

Rapport de Soutenance

Elderberried



Table des matières

1	Introduction	3
2	Reprise du cahier des charges	4
2.1	Notre entreprise et notre équipe	4
2.1.1	Notre entreprise	4
2.1.2	Notre équipe	5
2.2	Notre Projet	6
2.3	Répartition des tâches	7
3	Avancement et Planification	8
3.1	L'histoire	8
3.2	L'architecture	8
3.2.1	Utilité de l'architecture du projet	8
3.2.2	Base	8
3.2.3	Mouvements	9
3.3	Entités	10
3.3.1	Autre comportements	11
3.3.2	Problèmes rencontrés	11
3.4	Les Minis Jeux	11
3.4.1	Règles du jeu	12
3.4.2	TankMinigame	12
3.4.3	PushMinigame	13
3.4.4	Miner game	13
3.4.5	Spatialship game	13
3.4.6	Et bien d'autres	14
3.5	L'interface utilisateur	14
3.5.1	Menu de connexion multijoueur	14
3.5.2	Sélection de mini-jeux	14
3.5.3	Paramètres généraux	15
3.5.4	La couleur de la Team	15
3.6	Le multijoueur	16
3.6.1	Netcode for GameObjects	16
3.6.2	Mirror	16
3.6.3	Installation du serveur et des clients	17

3.6.4	Installation du système de Team sur le serveur . .	17
3.6.5	Chargement des joueurs entre les scènes	18
3.6.6	Synchronisation des Input	18
3.7	L'intelligence artificiel	19
3.8	Les graphismes	19
3.8.1	Création de base des personnages	19
3.8.2	Conception des boutons	20
3.8.3	Création du décor	21
3.8.4	Résolution et Pixel Perfect Camera	21
3.8.5	Shader Graph	22
3.8.6	Perspectives futures	23
3.9	Le son	24
3.9.1	La musique	24
3.9.2	Les effets sonores	24
3.10	Le site web	25
4	Conclusion	27
5	Annexe	28

1 Introduction

Dans le cadre de notre projet de la première année en classe préparatoire à l'EPITA, nous avons entrepris le développement d'un jeu vidéo intitulé "Another Recursion". Ce projet a marqué notre initiation dans le domaine complexe et fascinant du développement de jeux vidéo, notamment en utilisant Unity3D et le langage de programmation C. Avant le début de ce projet, notre expérience en matière de création de jeux en ligne et de programmation sur Unity était quasiment inexistante, nous plaçant ainsi devant un défi de taille dès le commencement de notre travail.

Durant les premiers mois, notre travail a principalement consisté à acquérir les connaissances de base nécessaires à la réalisation de notre projet. Nous avons consacré une grande partie de notre temps à l'apprentissage de Unity, un moteur de jeu puissant mais complexe, et à la compréhension des principes fondamentaux du développement de jeux en ligne. Malgré les obstacles rencontrés, notre équipe a réussi à mettre en place les éléments essentiels du jeu, notamment un menu fonctionnel, un système de multijoueur opérationnel, et les premiers mini-jeux qui forment la base de "Another Recursion".

Ce rapport de première soutenance décrit en détail les progrès réalisés depuis la validation du cahier des charges, mettant en lumière à la fois les avancées significatives et les retards éventuels par rapport au planning initial. Nous y présentons également la répartition des tâches au sein de l'équipe, illustrant comment chaque membre a contribué à l'avancement du projet. Ce rapport vise à fournir une vue d'ensemble des efforts déployés par notre équipe pour transformer notre vision initiale en une réalité, malgré un démarrage avec des connaissances limitées dans le domaine. Il souligne notre engagement envers le projet "Another Recursion" et notre détermination à poursuivre le développement du jeu en vue de créer une expérience ludique mémorable pour les joueurs.

2 Reprise du cahier des charges

2.1 Notre entreprise et notre équipe

2.1.1 Notre entreprise

Notre entreprise a pour nom "Elderberried".
Le choix de ce nom a été motivé par notre intérêt partagé pour le sureau, poussant notamment en Alsace et dans les Vosges, et par son fruit au goût amer.

Le choix du jeu de mot avec "Buried" a simplement été facilité par sa ressemblance avec le mot "Berry". Le choix peut également être vu comme un jeu de mot avec le mot "elder" (âgé en anglais), qui associé avec la jouvence d'une plante créerait cette image tout public que notre jeu veut donner.

Notre entreprise est spécialisée dans le développement de jeux-vidéos pour des clients. Cela veut dire qu'un client nous imposera un cahier des charges comprenant des critères, que l'on devra respecter en développant un jeu qui remplira les attentes du commanditaire, puis nous le leur vendons ensuite pour un prix prédéfini. Les commanditaires se feront un bénéfice à la vente des copies aux joueurs.

Elderberried est un jeu de mot entre le sureau, prononcé "Elderberry tree" en anglais, et l'action d'être enterré, "Buried", notre nom ferait donc référence au fait d'être enterré sous du sureau, d'où notre logo.



2.1.2 Notre équipe



FIGURE 1 –
Thomas.P



FIGURE 2 –
Asmae.H



FIGURE 3 –
Thomas.H



FIGURE 4 –
Loïc.V



FIGURE 5 –
Edgar.L

Thomas Paillereau est le leader de notre équipe. Sa passion pour le codage pur, développée après avoir découvert Scratch lors d’une colonie de vacances informatique, fait de lui un expert en gestion de GIT. Son ouverture d’esprit, son empathie et sa clarté dans ses propos font de lui un chef d’équipe efficace.

Asmae Himmi est le pilier de notre équipe. Sa passion pour les jeux vidéo et son expérience dans la création de mods graphiques pour les jeux font d’elle une source précieuse de créativité. Elle est également connue pour son soutien constant et sa capacité à encourager l’équipe.

