

Dtres



Rapport de soutenance n°2

CodeFall



Promotion 2029

Florian GAGNIER
Imane SARWAR
Florent FOSSATI
Alexandre SALGUEIRO

C2

Table des matières

Introduction	3
I Vue générale du projet	3
1 Répartition des tâches	3
2 Ressenti général	3
II Avancement du projet	4
1 Histoire	4
2 Gameplay et Multijoueur	4
A Multijoueur/Networking	4
B Gameplay et Enigmes	5
C Intelligence artificielle	10
3 Graphisme	12
A Modélisation	12
B Salles	13
4 Ambiance sonore	15
5 Interface	16
6 Site	17
III Bilan sur l'avancement	19
IV Développement futur	19
Conclusion	20

Introduction

Dès le début du projet, nous avons défini des objectifs clairs ainsi qu'une méthodologie adaptée pour mener à bien notre travail. Nous avons choisi d'héberger notre jeu et notre site web sur GitHub afin de faciliter la collaboration et le partage du code entre les membres de l'équipe. 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 pour la création et le design de notre jeu.

I. Vue générale du projet

1. Répartition des tâches

Pour optimiser notre organisation, nous nous sommes attribués des rôles, en fonction de ses compétences et affinités :

Répartition des tâches	Gagnier Florian	Fossati Florent	Salgueiro Alexandre	Sarwar Imane
Menu/Options			*	+
Multijoueur	*	+		
Audio		*	+	
IA		*		+
Enigmes et Mécaniques de jeu	+	*		
histoire et dialogues	+			*
Graphismes/Texture			*	+
Site Web	*		+	

Légende

*	Responsable
+	Aide directe

Grâce à cette répartition, nous avons pu avancer efficacement tout en maintenant une bonne coordination entre les différentes parties du projet.

2. Ressenti général

Globalement, le développement suit son cours et nous avons atteint plusieurs objectifs majeurs. Cependant, certains aspects restent complexes à gérer, notamment la synchronisation des objets en mode multijoueur et l'optimisation des performances ; ce qui nous amène à adapter nos méthodes de travail pour y faire face.

II. Avancement du projet

1. Histoire

L'histoire du jeu est bien avancée, avec plus de 70% du scénario finalisé. Nous avons rédigé les dialogues d'introduction ainsi que ceux des deux premières énigmes. Pour rendre l'expérience plus immersive, nous envisageons d'enrichir encore les interactions entre les personnages et d'ajouter des éléments narratifs dynamiques en fonction des choix des joueurs.

2. Gameplay et Multijoueur

A. Multijoueur/Networking

Pour cette soutenance, nous avons intégré un système de gestion des déconnexions afin d'assurer une meilleure stabilité en multijoueur. Lorsque deux joueurs sont dans le lobby, la fermeture du lobby dépend de qui se déconnecte. Si l'hôte quitte la session, alors les deux joueurs sont déconnectés et le lobby est fermé.

En revanche, si c'est le client qui se déconnecte, seul son départ est pris en compte, tandis que l'hôte reste connecté et peut continuer à attendre d'autres joueurs. Une fois la partie lancée, quelle que soit la personne qui quitte – hôte ou client –, la partie s'arrête automatiquement pour garantir une cohérence du jeu.

```
public void OnPlayerLeft(NetworkRunner runner, PlayerRef player)
{
    // Vérifiez si nous sommes dans la scène du lobby
    if (SceneManager.GetActiveScene().name == _lobbySceneName)
    {
        // Si nous sommes dans la scène du lobby, déconnectez uniquement le client
        runner.Disconnect(player);
        return;
    }

    // Disconnect the other player and redirect to the main scene
    foreach (var p in runner.ActivePlayers)
    {
        if (p != player)
        {
            runner.Disconnect(p);
        }
    }

    // Remove characters from the scene
    foreach (var obj in FindObjectsOfType<NetworkObject>())
    {
        if (obj.HasStateAuthority)
        {
            runner.Despawn(obj);
        }
    }

    // Shutdown the server and load the main scene
    runner.Shutdown();
    SceneManager.LoadScene(_menuSceneName);
}
```

Nous avons également travaillé sur un bug lié à la gestion de la caméra. Désormais, chaque joueur voit bien l'environnement à travers sa propre caméra, ce qui constitue une avancée significative. Cependant, un problème persiste : si l'hôte se déplace correctement en fonction de la direction indiquée par sa caméra, le client, lui, suit une logique différente. Son déplacement est influencé par la caméra de l'hôte, bien qu'un léger mouvement initial dans la bonne direction soit perceptible. Après de nombreuses recherches, consultations de forums et tests approfondis, ce problème demeure non résolu, et nous continuons à explorer des solutions possibles.

Nous avons remarqué un autre problème concernant la synchronisation des pseudos dans le lobby. Actuellement, chaque joueur ne voit que son propre pseudo, ce qui signifie que l'hôte ne voit que son nom et le client uniquement le sien, sans afficher celui de l'autre. Ce défaut nuit à l'expérience multijoueur et nous pousse à envisager une refonte partielle du système de gestion du lobby afin d'assurer un affichage correct des informations et une meilleure synchronisation entre les joueurs.

B. Gameplay et Enigmes

Dans un jeu d'énigmes, il est essentiel que le joueur puisse interagir avec certains éléments du décor, indispensables à la résolution des différentes énigmes. CodeFall étant un jeu basé sur la réflexion, nous avons choisi de limiter le nombre d'objets interactifs pour nous concentrer sur des mécaniques pertinentes et bien intégrées au gameplay. Parmi ces éléments, deux sont particulièrement essentiels à l'expérience du jeu.

Le premier est une caisse mobile, que les joueurs peuvent déplacer pour créer des plateformes ou des escaliers, leur permettant ainsi d'accéder à des endroits autrement inatteignables. Le principal défi a été d'implémenter un système de déplacement intuitif, où l'objet suit naturellement le regard du joueur et peut être saisi uniquement lorsqu'il se trouve à une distance réaliste.

