

# Projet "Echo of Nyxalis"

## Rapport de soutenance

PROMO 2029 Prepa Sup D1



### Étudiants :

Aïzabel AUTECHAUD  
Augustin BAZIN ROY  
Julien CHARLES  
Clément GOMEZ  
Angéline TAING

27 mai 2025

---

# Sommaire

<b>1</b>	<b>Présentation Générale</b>	<b>4</b>
1.1	Introduction . . . . .	4
1.2	Vision et objectif . . . . .	6
1.3	Origine et Nature du Projet . . . . .	6
1.4	Répartition et Organisation . . . . .	8
<b>2</b>	<b>Design Visuel et Environnemental</b>	<b>9</b>
2.1	Environnement . . . . .	9
2.2	Ambiance Visuelle . . . . .	10
<b>3</b>	<b>Scénario et Gameplay</b>	<b>11</b>
3.1	Scénario du jeu . . . . .	11
3.2	Évolutions et Fonctionnalités Implémentées . . . . .	12
3.3	Conception générale . . . . .	14
3.4	Relier les salles : Portails et transitions . . . . .	15
3.5	Exploration & Donjons . . . . .	15
3.6	Génération aléatoire et seed . . . . .	16
<b>4</b>	<b>Systèmes du Jeu</b>	<b>16</b>
4.1	Système de dialogue narratif . . . . .	16
4.2	Refonte de l'inventaire . . . . .	18
4.3	Système de combat 2D . . . . .	22
4.4	Progression & Histoire . . . . .	23
4.5	Ambiance sonore et musicale . . . . .	23
<b>5</b>	<b>Interfaces et Utilisabilité</b>	<b>27</b>
5.1	Menu principal . . . . .	27
5.2	Le Tutoriel . . . . .	28
<b>6</b>	<b>Esthétique</b>	<b>31</b>
6.1	Esthétique générale . . . . .	31
6.2	Création des personnages . . . . .	32
<b>7</b>	<b>Ressources et Techniques</b>	<b>37</b>
7.1	Les assets . . . . .	37



---

7.2	Canvas . . . . .	40
7.3	site internet . . . . .	41
7.4	Optimisation et performances . . . . .	45
<b>8</b>	<b>Ressenti final</b>	<b>45</b>
8.1	Angeline . . . . .	45
8.2	Clément . . . . .	46
8.3	Aizabel . . . . .	47
8.4	Augustin . . . . .	48
8.5	Julien . . . . .	48
<b>9</b>	<b>Explication du code</b>	<b>50</b>



---

# 1 Présentation Générale

## 1.1 Introduction

Face à l'évolution rapide du marché des jeux vidéo avec l'avancée de l'intelligence artificielle et la réalité virtuelle, l'entreprise Prisme, spécialisée dans le secteur informatique et le développement de jeux vidéos, a fait le choix audacieux d'aller à contre-courant et de lancer un projet plus rétro pour se démarquer de ses concurrents.

Porté par la meilleure équipe de Prisme - JC3A, cinq amis motivés et réunis autour d'une passion, celle de créer - Echo of Nyxaris, doit arriver, grâce à ses graphismes de type cartoon en 2.5D à raviver nos vieux souvenirs, en plus de proposer des mécaniques nouvelles pour plaire à son public nostalgique et plus exigeant que jamais.

Les joueurs incarnent des personnes qui se réveillent sur Nyxalis, une planète qui devrait leur être familière. Mais, un coup d'œil à travers le hublot abîmé leur suffit à dire qu'il y a une erreur. La planète verte est violette.

Au milieu de la confusion, quelqu'un de l'équipage découvre que la capsule cryogénique a en fait dysfonctionné, et que leur sommeil de glace a duré plusieurs millénaires au lieu d'une centaine d'années. Ils sont sur la bonne planète. Seulement, ils se sont trompés d'ère.



---

Soudain, la porte du vaisseau cède. Une ombre tombe sur nos voyageurs. Une horreur sans nom, libre de toute définition, les regarde. Elle, qui n'a jamais eu de visage, ne peut s'empêcher d'envier ces êtres. Elle avance. Ils reculent. Elle ne comprend pas. Ils ne veulent pas la comprendre.

Ou peut-être ne peuvent-ils pas la comprendre ? Pourront-ils transcender leurs peurs, le temps et leur humanité pour survivre ?



---

## 1.2 Vision et objectif

Tout au long du jeu, les joueurs seront invités à explorer et inter-agir avec ce monde aussi fascinant que terrifiant, avec un but, se battre pour survivre et comprendre ce qui s'est exactement passé sur cette planète hostile.

Echo of Nyxaris saura séduire les joueurs grâce à ses énigmes, son univers et son intrigue entraînante. Il s'inspire d'ailleurs de plusieurs jeux à succès comme Noman's sky et Outer Wilds.

L'aspect multijoueur permettra aussi aux utilisateurs de progresser aux côtés de leurs coéquipiers afin d'affronter les monstres les plus redoutables et débloquent les prochaines étapes de l'histoire.

Que vous choisissiez de jouer seul ou en équipe, vivez une aventure immersive dans un autre monde en quête de percer les secrets qui vous entourent.

## 1.3 Origine et Nature du Projet

Ce projet découle d'un intérêt partagé pour le surnaturel et le futurisme, ainsi que d'une volonté d'apprendre, car qu'il y a t-il de plus passionnant que de réimaginer le monde avec des "si" ?

Echos of Nyxalis est un RPG Lite, un jeu de rôle dans lequel les



---

joueurs sont invités à incarner un personnage qui va évoluer au fil de son aventure, que ce soit à travers les combats, les interactions avec les personnages non-joueurs, ou les énigmes.

Simple de prise en main et avec une esthétique rétro, Echo of Nyxalis privilégie l'exploration et la survie, avec une forte composante narrative, centrée sur la découverte d'une civilisation mystérieuse.



---

## 1.4 Répartition et Organisation

Pour vous rappeler voici les tâches qui nous étaient assignées.

Tâches	Responsable	Suppléant
3D / Assets	Aïzabel	Julien
Sound Design	Clément	Angeline
UX / UI Design	Julien	Aïzabel
Site Web	Clément	Julien
Multijoueur	Angeline	Augustin
Gameplay	Augustin	Aïzabel
IA	Augustin	Angeline
Qualité	Julien	Clément





---

## 2 Design Visuel et Environnemental

Notre jeu s'inscrit dans un univers spatial inventé, conçu grâce à des graphismes en 2,5 D. Pour enrichir le potentiel visuel de notre projet, nous mettrons en œuvre une série de techniques artistiques. Parmi elles, nous utiliserons des effets de post-processing qui permettront d'ajouter des filtres visuels pour améliorer l'esthétique générale et la dynamique des scènes. De plus, nous apporterons une attention particulière à l'accentuation des ombres et des lumières, surtout durant le mode nuit, afin de renforcer l'illusion de profondeur et de créer une ambiance atmosphérique captivante qui suit la thématique de notre jeu.

### 2.1 Environnement

Nyxalis étant une planète inconnue, va se distinguer par un environnement avec une nature unique et inconnue au joueur grâce à des écosystèmes étranges avec des végétaux au design inconnu. Les paysages de Nyxalis varient, comprenant des forêts, des plaines mystérieuses ou encore des villes assez modernes incitant à la découverte de ce nouveau monde.

Le jeu combine un univers futuriste avec des éléments primitifs, en particulier pour les habitants de Nyxalis, car ils portent des vêtements inspirés de combinaisons d'astronautes dans des couleurs grises, vertes et blanches qui reflètent leur mode de vie unique.



---

## 2.2 Ambiance Visuelle

La domination des couleurs dans les paysages et les éléments graphiques crée une ambiance permettant au joueur de se plonger pleinement dans cet univers extraterrestre. Des effets de post-processing, comme le flou, les lumières dynamiques ou les effets de couleurs, seront utilisés pour améliorer l'aspect visuel du jeu.



