



Compte Rendu de Soutenance

Projet Mobius

Un jeu d'aventure en pixel art avec voyage temporel

EPITA

École d'Ingénieurs en Informatique

Équipe de développement :

Émile • Madeleine • Maël • Georges

January 19, 2026

Contents

Introduction	3
1 Piloter des projets collaboratifs	3
1.1 Organiser son travail soi-même : définir ses objectifs, planifier leur atteinte et être fiable	3
1.1.1 Spécifier le projet	3
1.1.2 Planifier les étapes de la réalisation du projet	4
1.2 Expliquer son travail en fonction du contexte : avancement et difficultés, de manière concise, précise et claire	4
1.2.1 Présenter l'état d'avancement du projet	4
1.2.2 Fournir une explication technique sur une tâche réalisée	6
1.3 Identifier ses modalités de travail et ses modes de communication	7
1.3.1 Expliquer les modalités de travail et le mode de communication utilisés	7
1.3.2 Dynamique du groupe	8
1.4 Se situer soi-même dans le groupe de travail projet (connaître sa place / son périmètre d'action)	8
2 Produire une solution informatique efficace répondant à des contraintes de qualité, de sûreté et de sécurité	11
2.1 Produire des programmes informatiques de premier accès, impliquant une interaction avec l'environnement, la gestion des dépendances et la livraison d'un exécutable	11
2.1.1 Intégrer les outils de prise en main dans le produit	11
2.1.2 Adopter le rendu visuel à l'objectif et au style du jeu	12
2.1.3 Produire un exécutable de qualité	13

2.1.4	Fournir une documentation technique de qualité	14
Conclusion		14

Introduction

Le projet Mobius représente un projet de développement collaboratif ambitieux mené par une équipe de quatre étudiants. Il s'agit d'un jeu vidéo d'aventure développé en Python utilisant le moteur graphique Pygame, mettant en scène un univers de science-fiction centré sur le voyage temporel à travers différentes époques de l'histoire de l'humanité.

Ce compte rendu présente l'organisation du travail collaboratif, les choix techniques effectués, l'état d'avancement du projet ainsi que les différentes modalités de travail mises en place par l'équipe. Le projet se distingue par son approche méthodique et sa répartition des tâches selon les compétences de chacun des membres.

1 Piloter des projets collaboratifs

1.1 Organiser son travail soi-même : définir ses objectifs, planifier leur atteinte et être fiable

La gestion de projet est un élément fondamental pour assurer le bon déroulement du développement de Mobius. L'équipe a mis en place une structure organisationnelle claire permettant à chaque membre de contribuer efficacement au projet.

1.1.1 Spécifier le projet

La spécification du projet constitue la base de notre démarche de développement. L'ensemble des fonctionnalités du jeu Mobius ont été soigneusement documentées et consignées dans un fichier `README.md` présent sur le dépôt GitHub du projet, accessible à l'adresse suivante : https://github.com/Epigold/Mobius_EPITA/.

Ce document central détaille de manière exhaustive chaque aspect du projet, depuis les procédures d'installation du jeu jusqu'aux mécaniques de gameplay. Il comprend notamment des instructions détaillées sur les commandes de déplacement, les interactions avec l'environnement, et les différentes fonctionnalités disponibles pour les joueurs. Cette documentation permet non seulement aux membres de l'équipe de maintenir une vision cohérente du projet, mais elle facilite également la prise en main du jeu par de futurs utilisateurs.

1.1.2 Planifier les étapes de la réalisation du projet

La planification du projet s'est articulée autour d'une répartition stratégique des tâches en fonction des compétences et des affinités de chaque membre de l'équipe. Cette approche permet d'optimiser la productivité et la qualité du travail fourni.

Exemple concret de répartition : Émile et Madeleine, reconnus pour leur créativité et leur sens artistique, ont été assignés* aux aspects graphiques du jeu ainsi qu'à l'élaboration de l'univers narratif. Leur travail comprend la création de l'histoire, la conception visuelle du jeu, et le design des différents éléments graphiques.

Parallèlement, Maël et Georges, dotés de compétences techniques plus approfondies en programmation, se sont vu confier le développement des mécanismes de jeu, de l'architecture logicielle, de l'intelligence artificielle des ennemis, ainsi que l'implémentation du système multijoueur.

