



RAPPORT DE SOUTENANCE 2
PROJET DE PROGRAMMATION - SEMESTRE 2

Nyctalopia

Créé par :

Iñigo Aldaraborda Hoyos - Chef de projet

Hugo Meleiro

Guilhem Cros

Marin Godefroy

EPITA Toulouse

Mardi 26 Avril 2022

Table des matières

1	Introduction	1
2	Répartition des tâches	2
3	Progrès réalisé	3
3.1	I.A. - Intelligence Artificielle	3
3.2	Audio et Effets Sonores	4
3.3	Gameplay	4
3.4	Communication	8
3.5	Graphismes, Modèles et Terrain	9
3.6	Multijoueur	11
3.7	UI/UX - Interface	12
3.8	Manuels d'instructions et d'installation	14
3.9	Site Web	14
4	Objectifs pour la dernière soutenance	18
4.1	Sauvegardes	18
4.2	Gameplay	19
4.3	Carte - Jouabilité	19
4.4	Multijoueur	20
5	Planning	21
6	Conclusion	22
7	Annexe	23

1 Introduction

Dans le cadre du projet de S2 à EPITA, nous sommes poussés à mettre en pratique les différentes connaissances acquises en Travaux Pratiques et en cours, dans la réalisation d'un projet en groupe. Le projet de notre studio gameHUB est la création d'un jeu d'horreur : *Nyctalopia*.

À l'arrivée de la première soutenance, nous avons préféré nous concentrer sur l'aspect technique du jeu, le “backend”, toute la partie que le ou les joueurs ne verraient pas, ce qui nous a permis pour cette deuxième date clé de notre projet, d'avancer plus rapidement afin d'obtenir des résultats plus visibles surtout dans les parties graphiques et jouables.

Pour la seconde soutenance, nous avons eu moins de temps, par rapport à la première soutenance, donc nous avons décidé principalement de régler des bogues pouvant être très gênants pour l'utilisateur, ainsi que l'implémentation de quelques fonctionnalités.

Nous présentons, dans ce rapport de soutenance, tout le progrès réalisé dans notre projet lors de cette deuxième période de travail, comment les tâches ont été réparties, et les objectifs que l'on envisage d'atteindre pour la prochaine soutenance.

2 Répartition des tâches

Voici la répartition prévue des tâches entre les étudiants du groupe gameHUB. Celle-ci n'a pas changé depuis le cahier des charges.

Tâches	Iñigo	Hugo	Guilhem	Marin
AI - Intelligence Artificielle			◊	✓
Audio	✓			
Communication & Gameplay			✓	◊
Graphismes & Modèles	✓			
Manuels				✓
Map	◊	✓		
Multijoueur	✓	◊		
Saves			✓	
UI/UX & Réseaux		✓		
VFX		✓		
Website				✓

✓ - Responsable

◊ - Support - Suppléant

3 Progrès réalisé

3.1 I.A. - Intelligence Artificielle

Pour implémenter une intelligence artificielle, nous avons testé divers algorithmes de "pathfinding" dans un grand nombre de scènes comprenant des obstacles afin de pouvoir implémenter dans notre jeu une entité antagoniste qui suivra le joueur.



FIGURE 1 – *Entité*

Notre entité utilisera le pathfinding pour déterminer, dès lors que le joueur rentrera dans son champ d'action, un chemin à suivre pour atteindre le joueur et l'attaquer.

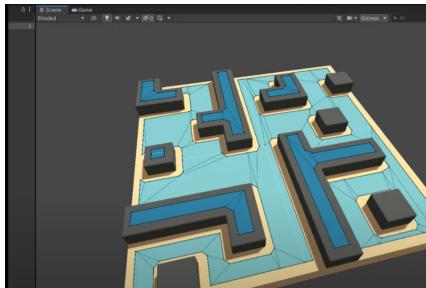


FIGURE 2 – *NavMesh : Tous les endroits possibles au déplacement de l'entité*

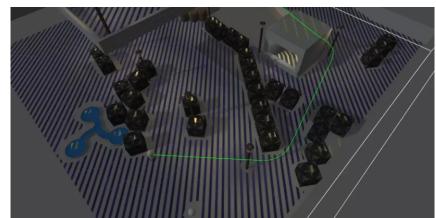


FIGURE 3 – *Pathfinding*

3.2 Audio et Effets Sonores

Pour l'audio, la cinématique principale et le menu d'accueil du jeu comptent déjà avec ses propres musiques, libres de droits et utilisables pour usage commercial, qui représentent bien l'aspect mystérieux et effrayant de notre jeu d'horreur.

