

Rapport de Soutenance n°2

We Don't See Sharp

Flush Hand Studio™

Membres de l'équipe :

Léo DESCHANEL-LOSCIUTO
Adrien GOMES
Laetitia BERKOUNE
Doria BOUZIANE
Emma RUCAY



Table des matières

Table des matières	1
1 Introduction	2
1.1 Contexte du projet	2
1.2 Objectifs pédagogiques et techniques	2
1.3 Structure du rapport	2
1.4 Outils de gestion principaux	2
2 Présentation générale du projet	3
2.1 Résumé du concept	3
2.2 Scénario résumé	3
2.3 Originalité et innovations techniques	4
2.4 Intérêt du projet pour les joueurs et l'industrie vidéoludique	5
3 Organisation et méthodologie	5
3.1 Répartition des tâches	5
3.2 Diagramme de Gantt (barres pour soutenances & taux avancement)	7
4 Réalisations techniques	8
4.1 Mécaniques de gameplay	8
4.2 Graphismes et animations	11
4.3 Intelligence artificielle	14
4.4 Multijoueur	16
5 Bilan intermédiaire	18
5.1 Avancées réalisées depuis la S1	18
5.2 Défis rencontrés et retards, solutions et/ou explications	18
5.3 Comparaison des progrès par rapport aux objectifs initiaux	19
6 Perspectives pour le rendu final	20
6.1 Fonctionnalités en cours de développement	20
6.2 Objectifs pour la dernière soutenance	20
6.3 Potentiels ajustements ou améliorations prévues	21
7 Conclusion	21
Annexes	23
Cahier des charges fonctionnel	23
Ressources utilisées	23
Références	24

1 Introduction

1.1 Contexte du projet

L'équipe de Flush Hand Studio™ vous propose, à travers ce Rapport, de découvrir l'organisation, le développement et les idées de conception de son premier jeu vidéo, *We Don't See Sharp*. Ce jeu d'horreur en vue à la première personne se distingue par ses mécaniques de puzzle et ses interactions environnementales immersives.

L'objectif fondamental de ce projet est de créer une expérience vidéoludique captivante, où chaque choix des joueurs impacte le déroulement de l'histoire. Pour atteindre cet objectif, l'équipe s'appuie sur Unity3D [1] et le langage C#, permettant d'explorer pleinement ses compétences techniques afin d'offrir un gameplay fluide et dynamique.

1.2 Objectifs pédagogiques et techniques

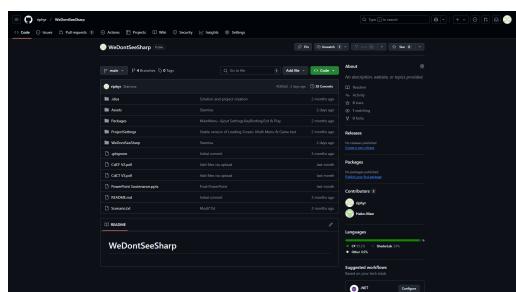
Ce projet valorise le travail collaboratif et l'acquisition de compétences en gestion de projet. Il intègre des concepts originaux comme l'utilisation de cartes de poker pour influencer l'issue du jeu et l'intégration potentielle de la reconnaissance vocale pour ajouter davantage d'interaction entre le joueur et son environnement. En combinant tension psychologique et réflexion stratégique, il offre une narration riche et interactive. Ce projet favorise également l'apprentissage des techniques de programmation en C# et la gestion efficace de ressources communes.

1.3 Structure du rapport

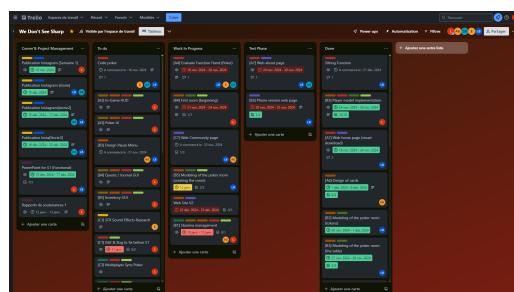
Ce rapport est structuré de manière à présenter successivement le contexte du projet, la présentation générale, l'organisation avec sa méthodologie ainsi que l'avancement, les réalisations techniques (récapitulatives et nouvelles), le bilan pré rendu final à l'état actuel du projet [2], les perspectives et dernières étapes avant l'ultime soutenance [3], avant de conclure avec une synthèse et des annexes détaillées.

1.4 Outils de gestion principaux

Pour mettre en place ce projet nous avons utilisé deux outils majeurs qui permettent à la fois de communiquer des informations mais aussi de gagner énormément en efficacité sur la gestion du travail de groupe : GitHub [4] et Trello [5]



- GitHub Project -



- Méthodologie kankan par Trello -

2 Présentation générale du projet

2.1 Résumé du concept

Notre projet consiste en la création d'un jeu vidéo jouable à la fois en solo et en multijoueur en vue à la première personne, combinant des éléments narratifs et des mécaniques de puzzle au sein d'un environnement 3D immersif. *We Don't See Sharp* commence dans un club de poker clandestin avant de basculer dans un univers mystérieux. Le joueur explore des zones thématiques représentant des moments clefs de la vie d'un personnage décédé au cœur du lore du jeu.

⇒ Une fois transporté dans le hub mystérieux, le joueur découvre un espace central entouré de portes, chacune menant à une salle thématique représentant une étape clef de la vie d'un personnage décédé plusieurs années auparavant dans le même club de poker clandestin que celui dans lequel le joueur apparaît au lancement de sa session de jeu. Pour progresser et espérer s'échapper de cet endroit oppressant, le joueur devra explorer ces salles, qui retracent des moments marquants de la vie de cette âme tourmentée. Ces espaces incluent des lieux symboliques, tels que la maison où il a vécu, son premier emploi, ou des événements charnières de sa vie d'adulte. Chaque salle est conçue pour dévoiler progressivement les éléments du récit, à travers des énigmes à résoudre, des objets à manipuler, et des indices à collecter. Au fur et à mesure de ces explorations, le joueur reconstitue le contexte tragique de la mort du personnage, tout en étant confronté à des manifestations paranormales et à des choix cruciaux qui influenceront le dénouement de la partie.

⇒ Le mode multijoueur est coopératif et se joue en ligne. Les joueurs travaillent ensemble pour résoudre des puzzles et découvrir des indices qui débloqueront des parties de l'histoire. Ils devront interagir avec l'environnement, utiliser des objets trouvés sur place, dont des cartes de poker qui ont un rôle déterminant dans le déroulement du scénario, et affronter des événements paranormaux de plus en plus intenses. L'aventure en mode solo consistera quant à elle à un jeu certes proche dans le fond mais dans la forme, il y aura plus d'intimité, de pression et de frayeurs que le joueur se fera à lui-même.

2.2 Scénario résumé

We Don't See Sharp! commence dans une ambiance calme et intrigante au sein d'un club de poker clandestin, caché au cœur d'une boîte de nuit, où le joueur prend place à une table avec divers personnages non joueurs (PNJ^[6]). Ce cadre tamisé est rapidement assombri par une atmosphère sinistre lorsque la partie commence. Les enjeux deviennent significatifs, et le joueur croit naïvement que la soirée se déroulera sans accroc. Cependant, la tension monte lorsque des événements inexplicables commencent à se produire : des éclats de rire étranges, des murmures inquiétants, ou encore des ombres furtives apparaissant dans la salle.

Soudain, le décor se transforme, la lumière s'éteint, et le joueur chute dans un espace alternatif (style backrooms^[7]), un hub mystérieux qui sera le centre névralgique de son aventure. Ce lieu désorientant ne présente aucune issue visible, à l'exception d'une porte entrouverte qui incite le joueur à l'explorer. Chacune des portes de ce hub mène à un niveau distinct, représentant une facette de la vie de l'entité hantée (comme précisé dans la section ci-dessus) qui le suit depuis qu'il est tombé dans cet environnement inconnu. Ces niveaux sont soigneusement conçus pour refléter des moments clefs de son existence, tels que :

- **La maison de sa vie** : Une reconstitution détaillée de son domicile, où le joueur peut interagir avec des objets du passé, revivre des souvenirs et découvrir des indices sur les événements tragiques qui ont conduit à sa mort. Le tout en évoluant progressivement dans cet espace à travers des énigmes.
- **Le labyrinthe d'Halloween** : Dans cette zone, le joueur se retrouve plongé dans un vaste labyrinthe extérieur, constitué de hautes haies taillées. L'atmosphère est marquée par une obscurité oppressante, avec des bruits inquiétants et des murmures étranges. Alors qu'il explore les recoins inhospitaliers du labyrinthe, le joueur doit naviguer habilement pour éviter des entités malveillantes qui surgissent des ombres, reflétant les peurs et les regrets de l'entité hantée.
- **Le lieu de travail** : Dans ces bureaux, le joueur pourra revivre l'expérience professionnelle du défunt aussi bien pour les bons comme les mauvais côtés. Cet espace qui semble n'être qu'un lieu de labeur, cache en réalité un aspect bien plus sombre derrière son apparence formelle.
- ... et d'autres encore ...

