

Rapport 1 - Shattered Mind

DEVONKHAIN

Noémie DA CUNHA LOBO, Chloé BAUER, Séléna CONSTANS

15 janvier 2025

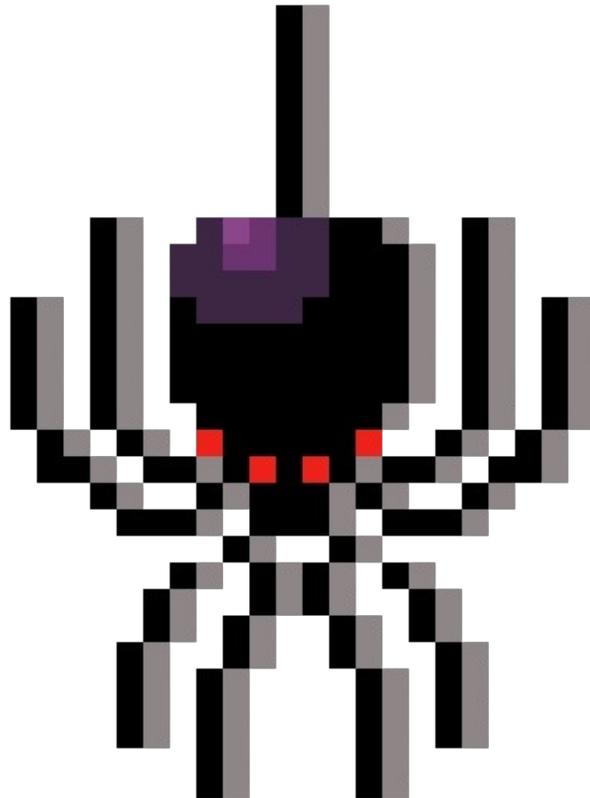


Table des matières

1	Introduction	3
2	Organisation et gestion du projet	3
2.1	Répartition des rôles	3
2.2	Méthodologie de travail	3
3	Avancements réalisés depuis la validation du cahier des charges	5
3.1	Résultats intermédiaires	5
3.2	Avances et retards	9
3.2.1	Avances	9
3.2.2	Retard :	11
3.3	Bilan technique	12
4	Plan d'action pour la prochaine soutenance	19
5	Annexes	19

1 Introduction

Le jeu que nous proposons se nomme Shattered Mind, il sera un mélange d’horreur, d’énigmes et de combats avec une progression en étages dans lequel deux protagonistes doivent tout faire pour se retrouver 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.

2 Organisation et gestion du projet

2.1 Répartition des rôles

Noémie DA CUNHA LOBO occupe le rôle de cheffe d’équipe et supervise la section développement. Elle est responsable de la mise en œuvre des éléments fondamentaux du projet, tels que les fonctionnalités multijoueurs et l’intelligence artificielle (IA). De plus, elle veille à la maintenabilité du code ainsi qu’au respect des normes et conventions de développement.

Chloé BAUER est en charge de la direction artistique et du design, de plus elle fait également partie de l’équipe de développement et vérifie que tous les visuels soient bien implémentés dans le jeu. Elle conçoit la majorité des éléments visuels du jeu, s’assure de leur intégration harmonieuse dans l’environnement numérique, et garantit la cohérence des proportions entre les différents éléments visuels. Tout design non accompagné par des crédits a été fait par Chloé BAUER.

Séléna CONSTANS, quant à elle, occupe le poste de responsable de la narration. Elle veille à ce que toutes les réalisations soient alignées avec l’histoire du jeu et que l’ensemble des éléments respecte une logique et une cohérence narrative.

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.
- Afin de visualiser et organiser notre travail, nous avons recours à un tableau numérique via le logiciel Trello. Ce tableau nous permet de suivre l'état des tâches, en les classant en trois catégories principales : tâches accomplies, tâches en cours et tâches à faire. Grâce à cette visualisation claire, nous pouvons également hiérarchiser les tâches, ce qui nous aide à concentrer nos efforts sur les éléments les plus importants du projet à chaque étape.