---

## 3 Scénario et Gameplay

Echo of Nyxalis est un RPG retro en 2,5D. Le joueur pourra se déplacer de gauche à droite, de haut en bas, mais aussi en profondeur avec les touches du clavier. Il pourra aussi interagir avec des objets fixes (portes, rocher, jarres, etc ...).

Les combats se feront au tour par tour. Avec la souris, le joueur devra choisir sa stratégie, soit se battre et choisir la bonne combinaison d'attaques et de défenses, soit s'enfuir. Il pourra aussi, s'il le souhaite, accélérer les animations de combats pour fluidifier le jeu.

Nous garderons un système très simple comme tout RPG, avec :

- Un inventaire,
- Une fiche de personnages, avec sa classe et ses statistiques, influençant par exemple, les dégâts subis,
- Un système de dialogue, qui permet de découvrir petit à petit la langue des monstres que nous rencontrons à travers d'énigmes.

### 3.1 Scénario du jeu

Un astronaute se réveille après un sommeil cryogénique. Il se rend compte qu'une voix lui parle. Cette voix, celle d'un inconnu auquel il découvrira le visage à la fin du jeu, lui explique ce qu'il fait sur cette planète et ce qu'il doit faire pour rentrer chez lui. Elle lui explique aussi à contrôler son corps après plusieurs années de sommeil. L'astronaute sort de sa capsule, et voit un monde nouveau s'ouvrir à lui, avec sur son casque, des indications pour remplir sa mission d'astronaute. Tout d'abord, aller voir un habitant qui



---

collabore assez facilement, étant donné qu'il a été informé par l'organisation des astronautes, de la présence de celui-ci.

L'astronaute a pour première mission de récupérer une clé dans une grotte. Il devra pour la première fois traverser un portail qui l'amènera vers cette grotte et un autre qui lui permettra de revenir à l'endroit où il était. Après cela, l'habitant lui indique qu'il a déverrouillé la porte du premier donjon. En effet, tant qu'il n'a pas récupéré la clé, le joueur ne peut pas entrer dans le donjon. L'astronaute va dans le donjon, un coffre est présent et une atmosphère glaciale l'entoure, il avance, ... et tout d'un coup un monstre vient vers lui, le touche et alors commence un combat entre lui et le monstre. Une fois tué, l'astronaute peut traverser le portail qui l'amènera dans une salle des coffres. Dans cette salle, se trouve un autre portail qui l'emmène dans une forteresse contrôlée par les monstres. Un commandant de la résistance l'accueillera lui disant de chercher un portail pour trouver une clé qui lui permettra de déverrouiller le dernier portail présent sur la grande place de la forteresse, synonyme de fin du jeu. Alors le joueur traverse la ville, trouve le portail, sauf que le portail ne l'emmène pas dans le second donjon, mais dans un labyrinthe. L'astronaute devra en sortir, avant de pouvoir entrer dans ce donjon. Une fois toutes ces épreuves passées, le joueur revient vers la forteresse et le commandant lui indique qu'il peut rentrer chez lui. Traverser le grand portail, lui ouvrira la voix pour rentrer chez. L'astronaute le traverse, et est accueilli par la voix du début. Il se rend compte d'un guet-apens, l'astronaute en sait trop sur cette planète et ils veulent le tuer pour garder des informations confidentielles. Le jeu se termine après la mort de l'astronaute.

## 3.2 Évolutions et Fonctionnalités Implémentées

Tout au long du développement d'Echo of Nyxalis, nous avons apporté de nombreuses modifications et améliorations qui ont transformé notre projet initial en une véritable aventure jouable, im-



---

mersive et riche en mécaniques.

Dès le début, nous avons effectué une refonte complète de l'interface de démarrage du jeu. L'ancien menu, trop complexe, a été remplacé par une interface plus simple dans un style rétro-cartoon coloré. L'objectif était clair : donner envie aux joueurs de découvrir notre univers dès l'écran titre. Les boutons ont été retravaillés, les polices adaptées à notre charte graphique, et l'ensemble a été rendu plus lisible et intuitif.

L'interface utilisateur (UI) au lancement du jeu a également été améliorée, pour mieux guider le joueur dans ses premiers choix : solo ou multijoueur, sélection de personnage, début d'aventure. Chaque interaction a été pensée pour être à la fois ergonomique et immersive.



---

### 3.3 Conception générale

Echo of Nyxalis repose sur une mécanique centrale : l'exploration d'un monde inconnu et fragmenté, à la recherche des souvenirs du personnage principal. Le joueur incarne un être mystérieusement réveillé sur une planète étrange et hostile, sans mémoire ni repère, et devra parcourir différents environnements interconnectés, en quête de vérité... et de survie.

Le cœur de cette expérience repose sur un système de donjons accessibles via des portails dimensionnels, chacun menant à une portion différente de la planète Nyxalis. Ces donjons forment une structure narrative non linéaire, dans laquelle chaque zone explore un thème visuel, sonore et scénaristique spécifique.

Le déroulement de l'aventure s'articule autour de plusieurs étapes :

- Le joueur apparaît dans un environnement spécifique (un "hub"), guidé par des PNJ
- Il est invité à explorer les lieux, récupérer des objets essentiels à sa progression, résoudre des énigmes, et se préparer aux combats à venir.
- Des coffres cachés sont disséminés dans les environnements : ils contiennent de quoi se soigner, renforcer ses chances contre les boss mais aussi des souvenirs perdus.
- Chaque environnement se conclut par un donjon contenant un boss. Une fois celui-ci vaincu, un portail vers une nouvelle zone s'ouvre, menant à la suite de l'histoire.

Le jeu comprend à terme :

- 4 à 5 environnements distincts à explorer



- 
- 2 boss principaux
  - Des labyrinthes générés aléatoirement
  - Des hubs scénarisés servant de base de préparation entre les affrontements

Ce cycle exploration → préparation → boss → narration constitue la structure de notre gameplay.

### 3.4 Relier les salles : Portails et transitions

Pour circuler d'un environnement à l'autre ou entre différentes scènes du même environnement, le joueur utilise des portails mystiques. Ceux-ci sont verrouillés par défaut et nécessitent généralement une clé ou une action spécifique pour être ouverts (finir un combat, parler à un PNJ, résoudre une énigme...).

Chaque salle ou scène contient un portail activable, ce qui nous a demandé une bonne gestion de la navigation inter-scènes dans Unity. Cela a représenté un défi technique : chaque scène devant conserver l'état du joueur, des objets, de l'inventaire, etc.

Les portails ont été conçus pour être à la fois anciens et mystérieux, leur esthétique s'intègre parfaitement à l'univers rétro-futuriste de Nyxalis : un mélange de magie ancienne et de technologie oubliée.

### 3.5 Exploration & Donjons

L'exploration repose sur un système de portails interconnectés, qui permettent aux joueurs de voyager entre plusieurs zones et donjons. En s'approchant des ennemis présents dans le donjon, une phase de combat se déclenche automatiquement et les monstres si vous les approchez, vous chasseront en retour.



---

Ces donjons renferment des coffres cachés disséminés dans les pièces, qui contiennent des objets de soin, de l'équipement ou des ressources utiles à la survie. Cela incite les joueurs à bien fouiller dans chaque recoin.

### 3.6 Génération aléatoire et seed

La structure des donjons repose sur une génération procédurale basée sur une "seed" (valeur aléatoire partagée). Cette approche permet à tous les joueurs d'un groupe multijoueur de parcourir le même labyrinthe, sans désynchronisation.

La seed contrôle :

- La disposition des salles
- La configuration des murs et couloirs
- L'apparition des coffres cachés
- La présence d'ennemis ou d'objets spéciaux

Cette méthode permet une grande variété d'expériences tout en garantissant une accessibilité et une cohérence structurelle (aucune salle ne reste bloquée ou inaccessible). Cela évite aussi une surcharge de calcul, car la topologie du donjon est préétablie en amont, dès le lancement de la scène.

## 4 Systèmes du Jeu

### 4.1 Système de dialogue narratif

Nous avons également intégré un système de dialogue immersif avec :





- 
- Une voix off narrative qui accompagne le joueur tout au long de l'histoire.
  - Des personnages non-joueurs (PNJ) récurrents qui enrichissent le lore du jeu.

