

Dtres



CodeFall



Promotion 2029

Florian GAGNIER
Imane SARWAR
Florent FOSSATI
Alexandre SALGUEIRO

C2

Table des matières

Introduction	3
I Gameplay et Multijoueur	4
1 Multijoueur/Networking	4
2 Gameplay et Enigmes	8
3 Histoire	17
II Graphisme	18
1 Style graphique	18
2 Robots	19
III Site	21
IV Objectifs	21
V Conclusion	22

Introduction

Dans un premier temps, nous avons défini les objectifs de notre projet ainsi que la méthodologie pour le réaliser. Afin de faciliter la collaboration entre les membres de l'équipe, nous avons décidé d'héberger notre jeu et notre site web sur GitHub, ce qui permet un accès centralisé et efficace pour tous.

Pour simplifier l'utilisation de GitHub, nous avons installé GitHub Desktop, un outil convivial pour la gestion des dépôts. Par ailleurs, nous avons également installé Unity et Blender, nécessaires à la création et au design de notre jeu.

Nous avons ensuite appris à maîtriser les outils nécessaires au projet. Bien que nous disposions déjà d'une expérience personnelle, les travaux pratiques réalisés auparavant nous avaient initiés aux bases d'Unity et de la programmation en C#, ce qui nous a permis de renforcer nos compétences et de progresser plus rapidement.

I. Gameplay et Multijoueur

1. Multijoueur/Networking

Afin de comprendre le fonctionnement du multijoueur en mode peer-to-peer (P2P), j'ai visionné plusieurs vidéos explicatives, notamment sur YouTube. Le modèle P2P repose sur le principe selon lequel chaque joueur agit à la fois comme client (pour envoyer des données) et comme serveur (pour recevoir des données).

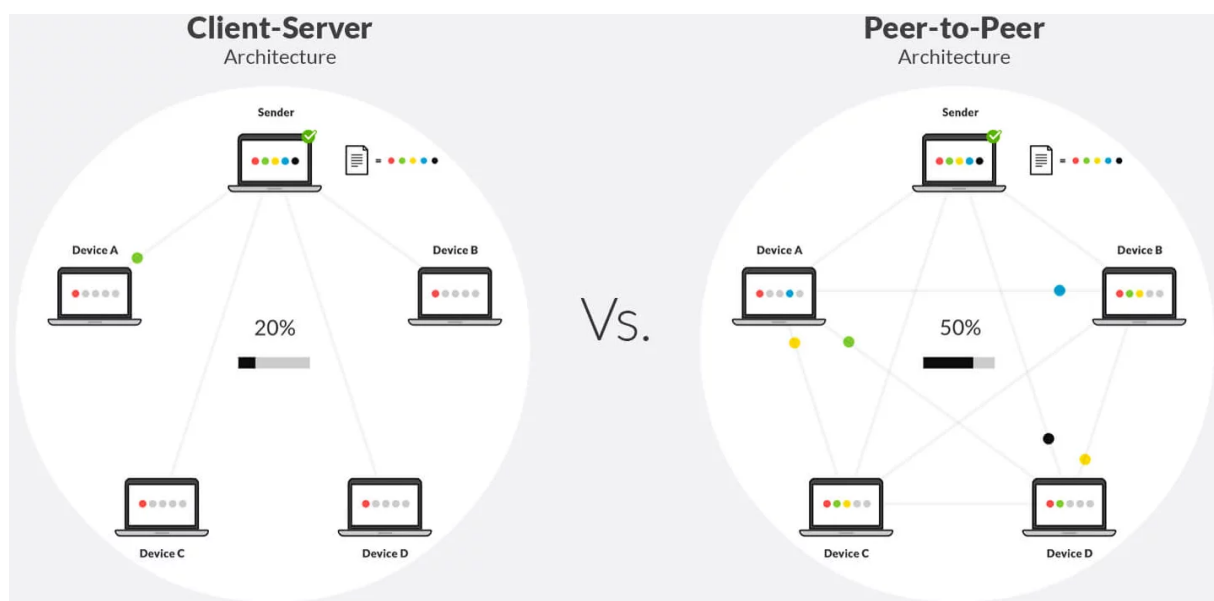
Parmi les différentes options disponibles pour gérer le networking, nous avons choisi d'utiliser l'outil Photon Fusion, réputé pour sa popularité et ses nombreuses fonctionnalités. Fusion propose un système dans lequel un joueur peut agir comme hôte tandis que les autres se connectent en tant que clients. Ce modèle P2P, associé à une architecture client-serveur décentralisée, nous semble particulièrement adapté au jeu multijoueur d'énigmes que nous développons.

Avantages du modèle P2P pour notre projet :

- Coûts réduits : Pas besoin d'héberger des serveurs dédiés.
- Simplicité de création de sessions : Les joueurs peuvent créer des parties sans dépendre de la disponibilité de serveurs officiels.
- Expérience sans latence significative : L'hôte bénéficie souvent d'une meilleure expérience de jeu.