Au cours de chaque niveau, des apparitions paranormales, telles que des silhouettes floues ou des voix chuchotées, sont conçues pour susciter la peur et le mystère. Les joueurs devront interagir avec des objets spécifiques, fouiller des documents cachés, et parfois résoudre des énigmes plus ou moins complexes pour progresser. Les choix que le joueur effectue dans ces niveaux peuvent également influer l'histoire, créant des chemins narratifs alternatifs. Par exemple, choisir d'interagir avec un certain objet peut révéler un indice crucial ou déclencher un événement clef.

L'objectif ultime du joueur est de découvrir la vérité sur la vie tragique de l'entité qui hante le hub, tout en cherchant un moyen de s'échapper de ce monde alternatif. Les joueurs collecteront des cartes de poker tout au long de leur parcours, qui serviront à débloquer un combat final de cartes contre une représentation de l'entité. Ce combat ne sera pas simplement une question de chance, mais exigera une compréhension approfondie des événements et des choix effectués tout au long du jeu.

En intégrant des éléments de gameplay de type "puzzle", chaque pièce présente des défis uniques. Parfois, le joueur devra participer à des jeux de cartes avec des PNJ, qui serviront à la fois de mécanisme de progression et de révélateurs de l'histoire. Ces rencontres permettent également d'en apprendre davantage sur les PNJ, leurs expériences et leurs liens avec le monde du jeu, enrichissant ainsi le récit.

Le jeu est conçu pour être une expérience immersive où le joueur ressent une véritable pression psychologique, alimentée par l'ambiance sonore, les événements inattendus et la profondeur narrative. Chaque action, choix ou interaction a un impact, rendant l'expérience personnelle et mémorable.

2.3 Originalité et innovations techniques

Le jeu introduit des concepts innovants tels que l'utilisation des cartes de poker pour influencer le déroulement de l'histoire et de potentiels mécaniques de reconnaissance vocale

permettront des interactions dynamiques. Ces éléments renforcent l'expérience immersive en ajoutant des aspects stratégiques et interactifs uniques. Il sera en effet possible, au cours de plusieurs moments clés du jeu, de participer à des parties avec des IA [8].

2.4 Intérêt du projet pour les joueurs et l'industrie vidéoludique

En s'inspirant de jeux comme *Phasmophobia*, *Visage* ou encore *Granny*, mais en y intégrant des mécaniques originales et remixées, *We Don't See Sharp* offre une expérience interactive et dans laquelle le joueur a une réelle incidence sur les événements. Le projet met l'accent sur la narration interactive où chaque choix des joueurs influence directement le scénario, renforçant l'idée que "vos choix ont des conséquences".



- Jeu d'horreur "**Phasmophobia**" -



- Jeu d'horreur "**Visage**" -

3 Organisation et méthodologie

3.1 Répartition des tâches

- **Léo DESCHANEL-LOSCIUTO** : Responsable du projet de groupe, du maintien de l'organisation technique et temporelle à travers les diverses étapes du projet [9] ainsi que de la rédaction de l'ensemble des documents officiels [10] nécessaires aux différentes représentations évaluées du PROJET S2. Responsable de l'aspect multijoueur/réseau, des mécaniques de jeu principales, ainsi que de la gestion GUIs [11]. Assiste également EMMA et ADRIEN dans la conception du scénario, dans les interactions avec l'environnement et aussi dans la conception des espaces de jeu.
- **Adrien GOMES** : Principalement en charge des graphismes [12], du level design [13], de la conception des espaces, et des interactions entre joueur(s) et environnement. Il joue également un rôle dans l'assistance auprès de LEO et de LAETITIA vis-à-vis des déplacements du joueur ainsi que de certains aspects esthétiques liés à la promotion du jeu.
- **Laetitia BERKOUNE** : Responsable spécialisée dans le développement web (création des sections du site, de son design et de l'ensemble de ses fonctions), elle gère également les assets du jeu permettant de faciliter le travail de conception des espaces et d'optimisation de l'espace mémoire du jeu. Elle apporte en même temps son soutien aux interactions environnementales à ADRIEN, ainsi qu'à DORIA et EMMA sur les aspects promotionnels du projet ou sur la conception des fonctionnalités relatives aux multiples IA [8][14].
- **Doria BOUZIANE** : Responsable de la conception des IA de poker [8] grâce notamment à son

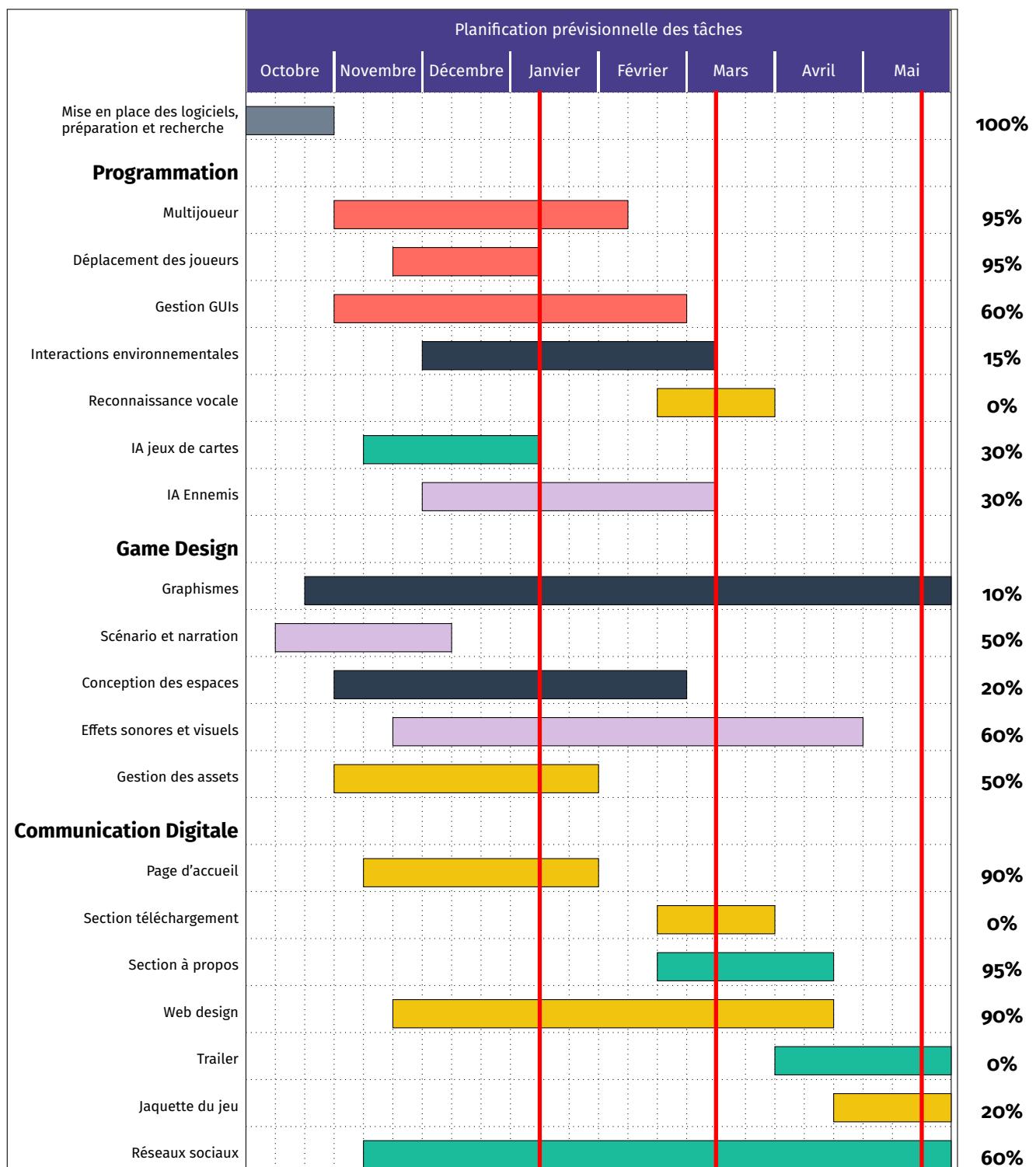
intérêt initial pour ce jeu de bluff, elle gère aussi en parallèle la promotion sur les réseaux sociaux et assistera dans la reconnaissance vocale ainsi que dans la mise en réseau finale de l'ensemble des composants du jeu (en particulier lors des phases de merging de scène entre membres). Elle intervient également dans le support des SFX^[15] divers.