Thomas Henry, bien qu’il soit plutôt solitaire, s’investit pleinement dans la planification et la répartition des tâches. Son expérience en reverse engineering et en programmation fait de lui un expert en C, ce qui lui confère le rôle de soutien technique au sein de l’équipe.

Loïc Vasile est notre logicien. Sa grande connaissance en création de sites web, sa patience inébranlable et son approche méthodique lui permettent de résoudre les défis les plus complexes. Malgré son manque d’expérience dans la création de jeux vidéo, sa curiosité et sa rigueur lui permettront de découvrir de nouvelles possibilités dans la création de notre jeu.

Edgar Lugnier est un innovateur efficace, à l’aise en solo comme en équipe. Sa grande connaissance des jeux massivement multijoueurs et son expérience dans la création de jeux multijoueurs font de lui un atout précieux pour notre projet. Il se concentrera principalement sur l’aspect multijoueur de notre jeu, tout en apportant une aide créative pour trouver de nouvelles idées.

2.2 Notre Projet

Ce Projet, proposé par EPITA, est un jeu vidéo nommé "Another Recursion" et il s'agira d'un jeu type mini-jeux, impliquant plusieurs niveaux pour chaque mini-jeux mais aussi, une possibilité de régler la difficulté de ceux-ci. Nous allons donc progresser de façon rigoureuse et constante sur notre projet pendant toute sa durée, dans l'optique de créer une expérience plaisante et inoubliable, ce que nous ferons par notre travail, notre motivation et notre rigueur.

Ce jeu sera programmé sur Unity3D, en C#. Il privilégiera l'affrontement en 1 contre 1, en réseau UNET Mirror, et intégrera une IA dans certains mini-jeux, avec des fonctionnalités diverses selon les cas, ainsi que pour choisir l'ordre de mini-jeux à jouer.

Les mini-jeux seront principalement en dimension 2D, avec un changement de perspective selon les cas. Pour rendre le jeu plus fluide, et amusant, les mini-jeux deviendront plus dur avec le temps.

Bien sur, le nom de ce jeu vient du terme "Recursion", un terme de l'informatique grandement relié à notre formation, dont la définition est "qui peut être répété un nombre indéfini de fois par l'application de la même règle".

Le nom du jeu et l'histoire (dont on parlera plus tard) ont donc été influencé par son gameplay, la répétition successive de mini-jeux, sans fin jusqu'à ce qu'un des joueurs gagne.

Ce jeu sera programmé sur Unity3D, en C. Il privilégiera l'affrontement en 1 contre 1, en réseau Netcode, et intégrera une IA dans certains mini-jeux, avec des fonctionnalités diverses selon les cas, ainsi que pour choisir l'ordre de mini-jeux à jouer.

Les mini-jeux seront en dimension 2D, avec un changement de perspective selon les cas. Pour rendre le jeu plus fluide, et amusant, les mini-jeux deviendront plus dur avec le temps.

2.3 Répartition des tâches

Nous avons révisé notre répartition des tâches comparé à notre dernier cahier des charges :

Rôle	Thomas.H	Thomas.P	Asmae	Loïc	Edgar
Design et Son					
Game Design	Responsable	Suppléant			
Graphismes			Responsable	Suppléant	
Style			Suppléant		Responsable
Histoire	Suppléant	Responsable			
Music/SFX	Suppléant		Responsable		
Programmation du jeu					
Logiciel		Suppléant			Responsable
Architecture	Responsable			Suppléant	
Fonctionnalités Principales	Responsable	Suppléant			
Fonctionnalités secondaires			Suppléante	Responsable	
User Interface			Responsable		Suppléant
IA		Responsable		Suppléant	
Multi-joueur					
Multi-joueur /NetCode	Suppléant				Responsable
Site web					
Site Web			Suppléante	Responsable	
Autres					
Manuel (installation)				Responsable	Suppléant
Manuel (utilisation)	Responsable	Suppléant			
GIT		Responsable			Suppléant

3 Avancement et Planification

3.1 L’histoire

Pour un jeu vidéo, l’histoire est un facteur important pour l’expérience du joueur. Cela lui donne l’impression de progresser, malgré une potentielle répétition des activités proposé. Il était donc important d’avoir une histoire intéressante pour captivé l’attention du joueur.

Cependant, nous voulions aussi faire une histoire que l’on peut comprendre et développer facilement, et nous avons décidé de faire référence à nos étude, en parlant de l’intelligence artificielle.

En effet, le scénario du jeu se déroule dans un laboratoire de recherche, où des IA en apprentissage (les joueurs) se battent pour savoir laquelle sera gardé pour continuer son apprentissage. Les IA se battront donc dans divers domaines, prenant la forme de robots, d’engins mécaniques, etc.

3.2 L’architecture

3.2.1 Utilité de l’architecture du projet

La programmation, tout le monde le sait, ça peut vite devenir désordonné. L’utilité des techniques lié à l’architecture dans le code tel que la hiérarchie des dossiers, des classes et la réutilisation de composants et de fonctions est donc crucial dans la réalisation de notre jeu.

Plus complexe encore, le type de **party game** exige une attention toutes particulière à la **modularité** car il rapproche des jeux qui ont des fonctionnements très différents.

3.2.2 Base

Le projet est modulé selon une classe **’NBPlayerBase’** qui fait partit de la "prefab" du joueur, celle ci est également raccorder à un gestionnaire d’input présent dans la class **’InputManager’**.

Tout composant étant nécessaire pour changer l’aspect et le fonctionnement du joueur seront donc directement relié à sa prefab pour éviter les complications.

Un schéma représentant ces compositions est présent en figure *Base Component*.