**Il est important de noter que cette assignation est de nature temporaire. Une fois leurs tâches créatives et graphiques accomplies, Émile et Madeleine rejoindront l'équipe de programmation pour contribuer au développement technique du jeu, permettant ainsi une meilleure polyvalence de l'équipe.*

1.2 Expliquer son travail en fonction du contexte : avancement et difficultés, de manière concise, précise et claire

La communication régulière sur l'état d'avancement du projet est essentielle pour maintenir la cohésion de l'équipe et assurer le respect des échéances. Cette section détaille les progrès réalisés dans chaque domaine du développement.

1.2.1 Présenter l'état d'avancement du projet

Le projet Mobius progresse de manière satisfaisante sur l'ensemble de ses composantes. Voici un état détaillé de l'avancement par domaine :

Histoire et univers narratif :

Une grande partie de l'histoire du jeu a été élaborée et consignée dans un document de référence. Le scénario principal est défini, incluant les différentes époques que le joueur traversera au cours de son aventure. L'univers de science-fiction créé pour Mobius

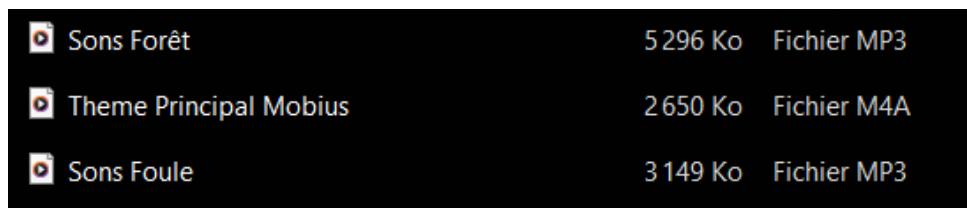
se veut riche et cohérent, avec une attention particulière portée aux détails de chaque période temporelle. Des modifications mineures restent possibles concernant certains détails narratifs afin d'améliorer la cohérence globale de l'histoire.

Intelligence artificielle :

Le système d'intelligence artificielle pour les ennemis est actuellement fonctionnel pour la première zone du jeu. Quatre types de monstres différents ont été implémentés, chacun avec des comportements et des patterns d'attaque distincts. Ces modèles d'IA permettent de créer des combats variés et intéressants pour le joueur. Le travail restant consiste à développer des modèles similaires pour les autres zones du jeu, en adaptant les comportements aux différents contextes et époques.

Musiques et environnement sonore :

L'aspect audio du jeu bénéficie d'une attention particulière. L'équipe a déjà produit une composition musicale originale pour le jeu, créée en collaboration avec un musicien professionnel, ce qui garantit une qualité sonore élevée. En complément, des effets sonores et bruitages libres de droits ont été intégrés pour enrichir l'expérience immersive du joueur. Ces éléments audio contribuent significativement à l'atmosphère du jeu.






 Sons Forêt	5 296 Ko	Fichier MP3
 Theme Principal Mobius	2 650 Ko	Fichier M4A
 Sons Foule	3 149 Ko	Fichier MP3

Figure 1: Différents sons prévus pour le jeu

Conception des cartes et niveaux :

Le développement des environnements de jeu est en cours avec deux cartes actuellement en phase de développement actif. Cependant, un travail important reste à accomplir concernant les directions artistiques des autres zones du jeu. Chaque époque visitée nécessitera un design visuel distinct et cohérent. De plus, l'ensemble des "salles" et des zones que le personnage devra parcourir restent à concevoir et à implémenter, représentant une part importante du travail futur.

Autres éléments graphiques :

Le développement graphique s'étend bien au-delà des cartes. De nombreux autres assets visuels sont en cours de création, notamment le sprite du personnage principal, les différents types d'ennemis, l'écran d'accueil du jeu, et les divers menus d'interface utilisateur.

Plusieurs prototypes ont déjà été réalisés et testés, permettant d'affiner progressivement le style visuel du jeu pour assurer une cohérence esthétique globale.

Développement et programmation :

Sur le plan technique, l'équipe dispose actuellement d'un prototype fonctionnel du jeu. Ce prototype intègre déjà le système de classes pour le personnage, permettant au joueur de choisir parmi différentes spécialisations ayant chacune des capacités et des caractéristiques uniques. Cette base technique solide facilite l'ajout progressif de nouvelles fonctionnalités.

Fonctionnalité multijoueur :

L'équipe ambitionne d'implémenter un système multijoueur pour enrichir l'expérience de jeu. Le système prévu fonctionnera via un hébergement de serveur en local, permettant une expérience de jeu coopérative à deux joueurs. Cette fonctionnalité représente un défi technique important mais apportera une valeur ajoutée significative au projet.