- **Emma RUCAY** : Membre particulièrement plongée dans le développement de l'intelligence artificielle, en particulier responsable pour les ennemis^[14] et assistante sur les jeux de cartes^[8]. Responsable des effets sonores et visuels, elle intervient également en très grande partie dans le scénario^[16] et la promotion via le trailer.

Tâches	Léo	Adrien	Laetitia	Doria	Emma
Programmation					
Multijoueur	R			S	
Mécaniques de jeu					
Déplacement des joueurs	R	S			
Gestion GUIs	R	S			
Interactions avec l'environnement	S	R	S		
Reconnaissance vocale			R	S	
Intelligence artificielle					
Jeux de cartes			S	R	S
Ennemis			S		R
Game Design					
Style					
Graphismes		R	S		
Scénario et narration	S	S			R
Environnement					
Conception des espaces		R	S		
Effets sonores et visuels				S	R
Gestion des assets	S		R		
Communication Digitale					
Web Development					
Page d'accueil			R	S	
Section téléchargement			R	S	
Section à propos			S	R	
Web design			R	S	
Promotion					
Trailer				R	S
Jaquette du jeu		S	R		
Réseaux sociaux (instagram)			S	R	

Répartition des tâches par membre de l'équipe.
R : responsable - S : suppléant(e)

3.2 Diagramme de Gantt (barres pour soutenances & taux avancement)



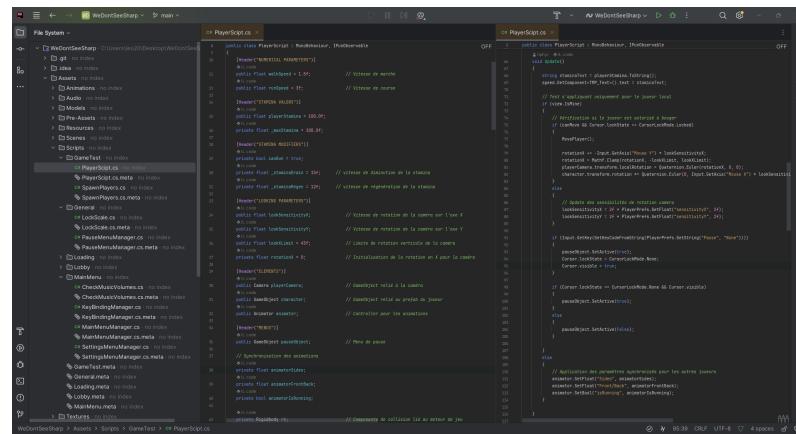
Couleur	Tâche	Responsable
Vert	IA jeux de cartes, Section à propos, Trailer, Réseaux sociaux	Doria
Rouge	Multijoueur, Déplacement des joueurs, Gestion GUIs	Léo
Noir	Interactions environnementales, Graphismes, Conception des espaces	Adrien
Vert clair	IA Ennemis, Effets sonores et visuels, Scénario et narration	Emma
Jaune	Reco. vocale, Gestion assets, Page d'accueil - téléchargement, Jaquette du jeu	Laetitia

4 Réalisations techniques

4.1 Mécaniques de gameplay

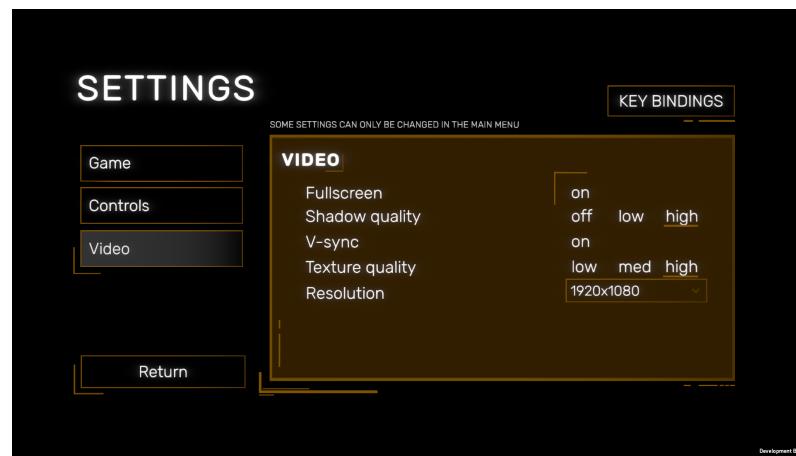
Présenté en S1

Les mécaniques de gameplay déjà mises en œuvre intégraient plusieurs fonctionnalités essentielles pour enrichir l'expérience du joueur. Le système de mouvements des personnages, par exemple, offre des contrôles fluides et personnalisables, avec la possibilité d'ajuster la sensibilité des axes horizontaux et verticaux directement depuis les paramètres du jeu. Ce système assure une navigation intuitive dans l'univers du jeu tout en respectant les préférences du joueur.



- Script de gestion des mouvements du joueur -

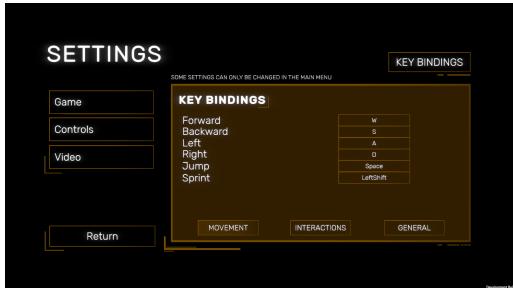
Le menu principal et les paramètres jouent également un rôle clef, offrant une structure claire pour naviguer entre les différentes options et menus secondaires. Des fonctionnalités avancées, comme la gestion des ombres, la résolution d'écran et les textures, permettent de personnaliser l'expérience graphique en fonction des capacités matérielles ou des préférences du joueur. Les ajustements sont instantanés et basés sur des sauvegardes via PlayerPrefs [17], garantissant une continuité dans les sessions de jeu.



- Interface de paramètres graphiques -

De plus, un système de configuration des raccourcis clavier a été implémenté pour s'assurer que chaque joueur puisse adapter les commandes à ses besoins spécifiques. Cette fonction-

nalité améliore l'accessibilité et l'immersion, en particulier dans les phases d'exploration et d'interaction.



- Interface d'attribution de touches -

The screenshot shows the Unity Editor with a script named 'KeyBindings.cs'. The code defines a dictionary where keys map to actions. It includes methods for setting up key mappings and handling events like 'OnEnable' and 'OnDisable'.

```

using System.Collections.Generic;
using UnityEngine;

public class KeyBindings : MonoBehaviour
{
    public static Dictionary<KeyCode, string> keybindings = new Dictionary<KeyCode, string>();
    public static void SetKey(KeyCode key, string action)
    {
        keybindings[key] = action;
    }
    void OnEnable()
    {
        foreach (var entry in keybindings)
        {
            if (entry.Key != KeyCode.None)
            {
                InputManager.OnInput += HandleInput;
            }
        }
    }
    void OnDisable()
    {
        foreach (var entry in keybindings)
        {
            if (entry.Key != KeyCode.None)
            {
                InputManager.OnInput -= HandleInput;
            }
        }
    }
    void HandleInput(string action, float value)
    {
        if (keybindings.ContainsKey(KeyCode.Escape))
        {
            if (action == keybindings[KeyCode.Escape])
            {
                Application.Quit();
            }
        }
        else
        {
            foreach (var entry in keybindings)
            {
                if (entry.Key == KeyCode.Escape)
                {
                    continue;
                }
                if (entry.Key == KeyCode.None)
                {
                    if (action == entry.Value)
                    {
                        entry.Key = KeyCode.Escape;
                        break;
                    }
                }
                else
                {
                    if (Input.GetKey(entry.Key) && entry.Key != KeyCode.Escape)
                    {
                        if (action == entry.Value)
                        {
                            entry.Key = KeyCode.Escape;
                            break;
                        }
                    }
                }
            }
        }
    }
}

```

- Script des attributions de touches -

Enfin, des efforts avaient été concentrés sur des transitions visuelles et sonores immersives, intégrées au gameplay, telles que des animations fluides et optimisées des caméras et des effets sonores dynamiques, renforçant l'engagement du joueur. Ces éléments contribuent encore aujourd'hui à la cohérence et à l'enrichissement global de l'expérience de jeu.