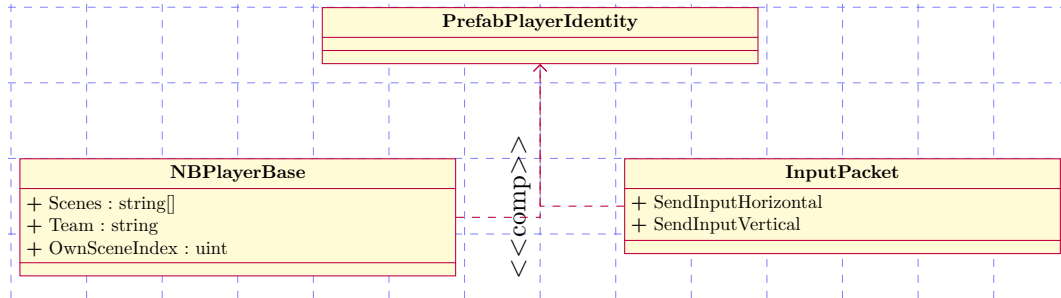


FIGURE 6 – Base Components

3.2.3 Mouvements

La gestion du mouvement se trouve être une des composantes essentielle du mouvement d'un joueur mais pas seulement. Celle-ci peut aussi être rajoutée aux différentes entités présentes dans des jeux.

Un problème se pose lors de l'implémentation de ce genre de fonctionnalité dans unity qui est son système de composition limitant la modularité, car dans un party game le mouvement peut fonctionner de deux différentes manière par rapport au jeu joué (un platformer aura par exemple besoin d'un **RigidBody** par exemple)

Voici donc en figure *Movement Components* une implémentation de classes gérant le mouvement à l'aide d'un **Transform** passé par référence.

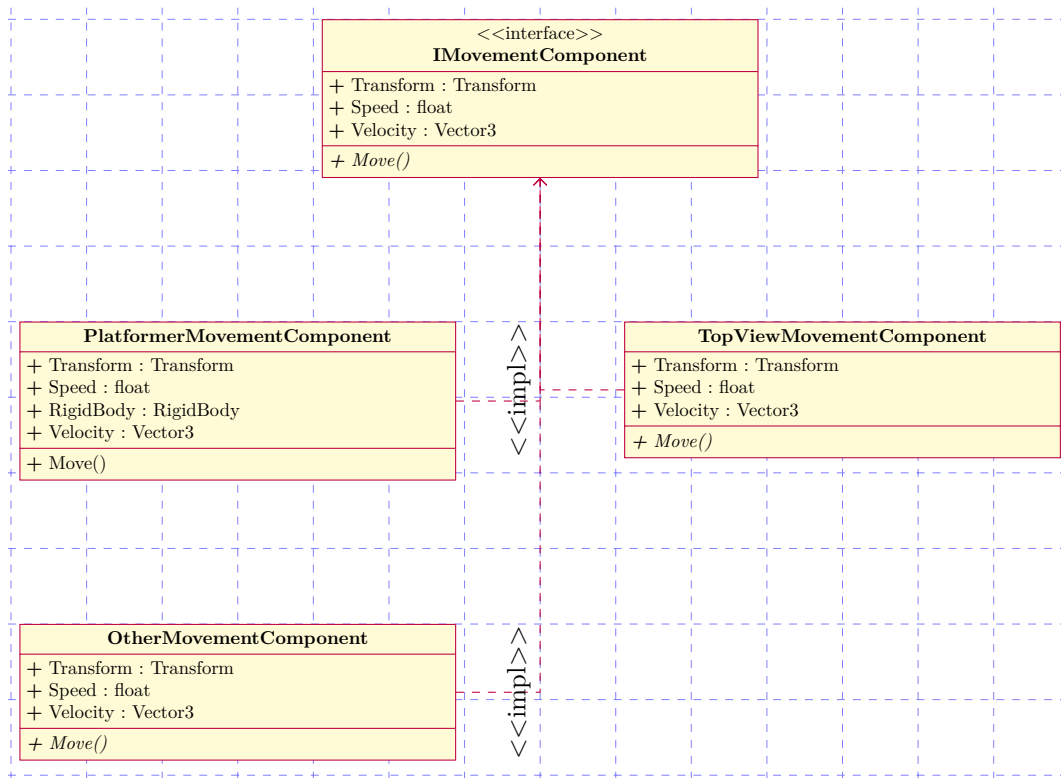


FIGURE 7 – Movement Components

3.3 Entités

Voici un exemple d'implémentation de script de base pour chaque entités :

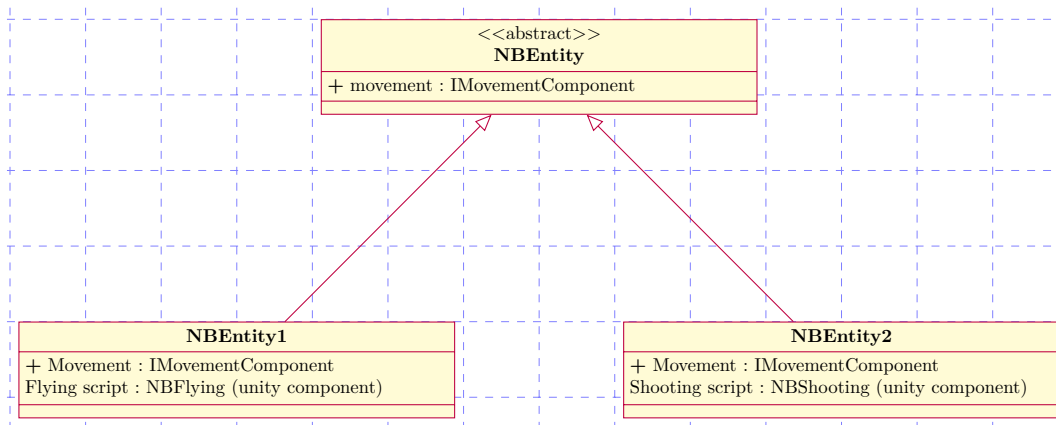


FIGURE 8 – Entity inhéritance

3.3.1 Autre comportements

Les autres comportements pourront également être calqués sur un modèle similaire mais suivront majoritairement une architecture de composant pouvant être ajoutée sur les entités et le joueur de manière intelligentes.