1.2.2 Fournir une explication technique sur une tâche réalisée

Approche de programmation et architecture logicielle :

Pour la réalisation du code du jeu Mobius, l'équipe devait respecter certaines contraintes techniques imposées dans le cadre du projet académique. Principalement, le développement devait être effectué en langage Python et utiliser le moteur graphique Pygame pour la gestion de l'affichage et des interactions.

Face à ces exigences, l'utilisation de la méthodologie POO (Programmation Orientée Objet) s'est imposée comme la solution la plus adaptée et la plus pertinente. Cette approche présente de nombreux avantages pour un projet de cette envergure.

Grâce à la POO, l'équipe a pu structurer le code de manière modulaire et maintenable. Chaque élément du jeu a été conçu comme une classe indépendante : les personnages jouables, les différents types d'ennemis, les systèmes de déplacement, les mécaniques de combat, et tous les autres mécanismes du jeu ont été isolés dans des modules distincts. Cette séparation permet une expérience de développement optimale, facilitant le débogage, la réutilisation du code, et l'ajout de nouvelles fonctionnalités sans risquer d'affecter les composants existants.

L'architecture orientée objet permet également une meilleure collaboration entre les membres de l'équipe, chacun pouvant travailler sur des composants spécifiques sans interférer

avec le travail des autres développeurs.

```
class Boss(Enemy):
    def __init__(self, player, wave):
        health = 1000 + (wave * 200)
        super().__init__(player, TAILLE_BOSS, 5, health, 25, (150, 50, 150))
        self.type = "BOSS"
        self.phase = 1
        self.attack_cooldown = 0
        self.charge_speed = 15
        self.charging = False
        self.charge_time = 0
        self.target_x = 0
        self.target_y = 0
        self.wave = wave
```

Figure 2: Exemple de code POO illustrant l'architecture du jeu

1.3 Identifier ses modalités de travail et ses modes de communication

L'organisation du travail et la communication au sein de l'équipe sont des facteurs déterminants pour le succès du projet. L'équipe a mis en place des processus efficaces pour coordonner les efforts de chacun.

1.3.1 Expliquer les modalités de travail et le mode de communication utilisés

Organisation des séances de travail :

L'équipe s'organise de manière à se retrouver physiquement chaque semaine lors d'heures creuses sur le site d'EPITA. Ces réunions hebdomadaires constituent des moments privilégiés pour la coordination et la synchronisation du travail. Lors de ces sessions, chaque membre présente ses avancées personnelles sur son périmètre de compétences et expose les objectifs qu'il se fixe pour les séances suivantes.

Ces réunions permettent également de planifier collectivement les actions à entreprendre pour respecter les échéances à venir. L'équipe peut ainsi identifier rapidement les éventuels blocages, réajuster la répartition des tâches si nécessaire, et s'assurer que tous les membres progressent de manière harmonieuse vers les objectifs communs.

Outils de communication :

Pour la communication quotidienne et les échanges en dehors des réunions physiques, l'équipe utilise exclusivement la plateforme Discord. Ce choix s'explique par la simplicité d'utilisation de cet outil et ses fonctionnalités adaptées au travail collaboratif.

Discord offre la possibilité de créer différents salons thématiques, permettant de structurer les discussions par domaine (programmation, graphismes, musique, organisation générale, etc.). Cette organisation garantit que les conversations restent centrées et que chaque membre peut facilement retrouver les informations pertinentes pour son travail. De plus, la conservation automatique d'une trace écrite de toutes les discussions permet de maintenir une mémoire collective du projet et de s'y référer en cas de besoin.

1.3.2 Dynamique du groupe

La dynamique de groupe constitue un élément essentiel du bon fonctionnement du projet Mobius. Les tâches ont été réparties de manière réfléchie et stratégique, afin que chaque membre de l'équipe puisse faire profiter le groupe de ses compétences spécifiques et de ses points forts.

Cette répartition des responsabilités est formalisée et détaillée dans le cahier des charges techniques du projet, document de référence qui définit clairement les rôles et les périmètres d'action de chacun. Cette formalisation évite les confusions et les chevauchements de responsabilités.

Au-delà de cette répartition initiale, le groupe maintient une approche collaborative dans la prise de décisions. Des consultations régulières sont organisées sur les différents aspects du projet, permettant de recueillir l'avis et les suggestions de chacun, même sur des domaines qui ne relèvent pas directement de leur responsabilité principale. Cette approche inclusive favorise l'appropriation collective du projet et enrichit la qualité des décisions prises.