Ce système renforce la dimension narrative du jeu, en donnant des objectifs, en distillant des indices, ou en racontant l'histoire de la planète Nyxalis et de ses habitants.

Pour implémenter le système de dialogue, nous avons utilisé un asset Unity appelé Yarn Spinner. Yarn Spinner est à la fois très pratique car les Yarn scripts (le code des dialogues) sont simples à rédiger. Les dialogues sont composés d'un composant Unity appelé Canvas qui permet l'affichage du texte et de plusieurs autres composants, le tout englobant ce qu'on appelle le Dialogue Runner. Pour simplifier, on dira que c'est le système de dialogue. Dans ce composant, on retrouve aussi un bouton "continuer" qui permet à l'utilisateur de passer à la suite des dialogues. L'affichage du texte se fait en temps réel de gauche à droite.

De plus, pour rendre le contenu plus interactif, l'utilisateur aura le choix de ses réponses grâce à un système de choix de dialogue. Ainsi, par exemple, s'il veut apprendre à jouer au jeu, le joueur peut décider de suivre le didacticiel ou non.

Les problèmes rencontrés avec les dialogues : Les difficultés rencontrées ont été nombreuses. Le système de dialogue nous a causé bien des soucis pour l'implémenter. D'une part, dans le jeu, un dialogue s'affiche avec un bouton "continuer". Sauf que ce bou-



---

ton "continuer" ne fonctionnait que quand le joueur n'était pas dans la scène, à cause du fait que la souris était comme bloquée par le joueur (astronaute). Il nous a fallu du temps pour comprendre le problème. Nous avons donc remédié à la situation en proposant une autre approche avec des dialogues simples qui tiennent sur une feuille de dialogue dans les scènes où le joueur contrôle l'astronaute, tandis que dans les autres scènes, nous nous sommes permis d'espacer le plus possible le texte afin de rendre le contenu plus interactif. De plus, nous avons rajouté bon nombre de choix pour l'utilisateur afin de rendre le jeu plus vivant. Enfin, nous avons rencontré des soucis avec le changement de scène entre le didacticiel et la deuxième scène composée uniquement de dialogues. En effet, une nouvelle fois, nous avons eu un problème à cause de la souris, car elle était présente mais n'interagissait pas avec le dialogue. Là encore, le problème fut assez difficile à cerner, car le joueur n'était présent que sur la scène du didacticiel et non sur l'autre. C'est là que nous avons compris que certaines choses, bien que techniquement uniquement présentes dans une scène, pouvaient en réalité impacter l'autre scène.

## 4.2 Refonte de l'inventaire

Concernant l'inventaire, nous sommes partis d'un prefab d'UI Unity, que nous avons totalement retravaillé. Nous avons écrit un nouveau code pour la gestion des objets, en veillant à ce que tout soit cohérent avec l'univers graphique et fonctionnellement fluide.

Le principal défi technique a été de synchroniser les inventaires dans le cadre du multijoueur : chaque joueur devait avoir son propre inventaire, même s'il interagit avec un coffre commun. Il fallait aussi que ces inventaires persistent même en changeant de scène, ce qui a nécessité une bonne structuration du code et des données.



---

## Implémentation technique

L'implémentation de l'inventaire a commencé par une structure simple : une classe Inventory contenant une liste d'objets (items) en possession du joueur. Cependant, dès que nous avons intégré différents types d'objets – comme des armes, des équipements ou des potions – nous avons dû enrichir considérablement la structure de base pour prendre en compte de nouvelles propriétés : type, effet, description.

Au fur et à mesure, cette fonctionnalité s'est révélée beaucoup plus complexe que prévu. Il a fallu prévoir :

- Une gestion dynamique des types d'objets
- Une logique d'affichage et d'interaction graphique
- Une synchronisation avec d'autres éléments du jeu (combat, interface, multijoueur)

Chaque objet devait être visuellement identifiable, ce qui nous a conduit à créer une icône spécifique pour chaque type d'item disponible. Par exemple :

- Un parchemin devrait afficher son modèle
- Une potion de soin son flacon distinctif

Les potions sont des objets à usage unique. Lorsqu'un joueur utilise une potion, celle-ci disparaît automatiquement de l'inventaire après avoir appliqué son effet (généralement un gain de points de vie).

Le système d'ouverture et de fermeture de l'inventaire a été mappé sur la touche E. Lorsqu'on appuie sur E, l'inventaire s'ouvre en surimpression. Une seconde pression sur E referme l'interface



---

et reprend le gameplay en temps réel.

Les coffres, quant à eux, ne peuvent être ouverts que si le joueur est suffisamment proche, et en appuyant sur la touche F. Une fois ouverts, le joueur doit glisser les objets dans leur inventaire en maintenant le clic gauche. Pour refermer le coffre, il suffit d'appuyer sur la touche F.

L'inventaire prend également en compte :

- Le clic droit sur un objet pour l'utiliser directement (ex. : consommer une potion)
- La possibilité de déplacer les objets d'une case à une autre, par drag & drop

## Conception graphique

Visuellement, l'inventaire est organisé sous forme de grille régulière. Il comporte :

- 12 emplacements organisés en 4 rangées de 3 cases
- Chaque case est un carré de même dimension, espacé de façon homogène
- Un petit rectangle marron dans chaque case représente la zone d'aperçu de l'icône

Le plus difficile dans cette partie a été de rendre les cases réactives aux clics :

- Lorsqu'un joueur clique sur une case, il fallait détecter précisément quelle case est cliquée
- Puis exécuter l'action liée à l'objet qu'elle contient (utilisation, déplacement, affichage)



---

Une fois cette interaction fonctionnelle, nous avons décliné la logique pour chaque type d'équipement, avec quelques ajustements spécifiques selon les objets.

## Informations contextuelles des objets

Pour rendre l'inventaire plus lisible et stratégique, nous avons mis en place un système d'affichage contextuel :

- Lorsque la souris survole une case, et que cette case contient un objet,
- Une fenêtre d'information apparaît, indiquant :
  - Le nom de l'objet
  - Une courte description
  - Et ses effets concrets (ex. : "Soigne 100 PV", "Augmente la défense de 20% pendant 10 secondes")

Cette fonctionnalité était cruciale pour aider le joueur à prendre des décisions éclairées, notamment pendant les phases de préparation avant un boss ou en plein combat.

## Objets utilisables

Nous avons développé une première série d'objets consommables utilisables pendant ou en dehors des combats. Par exemple :

- Une pomme : soigne une petite quantité de PV
- Une soupe : soigne davantage mais prend plus de temps à utiliser

Ces objets s'intègrent à l'inventaire et au système de combat, offrant au joueur plus de contrôle et de stratégie.

## Gestion des objets et inventaire

Nous avons repensé complètement le système d'objets. Désormais, les joueurs peuvent :

- Collecter des objets : soins, améliorations, équipements...
- Les visualiser dans une nouvelle interface d'inventaire plus claire et plus esthétique



- 
- Déplacer les objets via un système de glisser-déposer
  - Utiliser des objets pendant les combats, ce qui ajoute une dimension stratégique

Cette interface a été développée à partir d'un préfab Unity, mais totalement re-codée et stylisée pour s'adapter à notre univers. Le système prend aussi en compte la désynchronisation des inventaires entre les joueurs en mode multijoueur, et les objets restent persistants d'une scène à l'autre.

## Défis techniques

Le principal problème rencontré ici fut :

- La récupération précise de la position de la souris
- L'affichage correct du cadre d'information (notamment quand la case est trop proche du bord droit ou inférieur de l'écran)

Il a fallu prévoir des ajustements dynamiques pour éviter que le texte ne sorte de l'écran ou ne soit coupé. Ces détails ont demandé beaucoup de tests et de calibrages manuels, mais le résultat final est aujourd'hui fluide et propre à l'utilisation.