3.3.2 Problèmes rencontrés

Les problèmes rencontrés par rapport au changements d'architecture et de hiérarchie du projet sont majoritairement liés au travail fait chacun de son côté il faudra donc à l'avenir insister sur une hiérarchie plus stricte. Lors du raccord de deux branches les codes ne suivaient pas la même logique il a donc fallu changer quelques composants sur les prefabs. Dans tout les cas beaucoup de refactorisation sera nécessaire à l'avenir.

3.4 Les Minis Jeux

Dans cette partie nous parlerons des règles de jeux qui fonderont notre gameplay et de différents mini-jeux.

3.4.1 Règles du jeu

Les parties se dérouleront par sections de 5 mini jeux, 4 jeux qui permettront aux deux joueurs de se confronter avec la possibilité de perte de vie et 1 mini jeu mélangeant plusieurs mini jeux considéré comme "boss". Les sections de mini jeux seront de plus en plus difficiles et il sera possible de continuer à jouer de manière infinie et ainsi d'augmenter toujours plus la difficulté, notamment le temps autorisé pour réaliser le mini jeu. Certains paramètres comme le choix des mini jeux, le nombre de vie ou la difficulté pourront être choisis par les joueurs.

3.4.2 TankMinigame

Ce jeu est déjà partiellement implémenté sur notre prototype, il consiste en deux tanks représentés par nos deux joueurs. Le but est donc de tirer sur le tank de l'adversaire en évitant les objets. Pour cela le joueur peut utiliser la souris pour déplacer son canon puis le clic gauche pour tirer.

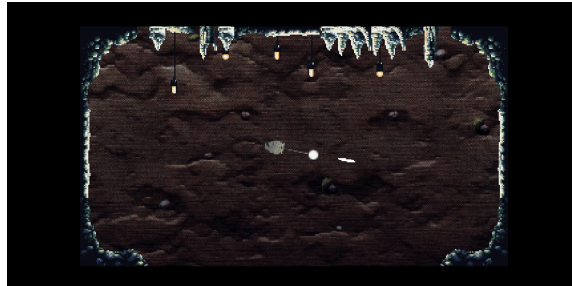


FIGURE 9 – Tank Game

3.4.3 PushMinigame

Ce jeu, pour l'instant non implémenté dans le prototype, est en fait un jeu inspiré de "Motos" un jeu de Namco sortit en 1989 dans lequel le joueur doit pousser des adversaires non joueurs en dehors d'une plateforme pour passer au niveau suivant, ici les deux joueurs devront pousser un certain nombre d'adversaires pour gagner.

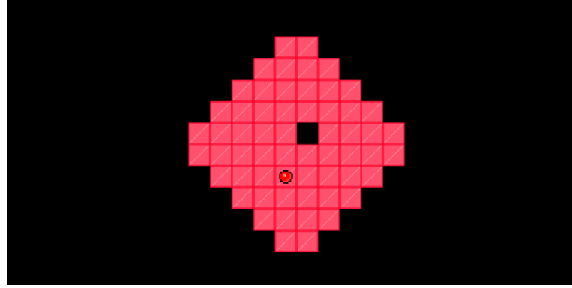


FIGURE 10 – Pusher Game

3.4.4 Miner game

Ce mini jeu se basera sur le concept de miner des blocs contenant des diamants qui sont générés à des emplacements aléatoires. Le joueur ayant récolté le plus de diamants dans le temps imparti gagne la manche. Il sera également possible de frapper le joueur adverse avec sa pioche pour bloquer ses mouvements pendant 1 à 2 secondes.

3.4.5 Spatialship game

Spatialship game est l'un de nos jeu les plus ambitieux, il proposera une phase de profil dans laquelle il faudra tirer des rayons laser avec notre vaisseau sur le boss qui sera une IA. De plus nous pourrons tirer sur le joueur adverse pour bloquer ses mouvements pendant 1 seconde. Par la suite après une phase de profil, le jeu basculera en vue de face et il faudra esquiver les obstacles lancés par le boss tout en tirant sur ses points faibles en cliquant dessus. Il sera également possible de rentrer volontairement dans des obstacles pour obtenir un bonus qui permettra de nuire aux actions du joueur adverse.

3.4.6 Et bien d'autres

Plusieurs autres mini jeux seront également présents qui, comme déjà établi ne se restreindront pas à un seul type de gameplay. Seront présents **point and clicks**, **platformers** et des niveaux **vue du dessus**.

3.5 L'interface utilisateur

3.5.1 Menu de connexion multijoueur

Sur le menu multijoueur, le premier élément est un bouton pour lancer en mode host. Le second est un bouton pour rejoindre en tant que client un serveur, le troisième est une case pour définir l'adresse IP du serveur que l'on veut rejoindre, et le dernier élément est une case pour définir la team du joueur. Lors de la connexion en tant qu'hôte ou en tant que client, les boutons sont bloqués pour empêcher tout type de manipulation durant la connexion qui pourrait créer un bug.



FIGURE 11 – Menu principal

3.5.2 Sélection de mini-jeux

La sélection personnalisée de mini-jeux n'a pas encore été implémentée mais ne saura tarder, elle permettra aux joueurs de sélectionner un ou des niveaux spécifiques avec une difficulté choisie et ainsi de pouvoir jouer dans les mini-jeux qu'ils préfèrent. Elle sera disponible dans le salon de connexion, autrement dit il faudra définir ce paramètre avant de lancer la partie.