Dans cette section, nous présenterons les parties du projet qui sont achevées ou presque terminées. Les explications techniques détaillées de ces différentes composantes seront disponibles dans la section "3.3 Bilan technique".
voici les tâches menées à bien jusqu'à présent :

— **Noémie Da-Cunha-Lobo :**

Elle s'est occupée de toute la partie programmation présente dans ce rapport, donc c'est-à-dire : faire le menu sur unity et l'implémentation de toutes les images. De plus, elle a également fait le parallèle avec Chloé Bauer pour que Unity et les designs concordent. C'est également Noémie qui s'est chargée de faire le site web que nous avons actuellement.

— **Chloé Bauer :**

Pour cette soutenance, elle était en charge de tout les graphismes présent ainsi que de créer un lien avec le jeu si tout concordait. Ce travail de coordination s'est effectué en parallèle avec Noémie Da-Cunha-Lobo qui elle était plus en charge de unity et des potentiels problèmes rencontrés avec le logiciel et les visuels.

— **Séléna Constans :**

Enfin, pour cette soutenance, elle a fait les énigmes ainsi que le discord communautaire.

De plus, le code est partagé entre les membres du groupe grâce à l'utilisation de Git, ce qui facilite la collaboration et la gestion des versions.

3 Avancements réalisés depuis la validation du cahier des charges

3.1 Résultats intermédiaires

— **Site Web et hébergement**

Le site web est désormais en ligne et hébergé chez OVH. Il est accessible via l'URL suivant : shatteredmind.fr.

Ce site, construit à l'aide de WordPress, se compose de plusieurs sections principales :

- **Page d'accueil** : Présente l'univers du jeu, l'équipe de développement, ainsi que les différents réseaux sociaux où suivre DEVONKHAIN.
- **Page d'actualités** : Informe sur l'évolution du projet avec des mises à jour régulières.
- **Page de rapports** : Sert de dépôt pour les différents livrables du projet.



FIGURE 1 – Accueil



FIGURE 2 – Présentation du jeu et de l'équipe

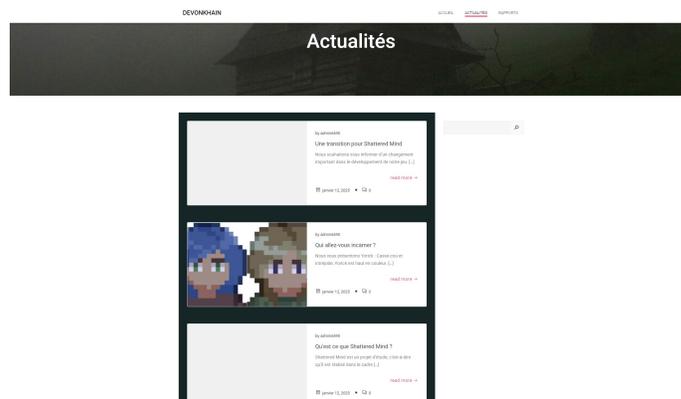


FIGURE 3 – Actualités

— Menu principal

Le menu principal du jeu offre quatre boutons fonctionnels : "Jouer", "Options", "Quitter" et "Crédits".

- **"Jouer"** : Permet d'accéder directement au jeu.
- **"Options"** : Ouvre une fenêtre où l'utilisateur peut ajuster le volume et activer ou désactiver le mode plein écran.
- **"Quitter"** : Permet de quitter le jeu.
- **"Crédits"** : Redirige vers une page web qui affiche notre site officiel, accessible à l'URL shatteredmind.fr.

Le menu principal est accompagné d'une musique de fond libre de droits, intitulée : *Haunted Woods (Horror Drone)* de Michael Klier, cette dernière a été récupérée sur le site opengameart.org. Cette musique a été soigneusement choisie pour enrichir l'expérience utilisateur tout en respectant les contraintes légales liées aux droits d'auteur.