De plus, l'implémentation de l'utilisation des objets en combat s'est révélée complexe, car je n'étais pas responsable de cette partie du projet. J'ai donc dû analyser le code existant, le modifier avec précaution pour éviter toute régression, et m'assurer que tout fonctionnait correctement. Par ailleurs, j'ai également dû améliorer la gestion de l'inventaire afin que l'objet utilisé disparaisse après usage, ce qui s'est avéré délicat à mettre en œuvre.

## 4.3 Système de combat 2D

Le cœur du gameplay repose sur un système de combat au tour par tour :

- Le joueur peut choisir son attaque, se défendre ou utiliser des objets



- 
- Chaque ennemi a ses propres caractéristiques et comportements
  - La stratégie est de mise pour survivre, surtout face aux boss

Ce système est déclenché lorsqu'un ennemi repère le joueur dans le donjon : il le pourchasse activement, grâce à une intelligence artificielle (IA) développée spécifiquement. Une fois le combat terminé, un coffre de récompense apparaît.

## 4.4 Progression & Histoire

Le joueur traverse les différents environnements via des portails dimensionnels, chaque zone ayant une esthétique et une ambiance propres. L'histoire progresse à mesure qu'il affronte des donjons, résout des énigmes, écoute les voix narratives et bat les boss principaux. L'objectif final est de comprendre ce qui est arrivé à la planète Nyxalis et trouver un moyen de retourner chez soi.

## 4.5 Ambiance sonore et musicale

Nous avons enrichi le jeu avec une ambiance sonore dynamique, composée de :

- Musiques de fond qui changent selon l'ambiance (exploration, danger, combat, boss...)
- Effets sonores pour les attaques, soins, interactions avec les objets et les environnements

Cela a été possible grâce à l'intégration de boucles issues de Logic Pro X et d'effets sonores conçus ou adaptés pour correspondre à notre univers narratif.

Un orchestre violet vous accompagnera pendant votre exploration :

- Une bande-son bleue, mystique, calme et mythique pour vous aider à résoudre les énigmes les plus complexes,



- 
- Une bande-son rouge, sanguine, rock et dynamique pour vous plonger dans les batailles les plus ardues.

Quant à nous, nous réalisons que créer un jeu est un projet complexe. Il ne s'agit pas que de programmer, mais d'écrire une histoire, trouver un design, définir une charte graphique, composer des bandes-sons, etc. Cette complexité fait que nous sommes plus que conscients qu'il s'agit d'une opportunité parfaite pour notre équipe de monter en compétences et de mener un projet informatique de bout en bout.

L'une des premières tâches a été de gérer la partie audio du jeu. Cela n'a pas été aussi simple qu'on pourrait le penser. Le principal défi a été de trouver des musiques cohérentes avec l'ambiance du jeu, qui changent selon les environnements — exploration, donjon, boss, etc. Nous ne voulions pas de simples boucles musicales génériques.

Pour composer et assembler les sons, nous avons utilisé Logic Pro X, un logiciel professionnel de création musicale développé par Apple. Ce choix s'est imposé car il propose une bibliothèque intégrée très riche appelée Apple Loops, qui offre de nombreuses boucles sonores prêtes à l'emploi. Ces ressources ont été précieuses pour établir une base sonore cohérente et immersive.

Nous avons ensuite ajusté les transitions entre les musiques selon les scènes et ajouté des effets sonores pour les interactions et les combats. L'ensemble vise à renforcer l'immersion et donner au jeu une vraie personnalité auditive.





---

Lors de la conception de l'ambiance sonore d'Echo of Nyxalis, nous avons principalement utilisé Logic Pro X, un outil professionnel de création musicale. Bien que ce logiciel offre une bibliothèque très fournie (Apple Loops), nous avons rapidement constaté une limitation importante : beaucoup de boucles sonores étaient génériques, peu nuancées, ou simplement déconnectées de l'ambiance étrange, mystique et rétro-futuriste que nous cherchions à transmettre.

Certaines boucles étaient trop lisses, manquaient de texture, ou évoquaient davantage une ambiance médiévale ou urbaine, mais pas l'atmosphère inquiétante, froide et mystérieuse de la planète Nyxalis.

Pour dépasser ces limites, nous avons opté pour une autre approche : Nous avons commencé par superposer plusieurs boucles, jouant sur les couches sonores pour créer des textures musicales uniques.

Nous avons utilisé les outils intégrés de Logic Pro X pour personnaliser chaque boucle :

- Ajustement des fréquences (EQ) pour mettre en avant ou atténuer certaines tonalités
- Ajout de filtres low-pass/high-pass pour créer des effets d'éloignement



---

ou de flou sonore

- Réverbérations, délais, pitch shifting pour donner de la profondeur et créer une ambiance onirique ou oppressante

Cette méthode nous a permis de transformer des éléments ordinaires en paysages sonores immersifs et cohérents avec notre direction artistique.

Par exemple :

- Une boucle de piano neutre est devenue un motif inquiétant en ralentissant son tempo et en lui appliquant une réverbération métallique.
- Un simple beat électronique a été transformé en ambiance rythmique tribale futuriste en le combinant à des échantillons filtrés de percussions synthétiques.

Malgré des ressources limitées, ces ajustements nous ont permis de composer des bandes-son originales, atmosphériques et bien ancrées dans l'univers de notre jeu.

Le résultat final est une ambiance musicale modulaire, où chaque environnement possède sa propre identité sonore, évoluant en fonction de l'action (exploration, combat, boss...).

À ce stade, les morceaux de musique de fond et de combat sont finalisés et parfaitement intégrés à l'univers du jeu. Les retours de l'équipe sur ces compositions sont positifs, notamment en ce qui concerne :



- 
- L'atmosphère futuriste et immersive des morceaux
  - La fluidité des transitions, qui contribuent à renforcer l'immersion du joueur
  - L'équilibre entre dynamisme et mystère, qui enrichit l'expérience globale

## 5 Interfaces et Utilisabilité

### 5.1 Menu principal

#### Fonctionnalités du menu

Le menu principal du jeu a été conçu pour être simple, intuitif et rapide à utiliser. Il comprend quatre boutons principaux :

- Solo : lancer une partie solo, avec choix du personnage
- Tuto : accéder à une introduction guidée au gameplay
- Quit : quitter le jeu

Notre objectif était d'offrir une expérience utilisateur fluide, en limitant les obstacles entre le lancement du jeu et le gameplay effectif.

#### Le fond visuel

L'arrière-plan du menu a été soigneusement choisi : il représente la première scène explorée, avec l'astronaute au centre, prêt à partir à l'aventure. Ce visuel permet d'ancrer immédiatement le joueur dans l'ambiance du jeu, en évoquant à la fois solitude, mystère et découverte.



---

## Le titre

Le titre du jeu est affiché avec notre logo et une police travaillée. Pour garantir un rendu de qualité quel que soit l'écran ou la résolution, nous avons utilisé un système d'anti-aliasing qui améliore la netteté et la lisibilité du texte.

## Thème graphique de l'interface

L'interface utilisateur utilise une palette de couleurs, symbole de mystère, d'inconnu et de magie. La police, simple et lisible, permet de conserver une identité visuelle forte tout en restant accessible à tous les joueurs.

## 5.2 Le Tutoriel

### Objectifs et utilité pédagogique

Suite aux premiers retours utilisateurs obtenus lors des phases de test, il est apparu que certains joueurs étaient perdus lors de leurs premières minutes de jeu, notamment en ce qui concerne les déplacements, l'interaction avec les objets ou l'utilisation de l'inventaire. Afin d'améliorer l'accessibilité de notre jeu et d'offrir une expérience plus fluide dès les premières minutes, nous avons décidé d'intégrer un niveau tutoriel.

Ce tutoriel a pour but :

- D'enseigner les mécaniques de base du jeu de manière progressive
- De familiariser le joueur avec les commandes (déplacements, saut, inventaire...)