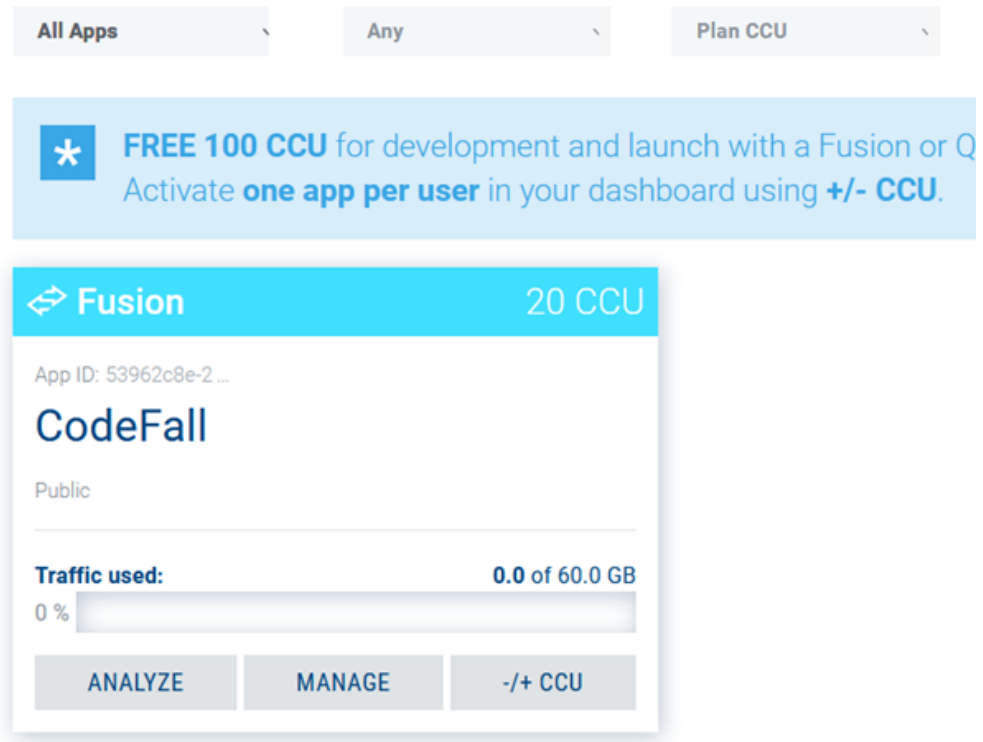
Inconvénients acceptables pour notre type de jeu :

- Échelle limitée : Le jeu est prévu pour deux joueurs seulement, ce qui n'est donc pas une contrainte.
- Avantage en termes de latence : Pour un jeu coopératif, cela n'est pas problématique.
- Facilitation de la triche : Ce risque est négligeable dans un jeu d'énigmes coopératif.



Après avoir installé Photon Fusion, j'ai consulté des vidéos explicatives et exploré la documentation officielle pour en comprendre le fonctionnement, notamment la configuration du projet à l'aide d'une AppID.

J'ai également étudié de nombreux projets existants développés avec Fusion, ce qui m'a permis de comprendre les concepts clés et d'ajouter des fonctionnalités au jeu. Par exemple, j'ai pu mettre en place un système permettant de faire apparaître un personnage après la création du serveur ou de la salle.



J'ai ensuite travaillé sur l'implémentation des mouvements du personnage en multijoueur, combinée à une caméra à la première personne (First Person View). Cela a nécessité de nombreuses recherches pour permettre aux joueurs de se déplacer selon la direction vers laquelle la caméra est orientée.

Le principal problème que j'ai rencontré concerne la gestion des caméras en multijoueur. Au lieu d'avoir une caméra propre à chaque joueur, comme prévu, la dernière caméra créée dans la scène est considérée comme la seule active. Ce bug, que je n'ai toujours pas résolu, provoque une inversion des caméras :

- L'hôte (joueur 1) voyait à travers les yeux du second joueur.
- Les déplacements du joueur 2 sont influencés par la direction de la caméra de l'hôte.

Bien que ce bug soit un obstacle, il m'a également inspiré une idée de gameplay. Nous avons décidé d'implémenter un mécanisme dans le jeu où, dans une salle spécifique, la caméra est volontairement inversée pour troubler les joueurs.

Pour permettre le bon fonctionnement du jeu, j'ai développé un menu principal dans lequel les joueurs peuvent :

- 1 - Saisir un pseudo et le nom de la salle.
- 2 - Créer un serveur en tant qu'hôte.
- 3 - Être redirigés vers une scène de lobby en attendant qu'un second joueur rejoigne la partie.

```
2 references
private async void StartGame(GameMode mode, string roomName, string sceneName)
{
    GameData.RoomName = roomName;
    _runnerInstance = FindObjectOfType<NetworkRunner>();
    if (_runnerInstance == null)
    {
        _runnerInstance = Instantiate(_networkRunnerPrefab);
    }

    _runnerInstance.ProvideInput = true;

    var startGameArgs = new StartGameArgs()
    {
        GameMode = mode,
        SessionName = roomName,
        ObjectProvider = _runnerInstance.GetComponent<NetworkObjectPoolDefault>(),
    };

    await _runnerInstance.StartGame(startGameArgs);

    if (_runnerInstance.IsServer)
    {
        _runnerInstance.LoadScene(sceneName);
    }
}
```

Player 5779

Room Name

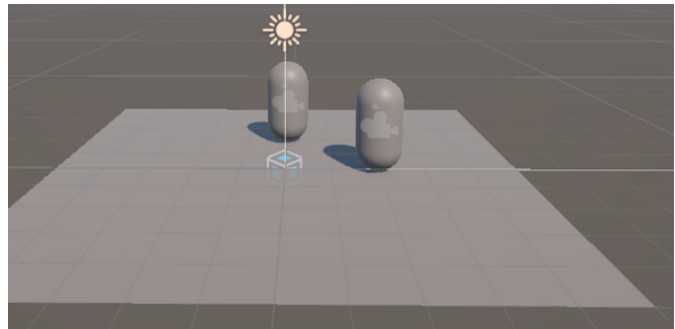
Start Host

Start Client

Dans le lobby, les fonctionnalités suivantes ont été implémentées :

- Affichage des pseudos des joueurs et du nom de la salle.
- Un bouton "Start" qui ne devient actif que lorsque les deux joueurs sont connectés.

Une fois le bouton "Start" pressé, les joueurs sont transportés dans la scène de jeu, où ils peuvent se déplacer et commencer à résoudre les énigmes.



2. Gameplay et Enigmes

Lors du jeu, nos deux joueurs devront évoluer de salle en salle et résoudre des énigmes pour avancer. La première étape pour concevoir ces énigmes a été d'en imaginer le déroulement. Les énigmes doivent répondre à quatre critères :

- La faisabilité : les énigmes doivent être équilibrées. C'est-à-dire qu'elles ne doivent pas être trop compliquées, mais pas trop simples non plus. En effet, les joueurs doivent réfléchir un minimum sur nos énigmes. De plus, la difficulté des énigmes doit être croissante au fur et à mesure du jeu.