1.4 Se situer soi-même dans le groupe de travail projet (connaître sa place / son périmètre d'action)

La clarté des rôles et des responsabilités de chaque membre est fondamentale pour la bonne marche du projet. Voici une description détaillée du rôle de chacun au sein de l'équipe :

Émile - Création narrative et développement interface :

Émile se distingue par sa créativité débordante et son attrait particulier pour l'univers de la science-fiction. Ces qualités l'ont naturellement conduit à se porter volontaire pour

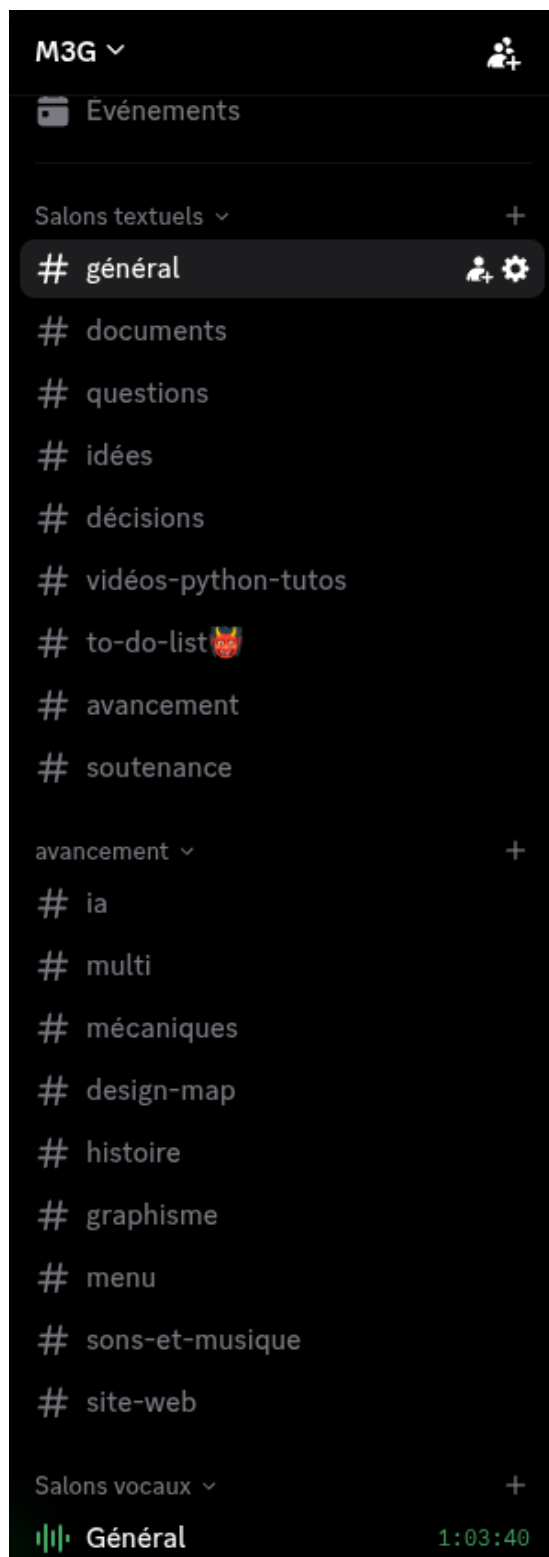


Figure 3: Discord - Salons thématiques

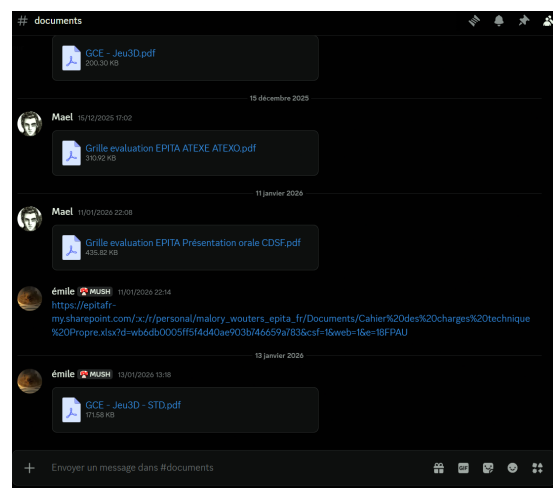


Figure 4: Discord - Vue d'ensemble

Répartition des tâches : deux personnes par tâche: (R)esponsable & (S)uppléant				
Tâches	Emile	Madeleine	Mael	Georges
Intelligence artificielle				R
Multijoueur			R	
Carte	S	R		
Musique et son			R	S
Graphisme	S	R		
Menu	R			
Programmation mecanique			S	R
Histoire du jeu	R	S		
Site Web			S	R

Figure 5: capture d'écran de la répartition des rôles dans le cahier des charges technique

la création de l'histoire et de l'univers de Mobius. Son rôle comprend la conception du scénario principal, le développement des différentes époques visitées, et la création du lore global du jeu.