3.5.3 Paramètres généraux

Les paramètres généraux du jeu seront implémentés à la fin du projet. Ils permettront de définir des touches personnalisées pour le mouvement, de basculer entre le mode plein écran, écran fenêtré et fenêtre sans bordure. À l'avenir, d'autres idées pourront être établies en accord avec d'autres éléments, comme par exemple un mode pour masquer l'adresse IP du serveur auquel on souhaite se connecter ou encore de mettre par défaut une adresse IP ou une Team.

3.5.4 La couleur de la Team

Les différentes teams associées aux différents joueurs sont décidées par les joueurs lors de leur première connexion au serveur. Dans la case à remplir pour définir la team, nous avons décidé de transformer tout type de texte en code couleur HTML pour permettre de définir la couleur comme le nom de la team.

Ce système de conversion se déroule de la manière suivante. Tout caractère minuscule est traduit en majuscule, tout caractère majuscule de A à F ou nombre est laissé comme telle, tout autre éléments est traduit avec un modulo en un nombre si il est inférieur à la valeur de "A" et est traduit avec un modulo en une majuscule de A à F pour un nombre supérieur à la valeur de "A". À la fin, il est rajouté un "#" au début du string. Il est possible dans la case à remplir de prévisualiser cette couleur pour plus de compréhension.



FIGURE 12 – Case de changement d'équipe

3.6 Le multijoueur

3.6.1 Netcode for GameObjects

Lors de la conception de ce projet, dès la remise du premier cahier des charges, nous nous sommes directement intéressés à l'aspect Multijoueur. En effet, ce projet avait pour condition d'inclure du multijoueur, et s'il n'était pas implémenté en premier, il faudrait ensuite refaire tous le code pour l'y intégrer.

En vue du défi presque impossible qu'est la conception de son propre système multijoueur, nous avons opté pour l'utilisation de "Netcode for GameObjects", un système de multijoueur intégré de base à Unity.

Cependant, ce système ne correspondait pas à notre projet, car en effet, Netcode for GameObjects est un système multijoueur utilisé principalement pour faire des jeux où les utilisateurs doivent coopérer, ne nécessitant généralement pas de gérer les problèmes d'inégalités de connexion entre les joueurs. Il nous fallait donc un système plus optimisé pour le face à face entre joueurs.

3.6.2 Mirror

Il fallait donc recommencer sans système host/client (de Netcode), et nous nous sommes penchés plutôt sur Mirror qui est hérité de UNET et qui était une ancienne technologie de Unity, qui permet d'avoir un système host/client avec un host qui contient le serveur mais également qui est un client, ce qui simplifie grandement le code.

Par ailleurs, la prise en main de Mirror offre également quelque avantages comme par exemple la possibilité de demander directement d'exécuter une fonction sur serveur ou sur tous les clients avec [Command] et [ClientRpc]. Il permet également tout comme Netcode de synchroniser les variables entre le serveur et le client avec [SyncVar] que ce soit du serveur vers les clients ou du client vers le serveur.

3.6.3 Installation du serveur et des clients

L'installation du système multijoueur n'a pas été une tâche facile mais il est complètement implémenté et il ne risque pas de changer grandement. Tout d'abord, nous n'avons pas eu de mal à avoir un bouton qui crée un hôte autrement dit, un serveur et un client connectés au serveur, à avoir un bouton qui rejoint le serveur connecté à l'adresse IP écrite dans la case.

Cependant pour résoudre tout problème qui pourrait survenir et qui sont arrivés à de nombreuses reprises durant le développement du projet, nous avons créé de nombreux systèmes. Premièrement, un système qui bloque toutes interaction avec l'interface, durant la connexion de notre joueur, pour empêcher tout bug de survenir. De plus, nous avons également créé un système qui tente à de nombreuses reprises de se connecter et nous pourrions peut-être à l'avenir mettre dans les paramètres généraux le choix des délais de connexions. Par ailleurs, nous avons aussi bloqué toutes connexions de client au serveur lorsqu'il est dans le chargement de création du serveur mais aussi lorsque le jeu est lancé et que le salon autrement appelé lobby est quitté.

3.6.4 Installation du système de Team sur le serveur

Pour différencier les équipes de joueurs même si le jeu se concentrera sur le 1 contre 1, nous avons décidé de proposer au joueur dès l'interface de connexion la possibilité de choisir une team. Cette team est un code couleur HTML pour simplifier son utilisation pour les graphismes. Durant le chargement de connexion, la team envoyée par le client est associée au GameObject du joueur sur le serveur puis envoyé à tous les clients avec [SyncVar].

3.6.5 Chargement des joueurs entre les scènes

Le chargement des joueurs entre les scènes a été une tâche très complexe et semée d'embûches. En effet, par défaut le joueur est supprimé à chaque scène, c'est pourquoi nous avons décidé de placer dans sa fonction de début la fonction `"DontDestroyOnLoad(gameObject);"` pour empêcher le joueur de maintenir son état pour les scènes suivantes.

Comme nous gardons le joueur, nous devons donc initialiser le `GameObject` du joueur à chaque scènes correspondant à chaque mini-jeu. Ainsi, à l'aide d'un bouton start, il est exécuté une fonction sur le joueur hôte de la partie qui va passer à la prochaine scène dans la liste des scènes qui sont triées dans un ordre précis. Dans un avenir proche, nous implémenterons un système qui prendra les sections de 5 mini-jeux et de façon aléatoire en lancera une puis une autre jusqu'à la mort d'un joueur ou jusqu'à arrêt volontaire. De plus, avant de charger la scène, les paramètres pour la scène sont chargés sur le `GameObject` de tous les joueurs sur tous les clients

3.6.6 Synchronisation des Input

L'une des dernières améliorations que nous avons apporté à notre jeu est la synchronisation des input du clavier ou de la manette. Le système fonctionne de la manière suivante, toutes les millisecondes, les touches du clients modifient la variable correspondant à leur touche en attribut et ces valeurs sont synchronisées avec le serveur grâce à `[SyncVar]` qui est mit en mode envoi de données du client vers le serveur. Ainsi le serveur peut parfaitement lire chacune des touches des différents joueurs et ainsi exécuter une action spécifique.