Les effets sonores du menu d'accueil ont déjà été mis en place et les effets extradiégétiques trouvables dans le jeu ont déjà été sélectionnés (ramassage d'objets, interaction quelconque).

Nous continuons d'effectuer des recherches concernant les effets sonores liés aux différents matériaux du jeu lorsque les joueurs y marchent dessus. Cependant, les effets sonores gratuits seront très accessibles grâce aux bibliothèques d'effets sonores gratuits *Zapsplat* ou *Freesound* auxquelles on a déjà jeté un coup d'œil.

3.3 Gameplay

Lors de la première soutenance nous avons ajouté des scripts permettant au joueur de se déplacer, de pouvoir s'accroupir et de pouvoir diriger la caméra.

Depuis nous avons ajouté deux scripts principaux à notre projet :

PlayerStatut - une classe permettant de stocker l'avancée du joueur dans sa partie.

Ce script permet de tenir à jour la réussite des différents puzzles et l'ouverture de différentes portes du jeu à travers la récupération des divers objets nécessaires à leur réussite ou leur ouverture. Ce script gardera aussi en mémoire les coordonnées du dernier point de sauvegarde utilisé pour y placer le joueur lors de sa prochaine mort ou du prochain lancement de la partie. Pour réaliser ce script, nous avons eu besoin de créer deux autres classes

- ***Item*** - qui contiendra l'identifiant de l'objet
- ***Environnement*** - qui contiendra l'identifiant des portes et énigmes résolues par le joueur

PlayerStatut stocke ces différents *Items* et *Environnement* dans des listes. Lors de l'ouverture d'une porte fermée on vérifiera si l'*Items* correspondant à la clé est bien dans le *PlayerStatut* du joueur et on ajoutera cette porte à une liste d'*Environnement*. Lors du prochain lancement de la partie, le jeu reprendra au même stade grâce aux coordonnées du point de sauvegarde et au suivi des différentes ouvertures. De plus une classe *Translate* comprend tous les équivalents nom/identifiant des objets, des portes, des puzzles pour faciliter la compréhension du code en plus des coordonnées des points de sauvegarde. Pour les points de sauvegarde : chaque point est lié à ses coordonnées dans le jeu.

SaveData - un script qui sera par la suite lié aux bornes de sauvegarde dissimilées à des endroits prédéfinis dans le jeu. Ce script permet de sauvegarder des données dans le répertoire créé lors de l'installation du jeu grâce à la méthode *Application.persistentPath* mise à disposition par Unity. Les données sauvegardées sont contenues dans le *PlayerStatut* (mentionné ci-dessus), et seront stockées dans le format JSON (1) et chiffrées grâce au chiffrement XOR (XOR Cipher) (2), empêchant ainsi le joueur de tricher en modifiant sa sauvegarde. Pour des raisons pratiques, lors des tests et des présentations, un mode développeur sera disponible permettant de modifier la sauvegarde et avancer à l'état souhaité du jeu.

- (1) : Le format JSON (JavaScript Object Notation) est un format couramment utilisé pour représenter de l'information structurée sous forme de texte, il est facilement lisible par des humains (malgré le manque d'espaces nécessaire pour alléger le fichier).

```
{"currentSave":{"x":50.0,"y":0.0,"z":100.0}, "item":[{"id":101}, {"id":102}], "envStatut":[{"id":201}]}
```

FIGURE 4 – Fichier JSON

- (2) : Le chiffrement XOR utilise l'opérateur binaire XOR (OU-exclusif), voici sa table de vérité :

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

FIGURE 5 – Table de vérité OU-exclusif

Lors du chiffrement nous allons passer à travers une porte logique XOR chaque lettre de notre chaîne de caractère avec la lettre au même indice, le tout modulo la longueur de notre clé, et ainsi reconstruire une chaîne de caractères chiffrée, de même longueur mais inintelligible. Lors du déchiffrement nous allons effectuer les mêmes actions mais avec la chaîne de caractères chiffrée à la place de celle en claire. Pour passer deux lettres à travers la porte XOR on passe en réalité les représentations binaires des codes ASCII (American Standard Code for Information Interchange) de ces lettres, le résultat sera donc une représentation binaire convertie en nombre et retranscrite grâce à la table ASCII. Exemple : Dans cet exemple on souhaite chiffrer le mot "MESSAGE" avec la clé "CLE" :