Exclusivités en S2

Depuis la dernière soutenance, plusieurs nouvelles mécaniques de gameplay ont été implémentées afin d'enrichir l'expérience du joueur et d'améliorer l'interaction avec l'environnement principalement. Ces ajouts concernent la gestion des objets, le contrôle du joueur et diverses améliorations liées à l'immersion et aux mécaniques de jeu.

Système d'inventaire et refonte du mouvement

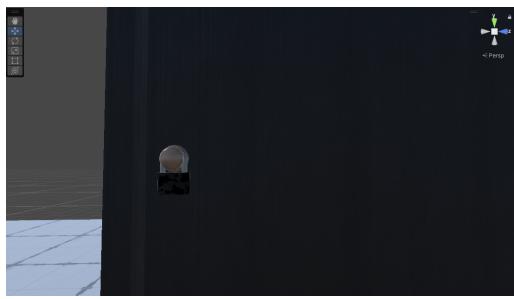
Un inventaire a été ajouté au joueur, permettant de stocker et gérer divers objets. Il intègre plusieurs fonctionnalités : ajout, suppression, vérification de possession, comptage et affichage (uniquement en mode debug Unity pour l'instant). Ce système servira à la gestion des objets interactifs comme des clefs, des journaux, des photos, des appareils divers...

Le mouvement du joueur a également été revu : le script basé sur un *Rigidbody* [18] a été remplacé par un *CharacterController* [19], offrant de meilleures performances et une gestion plus fluide des collisions. Cette modification a permis d'introduire un système de sprint limité, avec consommation et régénération d'endurance, apportant ainsi un aspect stratégique à la fuite face aux entités hostiles.

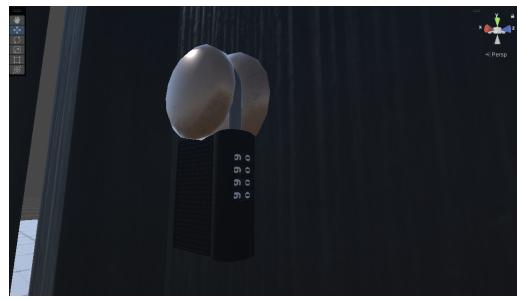
Système d'interaction avec les portes

Les portes du jeu sont désormais interactives selon trois modes :

- **Normale** : s'ouvre et se ferme librement par rotation.
- **Lock Key** : nécessite une clef spécifique présente dans l'inventaire.
- **Padlock** : équipée d'un cadenas numérique, nécessitant un code pour être déverrouillée.



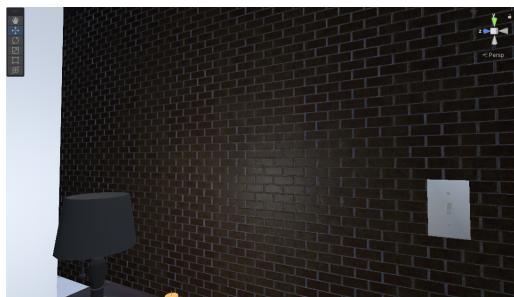
- Porte verrouillée pas cadenas à clef -



- Porte sécurisée par un code -

Fenêtres et interrupteurs

Les fenêtres peuvent désormais être ouvertes et fermées via une animation fluide, offrant des mécaniques de gameplay supplémentaires. En complément, un système d'interrupteur a été ajouté, permettant d'activer et désactiver des groupes d'objets comme les lumières.



- Interrupteur activable pour la lumière -



- Fenêtre ouvrable par animation -

Systèmes d'interaction avancés avec les meubles

Les meubles tels que les armoires ont été implémentés. Chaque porte peut être ouverte indépendamment et les tiroirs sont manipulables grâce à un script adaptatif permettant leur translation selon l'axe défini dans l'éditeur.



- Tiroirs interactifs -



- Armoire avec espace pour se cacher -

Système de bougies interactives et éclairage dynamique

Les bougies peuvent désormais être allumées par le joueur et possèdent une durée de vie dynamique : elles fondent progressivement. Elles peuvent être ramassées et replacées ailleurs, permettant un éclairage flexible adapté à la progression du joueur.

En complément, un système de briquet et d'allumettes a été ajouté. Le briquet est un objet persistant une fois récupéré, tandis que les allumettes sont consommables et peuvent permettre entre autres d'éclairer temporairement le joueur.



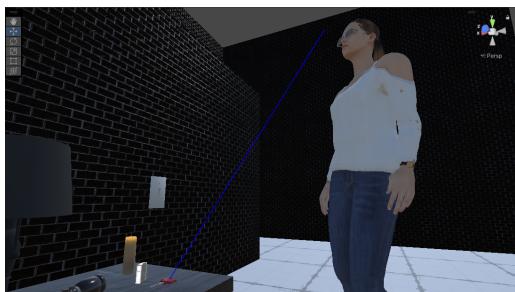
- Bougie allumable et transportable -



- Allumette utilisable pour un éclairage temporaire -

Système d'interaction du joueur avec son environnement

Un raycast [20] est désormais constamment associé à la caméra du joueur, permettant d'interagir avec les objets dans son champ de vision s'ils sont marqués comme interactifs. Ce système permet de limiter les interactions aux objets réellement atteignables et d'améliorer la fluidité du gameplay.



- Raycast détectant un objet interactif -



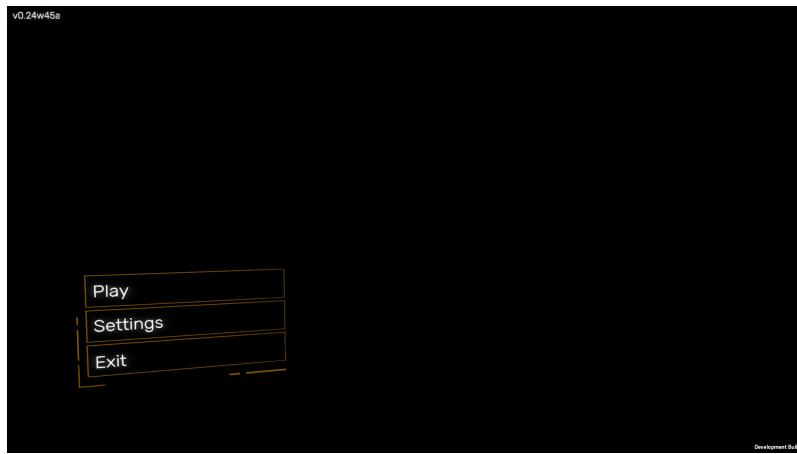
- Raycast ne détectant rien -

Ces ajouts techniques ont permis d'affiner l'expérience du jeu et d'optimiser les performances, tout en renforçant l'immersion du joueur dans un univers interactif et dynamique.

4.2 Graphismes et animations

Présenté en S1

Dans la phase initiale du projet, les éléments graphiques s'étaient concentrés sur les premières versions des menus du jeu. Une V1 fonctionnelle des menus, comprenant le menu principal, le menu des paramètres, et les interfaces essentielles, avait été mise en place avec un design simple mais efficace. Ces menus permettaient de naviguer entre les différentes options du jeu tout en offrant une structure solide pour des itérations que nous avons depuis développer et que nous continueront de faire évoluer.

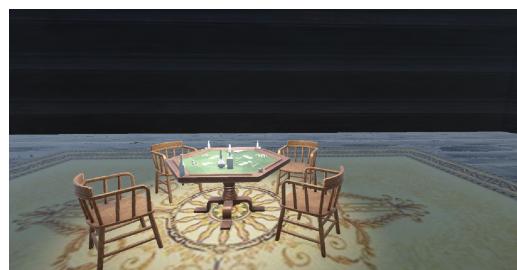


- Interface au lancement du jeu -

En parallèle, le game design de deux environnements clefs avait été entamé. Tout d'abord, une version conceptuelle de la maison, un espace central du jeu, avait été réfléchie, mettant en avant les zones importantes tout en laissant de la place à des améliorations esthétiques et des détails narratifs. De plus, la direction artistique de la salle de poker clandestine avait été définie, offrant une atmosphère sombre et mystérieuse qui correspond parfaitement au thème du jeu. Ces environnements ont permis de poser les bases pour nos futures intégrations plus riches, détaillées et immersives.