3.7 L'intelligence artificiel

Nous n'avons pas encore implémenté d'intelligence artificielle dans notre jeu mais nous en avons bien l'ambition d'en créer. En effet, que ce soit dans les mini jeux de petit taille avec quelques IA "zombie" qui pourrait nuire aux déplacements du joueur, ou que se soit avec les IA plus complexes telles que les boss, nous aurons des IA qui pourront interagir avec les joueurs en temps réel et qui contribueront aux scores des différents joueurs.

De plus, nous pensons inclure une intelligence artificielle gérant le choix de mini-jeux et leur difficulté. En effet, pour que notre jeu ne soit pas trop répétitif, il faudrait que le même jeu ne soit pas utilisé 2 fois de suite, qu'il y ait certaines variations entre chaque challenge, et aussi, si le mini-jeu possède une difficulté variable, de la choisir en fonction du temps de jeu de la partie des joueurs, permettant une augmentation de la difficulté avec le temps.

3.8 Les graphismes

3.8.1 Création de base des personnages

Étant donné que nous sommes relativement nouveaux dans le domaine du pixel art, nous avons consacré du temps à explorer et à nous familiariser avec les techniques, les outils et les ressources disponibles.

Nous avons étudié divers tutoriels en ligne, examiné des œuvres de jeux en pixel art tel que Pokémon et Stardew Valley , et nous avons expérimenté différents logiciels spécialisés dans le pixel art. Nous avons donc opté pour l'utilisation d'Aseprite, un logiciel permettant de créer différentes frames et donc , des sprites en Pixel Art et de pouvoir les importer sous forme de Gifs.

Notre but premier a été tout d'abord de créer la forme de base des personnages qui constitueront notre jeu. Nous avons alors opté sur un format 16-bits pour apporter assez de détails mais aussi pour limiter la complexité des animations. Nous nous sommes inspirés des jeux game-

boy pour produire l’animation, tandis que pour les couleurs et le design des personnages, nous allons faire en sorte qu’ils conviennent au thème futuriste et robotique de notre jeu.



FIGURE 13 – Sprites du petit robot

3.8.2 Conception des boutons

L’interface utilisateur constitue un élément essentiel de toute expérience de jeu, et la conception de boutons clairs et intuitifs est un aspect crucial de cette interface. Pour répondre à cette exigence, nous créerons dans un futur proche une série de boutons pixel art pour les différentes options du menu, telles que “Play” , “Options” et “Exit”.

Chaque bouton sera soigneusement conçu pour être à la fois esthétique et fonctionnel. Nous opterons pour des formes simples et des couleurs adaptées à notre thème futuriste afin de garantir leur lisibilité pour les joueurs. En accord avec le style pixel art du jeu, nous veillerons à ce que l’animation des boutons soit esthétiquement plaisante mais aussi adaptée à notre jeu.



FIGURE 14 – Sprites du bouton play

3.8.3 Création du décor

Le décor joue un rôle essentiel dans la création de l’ambiance et de l’immersion d’un jeu vidéo. Dans notre projet, nous consacrerons du temps à la conception d’arrière plans pixel art pour qu’ils soient futuristes et agréables.

Nous nous inspirerons d’images libre de droit d’auteurs pour produire les premiers décors. Chaque élément du décor sera soigneusement pensé pour contribuer à l’histoire et à l’univers du jeu. Nous travaillerons sur l’agencement des éléments, les variations de couleurs et les petits éléments présents sur l’arrière-plan, ce qui permet de renforcer l’ambiance futuriste d’Another Recursion. Chaque élément du décor sera conçu pour raconter une histoire et pour stimuler l’imagination des joueurs, contribuant ainsi à créer une expérience de jeu captivante et mémorable.

3.8.4 Résolution et Pixel Perfect Camera

Au niveau de la résolution de notre jeu nous avons choisi d’utiliser la pixel perfect camera afin de restreindre tout pixel à l’écran à une résolution fixe et de permettre corriger visuellement automatiquement tout élément qui ne serait pas en pixel art. Pour la résolution fixe nous avons choisi de nous baser sur le format le plus commun qui 1920 par 1080 autrement dit un ratio de 16 par 9.

Nous avons décidé de le diminuer pour avoir un jeu plus pixelisé, ainsi la résolution choisie fut dans un premier temps 400 par 225. Par la suite

étant donné nos tile map qui sont fixées sur une grille de 16 pixels par 16 pixels nous avons choisi la résolution 400 par 224 pour permettre d'avoir deux multiples de 16 et ainsi permettre aux tile map de mieux se coller à notre résolution.