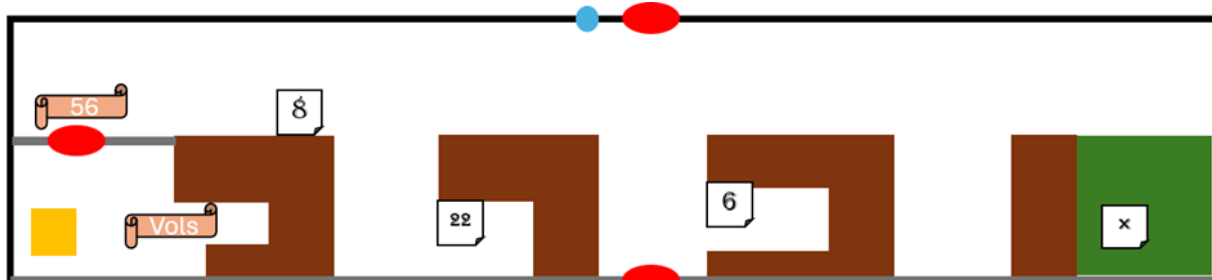
- L'utilisation de "bugs" : les énigmes de CodeFall doivent exploiter des "bugs", qui sont des mécaniques signatures du jeu. Ces derniers sont des failles dans le code du jeu, laissées intentionnellement pour permettre aux joueurs de résoudre les énigmes. L'utilisation de ces failles aura un impact sur la fin du jeu.

- L'originalité : nos énigmes doivent être originales pour éviter que le joueur ne s'ennuie ou qu'il ait la sensation de jouer à un jeu d'énigmes trop "générique".

- La coopération : CodeFall est un jeu à deux. De ce fait, il est essentiel que la solution des énigmes repose sur la coopération et la communication entre les deux joueurs.

J'ai donc conçu 3 salles d'énigmes aux totale qui apparaîtront dans le jeu dance ordre :

2ème partie :
Présentation globale :



(retrouver la légende du plan à la fin de la présentation de la salle)

- Le joueur se retrouve dans une bibliothèque vide et usée par le temps. Dans cette salle, nous trouvons une porte de sortie qui s'ouvre avec un code, une caisse et des bibliothèques vides.
- Le code pour ouvrir la porte est un code à 4 chiffres.
- La caisse est dans une partie fermée de la salle séparée du reste par un mur et une porte.
- Il y a également une plateforme sûre. - La caisse est buggée et il est possible de s'envoler avec. Le joueur devra découvrir cette mécanique.

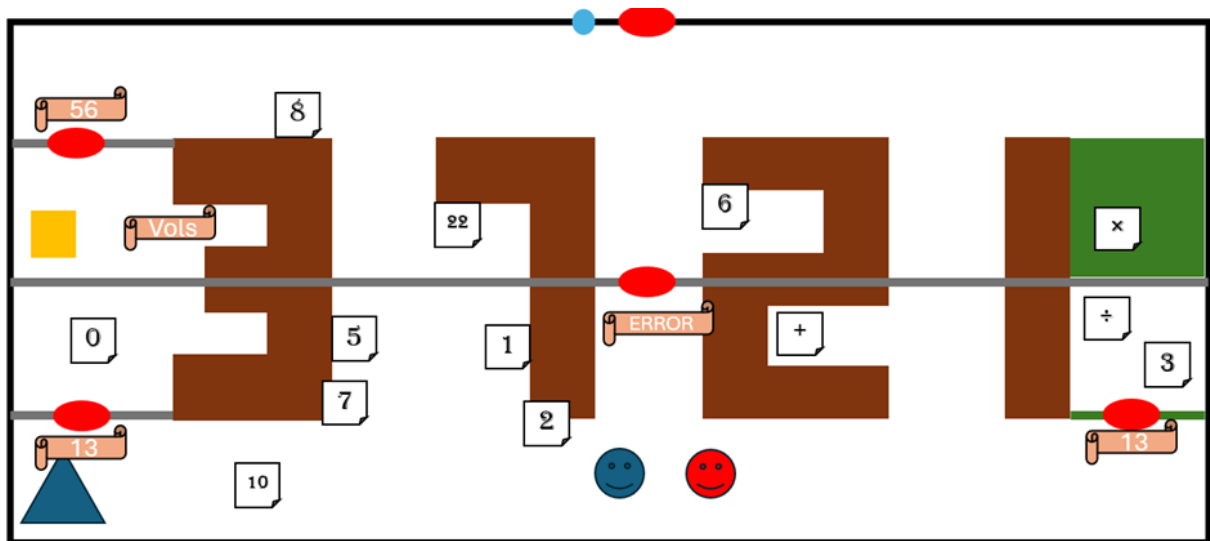
Solution :

Il faudra faire la courte échelle et récupérer la multiplication pour faire 7×8 sur la calculatrice afin d'accéder à la salle avec la caisse.