- V1 Scène de la maison -

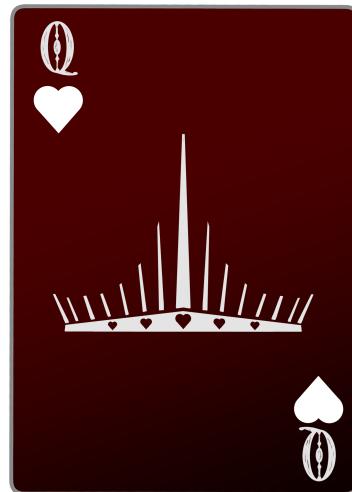


- V1 Scène de la salle de poker -

Les cartes de poker, un élément central du gameplay, avaient également été modélisées. Leur design allie simplicité et esthétique, tout en reflétant le thème global du jeu. Ces cartes jouent un rôle crucial dans le développement de l'histoire et des interactions entre les joueurs.



- Carte : As de trèfle -



- Carte : Reine de cœur -

Pour les animations, une première version fonctionnelle avait été implémentée pour le joueur. Les animations couvrant toutes les directions de déplacement (avant, arrière, gauche et droite), et s'adaptant dynamiquement à la vitesse du joueur. Cette flexibilité ont permis de rendre les déplacements plus fluides et réalistes dès la S1, contribuant à une expérience immersive dès les premières phases du développement.

Ces étapes ont posées les bases d'un univers visuellement cohérent, tout en laissant de nombreuses opportunités d'améliorations et de raffinements pour les prochaines étapes du développement que nous avons saisi pour prolonger le travail au cours de cette S2.

Exclusivités en S2

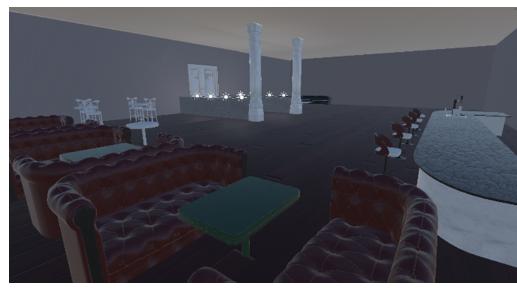
Depuis la dernière soutenance, quelques avancées ont été réalisées dans la direction artistique du projet. L'accent a été mis sur le développement des environnements, l'amélioration des effets visuels et l'intégration d'éléments immersifs permettant de se projeter un peu plus au sein du jeu.

Développement des premiers environnements

Un travail initial sur la conception des environnements a été entrepris, avec la modélisation d'une première maison servant de zone jouable. Cette maison constitue une des scènes de l'aventure du joueur. La scène d'introduction a elle aussi commencer à être élaboré avec des éléments pour l'instant évoquant un environnement proche du bar. L'objectif a été de poser les bases réelles de la direction artistique et de définir une atmosphère cohérente avec le ton du jeu. Désormais l'avancée sur ces scènes sera plus rapide et efficace.



- Scène de la maison -



- Scène d'introduction -

Ces différentes améliorations permettent de faire évoluer les perspectives du jeu vers des éléments concrets et cohérents avec l'univers du jeu. L'objectif suivant sera de peaufiner les détails et d'intégrer progressivement les nouveaux assets en fonction des besoins du level design.

4.3 Intelligence artificielle

Présenté en S1

Le développement de l'intelligence artificielle avait avancé lors de la dernière étape de manière progressive, mais il avait déjà à l'époque rencontré certains retards en raison de la complexité technique et de la nécessité de coordonner cette partie avec d'autres aspects du projet. Ces retards étaient dus à des imprévus dans l'intégration des mécaniques et des interactions avec les PNJ. Cependant, quelques avancées avaient été réalisées sur des fonctionnalités spécifiques et essentielles (en majorité relatives au jeu de poker).

Pour cet aspect de l'IA donc, de nombreuses fonctions individuelles avaient été codées mais non implémentées in-game. Cela incluait notamment la gestion du mélange des cartes, leur distribution équitable mais aléatoire aux joueurs, ainsi que la détection des mains de chaque joueur ou PNJ [6]. Ces fonctionnalités ont posé les bases du développement ultérieur, en veillant à rendre les adversaires réalistes et compétitifs, tout en offrant des interactions engageantes pour le joueur.

- Script de gestion des mises -

- Script de gestion du Deck -

Bien que l'IA pour les interactions avec les ennemis ou les événements paranormaux étaient encore en phase de réflexion et de prototypage, les avancées dans le domaine du poker avaient démontré une compréhension initiale des mécaniques nécessaires pour rendre le gameplay riche et interactif. Ces efforts permettront, à terme, de garantir une expérience immersive et cohérente dans l'ensemble des niveaux du jeu.

Exclusivités en S2

L'intelligence artificielle a connu plusieurs avancées, notamment sur le développement du pathfinding des ennemis. Ce travail est essentiel pour un niveau en cours de conception, où le joueur devra fuir une entité dans un labyrinthe.

Développement du pathfinding et conception du labyrinthe

Un prototype de pathfinding a été mis en place en utilisant le NavMesh d'Unity, permettant aux entités ennemis de se déplacer dynamiquement dans un environnement complexe. Ce système est destiné à gérer les mouvements d'un ennemi poursuivant le joueur dans un niveau de type labyrinthe.

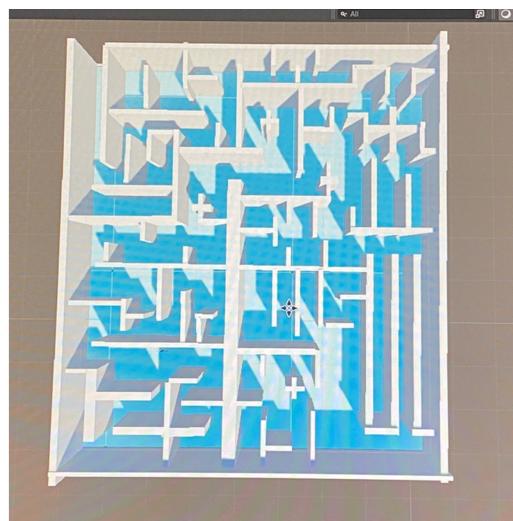
La première phase a consisté à modéliser le terrain, en positionnant les murs et les pièges du labyrinthe. Avant cette étape, plusieurs plans ont été dessinés afin d'obtenir un tracé équilibré en difficulté et challenge pour le joueur.

Correction des problèmes du NavMesh

Lors de l'implémentation du NavMesh, plusieurs problèmes ont été rencontrés, notamment une mauvaise attribution des surfaces, ce qui empêchait l'IA de se déplacer correctement. Après plusieurs ajustements, le terrain fonctionne désormais correctement et est prêt à être utilisé pour les tests de gameplay.



- NavMesh mal configuré -



- NavMesh bien configuré -

Ces avancées posent les bases du système de poursuite des ennemis, qui sera affiné au fil des prochaines semaines de développement. L'objectif final est en effet d'implémenter un comportement de traque dynamique, où l'ennemi pourra adapter ses mouvements et stratégies de poursuite en fonction des actions du joueur. Toutefois, ce développement est encore en phase d'expérimentation et nécessite plusieurs itérations avant son intégration définitive.

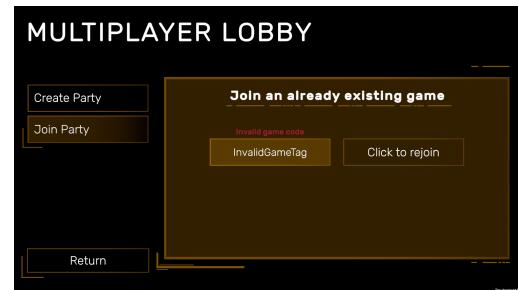
4.4 Multijoueur

Présenté en S1

Le mode multijoueur de *We Don't See Sharp* avait dès lors de la S1 atteint un stade pleinement opérationnel, offrant une synchronisation en ligne efficace grâce à l'intégration de Photon [21], une solution API [22] spécialisée pour les jeux multijoueurs. Contrairement à une configuration LAN [23] traditionnelle, le jeu permet aux utilisateurs de se connecter facilement via Internet, garantissant une accessibilité accrue et une expérience fluide pour les joueurs, quel que soit leur emplacement. Cette mise en œuvre a marqué une étape clef du développement.