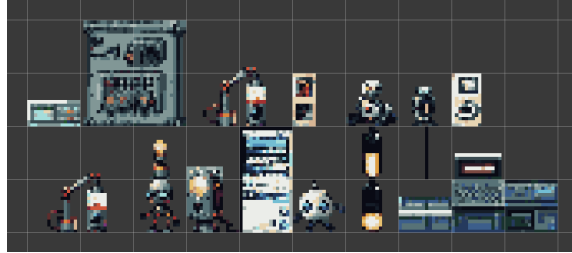


FIGURE 15 – Tile palette du décor

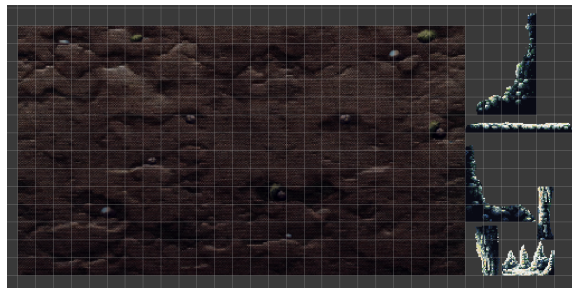


FIGURE 16 – Tile palette du terrain

3.8.5 Shader Graph

Lors de la conception des équipes, pour différencier chacune des équipes nous souhaitons rajouter une possibilité de modifier certaines des parties du sprite du joueur, ainsi pour le sprite du robot nous avons voulu modifier la couleur du bandeau en fonction de la couleur de l'équipe. Pour ce faire nous avons décidé d'utiliser un shader personnalisé.

Pour nous simplifier la tâche, nous avons utilisé l'asset Shader Graph qui permet de créer un shader avec des outils visuel. Comme nous ne sommes qu'au début de la création de sprite, pour l'instant le shader se limitera qu'à juste rajouter au dessus de notre sprite actuel une version du bandeau du robot avec une variable pour sa couleur qui sera définie à chaque fois que le sprite du robot apparaîtra.

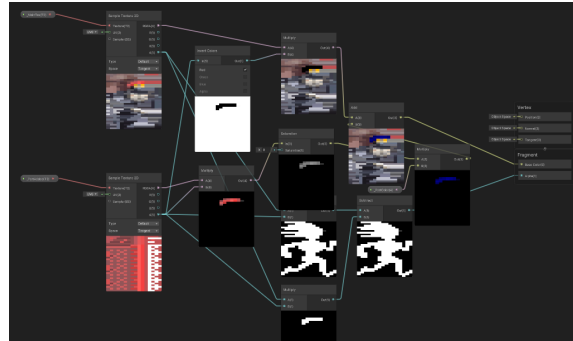


FIGURE 17 – Shader Graph du shader pour la couleur de l'équipe

3.8.6 Perspectives futures

Alors que nous continuons à progresser dans le développement de notre jeu , nous envisageons d'approfondir encore plus nos compétences en pixel art. Nous prévoyons d'ajouter encore plus de détails et de raffinements aux éléments graphiques existants, en s'inspirant de travaux déjà existants et en repoussant les limites de notre créativité.

Nous sommes déterminés à créer une expérience visuelle immersive qui enchante et captive les joueurs , tout en restant fidèles à notre esthétique futuriste et charmante. Nous sommes impatients de voir comment notre vision artistique évoluera au fil du temps et de partager notre création avec le monde.

3.9 Le son

3.9.1 La musique

La musique joue un rôle crucial dans la création de l’ambiance et de l’immersion d’un jeu vidéo. Notre équipe a entrepris de composer une bande-son originale qui captivera les joueurs et les plongera dans l’univers futuriste et robotique du jeu. Nous avons commencé par définir l’ambiance et le ton général du jeu, en tenant compte de son histoire, de son esthétique et de son gameplay. Nous utilisons une application de mixage et de créations appelée FL Studio pour permettre de créer des mélodies adaptées.

En nous appuyant sur ces éléments, nous avons commencé à composer des morceaux musicaux qui reflètent ces aspects. Nous avons exploré différents genres musicaux, des mélodies épiques aux ambiances atmosphériques, en cherchant à créer une bande-son variée et dynamique qui accompagnera efficacement les différents moments du jeu. Notre objectif est de créer une musique qui soit à la fois mémorable et évocatrice, et qui contribue à renforcer l’immersion des joueurs dans l’univers du jeu, en s’inspirant parfois de bandes sons connues telles que celles de Minecraft.

On a alors utilisé un système de boucle musicale, pour créer une musique qui est certes répétitive mais adaptée aux différents niveaux de notre jeu et qui va changer en fonction de la difficulté des niveaux.

3.9.2 Les effets sonores

Les effets sonores sont un autre élément essentiel pour créer une expérience de jeu immersive. Bien que nous n’ayons pas encore commencé à travailler sur les effets sonores, nous sommes conscients de leur importance pour enrichir l’expérience sonore du jeu. Nous prévoyons de créer une large gamme d’effets sonores, des bruits ambiants aux sons d’action, en passant par les bruits de pas et d’attaques des personnages.

Nous sommes déterminés à créer des effets sonores de haute qualité qui correspondent à l'esthétique et à l'atmosphère du jeu. Nous explorerons différentes techniques d'enregistrement et de manipulation sonore pour obtenir des résultats authentiques et convaincants. Notre objectif est de créer une palette sonore riche et immersive qui ajoute une nouvelle dimension à l'expérience de jeu.

3.10 Le site web

Dans le cadre de notre projet, nous avons eu pour responsabilité de concevoir et de développer le site web associé au jeu. En utilisant les langages HTML, CSS et JS, nous avons fait de notre mieux pour créer une interface utilisateur fonctionnelle et attrayante offrant aux joueurs un accès facile aux informations essentielles sur le jeu. Ce site était censé rendre possible le téléchargement du jeu et des diverses ressources utilisées pour sa création, ainsi que des informations sur le projet, la partie multijoueur, la partie IA et l'équipe. Ce site web se devait donc d'être complet.

L'un des aspects clés de la création du site web a été la conception et l'implémentation d'un menu latéral interactif. Ce menu permet aux joueurs de naviguer facilement entre les différentes sections du site, y compris les informations sur le projet, sur l'équipe, et aussi une section depuis laquelle le téléchargement du jeu, du manuel et de ce rapport seraient disponibles. La planification de l'architecture du menu et surtout son implémentation a pris du temps, mais nous nous sommes assurés qu'il était intuitif et facile à utiliser par tous les types de joueurs en plus d'être visuellement agréable.