- 
- De créer une première immersion douce dans l'univers de Nyxalis, avec un niveau à part, simple mais cohérent visuellement
  - D'éviter toute frustration inutile et d'assurer une prise en main agréable, même pour un joueur non habitué aux RPGs ou aux jeux 2.5D

## Conception du tutoriel : une progression par étapes

Le niveau tutoriel prend la forme d'un mini-parcours guidé dans un environnement simple mais cohérent avec l'univers du jeu. Chaque mécanique y est introduite progressivement, de façon naturelle, à travers des obstacles et éléments interactifs.

Voici les principales étapes :

**Étape 1 — Déplacements de base** Le joueur commence dans une zone dégagée, sans danger. Une voix narrative l'invite à se déplacer à l'aide des touches Z, Q, S, D (ou WASD selon la configuration clavier). Il ne peut progresser vers l'étape suivante que lorsqu'il a bien assimilé ces déplacements.

**Étape 2 — Le saut** Le joueur se retrouve face à des tables ou des obstacles bas qui bloquent le chemin. Pour continuer, il doit utiliser la touche Espace afin de sauter par-dessus. Cette section force l'expérimentation et garantit que le saut a été compris avant d'avancer.

**Étape 3 — Ouverture de coffre** Une fois l'obstacle franchi, le joueur découvre un coffre scellé. Il est invité à interagir avec celui-ci (par exemple avec la touche F) pour l'ouvrir et récupérer



---

un objet de soin ou une ressource.

**Étape 4 — Utilisation de l'inventaire** Après avoir obtenu un objet, une indication visuelle (ou sonore) invite le joueur à ouvrir son inventaire. Il y apprend :

- À naviguer dans l'interface
- À sélectionner un objet
- À utiliser un objet (clic droit, touche raccourcie...)

Cela permet de tester en condition réelle la gestion d'objets, essentielle pour survivre lors des futurs combats dans les donjons.

## Avantages pédagogiques

Le tutoriel agit comme une zone d'apprentissage sécurisé, où le joueur ne risque ni combat ni sanction. Cela crée un sentiment de contrôle et de compréhension, qui prépare à la complexité progressive du jeu.

Par ailleurs, le tutoriel est :

- Court (1 à 2 minutes)
- Narratif (introduit par la voix off d'un PNJ)
- Visuellement cohérent
- Réutilisable pour les sessions multijoueur si besoin



---

## 6 Esthétique

Dans cette section, nous vous présentons les modélisations 3D des différents assets utilisés pour la création de notre jeu vidéo dans le but d'ancrer notre univers dans une esthétique. Unique et immersive, nous avons choisi pour cela un style visuel à la fois mystérieux et fantastique, en opposition totale avec le réalisme. Tous les modèles 3D ont été créés dans cette optique, en accordant une attention particulière aux détails et à l'optimisation pour garantir de bonnes performances, une partie a été réalisée par nos soins directement, tandis que d'autres éléments ont été récupérés sur l'asset store de Unity, afin de nous faciliter la tâche, notamment en raison d'un manque de temps. Notre objectif était de créer un univers fantastique et aventureux. Nous avons donc tenté de transmettre cette ambiance à travers nos personnages ainsi que dans l'environnement global du jeu. Il est également important de préciser que tous les modèles, ainsi que les matériaux que nous avons créés, ont été réalisés avec Blender. Dans le cadre du projet, nous avons eu l'opportunité de créer en 3D les personnages. Pour cela, nous avons utilisé Blender, un outil puissant et très complet, qui a permis de modéliser avec précision les moindres détails : silhouette, combinaison, accessoires, visage... L'objectif était double : concevoir un personnage visuellement marquant et fonctionnel pour les animations. Nous avons accordé une grande attention à la cohérence des couleurs avec l'univers visuel de Nyxalis. Une fois la modélisation terminée, nous avons ajouté un squelette (rigging) pour animer le personnage : mouvements des bras, des jambes, des doigts, déplacements réalistes... Cela nous a permis de créer des animations naturelles pour les actions principales (marche, saut, interactions).

### 6.1 Esthétique générale

#### La végétation

L'un des plus gros défis techniques a été la modélisation des arbres. En tant que débutants sur Blender, nous avons rencontré des problèmes pour structurer efficacement les feuilles et gérer l'impact



---

visuel sans alourdir le rendu.

Les buissons sont une déclinaison simplifiée du modèle d'arbre, reprenant les mêmes techniques. Pour finir, nous avons ajouté de petites herbes colorées, qui renforcent la densité de l'environnement et rappellent la palette chromatique de Nyxalis.

## 6.2 Création des personnages

### Le personnage principale : l'astronaute

Nous avons commencé par créer le héros principal de notre jeu : un astronaute. Il s'agit du tout premier personnage que nous avons modélisé, ce qui représentait une étape cruciale dans le développement de notre projet. À ce moment-là, nous n'avions aucune expérience en modélisation 3D, ce qui a rapidement mis en évidence la complexité et le temps que cette tâche allait nécessiter. Cette première étape a donc été marquée par un apprentissage progressif, ponctué de nombreuses recherches, essais et erreurs.

Malgré notre manque de compétences initiales, nous avons réussi à concevoir un personnage dont l'apparence se rapproche assez fidèlement de l'idée que nous avions en tête au départ. Le style visuel que nous voulions donner à notre jeu un univers fantastique et immersif a guidé nos choix esthétiques, même si certaines concessions ont dû être faites en raison de nos limites techniques.

Une fois la modélisation terminée, nous avons abordé l'étape suivante : la mise en place d'un squelette pour permettre l'animation





---

du personnage. Là encore, aucun membre de l'équipe n'avait d'expérience dans ce domaine, ce qui a rendu l'exercice particulièrement complexe. Il a fallu plusieurs tentatives, corrections et ajustements avant de parvenir à un squelette complet et fonctionnel, capable de permettre des mouvements cohérents et naturels à notre astronaute.

L'étape de l'animation a elle aussi apporté son lot de défis. Parmi les problèmes rencontrés, nous avons notamment observé une désynchronisation entre les jambes lors de la marche l'une se déplaçait plus rapidement que l'autre ou encore un léger mouvement résiduel du personnage, même lorsqu'il était censé être immobile. Ces bugs, bien que frustrants au début, nous ont permis d'affiner notre compréhension du processus d'animation et, après quelques ajustements supplémentaires, nous avons réussi à obtenir une animation satisfaisante et stable.

## **Le guide du héros**

Vient ensuite un autre personnage clé de notre jeu : le guide du héros. Ce personnage occupe une place centrale dans l'univers que nous avons imaginé puisqu'il accompagne le héros tout au long de son aventure en lui apportant une présence rassurante. Sa création a représenté un défi encore plus grand que celle du héros principal. En effet, le guide est nettement plus complexe sur le plan visuel, notamment en raison de son apparence plus humanoïde, qui exige un niveau de détail plus élevé pour obtenir un rendu crédible.

Contrairement au héros, dont l'équipement d'astronaute permettait une certaine simplification, le guide se distingue par une esthétique très particulière. Ses vêtements sont en partie constitués de plantes, ce qui évoque directement l'aspect fantastique de notre



---

univers, fortement inspiré par la nature, les forêts mystérieuses et la magie. Ce choix de design visait à renforcer l'ambiance immersive de notre jeu, mais il a également complexifié la modélisation, car chaque élément végétal devait être intégré de manière cohérente et visuellement harmonieuse.

Un autre aspect ayant demandé une attention toute particulière est le visage du personnage. Contrairement au héros, dont le visage est caché sous un casque, le guide possède des traits visibles, ce qui a nécessité un travail plus fin au niveau de la topologie, des textures et de l'expression faciale. Donner vie à un visage crédible et expressif est un exercice délicat, surtout avec notre niveau d'expérience limité, mais cela faisait partie intégrante de l'identité de ce personnage.

La conception de ce personnage a donc représenté un véritable objectif pour notre équipe. Elle nous a demandé non seulement beaucoup de temps, mais aussi un effort d'imagination important pour concevoir une silhouette et une allure qui soient à la fois originales, cohérentes avec notre univers et techniquement réalisables avec nos compétences.