FIGURE 4 – Menu principal



FIGURE 5 – Menu principal avec la fenêtre des réglages

— **Image de chargement**

Pour l'écran de chargement au démarrage du jeu, nous avons choisi une animation simple et efficace : Yorick, le personnage principal, court sur un fond sombre. Cette conception minimaliste vise à maintenir une fluidité visuelle tout en reflétant l'univers du jeu.

— **Première pièce**

La première pièce à avoir été conceptualisée et implémentée a permis de tester les animations d'un personnage, les collisions entre le joueur et le décor, ainsi que la gestion de la physique, car avec les configurations par défaut, le personnage tombait hors de l'écran.

Les designs de toute la pièce donc il s'agit du plancher, des murs, des rampes ainsi que du contour en bois, tout cela provient du site <https://open-gameart.org/> et produit par bluecarrot16.



FIGURE 6 – Course de Yorick pour l'écran de chargement

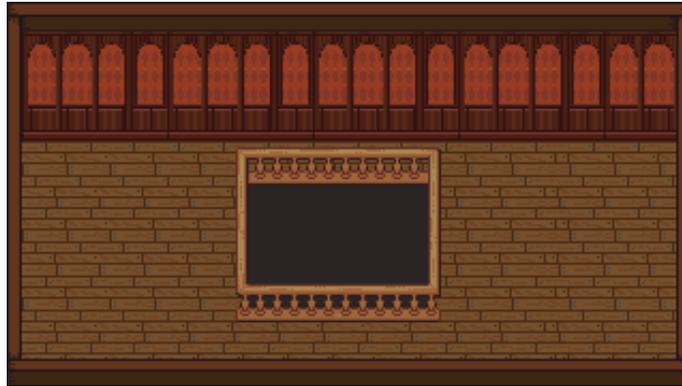


FIGURE 7 – Première pièce

Nous avons réussi grâce à la conception de la salle de créer l'inventaire ainsi qu'une fenêtre rendant le reste de la salle plus sombre. Cela permet d'avoir un peu plus d'éléments montrant le jeu et tout les éléments le constituant.

— Animation de marche de Yorick

— Enigmes

Les joueurs rencontreront différents types d'énigmes qui leur demanderont de collaborer entre deux personnages, Léon et Yorick. Ces énigmes seront de nature variée, chacune impliquant des mécanismes spécifiques pour avancer dans le jeu. Voici un aperçu des types d'énigmes que les joueurs devront résoudre :

- **Énigmes basées sur l'observation visuelle** : Les joueurs devront prêter attention à des symboles, des ombres ou des détails visuels pour en extraire des informations essentielles.



FIGURE 8 – Inventaire sans le personnage

- **Énigmes sonores** : La résolution dépendra de la capacité à mémoriser et à reproduire des séquences musicales ou des sons.
- **Énigmes temporelles** : Les joueurs devront manipuler le temps ou réagir à des événements qui se produisent à des moments spécifiques.
- **Énigmes de perspective et de manipulation d’objets** : Certaines énigmes nécessiteront de changer la perspective du joueur ou de déplacer des éléments dans l’environnement pour débloquer de nouvelles possibilités.
- **Énigmes interactives avec des objets ou des mécanismes** : L’interaction avec des objets physiques ou virtuels sera nécessaire pour avancer, comme le déplacement d’objets ou l’utilisation d’outils spécifiques.
- **Énigmes de communication et de transfert** : Les joueurs devront échanger des informations ou des objets entre deux personnages pour progresser.
- **Énigmes stratégiques ou de réflexion** : Certaines énigmes nécessiteront une réflexion logique ou stratégique pour faire les bons choix ou effectuer des mouvements dans un ordre précis.