Nous avons utilisé le CSS, qui est un langage permettant la mise en forme des différents éléments du site (paragraphe, images), rendant ainsi le site plus attrayant. Par exemple, lorsque les joueurs survolent les différents éléments du menu et du site, nous avons ajouté des effets d'assombrissement au survol, simple mais efficace, donnant directement l'impression que le site est interactif.

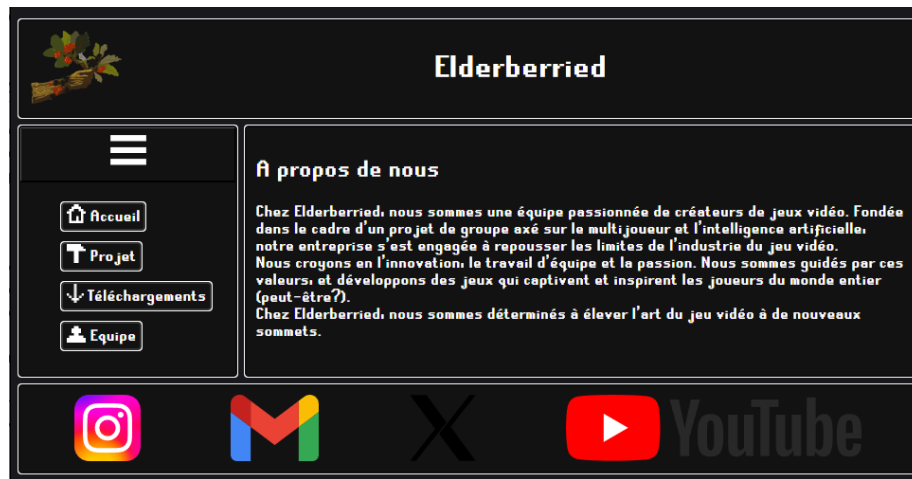


FIGURE 18 – Exemple de la page d'accueil du site

Le JavaScript a également été utilisé pour créer des transitions fluides notamment celles du menu latéral et des effets visuels attrayants. Les animations plus complexes ont été gérées avec JavaScript. Nous avons par exemple utilisé des boutons et des EventListener permettant de détecter quand un élément est cliqué, entraînant des changements sur le CSS (le style d'un élément) et donc sur le site web en général. Par exemple le menu latéral est rétractable, il suffit pour cela de cliquer sur l'icône de menu en haut du menu de navigation. Cette implémentation qui a l'air basique peut pourtant prendre des heures à ceux qui ne connaissent pas encore JS.

Grâce à notre travail sur le site web, nous avons pu offrir aux joueurs une expérience immersive qui complète parfaitement notre jeu vidéo, renforçant ainsi la cohérence de notre image.

4 Conclusion

En cette première phase importante de notre projet, nous nous situons dans une période significative du développement de "Another Recursion". Nous avons, avec un sentiment d'accomplissement, établi les fondations solides sur lesquelles notre jeu va continuer à s'élever. La mise en place réussie de la base multijoueur, du menu, et des interactions avec les scènes d'Unity montre une importante progression technique mais également une amélioration de la cohésion de l'équipe malgré notre projet ambitieux de création d'un Party Game avec plusieurs mini-jeux.

Cependant, il est clair que notre projet est loin d'être terminé. Un grand nombre de mini-jeux restent à être implémentés, chaque jeu nécessitant une attention particulière pour garantir qu'il enrichisse l'expérience offerte par "Another Recursion". De plus, l'amélioration des aspects visuels et auditifs, la finalisation de notre site web, et l'implémentation de paramètres modifiables, comme les réglages généraux du jeu et la personnalisation de l'ordre des mini-jeux, constituent des défis importants à relever dans les mois à venir.

Notre équipe, bien que composée de spécialistes dans différents domaines, reconnaît l'importance cruciale de la collaboration et de la polyvalence. Chacun de nous est appelé à contribuer à la programmation des mini-jeux, illustrant notre engagement envers un développement collaboratif du projet. En cette période de première soutenance nous avons pris conscience du chemin parcouru mais aussi des étapes qu'il nous reste à franchir.

Nos succès initiaux dans la mise en place des fonctionnalités de base du jeu nous motivent à poursuivre nos efforts, à améliorer notre maîtrise des outils et technologies en jeu, et à matérialiser notre vision ambitieuse pour chacun des mini-jeux.

5 Annexe



FIGURE 19 – Sprite du canon du tank

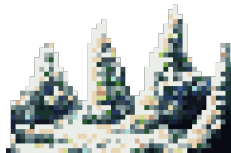


FIGURE 20 – Sprite stalagmite



FIGURE 21 – Cave background



FIGURE 22 – Sprite du bandeau robot

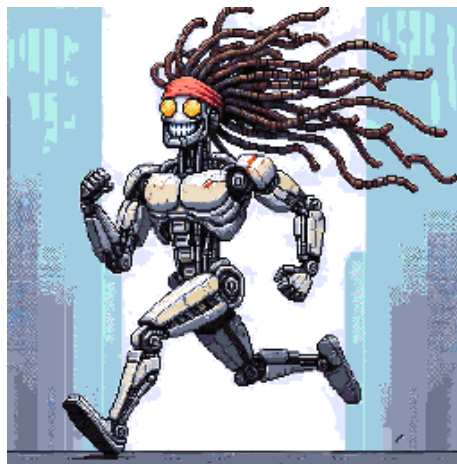


FIGURE 23 – Première version de l'icône du jeu



FIGURE 24 – Sprite du robot