Lettres	M	E	S	S	A	G	E
Code ASCII	77	69	83	83	65	71	69
Binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101

Lettres	C	L	E
Code ASCII	67	76	69
Binaire	01000011	01001100	01000101

FIGURE 6 – Conversion des lettres en binaire

Le chiffrement XOR va donc correspondre à ceci :

Message en binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101
Clé en binaire (répétée si nécessaire)	01000011	01001100	01000101	01000011	01001100	01000101	01000011
Message chiffré en binaire	00001110	00001001	00010110	00010000	00001101	00000010	00000110
Message chiffré en code ASCII	14	9	22	16	13	2	6
Message chiffré en lettre	SO	HT	SYN	DLE	CR	STX	ACK

FIGURE 7 – Conversion des lettres en binaire

Dans cet exemple le message chiffré est composé uniquement de caractères ASCII non imprimables, de caractères de contrôle, utilisés pour contrôler certains périphériques ou donner des indications sur le format du document.

Le chiffrement XOR est basique et le message chiffré peut être facilement décrypté si la clé est trop petite par rapport à la taille du message grâce à une analyse fréquentielle ou dans ce cas d'utilisation (sauvegarde) par une attaque à texte clair connue comme le joueur connaît par exemple les noms des objets.

Nous avons choisi de créer une clé de 50 caractères choisis aléatoirement parmi les caractères dite imprimeable de la table ASCII.

Il est donc ici préférable de se référer aux objets par des nombres plutôt que des noms, nous choisissons de représenter les objets, les portes et énigmes réussit par des identifiants. Pour un souci de lisibilité nous avons aussi ajouté un dictionnaire liant les identifiants des objets, portes et énigmes à leurs noms.

De plus, pour l'instant, la clé de chiffrement est unique pour toutes les sauvegardes et tous les joueurs mais il peut être intéressant d'en générer une nouvelle aléatoirement à chaque création d'une nouvelle partie en retranscrivant ce code en C#.

Nous avons aussi commencé les recherches et l'implémentation de la touche "utiliser" assignée par défaut à la touche "E". Nous utiliserons le système de *RayCast* (qui peut être traduit par "lancé de rayon") et le système *Event* proposé par Unity et découper ce système en deux scripts distincts : *Interactor* et *Interactable*. Ils seront respectivement liés au joueur et aux objets que l'on veut interactifs tels que les bornes de sauvegarde, les puzzles, les clés à récupérer, les portes...

Interactor - Ce script sera composé d'un RayCast dont la portée représentera la portée d'action du joueur dans le jeu et vérifiera si l'objet pointé par le joueur est étiqueté comme interactif et s'il n'y a pas d'objets entre celui-ci et le joueur rendant l'action impossible tel qu'un mur par exemple. Ensuite nous vérifierons si le joueur appuie bien sur la touche d'action.

Interactable - Ce script comportera l'évènement associé à l'objet interactif : récupération d'une clé (disparition de celle-ci et son ajout dans le *PlayerStatut*), ouverture d'une porte (vérification que la clé est bien en possession du joueur, si oui, ouverture de la porte).

Nous pourrons aussi par la suite implémenter un curseur changeant de forme si le joueur pointe et est en assez proche d'un objet interactif (passage d'une croix à une main par exemple).

Concernant le changement des touches, nous avons décidé pour l'instant de laisser le choix au joueur entre deux configurations "AZERTY" et "QWERTY" qui est plus rapide à implémenter. La personnalisation des touches prend du temps à implémenter et n'est pas vitale, nous avons donc décidé de repousser son implémentation à plus tard.

3.4 Communication

La police d'écriture temporaire du logo du jeu Nyctalopia (actuellement : “*Venom*”) a été remplacée par la police “*October Crow*”.



FIGURE 8 – Police d’écriture choisie pour Nyctalopia

3.5 Graphismes, Modèles et Terrain

L'histoire du jeu se basera sous forme de chapitres, le premier est le Chapitre 0, c'est une cinématique d'une minute trente secondes, où le joueur est immédiatement accueilli par un accident de voiture.