- Interface de création de salon -



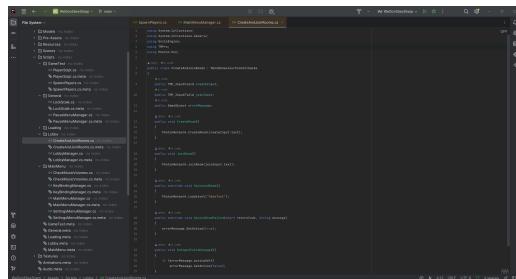
- Interface de connexion à un salon -

Le processus de connexion est géré par un script dédié, assurant la liaison initiale au serveur Photon [21] et dirige les joueurs vers un lobby commun une fois connectés. Ce système garantit une transition fluide et élimine les interruptions potentielles. D'autres scripts facilitent quant à eux la création et la gestion des salles de jeu, permettant aux joueurs de rejoindre ou de créer leurs propres sessions. Chaque salle offre un environnement cohérent pour les interactions multijoueurs.

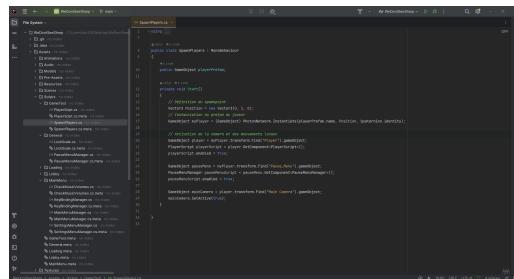


- Écran de chargement lors de la connexion au réseau Photon -

Une fois dans une partie, le programme assure automatiquement l'instanciation des avatars des joueurs sur le réseau. Chaque joueur étant équipé de contrôles locaux, avec une caméra dédiée et des menus spécifiques, tout en étant synchronisé avec les autres participants. Cette configuration garantit que les mouvements, interactions et actions locales soient visibles pour tous les joueurs en temps réel, renforçant la coopération et l'immersion dans le jeu.



- Script de mise en réseau des joueurs -



- Script d'instanciation des joueurs -

De plus, des éléments comme la gestion des entrées des joueurs ou la synchronisation des interactions environnementales et des animations ont été soigneusement intégrés. Ces fonctionnalités assurent une expérience stable et immersive, avec un équilibrage optimal entre performance et précision réseau. Malgré les défis techniques, l'équipe a réussi à créer une base multijoueur solide, qui servira de socle pour les futures améliorations et ajouts qui demeureront sans aucun doute plus complexes et nombreux.

Ces avancées ont démontré l'engagement déjà présent fin 2024 pour fournir une expérience multijoueur de haute qualité, essentielle pour captiver les joueurs et garantir le succès du jeu. Le multijoueur est non seulement une fonctionnalité opérationnelle, mais aussi un pilier central du gameplay collaboratif de *We Don't See Sharp*.

Exclusivités en S2

Le mode multijoueur a bénéficié de plusieurs améliorations visant à optimiser la gestion des animations, la synchronisation des objets interactifs et la cohérence de l'environnement en réseau. Ces mises à jour garantissent une meilleure stabilité et fluidité du jeu en ligne.

Optimisation de la gestion des animations sur réseau

Les animations des joueurs sont désormais synchronisées via un système simplifié de transmission de données en réseau. Plutôt que d'envoyer l'état de chaque animation en continu, elles sont gérées par un flux de requêtes en écriture/lecture, permettant de limiter la charge réseau et d'optimiser la réactivité des mouvements.

Intégration et synchronisation des objets interactifs

Tous les objets interactifs mentionnés dans la partie *Mécaniques de Gameplay* ont été intégrés dans l'environnement multijoueur. Ils sont instanciés dynamiquement et synchronisés entre les joueurs, garantissant une interaction fluide et cohérente, quel que soit le nombre de participants.

Gestion des états d'environnement et persistance des objets

Un système de mémoire des états a été mis en place afin d'assurer une cohérence globale des éléments du jeu en multijoueur. Chaque objet interactif essentiel dispose désormais d'un état enregistré en mémoire, permettant :

- De garantir la continuité des interactions, même en cas de déconnexion/reconnexion d'un joueur.

- D'assurer qu'un joueur rejoignant une partie en cours récupère immédiatement l'état actuel des objets et de l'environnement.

Ce fonctionnement repose sur l'utilisation de requêtes *Remote Procedure Call (RPC)* [24], qui servent de mémoire tampon et assurent la transmission des états en temps réel dès qu'un nouvel utilisateur rejoint la session.

Ces mises à jour permettent d'améliorer significativement l'expérience multijoueur, en rendant les interactions plus naturelles et en assurant une gestion robuste des objets et de l'environnement en réseau.

5 Bilan intermédiaire

5.1 Avancées réalisées depuis la S1

Divers progrès ont été accomplis dans plusieurs domaines du projet depuis la dernière soumission, permettant de faire le lien entre les bases posées il y a plusieurs mois et le futur rendu de mai.

Léo DESCHANEL-LOSCIUTO

- Mise à jour du système multijoueur avec l'optimisation de la synchronisation des animations et des objets interactifs.
- Intégration d'un état mémoire pour la gestion des objets en réseau, permettant une transmission cohérente en cas de reconnexion d'un joueur.
- Ajout de nombreuses interactions environnementales avec des objets instanciés et synchronisés entre les joueurs.
- Développement d'un menu in-game et d'un système d'inventaire avec gestion des objets récupérables, sans HUD [25] pour l'instant.

Laetitia BERKOUNE

- Travail sur la scène d'introduction, première interactions et applications de modèles/textures.

Emma RUCAY

- Développement du système de navigation IA basé sur le NavMesh d'Unity.
- Modélisation et intégration d'un niveau de type labyrinthe, destiné à une séquence de poursuite.
- Résolution de problèmes liés à l'application du NavMesh aux surfaces correctes, empêchant l'IA de marcher sur les murs.

5.2 Défis rencontrés et retards, solutions et/ou explications

Léo DESCHANEL-LOSCIUTO

- **Multijoueur :** Problèmes initiaux liés à la synchronisation des objets suite à une connexion tardive. Solution apportée via un système d'état mémoire et l'utilisation de RPC pour garantir une transition fluide lors des connexions.

- **Inventaire** : Difficultés avec les Canvas Unity et les différentes interactions sur le plan comme les boutons dont les hitbox [26] évoluent selon la résolution de son écran. Solutions toujours en cours de recherche pour certains cas mais éviter pour d'autres comme avec le cadenas à code qui utilise de fait un raycast.
- **Collaboration** : Inégalité de travail au sein du groupe, en tant que chef de projet et malgré toute la volonté du monde et toutes les méthodes (donner des instructions claires, donner des deadlines, laisser les membres se gérer...), aucune ne semble fonctionner pour mettre au pas ce groupe ce qui cause des différents entre les membres. Pour l'instant aucune réelle perspective de coordination stable n'est à l'horizon bien que ce soit l'objectif évidemment.

Adrien GOMES

- **Retard général** : Aucun travail effectué sur les tâches assignées depuis la dernière soutenance.

Laetitia BERKOUNE

- **Assistance sur IA** : Retard vis à vis des IAs sur lesquelle elle aide les responsables respectifs.

Emma RUCAY

- **NavMesh** : Difficultés techniques rencontrées lors de l'application du pathfinding sur le bon calque de surface, nécessitant des expérimentations et corrections.
- **Problèmes de dépôt Git** : Retard dans l'intégration du travail effectué sur l'IA à cause de problèmes liés au push des fichiers.

Doria BOUZIANE

- **Retard général** : Aucun travail effectué sur les tâches assignées depuis la dernière soutenance.