Pour cela, nous avons utilisé un Raycast, une technique qui permet de détecter les objets situés dans l'axe de vision du joueur. Cette approche nous a permis de rendre la prise en main fluide et immersive, tout en garantissant un contrôle précis sur les interactions avec l'environnement.



L'utilisation du Raycast a considérablement amélioré notre mécanique de jeu. Plutôt que d'intégrer un script de déplacement dans chaque objet interactif, nous avons directement géré cette fonctionnalité dans le script de la caméra du joueur.

Pour identifier les objets pouvant être déplacés, nous avons mis en place un tag spécifique : "CanBeMove". Seuls les objets portant ce tag peuvent être manipulés par le joueur.

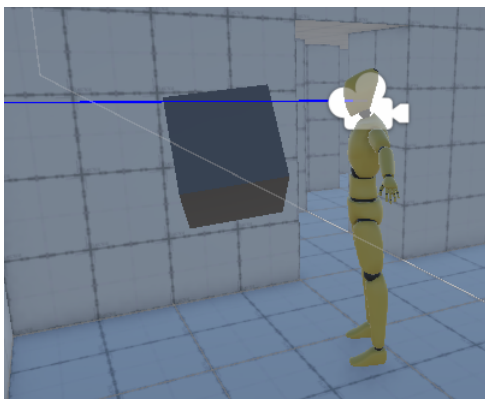
Concrètement, lorsque le joueur regarde un objet, un Raycast est projeté depuis la caméra dans la direction de son regard. Si le premier objet touché possède le tag CanBeMove, sa position est alors modifiée en fonction des mouvements du joueur, lui permettant ainsi de le déplacer librement dans l'environnement.

```
void Update()
{
    Vector3 mousePos = Input.mousePosition;
    mousePos.z = 3f;
    mousePos = cam.ScreenToWorldPoint(mousePos);
    Debug.DrawRay(transform.position, mousePos - transform.position,
    Color.blue);

    if (HoldingAnObject)
    {
        ObjectHold.transform.position = this.transform.position + Camera.main.transform.forward * 1.55f;
        if (Input.GetMouseButtonUp(0))
        {
            HoldingAnObject = false;
        }
    }
    else if (Input.GetMouseButtonUp(0) && !HoldingAnObject)
    {
        Ray ray = cam.ScreenPointToRay(Input.mousePosition);
        RaycastHit hit;

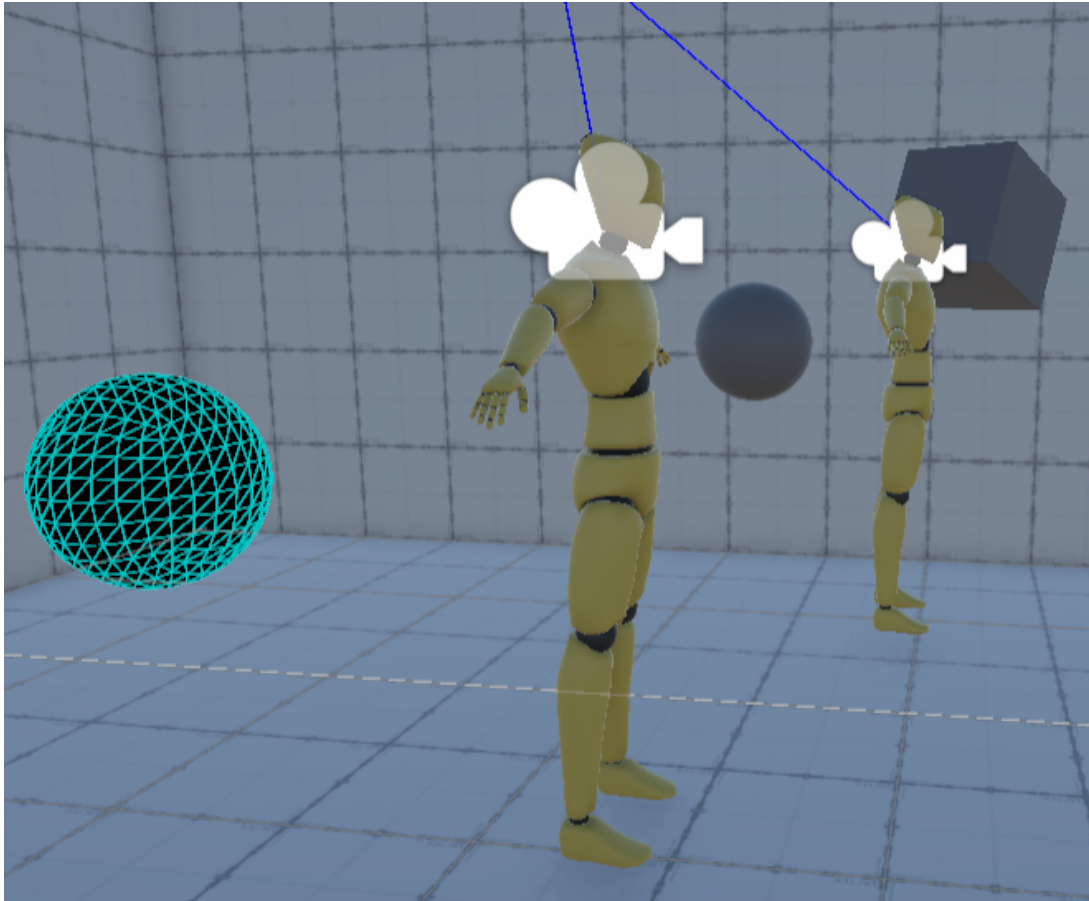
        if (Physics.Raycast(ray, out hit) && hit.distance < 3 && hit.collider.CompareTag("CanBeMove"))
        {
            ObjectHold = hit;
            HoldingAnObject = true;
        }
    }
}
```

Les ray font déjà partie de unity et nous ont grandement facilité la création de cette mécanique.