Ces types d’énigmes permettront de tester différentes compétences des joueurs, allant de la mémoire visuelle et auditive à la logique, en passant par la coordination entre les personnages.

3.2 Avances et retards

3.2.1 Avances

Dans le cadre de notre projet, un aspect a avancé plus rapidement que prévu, ce qui nous permet de gagner du temps et de nous concentrer sur d’autres priorités à venir.



FIGURE 9 – design de Léon et de Yorick



FIGURE 10 – Course de Yorick vers la droite

- **Énigmes :**
Nous devrions actuellement être aux alentours de 80 % des énigmes conceptualisées. Cependant, nous avons le plaisir d'annoncer que la conceptualisation de toutes les énigmes prévues pour le jeu est déjà terminée.
- **Map :**
Nous avons dû mettre en place pour cette soutenance des éléments qui ne devaient pas être commencé avant une certaine date. Il s'agit notamment de la map, donc ici nous avons mis en place un exemple type de pièce qui pourra être retrouvée dans le jeu final. De plus, la gestion des collisions



FIGURE 11 – Course de Yorick vers la gauche



FIGURE 12 – croquis 1ère image cinématique

et de la physique on été vu rapidement pour être capable de faire une présentation du jeu.

3.2.2 Retard :

Par rapport au diagramme de Gantt, nous sommes dans les temps et n'avons accumulé aucun retard.

```

using UnityEngine;
using UnityEngine.UI;

[ExecuteInEditMode]
0 références
public class SC_BackgroundScaler : MonoBehaviour
{
    4 références
    Image backgroundImage;
    4 références
    RectTransform rt;
    4 références
    float ratio;

    // Start is called before the first frame update
    0 références
    void Start()
    {
        backgroundImage = GetComponent<Image>();
        rt = backgroundImage.rectTransform;
        ratio = backgroundImage.sprite.bounds.size.x / backgroundImage.sprite.bounds.size.y;
    }

    // Update is called once per frame
    0 références
    void Update()
    {
        if (!rt)
            return;

        //Scale image proportionally to fit the screen dimensions, while preserving aspect ratio
        if (Screen.height * ratio >= Screen.width)
        {
            rt.sizeDelta = new Vector2(Screen.height * ratio, Screen.height);
        }
        else
        {
            rt.sizeDelta = new Vector2(Screen.width, Screen.width / ratio);
        }
    }
}

```

FIGURE 13 – Script pour redimensionner l’arrière-plan

3.3 Bilan technique

- **Menu principal** : Tous les fichiers cités sont rangés dans le dossier ”Assets” du projet Unity.

Dans un premier temps, nous devons créer un canvas pour contenir tous nos éléments. (GameObject - UI - Canvas). Il n’y a pas de modification apportée.

Ensuite il faut créer une image pour intégrer l’image du menu principal (Canvas - UI - Image), puis aller dans les propriétés (ou l’inspecteur) puis définir l’image que nous souhaitons dans ”Source Image”.

Il ne faut pas oublier de lier le script à l’image. Pour cela nous allons dans les propriétés de l’image, nous ajoutons un composant et sélectionnons le script. Ensuite il faut créer un objet vide (Canvas - Create Empty) afin d’y stocker nos boutons. Nous créons les quatre boutons.