5.3 Comparaison des progrès par rapport aux objectifs initiaux

Objectifs atteints

- **Multijoueur** : Fonctionnalités opérationnelles et poussées ce qui permettra de lacher du lest sur la dernière ligne droite de la conception du jeu. Instanciation [27], synchronisation [28] et optimisation, le réseau fonctionne correctement en très large globalité.
- **Déplacement des joueurs** : Fonction conformes aux attentes fixés et la large personnalisation disponible ne fait que rendre les déplacements/intéractions plus immersifs.
- **Gestion GUIs** : Les menus les plus lourds ont déjà une version fonctionnelle réalisée, léger retard sur les environnements in-game comme l'inventaire par exemple mais très largement passable.
- **IA Ennemis** : Après un départ hasardeux la voie se dégage pour cette fonctionnalité en passe d'être implémenté à l'ensemble du jeu jusqu'alors déjà produit.
- **Web et promotion** : Site web adapté et fonctionnel à ce jour.

Écarts et ajustements nécessaires

- **Interactions environnementales :** Manque de coordination et de leading sur cette tâche, trop peu de fonctionnalités encore.
- **IA Poker :** La vérification, la correction et l'implémentation du jeu de carte interactif dans le jeu ne sont pas encore faites.
- **Graphismes :** La refonte des menus et la participation à l'élaboration des menus in-game est impérative pour tenir les délais.
- **Conception des espaces :** Retard général car aucune scène n'est vraiment crédible à l'heure actuelle.

En conclusion, contrairement à ce que nous avions présumer dans la synthèse de notre dernier rapport, sur cette période de très nombreux retards pour diverses raisons ont freiné l'avancée du projet, et ralenti sa croissance par rapport à la dernière période de travail. Toutefois certaines parties sont parvenues à être poursuivis en se rapprochant le plus possible des délais fixées à l'origine. Le travail se poursuit et après une étape en cou de mou, l'ensemble de l'équipe souhaite se ressaisir pour parvenir à fournir en fin d'année un jeu fonctionnel, complet et à la hauteur de leurs attentes.

6 Perspectives pour le rendu final

6.1 Fonctionnalités en cours de développement

L'équipe se concentre actuellement sur plusieurs fonctionnalités afin de produire un ensemble satisfaisant et permettre le cheminement vers les dernières étapes avant la fin du projet :

- **Finalisation de l'IA des ennemis :** Amélioration du pathfinding et des comportements hostiles pour une meilleure réactivité et adaptation au joueur.
- **Système d'interaction avancé :** Ajout de nouvelles interactions avec l'environnement et optimisation des scripts existants pour une meilleure immersion.
- **Équilibrage du mode multijoueur :** Ajustement des synchronisations réseau et optimisation des requêtes pour garantir une meilleure fluidité de l'expérience en coopération.
- **Amélioration de la scène de départ :** Affinement du design du club de poker et premières intégrations d'intégrations à la scène déjà entamé, trappe, bouteille...

6.2 Objectifs pour la dernière soutenance

Actuellement, nos objectifs sont définis comme étant :

- **Système de sauvegarde et persistance des données :** Implémentation d'un mécanisme de sauvegarde permettant de conserver la progression du joueur entre les sessions.
- **Version jouable complète du mode solo :** L'objectif est d'avoir une expérience solo intégrant les mécaniques principales et un jeu abouti.
- **Ajout des mécaniques finales :** Finaliser toutes les mécaniques de gameplay essentielles pour rendre le jeu interactif et cohérent.
- **Refonte et finalisation des environnements :** Mise à jour des scènes principales avec davantage de détails graphiques et d'ambiance sonore pour améliorer l'immersion.

- **Tests et correction des bugs :** Phase intensive de tests internes pour identifier et corriger les principales erreurs impactant le gameplay.
- **Intégration de l'IA de poker :** Développement et implémentation sous Unity des fonctions et systèmes d'interactions pour le jeu de carte.

6.3 Potentiels ajustements ou améliorations prévues

Afin d'adapter le projet aux besoins émergents et aux retours obtenus, plusieurs ajustements sont envisagés :

- ★ **Réajustement des mécaniques de peur et d'horreur :** Modification du rythme et de l'intensité des événements paranormaux pour optimiser la tension.
- ★ **Équilibrage des puzzles et interactions :** Vérification de la difficulté des énigmes pour offrir une expérience cohérente et engageante.
- ★ **Optimisation des performances :** Amélioration des chargements et gestion des ressources pour éviter les ralentissements, notamment en multijoueur.
- ★ **Affinement du sound design :** Intégration de nouveaux effets sonores et musiques pour enrichir l'expérience immersive du jeu.
- ★ **Implémentation d'une première version du HUD :** Ajout d'un affichage minimaliste pour accompagner le joueur dans sa progression sans dénaturer son expérience.

Ces ajustements permettront de maintenir un haut niveau de qualité tout en respectant les contraintes de temps et de ressources, tout en assurant une expérience engageante pour les joueurs et un projet réussi pour l'équipe.

7 Conclusion

La conclusion de ce rapport marque un nouveau point d'étape pour l'équipe dans le développement de *We Don't See Sharp*, en soulignant les progrès réalisés, les défis affrontés et les enseignements tirés tout au long de cette phase intermédiaire du projet.

Synthèse de l'avancement

Le projet a avancé dans divers domaines tels que :

- ★ **Multijoueur :** Requêtes de synchronisation en temps réel et en mémoire tampon.
- ★ **Déplacement du joueur :** Refonte du système à la source pour optimiser les performances notamment sur le sprint.
- ★ **Interactions environnementales :** Premières réelles implantations fonctionnelles d'éléments interactifs dans une scène.

Site Web

Le projet possède également son propre site web accessible avec l'URL suivant : <https://bit.ly/WeDontSeeSharp>. Le site propose des sections de références au projet, de téléchargement du jeu ou encore des sections de documentations.



- Vue de l'interface du site web -

Leçons apprises

Le développement de *We Don't See Sharp* en S1 et S2 a permis à l'équipe d'acquérir de nombreuses compétences et de tirer des enseignements précieux :

- **Gestion de projet :** L'importance d'une méthodologie agile pour coordonner les efforts de l'équipe, prioriser les tâches et s'adapter aux imprévus a été confirmée. La plateforme Trello [5] s'est révélée efficace pour attribuer les tâches et suivre leur avancement.
- **Collaboration :** Les défis techniques et créatifs ont mis en lumière la nécessité d'une communication constante et claire entre les membres de l'équipe, favorisant un partage d'idées constructif. Ce qui a parfois du mal à être intégré réellement au sein du groupe.
- **Technologies :** L'apprentissage et la maîtrise d'outils tels que Unity3D [1], Blender [29], et Photon [21] ont renforcé les compétences techniques des membres s'étant le plus investis.
- **Résolution de problèmes :** Les défis rencontrés, comme les bugs liés au multijoueur ou les difficultés dans la modélisation 3D, ont permis de développer des compétences en recherche de solutions et en itération rapide.

Apports pédagogiques

Ce projet a offert une opportunité exceptionnelle de mettre en pratique les compétences techniques et organisationnelles acquises au cours de notre formation :

- ⇒ **Programmation avancée :** L'utilisation de C# pour les scripts et le développement des IA a permis de consolider nos bases en programmation tout en explorant des concepts avancés comme le pathfinding [30] ou encore la synchronisation et les requêtes réseau.
- ⇒ **Design de jeu :** La conception des niveaux et des mécaniques de gameplay a offert une perspective enrichissante sur l'importance de l'expérience utilisateur et de l'esthétique dans le développement vidéoludique.
- ⇒ **Innovation :** La réflexion visant à l'intégration de nos idées en jeu, comme les interactions avec l'environnement, des interfaces ou sur le fonctionnement pur de système intégré comme l'IA, a stimulé notre créativité et notre capacité à réfléchir efficacement.

Perspectives

À ce stade, le projet *We Don't See Sharp* est vers la bonne voie mais nécessite inévitablement une participation de chacun des membres de manière égale. La dernière étape est la plus

importe et nous allons donc devoir affiner nos idées une dernière fois en comparant nos capacités observées depuis septembre et nos attentes personnelles, optimiser les performances et enrichir l'expérience utilisateur...

L'innovation et la persévérance formeront les piliers de notre approche, tandis que nous continuons à transformer notre vision en une réalité captivante pour les joueurs. Ce projet représente une opportunité unique d'apprendre, de créer et de dépasser nos propres attentes.

Annexes

Cahier des charges fonctionnel

Contraintes techniques imposées et/ou retenues :

- Plateforme cible : PC sous Windows.
- Moteur de jeu : Unity3D ^[1] avec programmation en C#.
- API multijoueur : Photon ^[2].
- Optimisation des performances : gestion de la mémoire, utilisation de LODs ^[3] pour les modèles 3D.