Comme pour le héros, la phase de création du squelette et d'animation a été ponctuée de difficultés. Bien que nos compétences dans ce domaine se soient améliorées depuis la création du premier personnage, nous avons tout de même rencontré plusieurs problèmes. Certains mouvements ne se comportaient pas comme prévu, et l'intégration du squelette avec les nombreux éléments



---

du personnage, comme les plantes, a posé des défis particuliers. Malgré cela, nous avons su tirer parti de notre expérience précédente pour corriger la plupart des erreurs plus rapidement et obtenir une animation fonctionnelle.

## **Le méchant**

Enfin, nous arrivons à la création du dernier personnage majeur de notre jeu : le méchant présent dans les donjons. Ce personnage incarne l'antagoniste que le joueur rencontre au cours de son exploration, et représente un élément clé dans l'intensité et le rythme des phases de combat. Cependant, sa conception a été particulièrement difficile, non pas sur le plan technique, mais sur le plan créatif.

En effet, nous avons eu beaucoup de mal à imaginer une apparence convaincante pour ce personnage. Contrairement aux autres, nous ne disposions d'aucune idée précise de son design. Le manque d'inspiration, combiné à une contrainte de temps importante à ce stade du projet, nous a poussés à faire un choix différent pour ce personnage. Il s'agit donc du seul modèle pour les personnages que nous n'avons pas conçu nous-mêmes, mais que nous avons récupéré sur l'asset store de Unity. Cette décision nous a permis d'économiser un temps précieux tout en intégrant un ennemi visuellement adapté à l'univers du jeu.

Le monstre que nous avons choisi se présente sous la forme d'un œil géant, accompagné de cornes et d'une texture verdâtre, évoquant un aspect à la fois monstrueux et démoniaque. Ce design contribue à renforcer l'ambiance sombre et inquiétante des donjons, tout en marquant un contraste avec les autres personnages plus "humains" du jeu. Sa forme étrange et intimidante



---

correspond bien à ce que l'on peut attendre d'un gardien de donjon ou d'un ennemi surnaturel.

Contrairement aux deux autres personnages, ce monstre ne possède pas de squelette ni d'animations complexes. Il ne bénéficie pas d'un squelette créé par nous, car son comportement repose uniquement sur des mouvements simples gérés par l'intelligence artificielle. Il avance de manière directe vers le joueur dans le but de l'attaquer, ce qui, malgré sa simplicité, suffit à instaurer une certaine pression et à dynamiser les séquences de jeu dans les donjons.

Ce personnage, bien qu'emprunté et plus simple sur le plan technique, remplit pleinement son rôle dans le gameplay. Il complète efficacement notre galerie de personnages, et son ajout démontre aussi notre capacité à faire des choix pragmatiques en fonction des contraintes du projet.

## **Décorations et objets divers**

Notre univers de jeu est enrichi par la présence de divers objets qui contribuent à lui donner vie et à renforcer l'immersion du joueur. Parmi ces éléments, on retrouve notamment des portails, des coffres et une clé, chacun jouant un rôle dans la progression de l'aventure et son côté réaliste. Ces objets permettent au joueur d'interagir avec son environnement, de débloquent de nouvelles zones et d'avancer dans son parcours, tout en rendant l'univers plus dynamique et cohérent.

## **Difficultés rencontrées**

- Mon matériel informatique était parfois insuffisant pour gérer des scènes complexes sous Blender, ce qui ralentissait fortement mon travail.



- 
- J'ai dû consacrer beaucoup de temps à rechercher des tutoriels techniques et à poser des questions sur des forums pour progresser.
  - Le choix des couleurs pour la combinaison n'a pas été simple : il a fallu ajuster plusieurs fois la palette pour obtenir un résultat harmonieux avec l'environnement du jeu.

Malgré ces défis, cette tâche m'a permis de progresser énormément en modélisation 3D et d'acquérir des compétences précieuses que je pourrai réutiliser à l'avenir.

## 7 Ressources et Techniques

### 7.1 Les assets

Pour la conception de l'environnement global de notre jeu nous avons 4 assets spécifiques :

#### **Donjons**

Pour nos donjons, nous nous sommes fortement inspirés de l'imaginaire véhiculé par les films de fantasy, les jeux vidéo et diverses œuvres visuelles. Après avoir mené plusieurs recherches pour trouver des éléments correspondant à l'ambiance que nous voulions instaurer, nous avons identifié un certain nombre de modèles disponibles sur l'asset store. En combinant ces ressources avec nos idées initiales, nous avons pu créer des donjons uniques, fidèles à l'univers que nous souhaitions construire. Les donjons sont principalement composés de murs en pierre, d'arches massives et de portes en bois, ce qui correspond parfaitement à l'atmosphère médiévale et mystérieuse que nous cherchions à retranscrire. Ce mélange de textures et d'architectures renforce le sentiment d'exploration et d'immersion, tout en s'inscrivant dans les codes classiques du genre.



---

## Forteresse

Une forteresse qui constitue une nouvelle étape dans leur progression. Ce village sert de point de transition, permettant au héros de poursuivre ses quêtes, de rencontrer d'autres personnages, ou de se préparer à de nouvelles aventures afin d'accéder à encore bien d'autres donjons. Pour concevoir cet espace, nous recherchions un modèle simple et cohérent avec notre univers. Après plusieurs essais, nous avons finalement trouvé un modèle sur l'asset store qui a conquis l'ensemble de l'équipe. Son esthétique médiévale typique, avec ses petites maisons et des chemins, qui a été modifié pour avoir un style plus sombre en modifiant les murs en noirs, le sol en rouge, etc. . . pour qu'il s'intègre parfaitement à notre monde.

## Extérieurs

Les extérieurs, nous avons intégré un modèle de forêt féerique particulièrement réussi. Ce décor, riche en végétation, baigné de lumière douce et ponctué de détails magiques, s'est avéré idéal pour renforcer le côté enchanteur de notre univers. Il correspondait exactement à ce que nous avions en tête, voire au-delà de nos espérances, et nous avons estimé qu'il représentait parfaitement l'esprit de notre jeu. Ajoutez à cela nous avons mis divers installations comme un vaisseau spatial, deux portails ainsi qu'un hangar que nous verrons juste après.

## Bibliothèque

Une bibliothèque stylisée en couleur or, avec des motifs tels que la disposition de coffres pour en faire la salle des coffres.



---

Nous avons utilisé un asset de monstre auquel nous avons personnalisé ses animations pour qu'elles soient cohérentes avec notre jeu.

De plus, les portails proviennent eux aussi d'un asset.

## Personnalisation de l'environnement

La personnalisation de l'environnement, surtout la création d'une sorte de hangar isolé dans la forêt n'a pas été chose aisée. En effet, nous avons dû utiliser ProBuilder, un package inclus dans Unity, pour creuser un trou dans une montagne que nous avons déjà grâce à l'environnement. Ensuite, il nous a fallu créer un composant Arch, puis Plane, etc., pour rendre le tout harmonisé avec l'environnement de départ. Nous avons aussi mis un personnage pour accueillir le joueur ainsi qu'une maison (provenant d'un asset). Cet élément joue un rôle essentiel dans la structure du jeu : c'est depuis ce bâtiment que le joueur peut obtenir la clé permettant de débloquent l'accès aux donjons. Son emplacement et son aspect contrastent légèrement avec l'environnement naturel qui l'entoure, ce qui le rend facilement repérable. Ce choix de design vise à guider le joueur de manière intuitive, sans avoir besoin d'indications trop visibles ou intrusives car cela était un réel problème de savoir comment indiquer et guider notre héros.

Notre univers de jeu est enrichi par la présence de divers objets qui contribuent à lui donner vie et à renforcer l'immersion du joueur. Parmi ces éléments, on retrouve notamment des portails, des coffres et une clé, chacun jouant un rôle dans la pro-



---

gression de l'aventure et son côté réaliste. Ces objets permettent au joueur d'interagir avec son environnement, de débloquent de nouvelles zones et d'avancer dans son parcours, tout en rendant l'univers plus dynamique et cohérent. Nous avons aussi mis un personnage pour accueillir le joueur ainsi qu'une maison (provenant d'un asset).