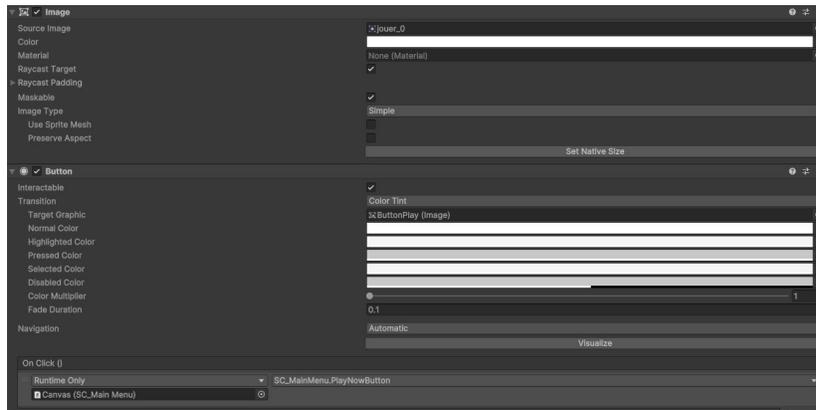


FIGURE 14 – Réglage type d'un bouton

Il faut lier le script au Canvas puis associer chaque fonction au bouton correspondant. C'est le même principe pour le menu réglage. Pour la musique, il suffit de créer un "audio" (GameObject - Audio - AudioSource), et d'y mettre le fichier audio souhaité.

- **Scène 1** : Dans cette section, nous aborderons la création d'une map, les collisions, la physique et l'animation du joueur. Il faut créer une Tile Palette (Window - 2D - Tile Palette). Nous allons y ajouter notre pièce.

Ensuite nous créons les boîtes de collision.

Il est temps d'ajouter notre personnage sur cette map. Pour cela nous pouvons le glisser directement sur l'interface de la scène. Pour que le personnage ne tombe pas, il faut faire attention à régler le Gravity Scale sur 0. De plus pour éviter que le personnage ne fassent de rotation sur lui-même lorsqu'il rencontre une boîte de collision, il faut également cocher Z dans "Freeze Rotation".

Pour créer les animations lorsque le personnage bouge, nous devons créer des nouveaux clips (Window - Animation - Animation - Create new clip). C'est ici que nous mettons les images destinés à l'animation du personnage.

Nous allons dans l'Animator (Window - Animation - Animator), c'est ici que nous allons définir l'état de base avec un clip, puis faire correspondre un mouvement à un clip. Chaque transition (flèche blanche) est associé à un nombre entier.

```

using UnityEngine;

0 références
public class SC_MainMenu : MonoBehaviour
{
    1 référence
    public GameObject MainMenu;
    4 références
    public GameObject SettingsMenu;

    // Start is called before the first frame update
    0 références
    void Start()
    {
        MainMenuButton();
    }

    0 références
    public void PlayNowButton()
    {
        // Play Now Button has been pressed, here you can initialize your game
        UnityEngine.SceneManagement.SceneManager.LoadScene("GameLevel");
    }

    0 références
    public void SettingsButton()
    {
        if (SettingsMenu.activeSelf)
            SettingsMenu.SetActive(false);
        else
            SettingsMenu.SetActive(true);
    }

    0 références
    public void CreditsButton()
    {
        // Open website in new tab
        Application.OpenURL("https://shatteredmind.fr");
    }

    1 référence
    public void MainMenuButton()
    {
        // Show Main Menu
        MainMenu.SetActive(true);
        SettingsMenu.SetActive(false);
    }

    0 références
    public void QuitButton()
    {
        // Quit Game
        Application.Quit();
    }
}

```

FIGURE 15 – Script définir les fonctions pour le menu principal

```
using UnityEngine;
using UnityEngine.Audio;

0 références
public class SC_SettingsMenu : MonoBehaviour
{
    1 référence
    public AudioManager audioMixer;
    0 références
    public void SetVolume(float volume)
    {
        audioMixer.SetFloat("volume", volume);
    }

    0 références
    public void SetFullScreen(bool isFullScreen)
    {
        Screen.fullScreen = isFullScreen;
    }
}
```