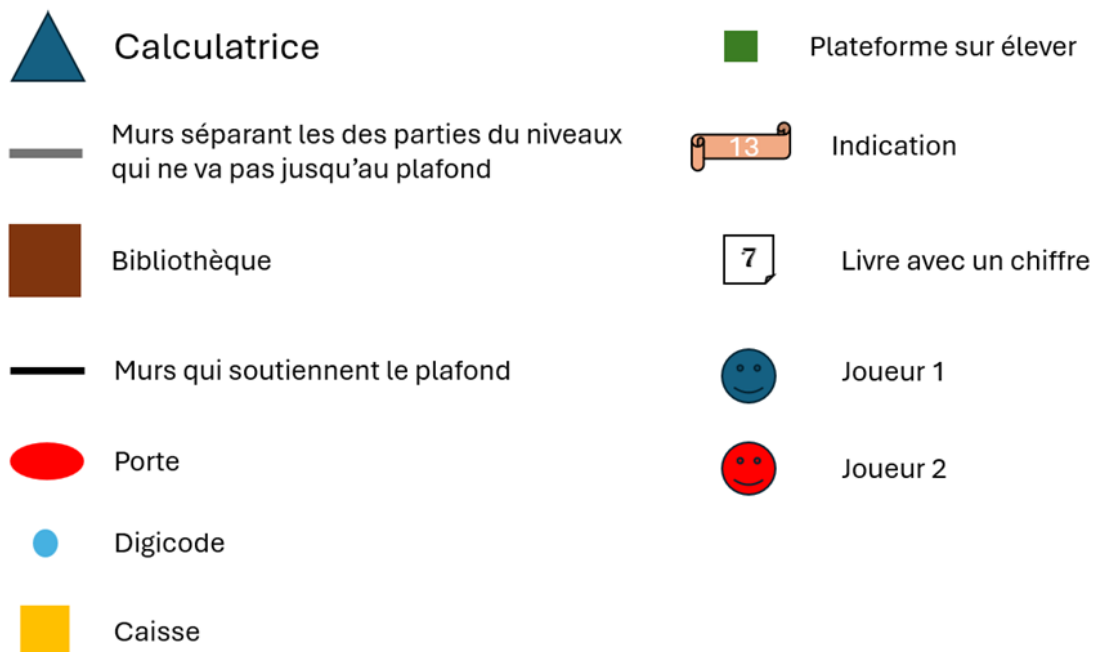
Ensuite, il faut utiliser la caisse pour prendre la hauteur et découvrir que les bibliothèques des deux salles forment un code à 4 chiffres.

Le code pourra être utilisé sur la porte pour l'ouvrir.

Plan Globale :



Légende :

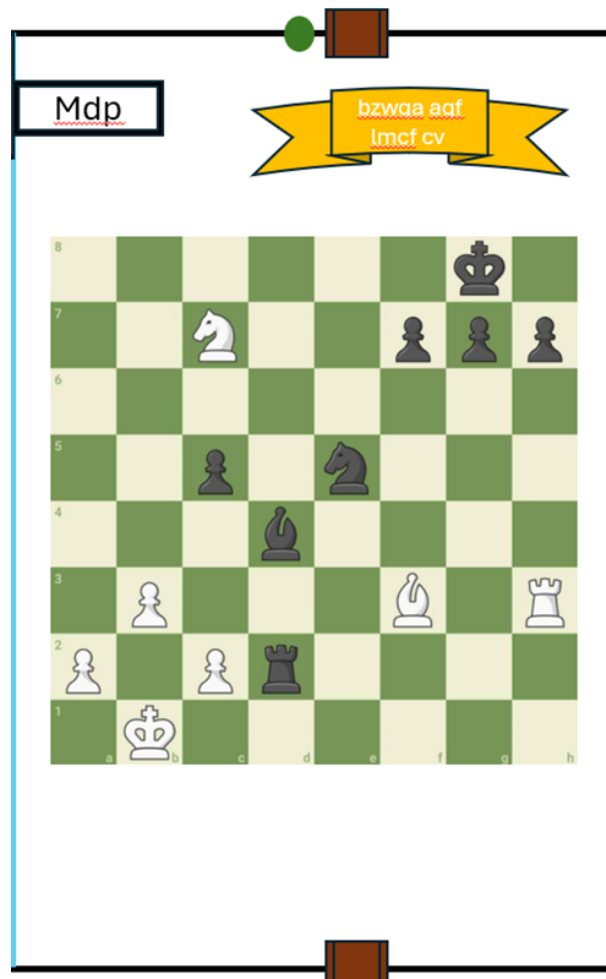


Échiquier :

Cette salle est répartie en deux parties séparé par une vitre, chaque joueurs se trouvent dans une partie différente :

1ère partie :

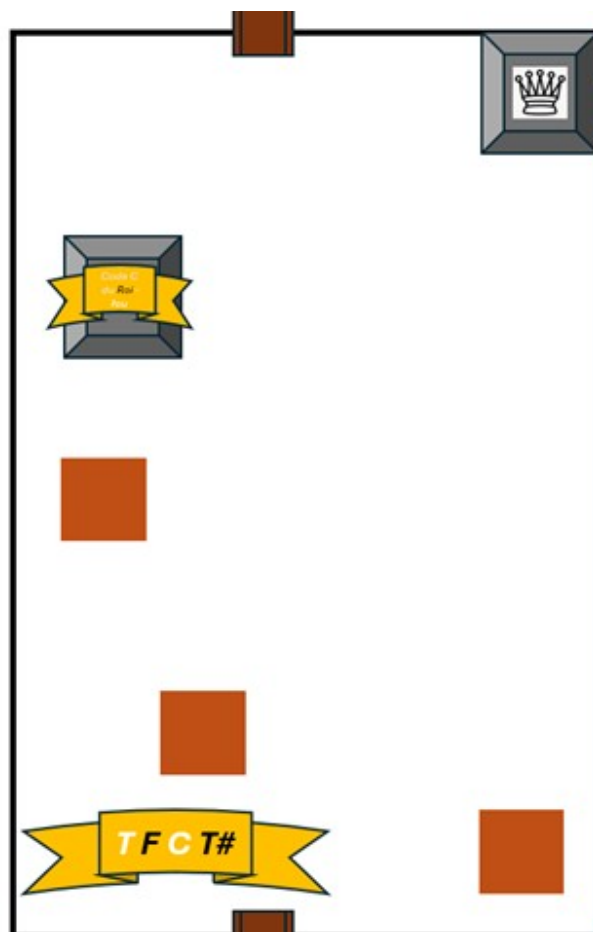
Présentation globale :



(retrouver la légende du plan à la fin de la présentation de la salle)

- La 1er moitié de la salle possède un échiquier géant avec des pièces géantes qui ne peuvent pas être dépassées, un indice avec marqué « bzwqa aqf lmcf cv » et deux keypads. Le premier avec des touches en alphabet-bizarre.(l'échiquier possède les coordonnées (a, b ,c ,etc. . .) mais en alphabet-bizarre). Le second avec des chiffres de 0 à 9 qui prend 4 digits.