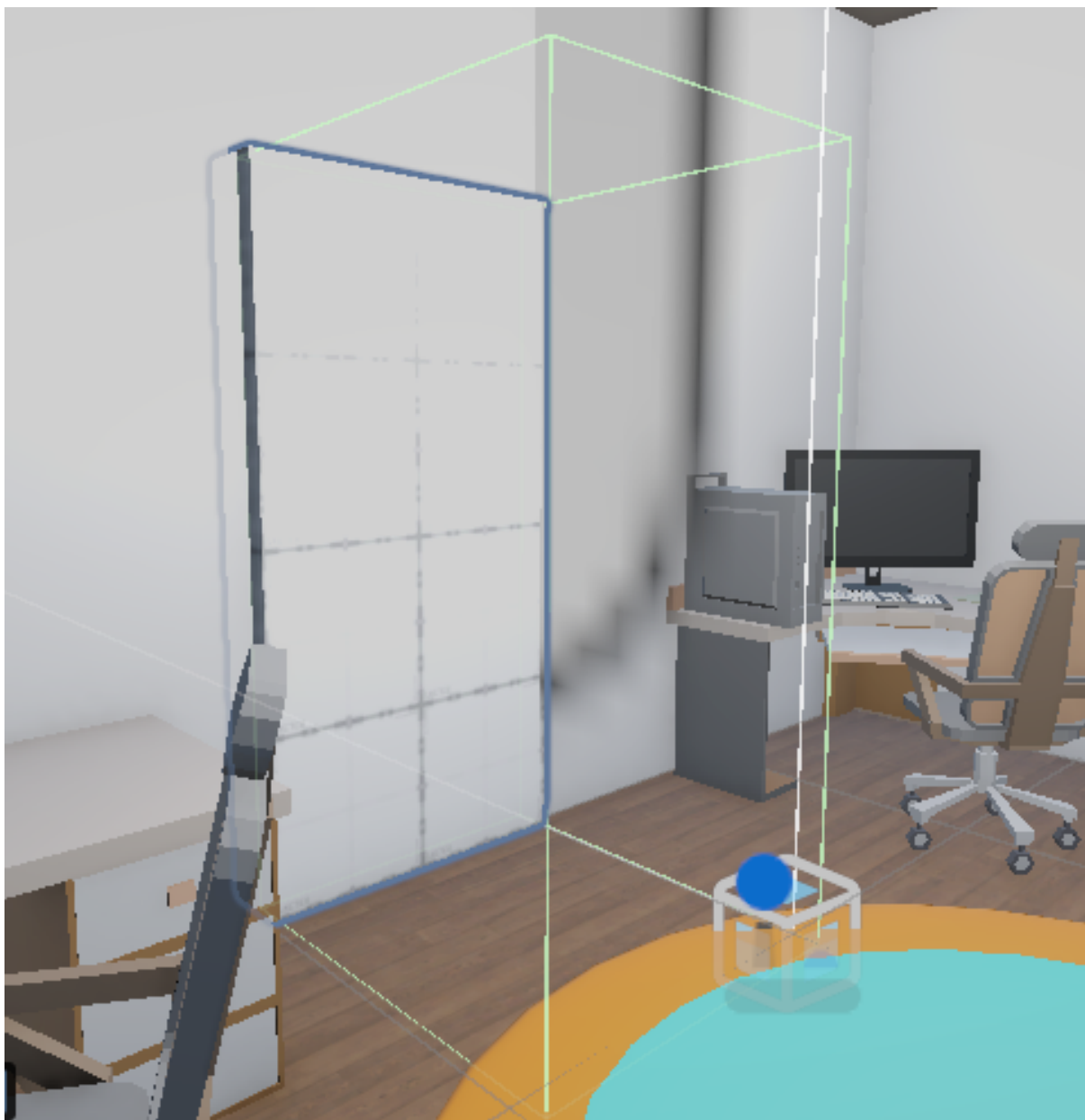
Cette mécanique ouvre de nouvelles opportunités de gameplay, car elle ne se limite pas uniquement aux caisses. À terme, nous pourrions l'adapter à d'autres objets interactifs, enrichissant ainsi les possibilités offertes aux joueurs.

Cependant, nous avons rencontré un problème de synchronisation des caméras en mode multijoueur, ce qui entraîne des incohérences. Lors des tests, nous avons constaté qu'un seul des deux joueurs peut déplacer la caisse correctement. Si l'autre joueur tente de la manipuler, l'objet se repositionne en fonction de la caméra du premier joueur, ce qui perturbe l'expérience.



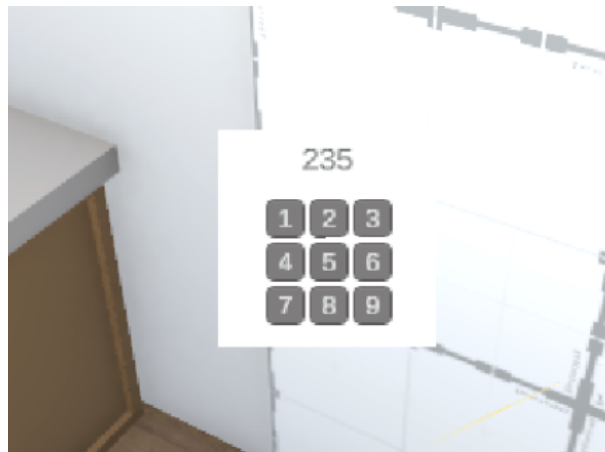
Pour résoudre ce problème, la priorité est d'ajuster la gestion des caméras en multijoueur. Une fois cette étape corrigée, nous pourrons nous assurer que chaque joueur peut interagir avec les objets indépendamment et sans interférences.

Le second élément interactif essentiel que nous avons intégré dans CodeFall est une porte à code. Cet élément joue un rôle clé dans le jeu, car les joueurs devront à plusieurs reprises décrypter, assembler et entrer des codes pour progresser. L'objectif est d'ajouter une dimension logique et stratégique, où chaque code découvert devient une clé pour ouvrir une porte et avancer dans l'aventure.



Pour créer la porte, nous avons mis en place une zone de détection permettant de vérifier si le joueur s'y trouve.

Une fois dans cette zone, un script vérifie s'il appuie sur la touche E. Si c'est le cas, un clavier numérique (keypad) s'affiche à l'écran. Celui-ci est composé de neuf boutons, chacun renvoyant une valeur différente au script.



Si le code entré est correct, la porte déclenche alors une animation d'ouverture.