Ressources utilisées

Outils et Logiciels

- ★ Unity3D : Utilisé pour le développement global (jeu, IA, multijoueur).
- ★ Blender : Modélisation et animation 3D pour les environnements et objets interactifs.
- ★ Photon : Gestion du multijoueur en ligne.
- ★ Photoshop : Retouche graphique pour les textures et les interfaces.
- ★ GitHub Desktop : Outils de mise en réseaux des fichiers pour programmer de manière collaborative.
- ★ Make Human : Logiciel de création de personnage 3D avec des modèles définis.
- ★ Mixamo : Ressource utilisé afin de concevoir des animations applicables aux modèles de joueur.

Licences

- ★ Unity3D : Version étudiante gratuite.
- ★ Photoshop : Licence annuelle à 232,84 €.
- ★ Jetbrains Rider / VS Code : Version étudiante gratuite.
- ★ Photon : Licence gratuite pour les petites équipes.
- ★ Chat-GPT 4-0 : Licence Plus mensuelle à 20 \$.
- ★ GitHub : Version étendue en mémoire pour étudiants gratuite.

Bibliographie

- ★ Tutoriels Unity : Documentations officielles et vidéos en ligne (YouTube).
- ★ Blender : Guides en ligne pour modélisation 3D (forum, Chat-GPT, YouTube...).
- ★ Photon API : Documentation pour développement multijoueur sur le site même.

Références

- [1] Moteur de jeu polyvalent permettant de créer des jeux en 2D et 3D. Utilisé pour le développement de *We Don't See Sharp*, il offre des outils pour le rendu graphique, les animations, la physique, et les interactions en temps réel.
- [2] Date actuelle du projet : *07 Mars 2025*.
- [3] Date de la dernière soutenance : *Mai 2025*.
- [4] Plateforme de gestion de versions et de collaboration pour le développement de logiciels. Permet de suivre les modifications du code, de collaborer en équipe et de gérer les dépôts (repositories). Utilisé pour centraliser les fichiers et les scripts du projet.
- [5] Trello est un outil de gestion de projet basé sur la méthode Kanban, qui repose sur l'utilisation de colonnes et de cartes pour visualiser et organiser les tâches à réaliser, leur progression et leur achèvement.
- [6] Accronyme : Personnage Non-Joueur (*personnage du jeu ayant une apparence semblable à celle du véritable joueur ou bien simplement une entité indépendante mais qui n'est pas directement contrôlable par le joueur, donc soit par une IA soit par un comportement alternatif et/ou régulier*).
- [7] Univers fictif d'horreur basé sur un concept d'espaces interminables, labyrinthiques, souvent associés à une ambiance oppressante et des dangers inconnus. Inspiré des creepypastas et utilisé comme cadre dans certains jeux d'horreur.
- [8] Accronyme : Intelligence artificielle (*ayant la capacité de participer à une partie de poker*).
- [9] Environnements fonctionnels mis en place à ce jour : GitHub (*service de mise en réseau de fichiers partagés*) et Trello (*KANBAN : système de gestion méthodologique agile par attribution de tâches spécifiques*).
- [10] Documents réalisés à ce jour : CDCT (*cahier des charges technique*), CDCF (*cahier des charges fonctionnel*), PDS (*plans de soutenance*), et RDS1-2 (*rapport de soutenance S1 et S2*).
- [11] Accronyme : Graphical User Interface (*éléments visuels interactifs*) ≠ UI (*User Interface*).
- [12] Ensemble des éléments relatifs : Menus (*fonds, boutons, icônes...*), Interfaces (*Inventaires, Barre de vie, Barre de stamina...*), et Textures (*dessins, cartes, éléments de décor...*)
- [13] Art de concevoir et d'adapter les espaces du jeu afin de refléter la vision des concepteurs, tout en préservant l'autonomie et la liberté du joueur, subtilement guidé par des détails et des mécanismes ingénieux qui orientent son exploration vers les objectifs souhaités.
- [14] Accronyme : Intelligence artificielle (*ayant la capacité de participer de se déplacer en autonomie (pathfinding) et d'interagir avec le joueur (dégats, conversations...)*)
- [15] Accronyme : Sound Effects (*effets sonores utilisés dans un jeu vidéo pour renforcer l'immersion, comme des bruits d'environnement, des interactions, ou des sons liés aux actions du joueur ou des entités du jeu*).
- [16] Élaboration d'une vision globale de la trame narrative et des éléments de gameplay à développer, en harmonie avec le lore, qui enrichira et complétera l'histoire pour renforcer son caractère immersif.
- [17] Sauvegarde de données intrinsèque à l'ordinateur sur lequel le jeu est sauvegardé, même après suppression du jeu. Accessible depuis l'éditeur de registre et modifiable par valeurs binaires à l'emplacement :
`HKEY_CURRENT_USER\Software\Unity\UnityEditor\FlushHandStudio\WeDontSeeSharp.`

- [18] Composant de Unity permettant de simuler la physique d'un objet 3D ou 2D. Il permet à un objet d'être affecté par la gravité, les forces et les collisions avec d'autres objets physiques. Utilisé notamment pour les déplacements réalistes des entités dans un jeu.
- [19] Composant Unity permettant de gérer le déplacement d'un personnage sans utiliser directement la physique du moteur. Contrairement au *Rigidbody*, il ne subit pas les forces comme la gravité par défaut et offre un contrôle plus précis sur les mouvements du joueur.
- [20] Technique utilisée en programmation de jeux pour envoyer un rayon invisible depuis un point donné dans une direction spécifique. Permet de détecter les objets qu'il rencontre, souvent utilisé pour les interactions avec l'environnement ou les mécaniques de tir.
- [21] API de mise en réseau spécialisée pour les jeux multijoueurs. Elle permet de gérer la communication en temps réel entre les joueurs, la création de salles (rooms), et la synchronisation des interactions dans des environnements en ligne.
- [22] Accronyme : Application Programming Interface (*ensemble de règles, de protocoles et d'outils qui permet à différentes applications ou logiciels de communiquer entre eux en définissant les moyens par lesquels les composants logiciels interagissent et offre des fonctionnalités préétablies pour simplifier le développement et l'intégration d'applications, comme la mise en réseau dans un jeu vidéo par exemple*).
- [23] Accronyme : Local Area Network (*réseau local permettant à des ordinateurs ou appareils situés à proximité de communiquer entre eux. Dans le contexte des jeux vidéo, un LAN est souvent utilisé pour des parties multijoueurs locales, ne nécessitant pas d'accès à Internet*).
- [24] Accronyme : Remote Procedure Call (*appel de procédure à distance*). Utilisé dans les jeux multijoueurs pour exécuter une fonction sur plusieurs clients ou serveurs, permettant de synchroniser des actions entre joueurs via le réseau.
- [25] Accronyme : Heads-Up Display (*affichage tête haute*). Interface graphique visible en permanence sur l'écran, affichant des informations essentielles au joueur comme la barre de vie, la jauge d'endurance, la carte, ou les objectifs.
- [26] Zone invisible associée à un objet ou un personnage dans un jeu vidéo, utilisée pour détecter les collisions. Les *hitboxes* servent à déterminer si un projectile touche un ennemi, si un joueur subit des dégâts, ou si une interaction est possible avec une interface par exemple.
- [27] Terme utilisé en programmation et dans Unity pour créer dynamiquement des objets pendant l'exécution du jeu. La fonction *Instantiate()* permet de générer des copies d'objets ou de préfabriqués (Prefabs) dans une scène.
- [28] Processus par lequel plusieurs éléments d'un jeu (joueurs, objets, animations, variables, etc.) sont maintenus à jour et cohérents entre différents clients en multijoueur. Dans Unity, cela se fait via Photon, des RPC, ou des scripts de mise à jour réseau.
- [29] Logiciel open-source de modélisation, d'animation et de rendu 3D. Utilisé pour créer des assets graphiques, des environnements et des animations pour les jeux vidéo. Il offre une flexibilité et une puissance adaptées aux projets professionnels comme amateurs.
- [30] Technique utilisée en informatique et en développement de jeux vidéo pour trouver le chemin optimal entre deux points dans un environnement, souvent utilisé pour les déplacements des IA dans un jeu.
- [31] Accronyme : Level of Detail (*Technique d'optimisation graphique qui adapte dynamiquement la qualité des modèles 3D en fonction de la distance par rapport à la caméra, afin d'améliorer les performances sans compromettre la qualité visuelle perçue*).