2ème partie :
Présentation globale :



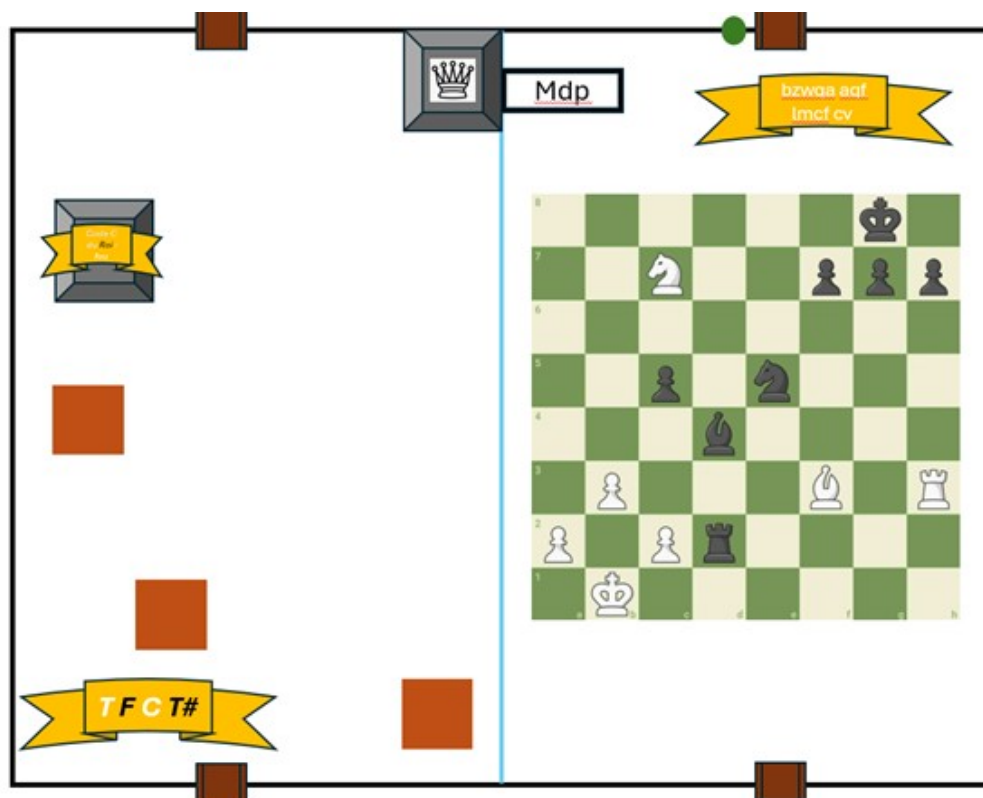
(retrouver la légende du plan à la fin de la présentation de la salle)

- La 2ème moitié de la salle possède 3 caisses, un indice avec marqué : T F C T# . Il y a aussi 2 coffres forts, le premier contient une reine géante, le second contient un indice avec marqué : «Code C du Roi(en noir) – fou(en blanc) ».








Solution globale :

- Pour résoudre cette énigme, il va falloir utiliser l'indice. En effet, il contient 4 éléments qui renvoient chacun à une pièce de l'échiquier. Il faut mettre les coordonnées des pièces dans le keypad pour ouvrir le coffre fort. Une fois le coffre ouvert, il faut faire passer la reine de l'autre côté du mur. Une fois cela fait, il est possible de la placer de telle sorte à faire échec et mat. Une fois le mât accompli, le deuxième coffre s'ouvrira et avec l'indice les joueurs devront déduire qu'il faut utiliser le code Caesar avec un décalage de 5 sur le message codé. Ils auront donc un code permettant d'ouvrir la porte de sortie.

Plan Globale :



Légende :

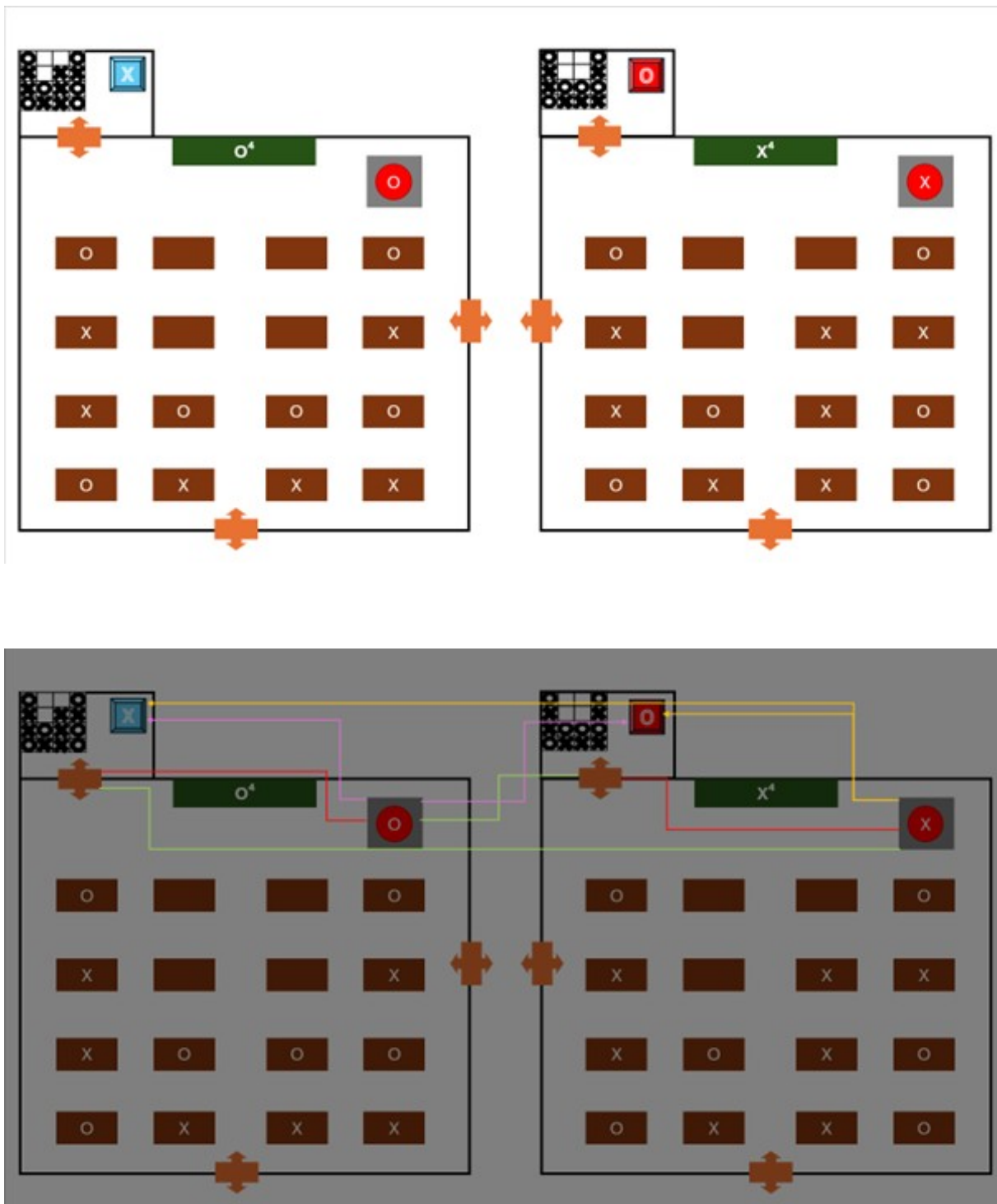
	Coffre fort		Digicode
	Keypad connecté au coffre fort		
	Caisse		
	Vitre/murs		
	Piece d'échec géant		
	Indice/instruction		