De plus afin d'éviter que le joueur soit bloqué s'il tape le mauvais code, nous avons rajouté ce script :

```
if (codeTextValue.Length >4)
{
    codeTextValue="";
}
```

Son rôle est de vérifier si le joueur tente de dépasser la limite de chiffres autorisée. Si c'est le cas, le code saisi est réinitialisé, permettant ainsi au joueur d'essayer une autre combinaison. Ce système présente l'avantage d'éviter au joueur de devoir fermer le keypad à chaque tentative de code.

La création de cette porte a été fastidieuse, car nous avons dû la recommencer deux fois. De plus, nous avons dû apprendre le fonctionnement des animations sur Unity afin d'assurer une ouverture fluide et sans problème de collision.

L'implémentation de cette porte nous permettra d'en créer des copies avec différents codes d'ouverture et de tailles variées. En effet, certaines portes à code pourront être utilisées pour l'ouverture de coffres-forts ou d'autres mécanismes similaires.

C. Intelligence artificielle

Nous avons réalisé une IA pathfinding pour notre robot. Pour cela nous avons créé 3 paramètres qui peuvent être changé :

```
private Transform _target;  
private float _followDistance = 2.0f;  
private float _moveSpeed = 3.0f;
```

- “target” est la cible que le robot va suivre, il va s’agir d’un joueur.
- “_followDistance” est une constante qui va nous aider dans le calcul de la position du robot plus tard.
- “_moveSpeed” représente la vitesse du robot, nous l’avons ajustée pour que le robot ne soit ni trop rapide, ni trop lent.

Nous avons conçu l’algorithme suivant pour calculer à quelle position devrait se trouver le robot :

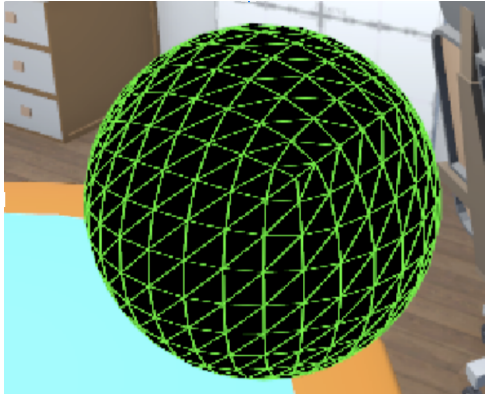
```
if (_target != null)  
{  
    Vector3 targetPosition = _target.position - _target.forward * _followDistance;  
    Vector3 direction = (targetPosition - transform.position).normalized;  
    Vector3 vecteurOriginal = (direction * _moveSpeed * Runner.deltaTime);  
    transform.position += vecteurOriginal;  
}
```

Cependant, cet algorithme nous a posé des problèmes. En effet, ses calculs étaient trop précis, ce qui entraînait un changement constant de position du robot, même lorsque le joueur était immobile. Ce dysfonctionnement donnait l’impression que le robot tremblait.

Toutefois, nous avons réussi à résoudre ce problème en ajoutant :

```
Vector3 targetPosition = _target.position - _target.forward * _followDistance;  
Vector3 direction = (targetPosition - transform.position).normalized;  
if (Vector3.Distance(transform.position, _target.position) <= 2.2f)  
{  
    _moveSpeed=0;  
}  
else  
{  
    _moveSpeed=3;  
}  
Vector3 vecteurOriginal = (direction * _moveSpeed * Runner.deltaTime);  
transform.position += vecteurOriginal;
```

Grâce à cet ajout, le robot ne peut plus se déplacer s'il est trop proche du joueur, donnant ainsi l'impression qu'il attend à proximité de celui-ci.



Nous avons également décidé de désactiver la hitbox du robot. En effet, afin d'éviter qu'il ne bloque le joueur, nous avons choisi de faire en sorte qu'il ne puisse ni le pousser ni interagir physiquement avec d'autres objets du jeu.

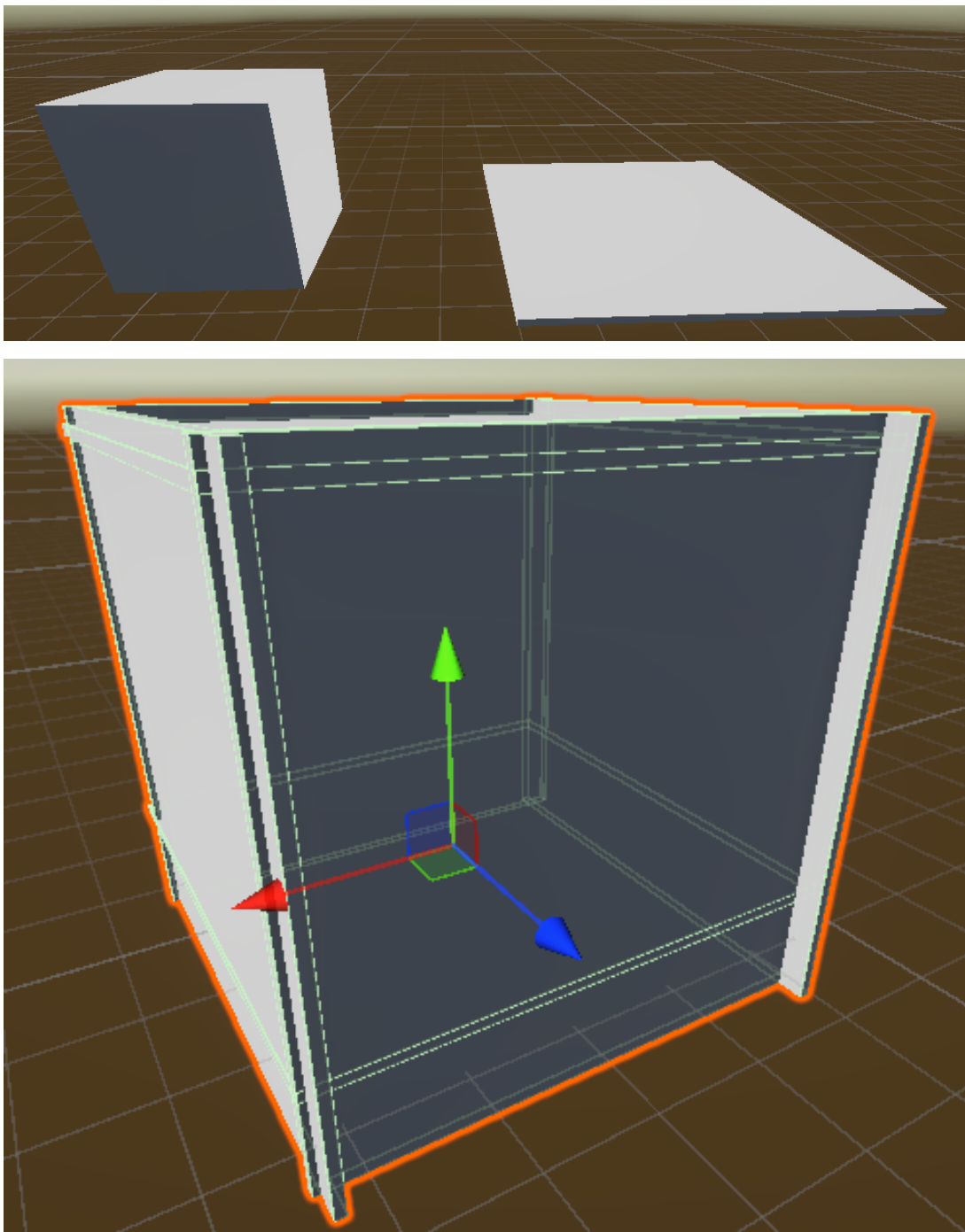
Pour conclure, nous avons réussi à concevoir un robot qui suit le joueur tout en restant discret et non intrusif.