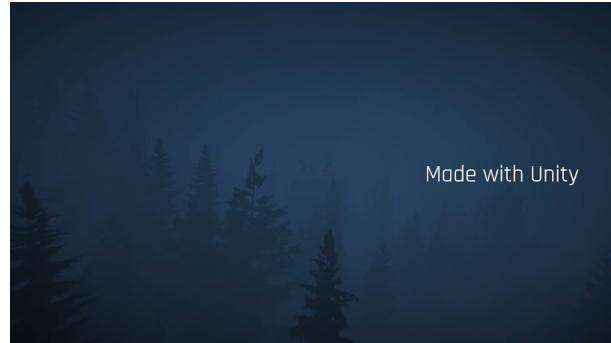


FIGURE 9 – *Chapitre 0*

Une grande variété de modèles 3D d'arbres, arbustes et herbe ont déjà été sélectionnés et utilisés dans le jeu pour couvrir ce terrain et créer une forêt riche et variée. La plupart de ces modèles de végétation proviennent de la bibliothèque de modèles 3D *Quixel's Megascans* et du site officiel de modèles Unity *Unity Asset Store*. Plusieurs modifications et conversions ont été nécessaires pour les rendre compatibles avec notre projet qui est basé sur le système de rendu HDRP (*High-Definition Render Pipeline*) qui offre des résultats graphiques plus attractifs.

Nous aurons ensuite une partie qui se déroulera dans la forêt (Chapitres 1 et 2) avec des énigmes et puzzles à résoudre tout au long de ces chapitres.



FIGURE 10 – *Représentation du Chapitre 1*



FIGURE 11 – *Représentation du Chapitre 2*

Enfin un troisième chapitre est déjà en cours de production, qui se déroulera, dans des souterrains. Dans ce nouvel espace jouable, le joueur devra, dans le noir et grâce à une lampe torche, trouver un moyen de s'échapper le tout en évitant l'entité antagoniste qui sera enfermée avec lui et qui le suivra avec un algorithme de *pathfinding*.



FIGURE 12 – *Chapitre 3 - Les égouts*



FIGURE 13 – *Chapitre 3 - Les égouts*



FIGURE 14 – *Système de lampe torche*

Concernant les personnages jouables, nous comptons actuellement avec deux modèles de personnages complètement animés, un homme et une femme, provenant de la bibliothèque d'Adobe *Mixamo*.



FIGURE 15 – *Personnage masculin*



FIGURE 16 – *Personnage féminin*

Pour l'entité qui suivra le joueur plus tard dans le jeu, un modèle a déjà été choisi. (c.f. I.A. - Intelligence Artificielle)x

3.6 Multijoueur

Pour le multijoueur nous avons décidé d'utiliser le SDK Steamworks fourni par Steam Inc. N'étant pas nativement compatible avec C#, nous avons dû utiliser une bibliothèque tierce nommée *Steamworks.NET*. Ce SDK permettra de créer des lobbys et d'intégrer une liste d'amis, qui facilitera la communication entre les deux joueurs étant donné que Steam est la plateforme de vente de jeux vidéos la plus populaire au monde avec 100+ millions de connexions mensuelles.

Avec tout ces outils en place, il nous a suffi de lire la documentation officielle de Steam, et de mettre en place un script consistant à créer et à joindre des lobbys grâce aux boutons présents au sein de l'interface (c.f. UI/UX).

Pour la deuxième soutenance, nous avons décidé d'implémenter un chat vocal à l'aide de SteamVoice, fourni par Steam.

3.7 UI/UX - Interface

L'interface utilisateur possède trois parties principales, le menu principal, le menu “Play” et le menu “Paramètres”. Ce dernier permet à l'utilisateur de pouvoir régler la résolution, la taille de la fenêtre (fenêtré, borderless ou plein écran), mais également le son et le choix des touches (AZERTY ou QWERTY). Le menu “Play” possède trois boutons, un placé à gauche, permettant au joueur 1 de créer un lobby et de le rejoindre, et à droite deux boutons permettent au joueur 2 de soit, joindre le lobby avec un *CSteamID* (code de lobby délivré par Steam) ou bien en utilisant sa liste d'ami Steam. Enfin, le menu principal possède deux grands boutons appelant le joueur à soit débuter une nouvelle campagne ou bien de continuer là où il avait sauvegardé pour la dernière fois (c.f. Sauvegardes)