FIGURE 16 – Script définir les fonctions du menu réglage

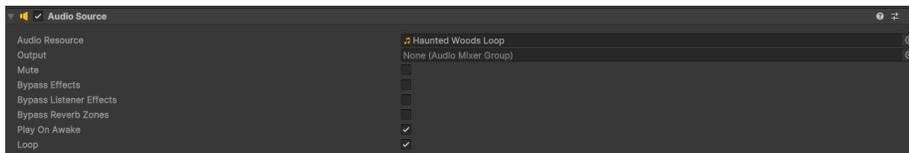


FIGURE 17 – Réglage pour la musique

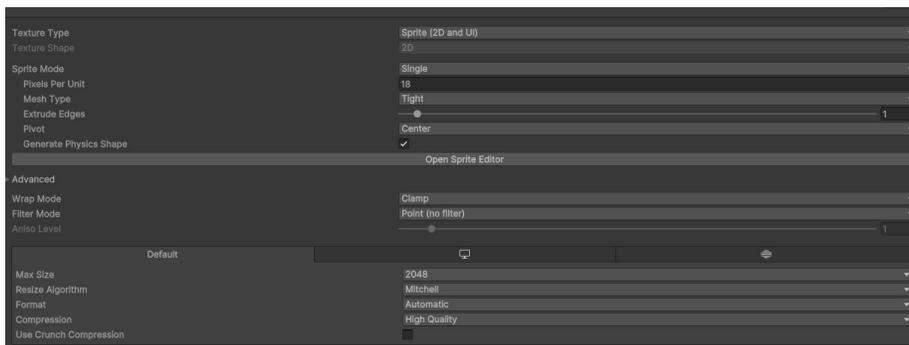


FIGURE 18 – Réglage de la pièce

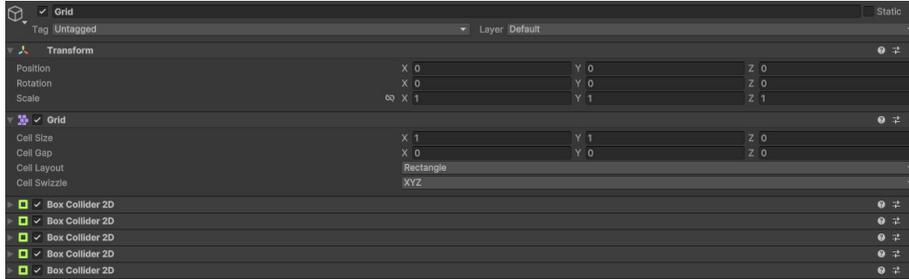


FIGURE 19 – Création des boîtes de collision

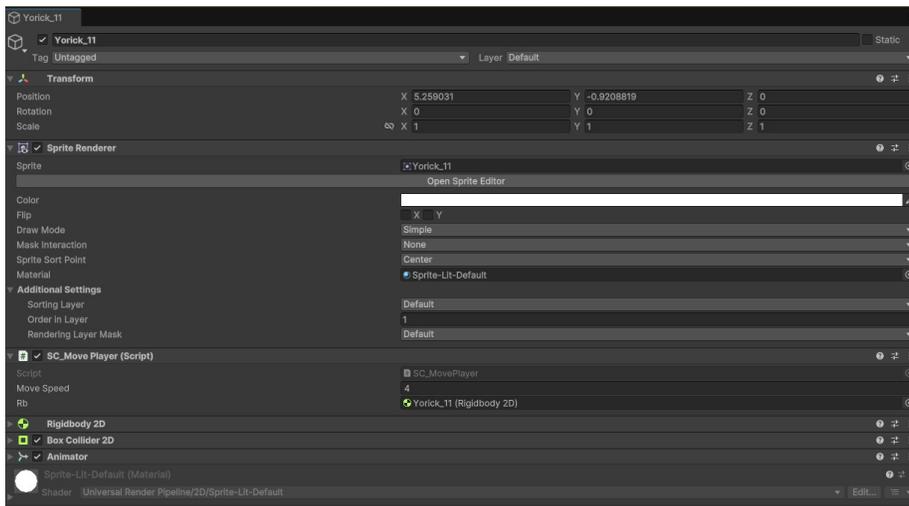


FIGURE 20 – Réglage Yorrick



FIGURE 21 – Création d'un clip

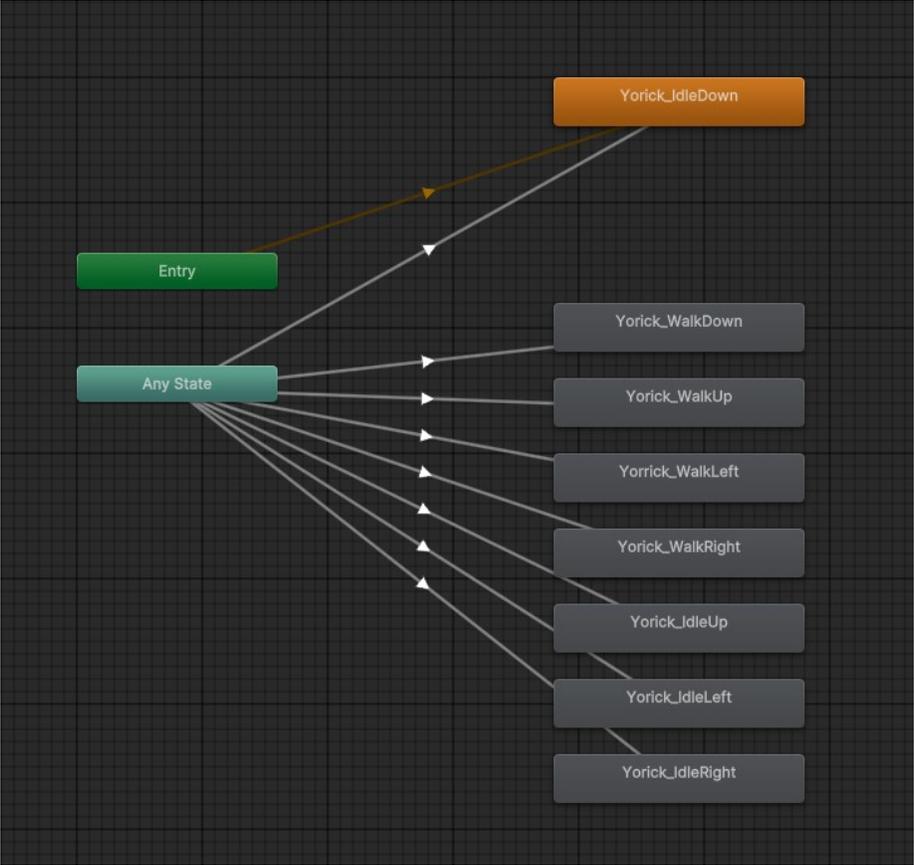


FIGURE 22 – Animator

```

using UnityEngine;

0 références
public class SC_MovePlayer : MonoBehaviour
{
    1 référence
    public float moveSpeed = 5f;
    3 références
    public Rigidbody2D rb;
    9 références
    Vector2 dir;
    6 références
    Animator anim;

    5 références
    int lastPosition;

    0 références
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        anim = GetComponent<Animator>();
    }

    0 références
    void Update()
    {
        dir.x = Input.GetAxisRaw("Horizontal");
        dir.y = Input.GetAxisRaw("Vertical");
        rb.MovePosition(rb.position + dir * moveSpeed * Time.fixedDeltaTime);

        SetParam();
    }

    1 référence
    void SetParam()
    {
        if (dir.x == 0 && dir.y == 0)
            anim.SetInteger("dir", lastPosition);

        else if (dir.y < 0) // Down
        {
            anim.SetInteger("dir", 1);
            lastPosition = 0;
        }

        else if (dir.y > 0) // Up
        {
            anim.SetInteger("dir", 2);
            lastPosition = 5;
        }

        else if (dir.x > 0) // Right
        {
            anim.SetInteger("dir", 3);
            lastPosition = 6;
        }

        else if (dir.x < 0) // Left
        {
            anim.SetInteger("dir", 4);
            lastPosition = 7;
        }
    }
}