3. Graphisme

A. Modélisation

Lors de la première soutenance, seul le développement des robots avait été réalisé, laissant le jeu pratiquement vide. L'objectif de cette deuxième soutenance était donc de mettre en place le spawn des joueurs ainsi que les deux premières salles du jeu.

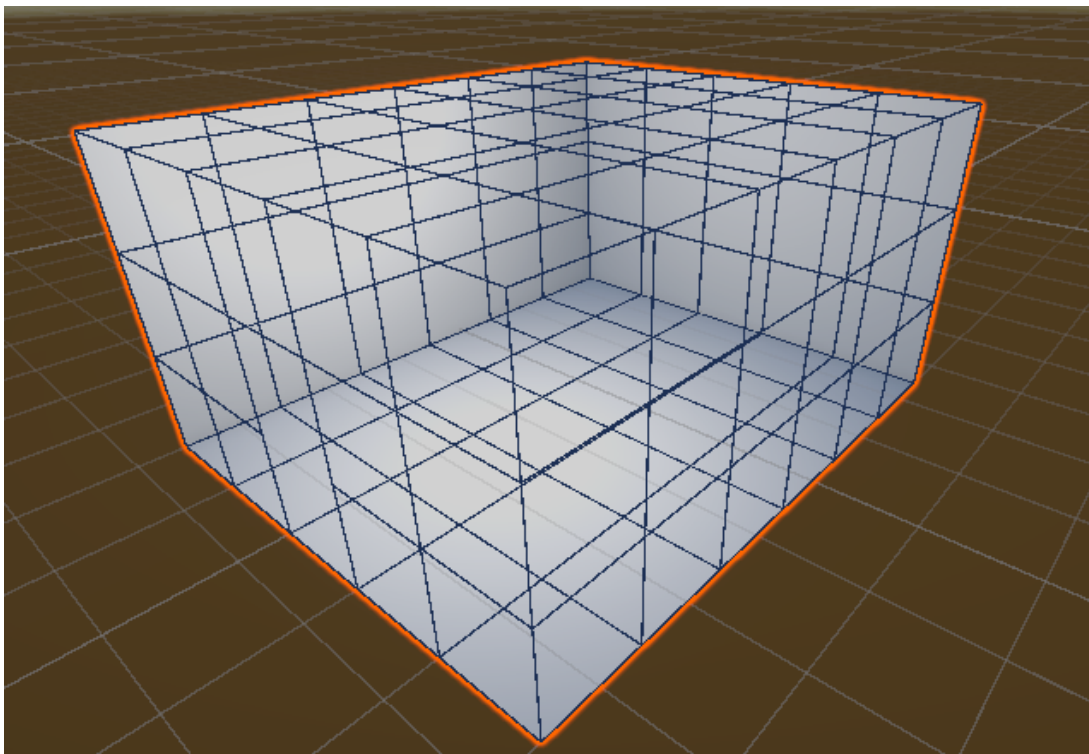
Dans un premier temps, il a fallu trouver une méthode efficace et simple pour concevoir la structure principale de chaque salle. Initialement, nous avons envisagé une approche naïve consistant à créer un cube, l'agrandir, puis l'aplatir afin d'en faire des murs, un sol et un plafond. Cependant, nous avons rapidement constaté que cette méthode était loin d'être optimale.



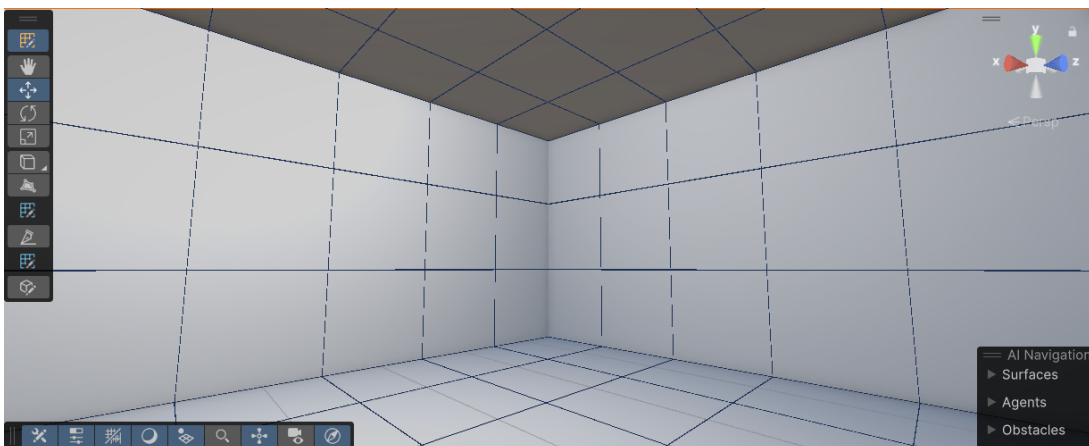
En effet, cette méthode limitait considérablement les possibilités de modification et d'expansion tout en manquant de précision. Après quelques recherches, nous avons identifié ProBuilder, un outil de Unity, comme la solution idéale à ce problème.