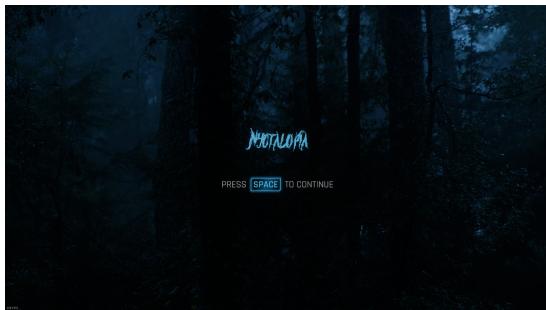


FIGURE 17 – Écran d'accueil

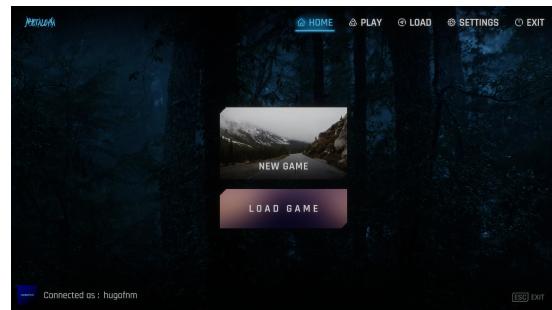


FIGURE 18 – Menu principal

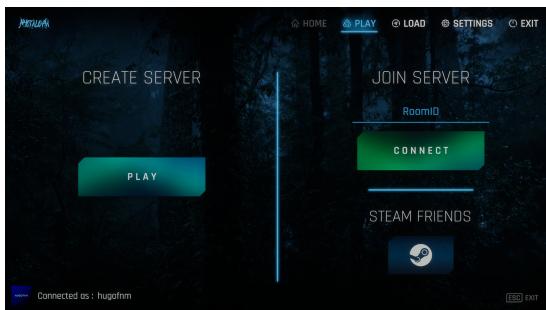


FIGURE 19 – Menu “Play”

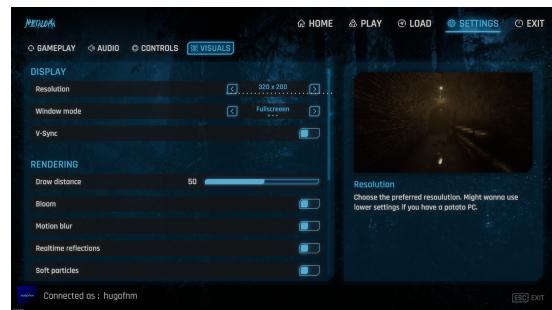


FIGURE 20 – Menu “Paramètres”

Il est désormais possible aussi de changer la langue du jeu. Trois options sont possibles : français, anglais ou espagnol.



FIGURE 21 – Français



FIGURE 22 – Anglais



FIGURE 23 – Espagnol

Si l'utilisateur n'a pas effectué de choix, la langue par défaut sera réglée par rapport à la langue système (le système d'exploitation envoie au jeu la langue principale du système).



FIGURE 24 – Sélection de la langue dans le menu Paramètres

Nous avons également mis en place un installateur graphique “.exe” qui est accessible à tout utilisateur possédant une machine Windows ou Linux 64 Bits sur le site get.nyctalopia.games (c.f Site Web). Ainsi qu'un script permettant d'informer ses amis que l'on joue à Nyctalopia sur la plateforme Discord a été créé.

