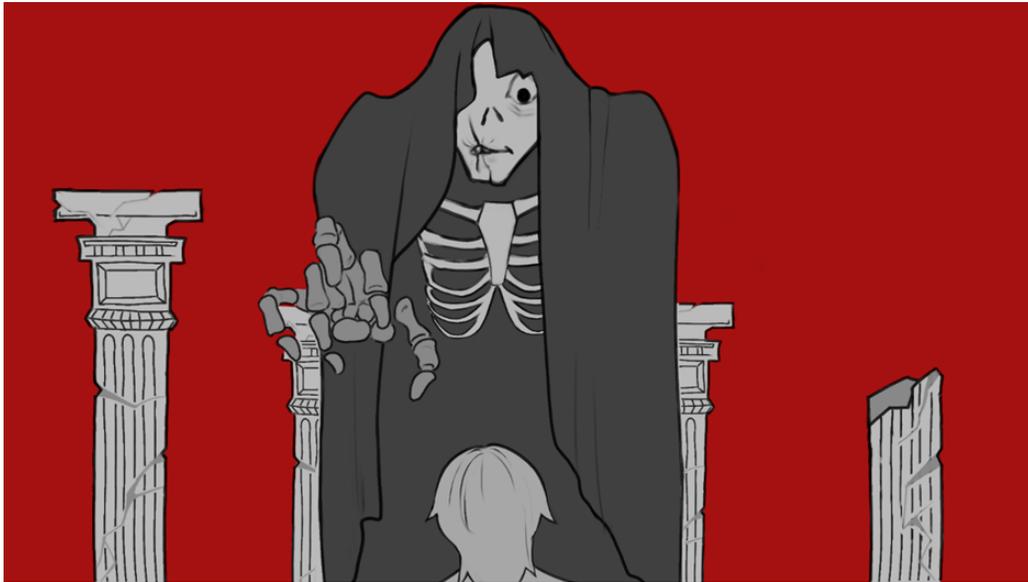


FIGURE 36 – 1ère esquisse de la scène de fin

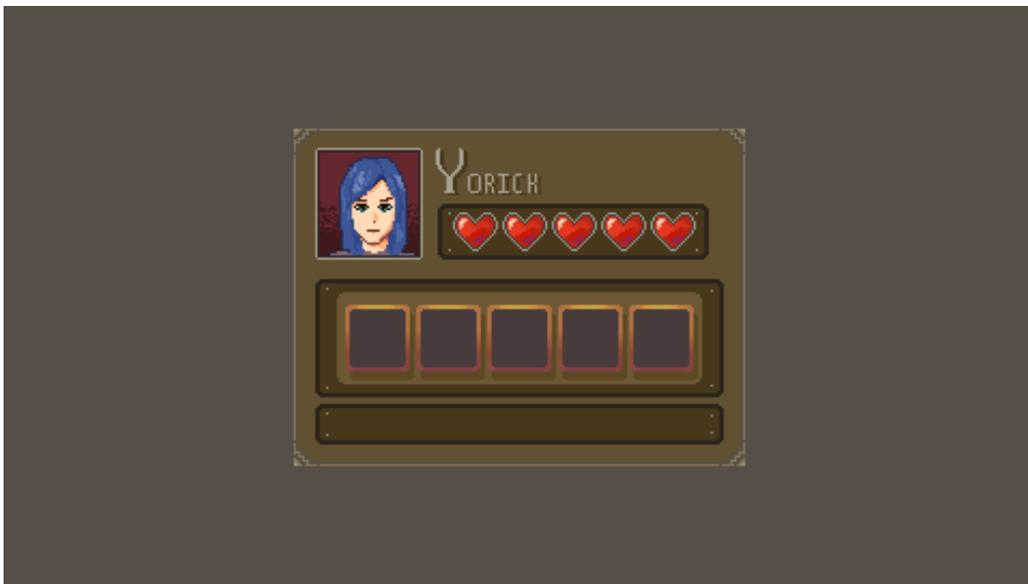


FIGURE 37 – Inventaire de Yorick



FIGURE 38 – Inventaire de Léon

## Note d'intention et Références

Notre ambition pour *Shattered Mind* est de vous plonger dans une expérience de jeu coopérative où la communication n'est pas seulement un atout, mais la clé de votre survie. Nous avons conçu ce jeu autour de l'idée que deux esprits travaillant en parfaite synergie peuvent surmonter les défis les plus complexes.