Simple d'utilisation et intuitif, ProBuilder permet de créer une grande variété de volumes en quelques instants et offre de nombreux outils de modification. Bien que son apprentissage ait initialement ralenti l'avancement graphique du jeu, cet investissement en temps a rapidement été compensé. Désormais, la création des salles est bien plus rapide et efficace.

B. Salles



De plus, cet outil nous permet d'avoir un aperçu extérieur de chaque salle, ce qui facilite le placement des objets tout en offrant une immersion totale une fois à l'intérieur de la pièce.



Afin de donner vie à chaque pièce, nous avons décidé d'utiliser principalement des assets issus du Unity Asset Store. Bien que nous ayons initialement sélectionné des modèles 3D sur divers sites, nous avons rapidement rencontré des problèmes de compatibilité avec Unity. En effet, de nombreux modèles se chargeaient correctement, mais leurs textures associées ne s'affichaient pas, laissant place à une texture rose par défaut.

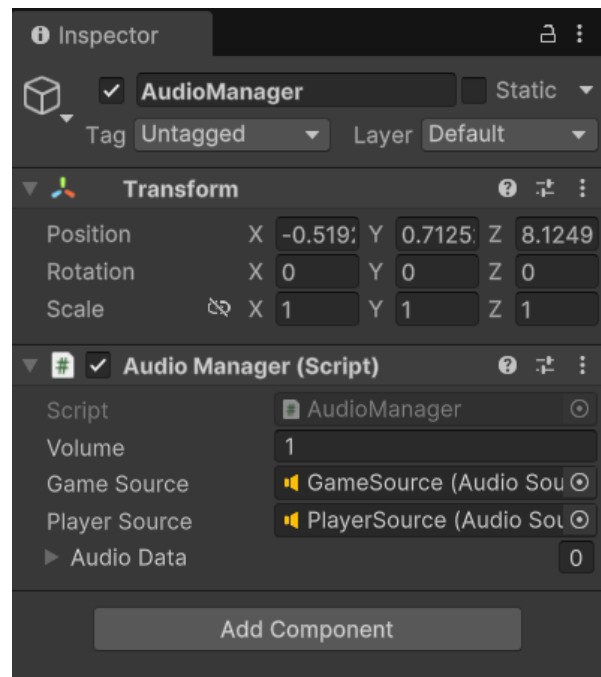
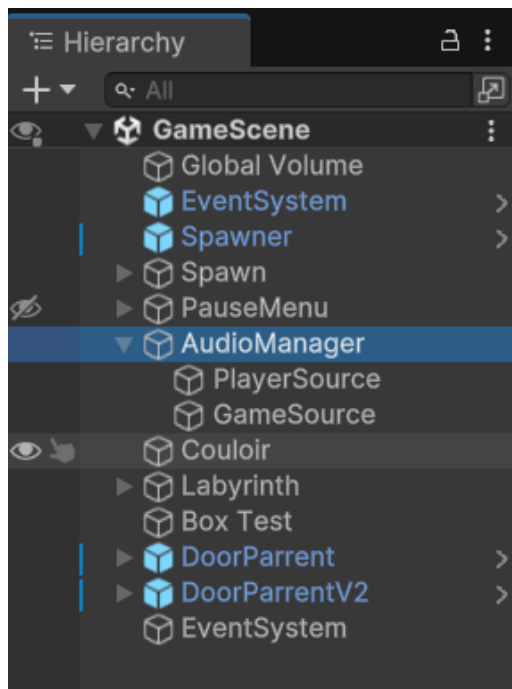
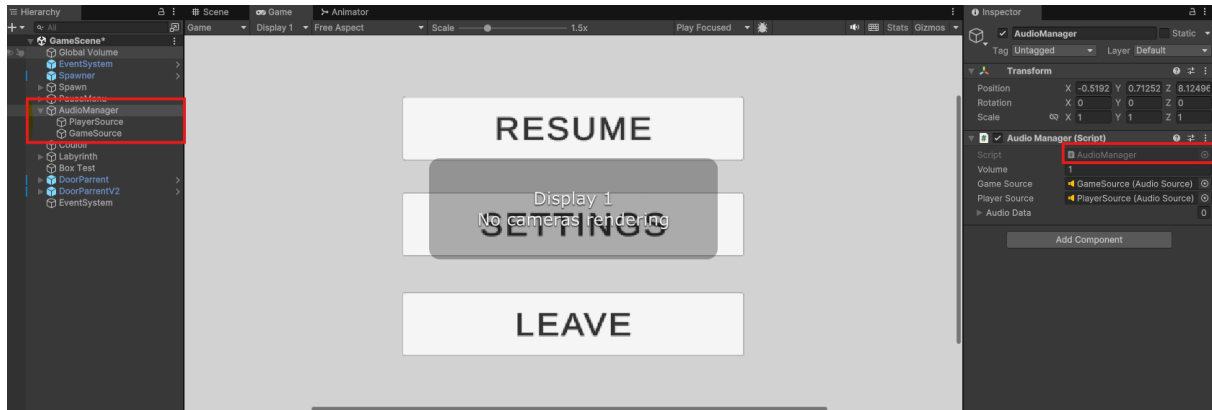
Face à ces limitations, nous avons opté pour une approche plus fiable en utilisant exclusivement le Unity Asset Store. Ce choix nous permet non seulement d'accéder à une large variété de modèles, mais aussi de nous assurer d'une compatibilité optimale avec Unity. Toutefois, lorsque les modèles disponibles ne correspondent pas à nos attentes, nous les créons nous-mêmes sur Blender afin de garantir un rendu fidèle à notre vision du jeu.