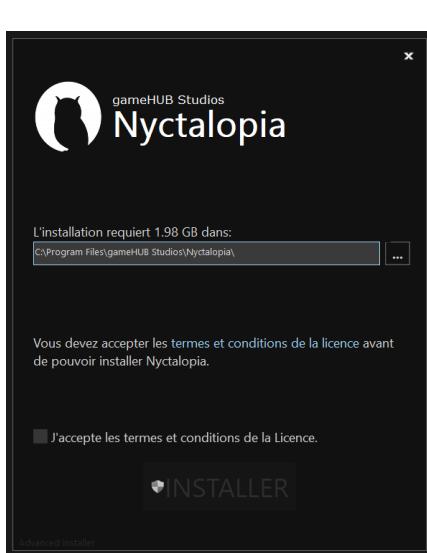


FIGURE 25 – Installateur Graphique .exe

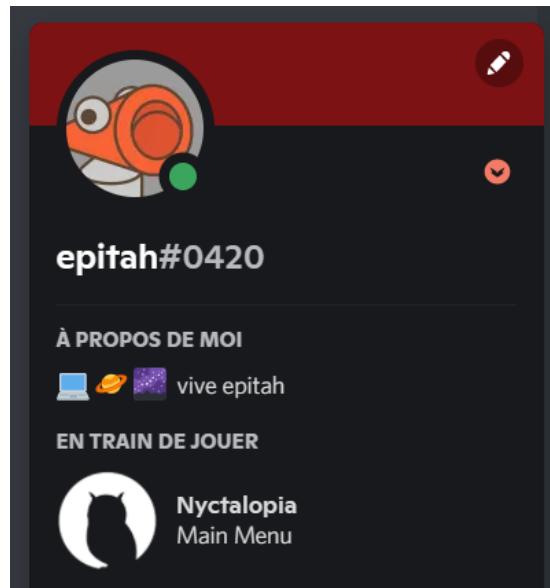


FIGURE 26 – Discord Rich Presence

3.8 Manuels d'instructions et d'installation

Pour la deuxième soutenance nos manuels d'instructions et d'installation sont entièrement complets et vont permettre aux utilisateurs qui rencontreront des problèmes d'avoir tout de suite des réponses à leurs questions. On y retrouve une version en français mais aussi en anglais.



EPITA Toulouse
2021 - 2022

FIGURE 27 – Première page

3.9 Site Web

Pour cette deuxième soutenance, nous avons décidé d'achever notre site web, qui est prêt au public.

Nous avons essayé de rendre le site attractif. On peut le voir avec les effets qui permettent que l'arrière-plan et les images au premier plan ne défile pas à la même vitesse ou encore avec le curseur qui suit la souris. Pour ces effets, on a fait le choix d'utiliser des scripts JS fournis par Internet pour se rapprocher le plus possible d'un vrai site professionnel. Le code HTML est très bien organisé pour nous permettre d'améliorer ou de modifier le site à tout moment.

Nous avons choisi un fond sombre pour tout de suite mettre le joueur dans l'ambiance du jeu où il sera dans l'obscurité. Enfin, le site est adapté pour tout type d'écrans ce qui permet au joueur de nous consulter n'importe quand. A travers ce site nous voulons mettre en confiance l'utilisateur et lui donner envie de jouer en montrant des informations intéressantes sur Nyctalopia et en montrant notre professionnalisme.

Pour cette soutenance nous avons apporté quelques modifications à notre site pour le rendre plus agréable à parcourir et à utiliser.

En terme de contenu sur le site on peut retrouver :

- Une page d'accueil : forêt animée



FIGURE 28 – Accueil du site

- Image du jeu : tels que la forêt ou encore l'interface utilisateur mais également la scène d'introduction

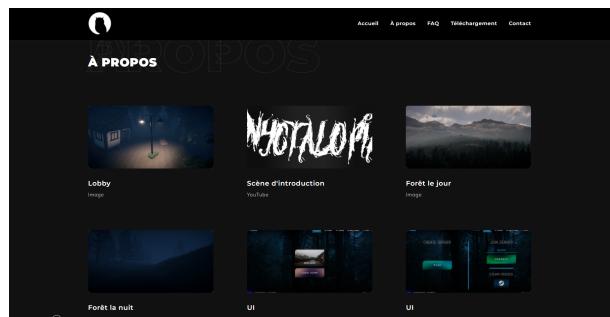


FIGURE 29 – Image du jeu

- Contact : Formulaire + courriel du studio : on peut retrouver maintenant un bouton "envoyer" qui nous renvoie vers notre boite mail. Pour l'instant il est inutile de remplir le formulaire mais pour la dernière soutenance nous implémenterons un script permettant de récupérer ce que l'utilisateur a écrit dans le formulaire pour le mettre directement dans un mail.

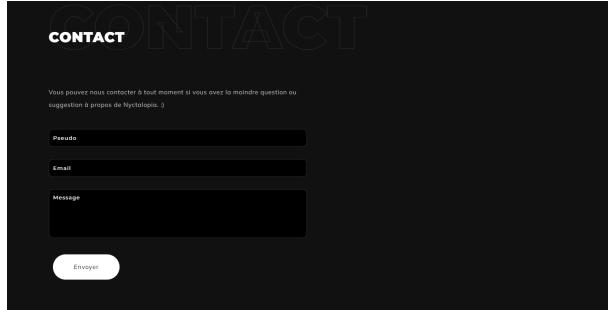


FIGURE 30 – *Contact*

- Équipe : rôles de chaque membre

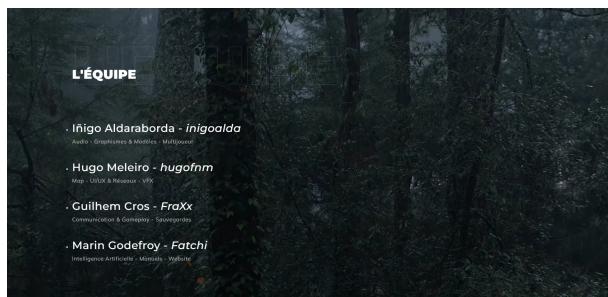


FIGURE 31 – *Membres du studio GameHub*

- FAQ : quelques réponses à des questions courantes.