```

FIGURE 23 – Script pour les mouvements du personnage

4 Plan d'action pour la prochaine soutenance

Dans le cadre de la prochaine soutenance, nous avons élaboré un plan d'action rigoureux visant à assurer des progrès significatifs dans le développement de notre jeu. Ce plan repose sur les points suivants :

- **Design et Animations :**
Tous les designs et animations seront terminés. Chaque élément visuel sera prêt à être intégré dans le jeu.
- **Musique :**
La composition musicale est sur le point d'être finalisée. Cet aspect apportera une ambiance sonore cohérente pour accompagner le gameplay.
- **Cinématiques :**
Les cinématiques seront prêtes à être présentées. Elles joueront un rôle clé dans la narration et l'immersion des joueurs.
- **Ennemis et IA :**
La conception des ennemis et de l'intelligence artificielle débutera. Ce développement permettra de poser les bases pour des interactions et des défis adaptés.

Ces étapes représentent des progrès importants dans notre processus de développement, et nous sommes impatients de partager ces avancées lors de la prochaine soutenance. Restez connectés pour suivre l'évolution de notre projet.

5 Annexes

Nous avons mis en place des réseaux sociaux où vous pouvez nous retrouver et suivre les avancées du jeu Shattered Mind.

- **Discord :**
discord.gg/RFQ5CZMFJq*
- **L'instagram :**
[@devonkhain](https://www.instagram.com/devonkhain)
- **Linkedin :**
www.linkedin.com/in/devonkhain-inc-b4a4ab340
- **Trello :**

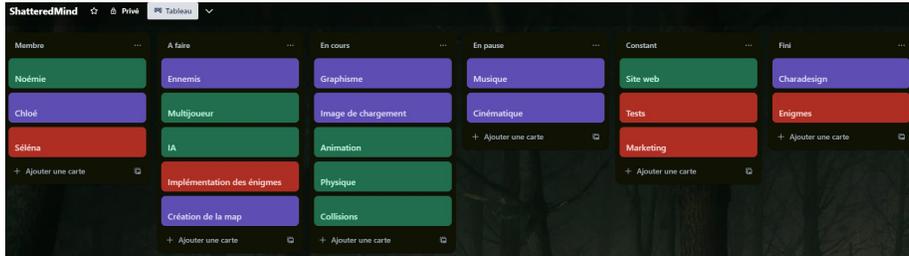


FIGURE 24 – Organisation actuelle - Trello

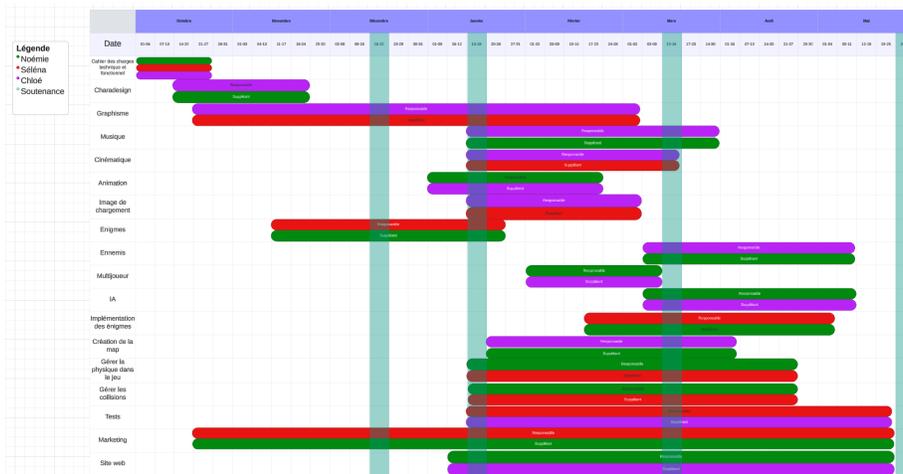


FIGURE 25 – Diagramme de Gantt