Désormais, les joueurs apparaissent mystérieusement dans une chambre, où ils font la rencontre de deux robots. Nous avons également conçu une salle dédiée à l'énigme de la bibliothèque, mais en raison de problèmes de compilation, nous n'avons pas pu l'intégrer dans cette version du projet.



4. Ambiance sonore

Nous avons intégré un système audio à notre jeu. Dans cette version, nous avons mis en place un Audio Manager qui joue une musique de fond pour les deux joueurs. Cette musique se répète en boucle et peut avoir une durée quelconque.



Comme illustré sur l'image, un script a été attaché à la gestion du son. Ce script servira à l'avenir à intégrer des musiques dynamiques, adaptées aux différentes situations du jeu, ainsi qu'à ajouter des bruitages pour les actions des joueurs (ex. : bruits de pas, saut, interactions, etc.).

5. Interface

Nous avons implémenté un menu principal au démarrage du jeu, permettant au joueur de choisir entre jouer, accéder aux options ou quitter. Pour l'instant, l'interface et le style visuel restent très simplistes, car notre priorité a été de nous concentrer sur le bon fonctionnement du menu avant d'améliorer son apparence.



Le menu est opérationnel et accessible à tout moment en jeu en appuyant sur Échap. Les options actuellement disponibles permettent de modifier la sensibilité de la souris et d'activer/désactiver le mode plein écran. D'autres fonctionnalités seront ajoutées dans les futures mises à jour.

Problèmes d'affichage rencontrés

Un souci persiste lors de l'ouverture du menu en jeu. Actuellement, lorsque le joueur appuie sur Échap, trois choix s'affichent :

- Reprendre la partie (ferme le menu)
- Options (ouvre le menu des options)
- Quitter (redirige vers le menu principal)

Le menu peut être refermé en appuyant à nouveau sur Échap. Cependant, si le joueur se trouve dans le menu des options et appuie sur Échap, la prochaine fois qu'il ouvrira le menu, il restera bloqué dans l'écran des options. Ce problème vient du fait que pour fermer le menu des options, il faut obligatoirement cliquer sur un bouton à l'écran, ce qui empêche un retour automatique au menu principal du jeu.

Origine du problème et solutions envisagées

Sur Unity, l'affichage de l'UI (interface utilisateur) peut être contrôlé de deux manières :

- Par un script : un code qui définit les actions des boutons lorsqu'un utilisateur clique dessus.
- Par l'interface graphique de Unity : qui permet de gérer l'affichage de manière plus visuelle et intuitive.

Dans notre cas, nous avons utilisé un mélange des deux méthodes, ce qui a complexifié la gestion de l’affichage. Ce mélange est dû au fait que nous avons suivi différents tutoriels avant de bien comprendre le fonctionnement de l’UI sur Unity, ce qui a conduit à une implémentation hybride difficile à manipuler.

Pour corriger ce problème, une refonte partielle du système de menus sera nécessaire. Cela permettra d’uniformiser la gestion de l’UI et d’améliorer la fluidité de navigation entre les différentes interfaces du jeu.

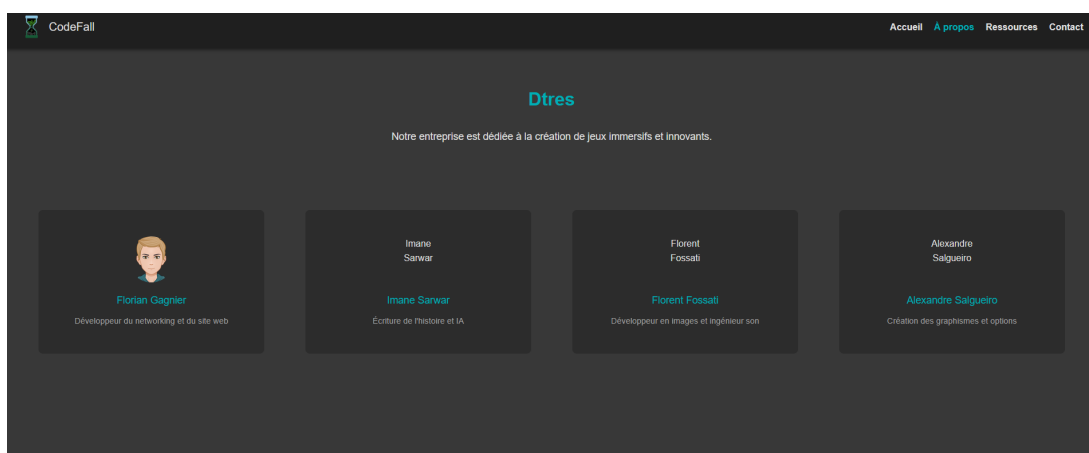
6. Site

Le site a été amélioré et comporte désormais quatre pages principales : “Accueil”, “À propos”, “Ressources” et “Contact”.

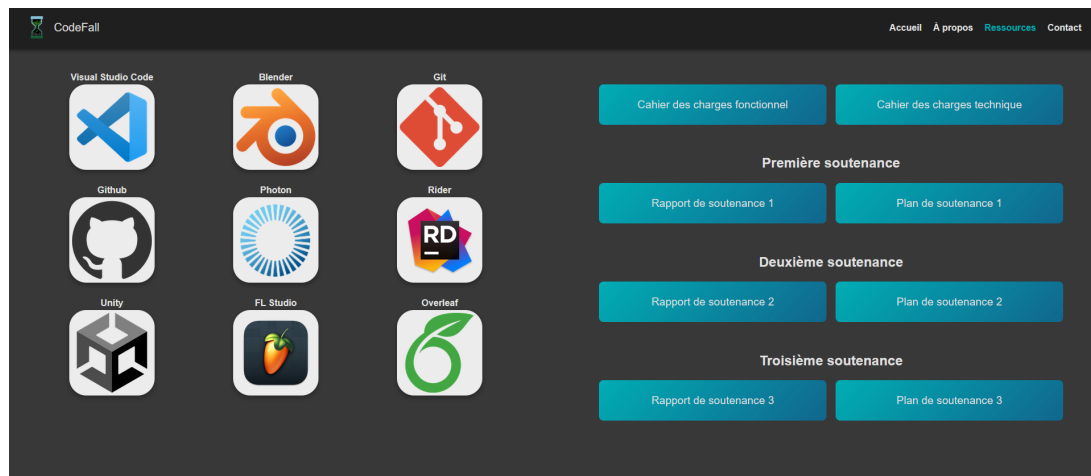
Sur la page “Accueil”, nous avons ajouté un carrousel d’images, dont le contenu sera intégré ultérieurement. Une courte phrase introduisant CodeFall est également affichée, accompagnée d’un bouton de téléchargement, permettant aux utilisateurs d’accéder directement au jeu.