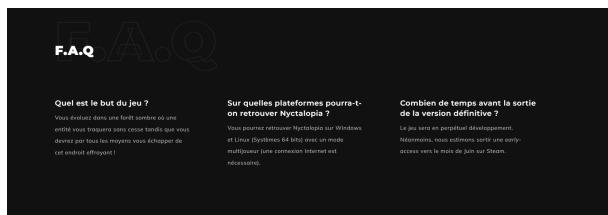


FIGURE 32 – *F.A.Q*

- Documents : Cahier des charges, rapport de soutenance
- Liens de téléchargements et Configurations : minimale et recommandée



FIGURE 33 – Site sécurisé SSL

Le site web est hébergé sur CloudFlare Pages, un service gratuit de l'entreprise de gestion DNS CloudFlare, permettant de déployer rapidement à partir d'un repo Git, un site web statique rapide grâce aux serveurs CDN disposés partout dans le monde et sécurisé par un certificat SSL et une protection anti-DDOS. Le nom de domaine choisi est *nyctalopia.games*. Il est simple et facile à retenir pour l'utilisateur.

Gestion DNS pour nyctalopia.games						
Type	Nom	Contenu	État du proxy	Durée TTL	Actions	
A	gitlab	[REDACTED]	DNS uniquement	Automatique	Modifier	
CNAME	get	gitlab.nyctalopia.games	Proxied	Automatique	Modifier	
CNAME	linux.get	gitlab.nyctalopia.games	Proxied	Automatique	Modifier	
❶ CNAME	nyctalopia.games	nyctalopia.pages.dev	Proxied	Automatique	Modifier	

FIGURE 34 – Interface CloudFlare

Nous avons également décidé de mettre en place un GitLab (*gitlab.nyctalopia.games*) privé sur un serveur nous appartenant, ce qui nous a permis d'utiliser la puissance brute de Git LFS et des pipelines CI/CD, améliorant notre productivité sans perdre de temps.

4 Objectifs pour la dernière soutenance

Voici la répartition prévue des tâches entre les étudiants du groupe gameHUB pour la dernière soutenance.

4.1 Sauvegardes

Concernant les sauvegardes, nous avons décidé de placer des bornes de sauvegarde, représentées par une ancienne radio, qui seront placées à des endroits stratégiques du jeu. Cette mécanique permet de simplifier le système de sauvegarde car ne demande pas de connaître l'emplacement du joueur et rendra le jeu plus stressant, car empêche le joueur de sauvegarder sa progression à tout instant.



FIGURE 35 – *Borne de sauvegarde*

Concernant les contrôles en jeu, nous devrons aussi implémenter une touche d'action pour interagir avec les objets du jeu, tel que des portes, des bornes de sauvegarde ou des objets en notre possession nécessaire à l'avancée du jeu. Ceci nous demandera aussi de définir la portée d'action du personnage. L'ajout de l'apparition du menu lors de l'appui sur la touche "escape" doit aussi être implémenté.

Le script, servant à redéfinir les touches de contrôles, n'est pas encore terminé, certains cas ne sont pas pris en compte, notamment le fait de d'empêcher le joueur d'assigner la même touche à plusieurs actions. Nous allons aussi devoir, par la suite, adapter ce script pour l'ajouter au menu en jeu. Le script peut déjà être adapté pour la touche « utiliser », qui sera implémentée plus tard. Une sauvegarde des touches redéfinies doit aussi être implémentée.

4.2 Gameplay

Concernant le gameplay, nous avons déjà commencé à réfléchir à certaines énigmes telles que la récupération de fusibles pour ouvrir des portes débloquant ainsi la suite de l'histoire.



FIGURE 36 – Boîte à fusibles

4.3 Carte - Jouabilité

Nous envisageons pour la dernière soutenance de finir les chapitres 1,2 et le troisième se déroulant dans les égouts en implémentant une partie jouable.