Compte tenu de l'importance de ce travail dans le processus initial de création du jeu et du fait qu'il s'agit d'une tâche concentrée sur les premières phases du projet, Émile a prévu de contribuer également au développement technique une fois la base narrative établie. Il se concentrera particulièrement sur la programmation du menu et de l'interface utilisateur.

Madeleine - Direction artistique et design graphique :

Madeleine possède de grandes capacités dans le domaine des arts graphiques et du design visuel. Elle assume donc naturellement la responsabilité de l'ensemble des graphismes du jeu. Son travail comprend la création des sprites de personnages, des ennemis, des éléments d'interface, ainsi que la conception et la réalisation des différentes cartes et environnements du jeu. Son sens esthétique contribue grandement à définir l'identité visuelle de Mobius.

Maël - Audio, multijoueur et programmation :

Maël apporte au projet ses compétences musicales, étant lui-même musicien. Il prend en charge l'ensemble de l'environnement sonore du jeu, incluant la composition musicale et l'intégration des effets sonores. Au-delà de cet aspect audio, Maël contribue également significativement au développement technique du projet. Il est particulièrement responsable de l'implémentation du système multijoueur, un défi technique important, ainsi que de certaines parties critiques de la programmation du jeu.

Georges - Intelligence artificielle et mécaniques de jeu :

Georges s'est familiarisé en profondeur avec le module Pygame, développant une expertise technique précieuse pour le projet. Il assume la responsabilité du développement de l'intelligence artificielle des ennemis, un composant essentiel pour assurer une expérience de jeu engageante et variée. Georges est également en charge de l'implémentation des différentes mécaniques de jeu, des systèmes de combat aux interactions avec l'environnement, garantissant ainsi la qualité du gameplay de Mobius.

2 Produire une solution informatique efficace répondant à des contraintes de qualité, de sûreté et de sécurité

Cette section aborde les aspects techniques et qualitatifs du développement du jeu Mobius, en mettant l'accent sur les bonnes pratiques de développement logiciel et la satisfaction de l'utilisateur final.

2.1 Produire des programmes informatiques de premier accès, impliquant une interaction avec l'environnement, la gestion des dépendances et la livraison d'un exécutable

2.1.1 Intégrer les outils de prise en main dans le produit

L'accessibilité et la facilité de prise en main du jeu sont des priorités pour l'équipe de développement. Dans cette optique, un effort particulier a été consacré à la création d'une documentation complète et claire destinée aux joueurs.

Un guide détaillé sur les contrôles en jeu a été élaboré, expliquant de manière claire et précise toutes les commandes disponibles et leurs fonctions respectives. Ce guide permet aux joueurs de comprendre rapidement comment interagir avec le jeu, comment déplacer leur personnage, comment accéder aux différents menus, et comment utiliser les diverses capacités disponibles.

Parallèlement, un guide d'installation complet a été rédigé pour faciliter la mise en place du jeu sur différentes configurations. Ce guide détaille pas à pas les prérequis techniques, les dépendances à installer, et la procédure d'installation, minimisant ainsi les obstacles

techniques qui pourraient empêcher un joueur de profiter du jeu.

Ces deux guides sont disponibles dans le fichier README du dépôt GitHub du projet, assurant leur accessibilité immédiate. Cette documentation permet une prise en main naturelle et intuitive du produit, dans l'objectif d'optimiser l'expérience utilisateur dès les premiers instants de jeu.