Salles de classe :

Ces salles sont séparées par un couloir, chaque joueur se trouve dans une salle différente. La spécificité de ce niveau est l'inversion des caméra des joueurs.

Salles :

Présentation globale :



- Légendes



- Dans ce niveau la caméra des 2 joueurs est inversée mais pas leurs contrôles, ils devront donc communiquer pour résoudre les énigmes.
- Les 2 salles possèdent chacune un interrupteur. L'un avec un « X » et l'autre avec un « O ». Les interrupteurs ouvrent une porte, en ferment une autre et modifient un écran.
- Chaque salle possède 16 tables avec soit un « X », ou un « O », ou rien dessus et un tableau.
- Chaque Salles possède une petite salle adjacente accessible par une porte (qui se ferme et s'ouvre en fonction d'un bouton). Les petites Salles contiennent une grille et un écran avec soit X, soit O dessus.

Solution :

- Les deux joueurs vont devoir récupérer les écrans et leur donner la bonne forme (X ou O en fonction du joueurs) pour faire un puissance 4.
- Pour cela, ils vont placer une caisse pour bloquer la porte et récupérer l'écran avec le bon signe. ils vont devoir ensuite les placer simultanément et appuyer plusieurs fois d'affilée sur les boutons ce qui va créer un bugs qui va ouvrir les 2 porte et laisser les joueurs avancer.

3. Histoire

Dans le cadre du projet CodeFall, je suis en charge de l'écriture de l'histoire et des dialogues, un élément essentiel pour donner du sens au jeu et permettre aux joueurs de s'y investir émotionnellement. Mon objectif est de créer une narration immersive qui accompagne le gameplay tout en faisant réfléchir les joueurs sur les conséquences de leurs actions sur l'environnement, à travers une métaphore du réchauffement climatique et des choix humains.

Pour l'instant, j'ai écrit un peu plus de 30 de l'histoire. Je me suis concentrée sur l'introduction de l'univers, la mise en place des bases narratives et les premières énigmes. Le scénario évolue en parallèle avec le gameplay : on commence dans un monde harmonieux, mais il se dégrade au fil des actions des joueurs lorsqu'ils exploitent des "bugs". Les dialogues, portés par les robots assistants, sont là pour guider les joueurs, leur fournir des indices et les pousser à réfléchir sur le message moral et philosophique du jeu.

L'un des plus grands défis a été de trouver le bon équilibre entre la narration et le gameplay. Les dialogues doivent être immersifs et intuitifs, expliquer les mécaniques du jeu sans ralentir l'action, et surtout enrichir l'expérience globale. Travailler avec l'équipe pour intégrer ces éléments tout en respectant les contraintes techniques d'Unity et des mécaniques de bugs a aussi demandé beaucoup d'adaptabilité. En parallèle, j'ai commencé à me former sur Blender pour mieux comprendre la création d'éléments visuels et collaborer plus efficacement avec le responsable des graphismes .

Pour l'écriture, je m'inspire de jeux comme Myst, pour la découverte progressive d'un univers captivant, Portal 2, pour son humour et l'ingéniosité de ses dialogues, et We Were Here Together, pour son aspect coopératif. Je m'appuie aussi sur le Principe de responsabilité de Hans Jonas, qui m'aide à donner une dimension philosophique et éducative à l'histoire.

Les prochaines étapes seront d'optimiser l'intrigue principale, d'approfondir les dialogues pour les niveaux avancés et de tester leur intégration dans le jeu afin d'en garantir la fluidité et l'impact. En parallèle, je vais continuer à progresser sur Blender pour contribuer davantage à la partie graphique.