De plus, l'importation d'assets, surtout ceux de l'environnement principal et du village, nous a posé des soucis dus à des problèmes de matériaux et de textures. En effet, tout apparaissait en rose. Ce n'est qu'après mûre réflexion que nous avons compris qu'il fallait changer les matériaux des préfab (de l'asset importé), de Standard à Universal Pipeline Renderer/Lit. Ensuite, les textures n'étaient la plupart du temps pas présentes, nous avons donc dû improviser, créer nos propres matériaux ou reprendre ceux qui étaient déjà là pour les appliquer à tous les préfab.

Enfin, des murs invisibles sont présents dans l'environnement afin que le joueur ne tombe pas dans le vide.

## 7.2 Canvas

Ensuite, nous avons eu des problèmes avec l'affichage de texte sur l'écran, comme lorsqu'il est écrit `Trouver la clé` par exemple. Le plus dur était de changer ce canvas au bon moment. Le moment qui nous a le plus posé problème était lorsque le joueur revient de la grotte avec la clé, puis parle avec l'habitant, ce qui déclenche un changement de texte. Ce changement de texte fut compliqué, car nous ne trouvions pas de solution au problème et, bien souvent, à force de modifier nos scripts, les dialogues avec le personnage ne fonctionnaient même plus.





---

Au début, nous voulions créer une fonction qui serait appelée dans notre script Yarn Spinner (l'asset que nous avons utilisé pour les dialogues), sauf que rien ne se passait. Nous avons donc dû changer notre façon de l'implémenter avec un simple trigger, lorsque le joueur touche la box collider du personnage.

## 7.3 site internet

Le site a été développé dans le but de promouvoir ce jeu mais aussi notre entreprise Prisme. Le site est organisé de manière à guider facilement les utilisateurs à travers les différentes sections. Sur la page d'accueil, vous trouverez un aperçu de ce que propose le site, comme télécharger le jeu. Chaque page a été construite de manière à offrir une expérience utilisateur optimale, tout en mettant en valeur le contenu principal. Sur la page d'accueil, par exemple, vous pourrez découvrir un gradient de couleur qui évolue, passant d'un assortiment de teintes plus claires à des nuances plus sombres en haut et en bas de la page, créant ainsi une ambiance plus interactive et moderne.

De plus, nous avons décidé d'intégrer plusieurs fonctionnalités essentielles afin de rendre le site web le plus accessible possible. C'est dans cette optique que nous avons ajouté un menu de navigation en haut de la page et présent sur chacune des pages afin que l'utilisateur puisse à sa guise, aller d'une page à l'autre sans avoir à trop chercher. Qui plus est, notre site web est responsive design, c'est-à-dire qu'il s'adapte aussi bien sur portable que sur ordinateur.



---

À l'avenir, nous prévoyons d'ajouter des images du jeu ainsi que des animations pour ajouter de la dynamique à l'ensemble du site et rendre notre jeu plus attractif.

Nous avons décidé de fragmenter notre site web en de multiples pages afin de permettre à l'utilisateur de trouver des informations plus facilement. En outre, nous avons ajouté de nombreuses pages dont voici un récapitulatif détaillé :



- 
- **Accueil** : Cette page présente une vue d'ensemble du site, avec un aperçu des sections principales et des informations essentielles pour aider l'utilisateur à naviguer rapidement.
  - **Présentation de l'histoire du jeu** : Cette section raconte l'intrigue du jeu et présente les différents personnages présents dans Echo of Nyxalis.



- 
- **Chronologie** : Une ligne du temps détaillant les étapes clés du jeu depuis sa création. Cette page permet de suivre l'évolution du projet, les mises à jour majeures, ainsi que les moments importants qui ont influencé son parcours.
  - **Outils utilisés** : Elle présente les logiciels utilisés pour créer le jeu. Ainsi, vous y trouverez des informations sur chacun des logiciels utilisés, ainsi qu'un petit descriptif expliquant la pertinence d'utiliser ces logiciels dans une logique de créer le meilleur jeu possible.
  - **Sons** : Une section dédiée à la bande sonore du jeu, comprenant une classification des différents sons du jeu.
  - **Membres** : Cette page met en avant les membres de l'équipe de développement. Chaque membre est présenté avec son rôle dans le projet, son parcours et sa contribution spécifique au jeu.
  - **Entreprise** : Une page dédiée à l'entreprise qui a développé le jeu. Vous y trouverez des informations sur l'histoire de l'entreprise, sa mission, ses valeurs et ses projets à venir. Elle permet de mieux comprendre le contexte dans lequel le jeu a été créé et de découvrir d'autres projets de l'entreprise.
  - **Télécharger** : Comme son nom l'indique, cette section permet le téléchargement du jeu ainsi qu'un aperçu détaillé des modifications apportées par chacun des membres de l'équipe à chaque soutenance technique.



---

## 7.4 Optimisation et performances

Bien que richement décorées, les salles ont été pensées pour préserver les performances :

- Mobilier simple et modélisation low-poly
- Textures optimisées, en particulier sur les murs, sols, torches
- Éclairage intelligent : seules les lumières proches du joueur sont actives
- Culling : les objets hors de la vue de la caméra ne sont pas rendus

Ces choix techniques nous ont permis de maintenir une fluidité constante, même en multijoueur, dans des environnements complexes et dynamiques.

## 8 Ressenti final

### 8.1 Angeline

Je n'avais jamais programmé auparavant. Ce projet m'a permis de découvrir le monde du développement de jeu vidéo sous tous ses aspects : code, design, musique, architecture, test...

J'ai appris à lire de la documentation, à corriger des bugs, à structurer un projet... et surtout à travailler en équipe malgré les difficultés.

Notre projet était très ambitieux. Nous avons dû faire des choix difficiles, comme renoncer à certaines idées originales (par exemple, intégrer la langue des signes dans le système de dialogue). Mais nous avons gardé l'essentiel : l'âme du jeu, son univers, son



---

gameplay.

L'ambiance de travail a été parfois tendue, il y a eu des désaccords, du stress. . . mais nous avons toujours su rester soudés pour livrer un jeu complet, jouable, et avec une vraie identité.

Et pour ça, je suis extrêmement fière de ce que nous avons accompli.

## 8.2 Clément

J'ai appris beaucoup de choses durant ce projet. Ce projet m'a permis de développer beaucoup d'aptitudes importantes pour la suite de ma vie. La première fut la patience. En effet, travailler en équipe demande parfois beaucoup de patience car on doit parfois attendre qu'une personne ait fini sa tâche avant de commencer la nôtre. De plus, je mettrai dans la catégorie de la patience : la persévérance. En effet, la persévérance est indispensable pour la réussite du projet. Il a fallu que j'échoue plusieurs fois, pour finalement trouver une solution au problème ou le contourner. Ma vision de l'échec a évolué avec le projet. Au début, je voyais d'un mauvais œil celle-ci mais finalement je me rends compte aujourd'hui qu'elle est importante et que sans échec il n'y a pas de réussite. J'ai aussi appris que travailler en équipe exigeait un travail de planification et de gestion des tâches indispensable à la réussite du projet. Enfin, voir le jeu, réussir à implémenter telle chose ou juste réaliser une tâche banale, sont autant de sources de satisfaction pour moi.

Je me suis senti à l'aise dans l'équipe, tout le monde était



---

compréhensif, chacun essayait de faire de son mieux pour donner vie au projet. Même si parfois, il nous est arrivé de faire preuve de pessimisme, l'ensemble de l'équipe a continué à persévérer même quand la situation semblait désespérée. Pour ma part, le projet m'a effectivement beaucoup stressé car il y avait une part d'inconnue, notamment la peur de ne pas finir au bon moment le projet. Mais ça reste, tout de même, une bonne expérience dans l'ensemble.

## 8.3 Aizabel

Avant ce projet, je n'avais jamais travaillé sur un jeu vidéo, ni même sur un projet informatique d'une telle envergure. Ce qui m'a le plus marquée, c'est à quel point chaque détail compte : la cohérence visuelle, l'ergonomie, le rythme du jeu, le son... rien ne doit être laissé au hasard.