FIGURE 37 – *Aperçu des égouts*

4.4 Multijoueur

Pour la troisième soutenance, nous espérons avoir terminé le multijoueur. Voici la liste de tâches à réaliser :

- Corriger un certain nombre de bogues notamment liés à la synchronisation des clients (scènes qui se déclenchaient trop tard, évènements asynchrones, ...)
- Implémenter l'apparition automatique des personnages selon le nombre de joueurs.

5 Planning

Voici le planning (modifié par rapport au cahier de charges et au premier rapport de soutenance) d'avancement des tâches par période (temps séparant deux soutenances). La prochaine période réfère à la dernière soutenance prévue du 6 au 17 juin 2022.

Tâches	Soutenance 1	Soutenance 2	Soutenance 3
AI - Intelligence Artificielle	10%	40%	80%
Audio	50%	80%	100%
Communication & Gameplay	30%	50%	80%
Graphismes & Modèles	50%	70%	90%
Manuels d'instruction	0%	100%	100%
Carte	15%	50%	90%
Multijoueur	80%	90%	100%
Sauvegardes	5%	40%	90%
UI/UX	80%	90%	100%
VFX	10%	50%	90%
Site web	90%	100%	100%

Code couleur :

- **Vert** : En avance
- **Rouge** : En retard

6 Conclusion

Le développement de ce projet représente pour nous une grande première. Certains d'entre nous ont déjà un antécédent de programmation, mais cela semble dérisoire face à tous les besoins de ce projet. Beaucoup de temps et de travail a été et sera nécessaire, mais l'objectif est important. En effet, il est inévitable que nous sortirons tous beaucoup plus expérimentés que nous ne le sommes maintenant, ce qui est le but recherché. Nous espérons donc mettre à bien toutes les idées proposées dans ce document et que le résultat de ce projet soit de bonne qualité. Pour nous, avoir un résultat intéressant sera un accomplissement personnel car le développement sera une épreuve mais donnera un jeu amusant auquel nous serons, en tant que joueurs, satisfaits en y jouant.

Enfin, l'esprit de groupe sera également un point important de l'apprentissage c'est à dire savoir s'entraider, communiquer, s'organiser et planifier nos objectifs. Ce sont des points que l'on attend d'un ingénieur et qui nous seront demandés durant nos carrières professionnelles. Il est donc évident que la qualité de ce projet sera un reflet de nos capacités à évoluer au sein d'un groupe pour aller vers un même but.

7 Annexe

Nyctalopia : Traduction anglaise du mot héméralopie, qui est la cécité nocturne, ou l'incapacité à bien voir dans un éclairage sombre.

Survival horror : Le survival horror est un genre de jeu vidéo, sous-genre du jeu d'action-aventure, inspiré des fictions d'horreur. Bien que des aspects de combats puissent être présents dans ce type de jeu, le gameplay fait généralement en sorte que le joueur ne se sente pas aussi puissant qu'il ne le serait typiquement dans un jeu d'action, et ce en limitant par exemple la quantité de munitions, d'énergie ou de vitesse. Le joueur doit parfois chercher certains objets pour avoir accès à un passage vers une nouvelle zone, et résoudre des énigmes à certains moments. Les jeux utilisent des thèmes d'horreur, et le joueur est souvent confronté à des environnements obscurs et à des ennemis qui peuvent surgir de nulle part.

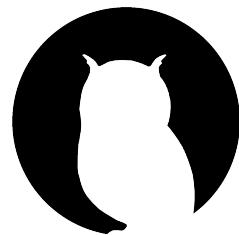
Jump scare : Un jump scare est un principe qui recourt à un changement brutal intégré dans une image, une vidéo ou une application pour effrayer brutalement le spectateur ou utilisateur. Ce principe s'est développé dans les années 1990 notamment au cinéma.

UI - Interface Utilisateur : L'interface utilisateur est un dispositif matériel ou logiciel qui permet à un usager d'interagir avec un produit informatique. C'est une interface informatique qui coordonne les interactions homme-machine, en permettant à l'usager humain de contrôler le produit et d'échanger des informations avec le produit.

IP : Une adresse IP est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque périphérique relié à un réseau informatique qui utilise l'Internet Protocol. L'adresse IP est à la base du système d'acheminement des paquets de données sur Internet. Il en existe deux versions : IPv4 et IPv6.

DNS : Le Domain Name System, généralement abrégé DNS, qu'on peut traduire en « système de noms de domaine », est le service informatique distribué utilisé pour traduire les noms de domaine Internet en adresse IP ou autres enregistrements.

Source : Wikipédia



Nyctalopia - gameHUB Studios

2022

EPITA Toulouse