Figure 6: README - Guide d'installation

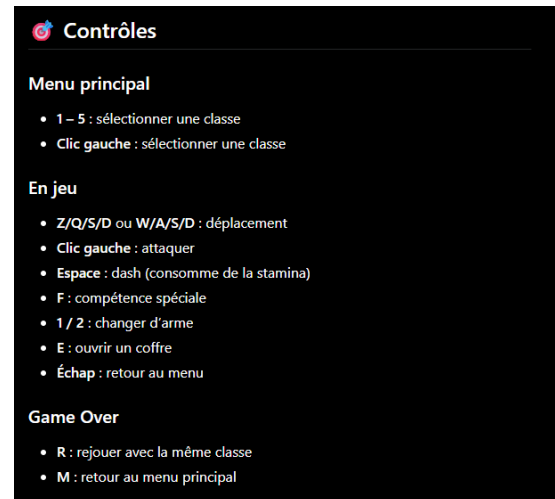


Figure 7: README - Commandes de jeu

2.1.2 Adopter le rendu visuel à l'objectif et au style du jeu

Le choix de la direction artistique constitue un élément fondamental de l'identité du jeu Mobius. L'équipe a opté pour un style visuel pixelisé, également appelé pixel art, un choix esthétique qui présente plusieurs avantages.

Le jeu se déroule à travers plusieurs époques et lieux de l'histoire de l'humanité, permettant au joueur de voyager dans le temps et de découvrir différentes périodes historiques. Cette mécanique de voyage temporel constitue le cœur de l'expérience de jeu et a fortement influencé les choix artistiques.

Pour refléter cette thématique centrale, l'équipe a décidé que la direction artistique du projet serait orientée vers le concept de voyage temporel, avec des représentations visuelles distinctes pour chaque époque visitée. Madeleine, qui assume la responsabilité des

graphismes du jeu, s'est inspirée de ce concept pour créer les différents éléments visuels du projet.

Cette approche se manifeste dans tous les aspects du design visuel, depuis les environnements jusqu'aux personnages, en passant par l'interface utilisateur. Par exemple, le logo de Mobius lui-même incarne cette philosophie, avec un design qui évoque le mouvement et la traversée du temps.



Figure 8: Logo de Mobius

Le style pixel art, au-delà de son aspect esthétique attrayant et nostalgique, présente également des avantages pratiques pour une équipe de développement de cette taille, permettant une création d'assets plus rapide tout en maintenant une qualité visuelle élevée.

2.1.3 Produire un exécutable de qualité

La distribution et l'accessibilité du jeu ont été soigneusement considérées par l'équipe. La première version jouable de Mobius est d'ores et déjà disponible sous forme d'exécutable sur le dépôt GitHub du projet.

Cet exécutable a été conçu pour être autonome et complet. Il regroupe en un seul et unique fichier l'intégralité du code source du jeu ainsi que tous les graphismes nécessaires. Cette approche simplifie grandement la distribution et l'installation du jeu pour les utilisateurs finaux, qui n'ont pas besoin de gérer manuellement les dépendances ou les fichiers d'assets séparés.

La production de cet exécutable démontre la maturité technique du projet et permet déjà à des testeurs externes de découvrir le jeu, fournissant ainsi des retours précieux pour guider le développement futur.

2.1.4 Fournir une documentation technique de qualité

La documentation technique représente un pilier essentiel de tout projet logiciel de qualité. L'équipe de Mobius a accordé une attention particulière à cet aspect.

Comme présenté précédemment, l'entièreté de la documentation liée au produit se trouve centralisée dans le fichier README disponible sur le dépôt GitHub du projet, dont le lien a été fourni au début de ce document. Ce choix de centralisation facilite l'accès à l'information et évite la dispersion de la documentation.

Dans ce document de référence, toutes les informations liées à la documentation technique sont disponibles et présentent en détail chaque aspect du projet. Cela inclut non seulement les guides d'utilisation destinés aux joueurs, mais également des informations sur l'architecture du code, les dépendances utilisées, et les procédures de développement, utiles pour d'éventuels contributeurs futurs ou pour la maintenance à long terme du projet.

Cette documentation de qualité témoigne du professionnalisme de l'équipe et de sa volonté de produire un projet complet et bien documenté, conforme aux standards de l'industrie du développement logiciel.

Conclusion

Le projet Mobius illustre la capacité d'une équipe étudiante à mener un projet de développement logiciel ambitieux en respectant les bonnes pratiques de gestion de projet et de développement collaboratif.

La répartition claire des rôles selon les compétences de chacun, la mise en place de processus de communication efficaces, et l'attention portée à la qualité technique et documentaire du produit démontrent la maturité de l'équipe dans son approche du développement.

Le projet progresse de manière satisfaisante sur l'ensemble de ses axes de développement, avec des réalisations concrètes déjà disponibles et une vision claire des étapes restantes pour atteindre les objectifs fixés. L'équipe reste mobilisée et déterminée à livrer un produit final de qualité qui saura offrir une expérience de jeu engageante et mémorable.