II. Graphisme

1. Style graphique

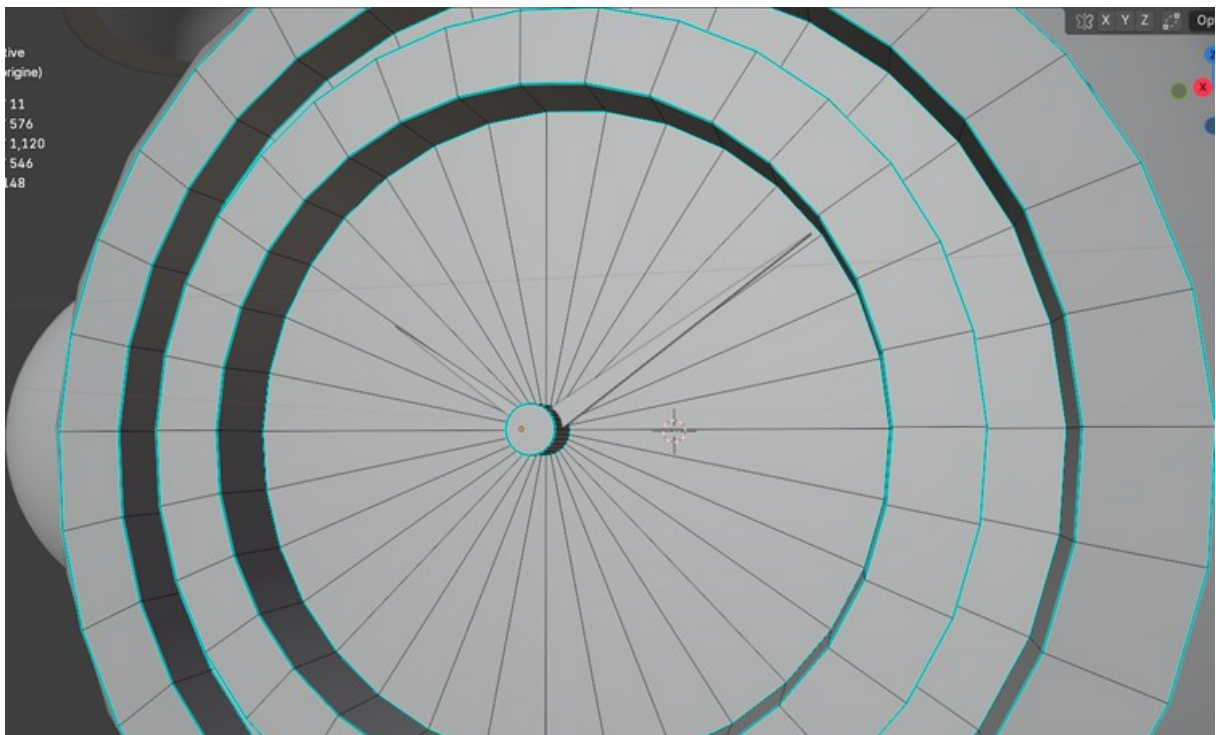
La réalisation du monde de CodeFall est plutôt difficile à imaginer. En effet, chaque salle du jeu représente un époque différente, ce qui veut dire qu'il peut-être est nécessaire de posséder plusieurs fois le même objet avec un style différent. Cela s'applique notamment aux appareils électroniques qui évoluent énormément en peu de temps. Nous avons donc sélectionné de nombreux modèles 3D misent à disposition sur des sites tels que : Turbosquid, Sketchfab ou encore Blendswap.



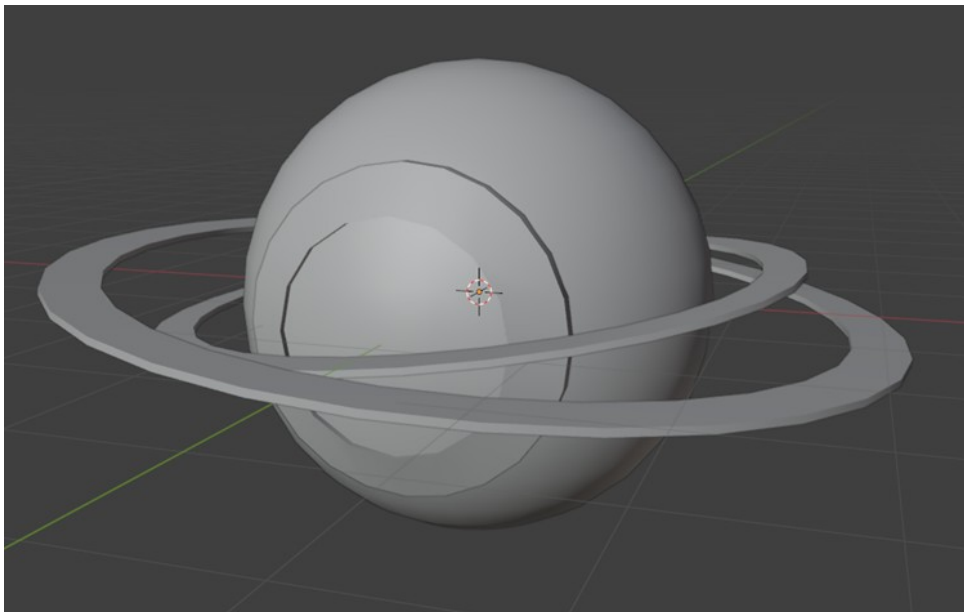
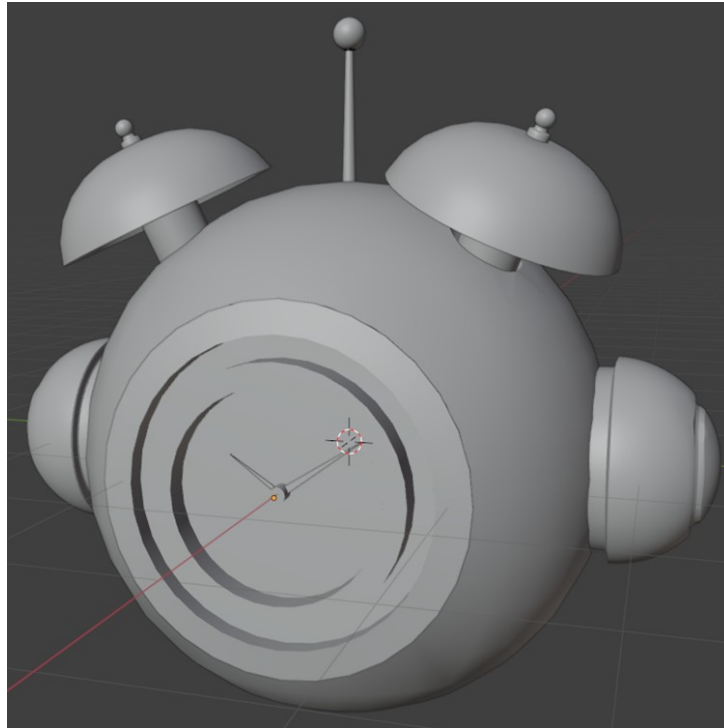
2. Robots