J'ai appris à collaborer avec des profils très différents, à défendre mes idées tout en restant ouverte aux critiques. Ce n'est pas toujours évident, surtout quand on s'investit autant. Il y a eu des remises en question, parfois des frustrations, mais toujours avec l'envie d'avancer.

Ce projet m'a aussi permis de prendre confiance en moi. J'ai osé sortir de ma zone de confort, tester des outils que je ne connaissais pas, proposer des idées, corriger du code...

Aujourd'hui, je suis fière de voir un jeu concret, terminé, que nous avons construit de toutes pièces ensemble.



---

## 8.4 Augustin

Ce projet m'a beaucoup apporté, surtout en termes d'organisation personnelle. J'ai compris que coder ne suffit pas : il faut savoir gérer son temps, prioriser, s'adapter aux imprévus. Le travail en équipe m'a aussi appris l'importance de la communication. Parfois, une simple incompréhension peut bloquer tout le monde.

Techniquement, j'ai progressé sur Unity, mais aussi en scripting, en logique de gameplay et en debugging. J'ai compris que l'itération est essentielle : on essaye, ça casse, on recommence, et un jour ça marche. Et cette satisfaction, ce moment où tout fonctionne enfin, vaut vraiment le coup.

Il y a eu des hauts et des bas, évidemment, mais je pense qu'on peut tous être fiers du résultat. Ce jeu n'est pas parfait, mais il nous ressemble. Il porte nos idées, nos efforts, nos erreurs aussi, et c'est ce qui le rend unique.

## 8.5 Julien

Participer à ce projet a été une des expériences les plus exigeantes et enrichissantes de ma formation. J'ai touché à tout : modélisation 3D, intégration, animations, test utilisateur... Et chaque étape avait ses défis.

J'ai aussi beaucoup appris sur moi-même. J'ai tendance à vouloir tout bien faire du premier coup, mais ce projet m'a obligée à accepter que l'erreur fait partie du processus. L'important, c'est de





---

corriger vite, de ne pas rester bloquée, de demander de l'aide.

Le plus fort pour moi a été de voir notre jeu prendre vie petit à petit. Au départ, ce n'étaient que des idées sur un document, et aujourd'hui c'est un univers entier qu'on peut explorer. Je suis fière du travail qu'on a accompli ensemble, et encore plus d'avoir gardé une bonne entente malgré la pression.



## 9 Explication du code

Voici une explication détaillée du code de contrôleur de joueur en C :

```
1 public class PlayerController : MonoBehaviour
2 {
3     [Header("Base setup")]
4     public float walkingSpeed = 7.5f;
5     public float runningSpeed = 11.5f;
6     public float jumpSpeed = 8.0f;
7     public float gravity = 20.0f;
8     public float lookSpeed = 2.0f;
9
10    [Header("Camera Settings")]
11    public float cameraDistance = 5f;
12    public float cameraHeight = 2f;
13    public float cameraSmoothness = 0.125f;
14    public Vector3 cameraOffset;
15    public float minVerticalAngle = -30f;
16    public float maxVerticalAngle = 60f;
17
18    [Header("Animation")]
19    private Animator animator;
20    private Alteruna.AnimationSynchronizable animSync;
21
22    private CharacterController characterController;
23    private Vector3 moveDirection = Vector3.zero;
24    private float rotationX = 0;
25    private float currentRotationY = 0;
26
27    [HideInInspector]
28    public bool canMove = true;
29
30    private Camera playerCamera;
31    private Vector3 targetCameraPosition;
32
33    private Alteruna.Avatar _avatar;
34
35    void Start()
36    {
37        _avatar = GetComponent<Alteruna.Avatar>();
38
39        if (!_avatar.IsMe)
40            return;
41
42        characterController = GetComponent<CharacterController>();
43        playerCamera = Camera.main;
44        animator = GetComponent<Animator>();
45        animSync = GetComponent<Alteruna.AnimationSynchronizable>();
46
47        cameraOffset = new Vector3(0, cameraHeight, -cameraDistance);
48        UpdateCameraPosition();
49    }
50
51    void Update()
```



```

52 {
53     if (!_avatar.IsMe)
54         return;
55
56     HandleMovement();
57     if (canMove)
58     {
59         HandleRotation();
60     }
61 }
62
63 void LateUpdate()
64 {
65     if (!_avatar.IsMe || playerCamera == null)
66         return;
67
68     UpdateCameraPosition();
69 }
70
71 private void HandleMovement()
72 {
73     bool isRunning = Input.GetKey(KeyCode.LeftShift);
74
75     // Utiliser la rotation du joueur au lieu de la cam ra pour le
76     // mouvement
77     Vector3 forward = transform.forward;
78     Vector3 right = transform.right;
79
80     float curSpeedX = canMove ? (isRunning ? runningSpeed : walkingSpeed) *
81         Input.GetAxis("Vertical") : 0;
82     float curSpeedY = canMove ? (isRunning ? runningSpeed : walkingSpeed) *
83         Input.GetAxis("Horizontal") : 0;
84     float mouvementDirectionY = moveDirection.y;
85
86     moveDirection = (forward * curSpeedX) + (right * curSpeedY);
87
88     // Mettre jour l'animation de marche
89     if (animator != null)
90     {
91         bool isMoving = (Mathf.Abs(curSpeedX) > 0.1f || Mathf.Abs(curSpeedY)
92             > 0.1f);
93         animator.SetBool("IsMoving", isMoving);
94     }
95
96     if (Input.GetButton("Jump") && canMove && characterController.
97         isGrounded)
98     {
99         moveDirection.y = jumpSpeed;
100     }
101     else
102     {
103         moveDirection.y = mouvementDirectionY;
104     }
105
106     if (!characterController.isGrounded)
107     {
108         moveDirection.y -= gravity * Time.deltaTime;
109     }
110 }

```



```

105     characterController.Move(moveDirection * Time.deltaTime);
106 }
107
108
109 private void HandleRotation()
110 {
111     rotationX += -Input.GetAxis("Mouse Y") * lookSpeed;
112     rotationX = Mathf.Clamp(rotationX, minVerticalAngle, maxVerticalAngle);
113
114     currentRotationY += Input.GetAxis("Mouse X") * lookSpeed;
115
116     transform.rotation = Quaternion.Euler(0, currentRotationY, 0);
117 }
118
119 private void UpdateCameraPosition()
120 {
121     Vector3 desiredPosition = transform.position + Quaternion.Euler(
122         rotationX, currentRotationY, 0) * cameraOffset;
123
124     RaycastHit hit;
125     if (Physics.Linecast(transform.position, desiredPosition, out hit))
126     {
127         targetCameraPosition = hit.point;
128     }
129     else
130     {
131         targetCameraPosition = desiredPosition;
132     }
133
134     playerCamera.transform.position = Vector3.Lerp(playerCamera.transform.
135         position, targetCameraPosition, cameraSmoothness);
136     playerCamera.transform.LookAt(transform.position + Vector3.up *
137         cameraHeight * 0.5f);
138 }
139 }

```

Listing 1: PlayerController.cs

- **Variables de configuration** : Les variables comme `walkingSpeed`, `runningSpeed`, `jumpSpeed`, etc., définissent les paramètres de base pour le mouvement et la caméra.
- **Initialisation** : La méthode `Start()` initialise les composants nécessaires comme `CharacterController`, `Animator`, et la caméra principale.
- **Mise à jour du mouvement** : La méthode `Update()` gère les entrées utilisateur pour le mouvement et la rotation.
- **Gestion du mouvement** : La méthode `HandleMovement()` calcule la direction du mouvement en fonction des entrées utilisateur et applique la gravité si le personnage n'est pas au sol.



- 
- **Gestion de la rotation** : La méthode `HandleRotation()` met à jour la rotation du joueur en fonction des entrées de la souris.
  - **Mise à jour de la position de la caméra** : La méthode `UpdateCameraPosition()` ajuste la position de la caméra pour suivre le joueur tout en évitant les obstacles.