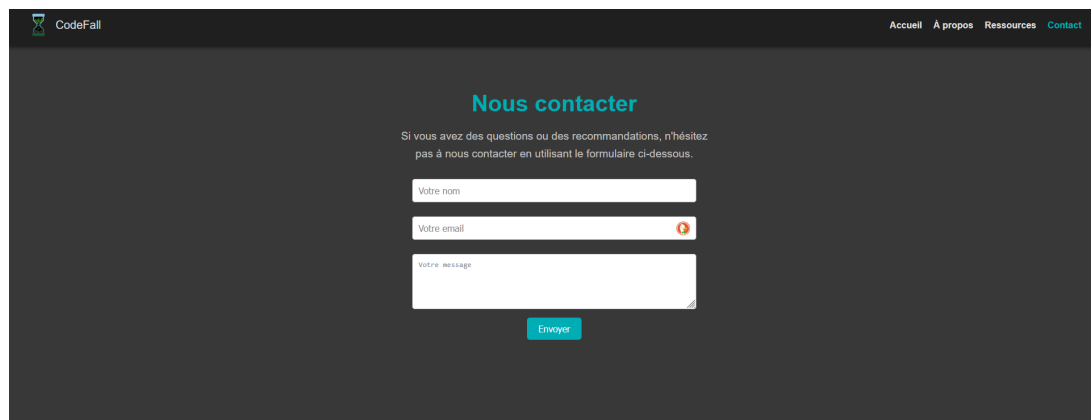
La page “À propos” présente l’équipe de développement de CodeFall, issue du studio Dtres et composée de 4 membres.



La page “Ressources” informe les joueurs sur les outils utilisés pour la création du jeu, tels que Unity ou Blender. Elle propose également un accès aux différents documents liés au projet, comme le cahier des charges, les rapports de soutenances ou encore les plans.



Enfin, une page “Contact” permet aux joueurs de nous signaler d’éventuels bugs, de suggérer de nouvelles fonctionnalités ou simplement de partager leur avis sur le jeu.



III. Bilan sur l'avancement

Tâches	Prévu	Réalisé
histoire et dialogues	70%	70%
Graphismes/Texture	60%	40%
Enigmes et Mécaniques de jeu	70%	75%
Multijoueur	80%	75%
Menu/Options	60%	60%
Site Web	50%	85%
IA	60%	75%
Audio	65%	60%

IV. Développement futur

Tâches	3ème soutenance
histoire et dialogues	100%
Graphismes/Texture	100%
Enigmes et Mécaniques de jeu	100%
Multijoueur	100%
Menu/Options	100%
Site Web	100%
IA	100%
Audio	100%

Pour cette troisième soutenance, notre objectif est de présenter un jeu et un site web entièrement terminés et fonctionnels.

- Améliorations à apporter au site web

- Assurer le bon fonctionnement de tous les boutons.
- Ajouter et afficher les images du jeu ainsi que nos avatars.
- Mettre à jour les informations relatives à l'entreprise et au jeu.
- Finaliser le style visuel pour une présentation plus aboutie.

- Améliorations de l'interface et du gameplay

- Finaliser l'interface du jeu avec un menu ergonomique et agréable.
- Ajouter des options permettant de modifier le volume et la résolution directement dans les paramètres.
- Corriger le bug de la caméra en multijoueur, qui impacte actuellement l'expérience des joueurs.
- Régler le problème de synchronisation des pseudos dans le lobby, afin que chaque joueur voie correctement les informations des autres participants.
- Assurer la synchronisation des objets interactifs, comme la porte et les caisses, pour garantir un bon fonctionnement en multijoueur.

- Améliorations audio

- Intégrer des ambiances sonores dynamiques, variant en fonction du contexte du jeu.
- Ajouter des bruitages liés aux mouvements du joueur (pas, sauts, interactions, etc.).
- Exploiter l'architecture audio déjà en place pour faciliter l'ajout et la gestion des différents sons.

Ces améliorations nous permettront de proposer un jeu plus fluide, immersif et abouti, aussi bien sur le plan technique que visuel et sonore.

Conclusion

Au cours de cette soutenance, nous avons accompli des avancées significatives dans toutes les dimensions du projet, qu'il s'agisse de l'aspect technique, graphique ou fonctionnel. Nous avons progressivement mis en place les mécaniques de jeu essentielles, intégré les éléments graphiques nécessaires et structuré les différentes fonctionnalités permettant d'assurer la cohérence et la fluidité du développement.

Toutefois, un défi majeur subsiste : la gestion de la caméra en mode multijoueur. Ce dysfonctionnement entraîne des incohérences dans les déplacements et l'affichage des joueurs, altérant ainsi l'expérience de jeu. Son correctif constitue une priorité, car il conditionne la viabilité du mode multijoueur.

Néanmoins, en dehors de cette difficulté, nous disposons désormais d'une architecture technique stable et d'un cadre fonctionnel bien défini. Autrement dit, les fondations du projet sont désormais en place, ce qui nous permet d'envisager les prochaines étapes du développement avec plus de sérénité et d'efficacité. L'essentiel du travail à venir consistera en l'optimisation des systèmes existants, l'enrichissement du contenu et l'amélioration de l'expérience utilisateur.

Ainsi, bien que certains défis techniques persistent, nous restons confiants quant à la finalisation du projet dans les délais impartis.