De plus, afin de personnaliser un maximum CodeFall et de retranscrire au mieux notre vision du jeu, nous utilisons le logiciel de modélisation 3D : Blender. Nous avons sélectionné ce logiciel car Alexandre avait déjà un peu appris à manipuler ce logiciel par le passé. Son premier objectif a été de créer l'un des deux robots du jeu. Mais avant de pouvoir le créer, il a fallu dans un premier tant trouver une idée de design afin d'avoir une base sur laquelle s'appuyer. J'ai donc réfléchi aux thèmes de CodeFall, ce à quoi allait servir le robot et ce qu'il représenterait. Finalement, il a été décidé que ce robot représenterait le temps. Suite à de nombreuses recherches sur le thème et après m'être inspiré de créations mis à disposition par des créateurs sur des réseaux sociaux tel que Pinterest, j'ai imaginé un design qui correspondait à mes attentes. J'ai donc commencé à le créer sur Blender mais cela était difficile de mes connaissances qui étaient limitées. J'ai donc dû suivre de nombreux tutoriels sur Youtube, consulté et appris les raccourcis des commandes utilisés de manière récurrentes. Afin de conserver toutes ces connaissances, j'ai créé un dossier dans lequel j'ai retranscrit tout ce que j'utilisais pour maîtriser au mieux, petit à petit ce logiciel. Toutefois, retranscrire ces idées est difficile puisqu'il n'existe pas d'aide pour chaque cas précis, il est donc nécessaire parfois de repenser ces idées.

Lorsque l'on modélise des objets 3D, il est essentiel de respecter certaines règles de topologie pour garantir que les modèles soient compatibles avec les moteurs de jeu et qu'ils soient faciles à manipuler. Une règle importante consiste à utiliser uniquement des faces triangulaires ou quadrangulaires (quads), cette pratique est cruciale pour plusieurs raisons. En effet, les moteurs de jeu comme Unity convertissent automatiquement toutes les faces en triangles pour le rendu. Si des polygones avec plus de quatre côtés (appelés N-gons) sont utilisés, la conversion peut produire des erreurs imprévisibles lors de l'affichage du modèle. De plus, cela permet de manipuler beaucoup plus facilement la création d'objet sur les logiciels de modélisations 3D. Enfin, les triangles sont les formes géométriques les plus simples à traiter par une carte graphique.



J'ai donc terminé la création du premier robot et commencé la création du deuxième.



Pour le moment, les textures n'ont pas encore été ajoutées car je n'ai pas encore tout à fait compris comment les exportés correctement sur Unity. En effet, il est simple d'exporter le modèle créer dans Unity mais ce n'est pas le cas pour les textures. Elles doivent exporter séparément et parfois certaines s'affiche tandis que d'autres non.

III. Site

Nous avons créé et mis en ligne un site web en utilisant les langages HTML et CSS. Ce site est accessible à l'adresse suivante :
<https://soiariis.github.io/Dtres/>

Le site est structuré avec un dossier dédié aux assets (images et fichiers) et un autre contenant trois fichiers .css, utilisés pour améliorer la lisibilité et l'ergonomie du site. De plus deux pages sont pour l'instant créées grâce au fichier .html ou index.html est la principale page (Accueil).

IV. Objectifs

2ème soutenance :

Pour la deuxième soutenance, l'objectif est de rendre le jeu jouable tout en résolvant les bugs techniques rencontrés jusqu'à présent, notamment le problème de la caméra qui n'est pas assignée au bon personnage.

Il est aussi nécessaire de gérer la déconnexion d'un joueur : si l'un des joueurs quitte la partie, celle-ci doit automatiquement s'arrêter. Par ailleurs, il faudra implémenter un système d'affichage de messages, indiquant si la partie est déjà complète ou si le salon n'existe pas, lorsqu'un joueur tente de se connecter.

Nous devons par ailleurs développer un système d'options permettant de mettre le jeu en pause et de modifier certains paramètres, tels que le volume du son, les touches de contrôle, etc.

Il sera nécessaire de concevoir une IA pour les robots. Ces derniers devront suivre chacun un joueur, en trouvant le chemin le plus optimal pour les rejoindre.

Des améliorations sont également prévues pour le site, notamment en complétant la page d'accueil et en optimisant le visuel de la section "À propos". Il faudra y rédiger une description complète de l'équipe, de l'entreprise et du jeu. De plus, il conviendra d'afficher les ressources utilisées (logiciels et assets) ainsi que les moyens de contacter l'équipe.

Nous devons aussi trouver des musiques pour le menu, les crédits et une musique par salle. Les musiques seront personnalisées pour créer des "effets" de bugs.

Enfin, nous allons concevoir les deux salles dans lesquelles les joueurs apparaîtront : la première salle d'énigmes ainsi que le début de la deuxième salle d'énigmes.

3ème soutenance :

Pour cette troisième soutenance, nous devons présenter un jeu et un site web pleinement terminés et fonctionnels.

Concernant le site, il est nécessaire de faire fonctionner tous les boutons, d'afficher les images du jeu, d'ajouter les informations de contact et de finaliser le style visuel du site.

Le jeu devra inclure un menu permettant de se connecter au mode multijoueur, limité à deux joueurs par salle. Les salles et les énigmes doivent être complètes, ainsi que les robots et leur IA pour suivre les joueurs. Enfin, l'audio et les dialogues devront être intégrés et fonctionnels.

V. Conclusion

En conclusion, nous avons déjà réalisé des progrès significatifs dans le développement du mode multijoueur, ainsi que dans la création des deux robots. De plus, les énigmes et le gameplay ont presque entièrement été conçus et attendent d'être implémentés lors de la deuxième soutenance.

Le site nécessite également des améliorations, et il reste à développer l'IA ainsi qu'un menu pour implémenter un système d'options.

Enfin, notre objectif pour la troisième soutenance est de finaliser le jeu et le site, afin de vous présenter le résultat final de notre travail.