L'inspiration principale derrière cette approche vient de jeux tels que *We Were Here*. Cette série a brillamment démontré comment une expérience entièrement coopérative peut être immersive et captivante, en forçant les joueurs à interagir constamment, à décrire ce qu'ils voient et à résoudre des énigmes en partageant leurs informations. C'est cette philosophie que nous avons cherché à adapter et à approfondir dans *Shattered Mind*.

Chaque énigme, chaque pièce du manoir d'Oakberry et chaque interaction a été pensée pour encourager les joueurs à échanger des informations précises et à travailler main dans la main. Loin des jeux où la coopération est optionnelle, ici, la collaboration est le cœur même du gameplay. Vous ne pourrez pas progresser sans votre partenaire, ce qui crée une atmosphère de tension palpable, mais aussi une immense satisfaction lorsque vous parvenez à débloquer un passage ou à comprendre un indice grâce à votre coordination.

Nous voulons que les joueurs ressentent le poids de l'histoire du manoir et de la famille Anderton non pas en la lisant, mais en la découvrant à travers les yeux de leur partenaire, en échangeant des indices fragmentés et en reconstituant le puzzle ensemble. C'est une danse délicate entre ce que vous voyez, ce que vous entendez et ce que vous devez communiquer pour avancer. Nous espérons que cette approche renforcera non seulement le sentiment d'accomplissement, mais aussi le lien entre les joueurs, faisant de Shattered Mind une aventure mémorable partagée.

## 8.2 Réseaux Sociaux

- Discord : [discord.gg/RFQ5CZMFJq](https://discord.gg/RFQ5CZMFJq)
- Instagram : <https://www.instagram.com/devonkhain>
- LinkedIn : <https://www.linkedin.com/in/devonkhain-inc-b4a4ab340>

## 8.3 Diagramme de Gantt

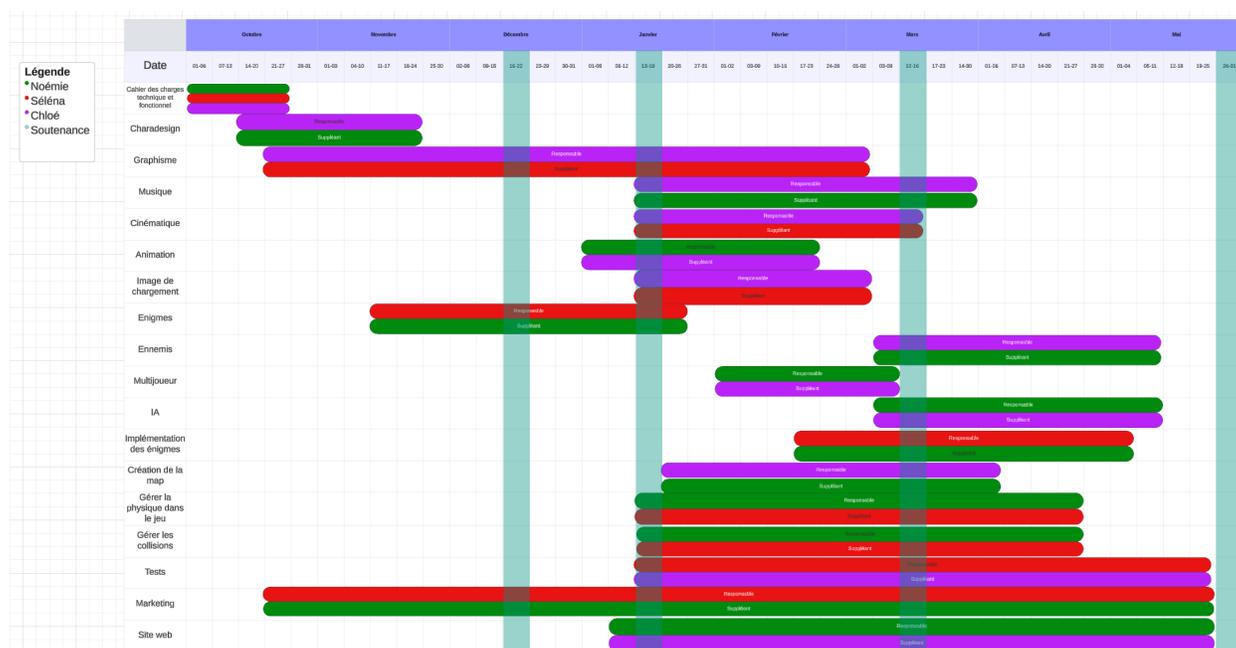


FIGURE 39 – Diagramme de Gantt du projet

## 8.4 Références et crédits

- Designs de la première pièce (plancher, murs, rampes, contour en bois) : <https://opengameart.org> par bluecarrot16.
- Musique : Michael Klier sur <https://opengameart.org>.