



2023 / 2024

16 mai 2024

CIMENT DOUX

Master 1 - IMAGINE

SERVA Benjamin - JEAN Louis - COMBOT Evan

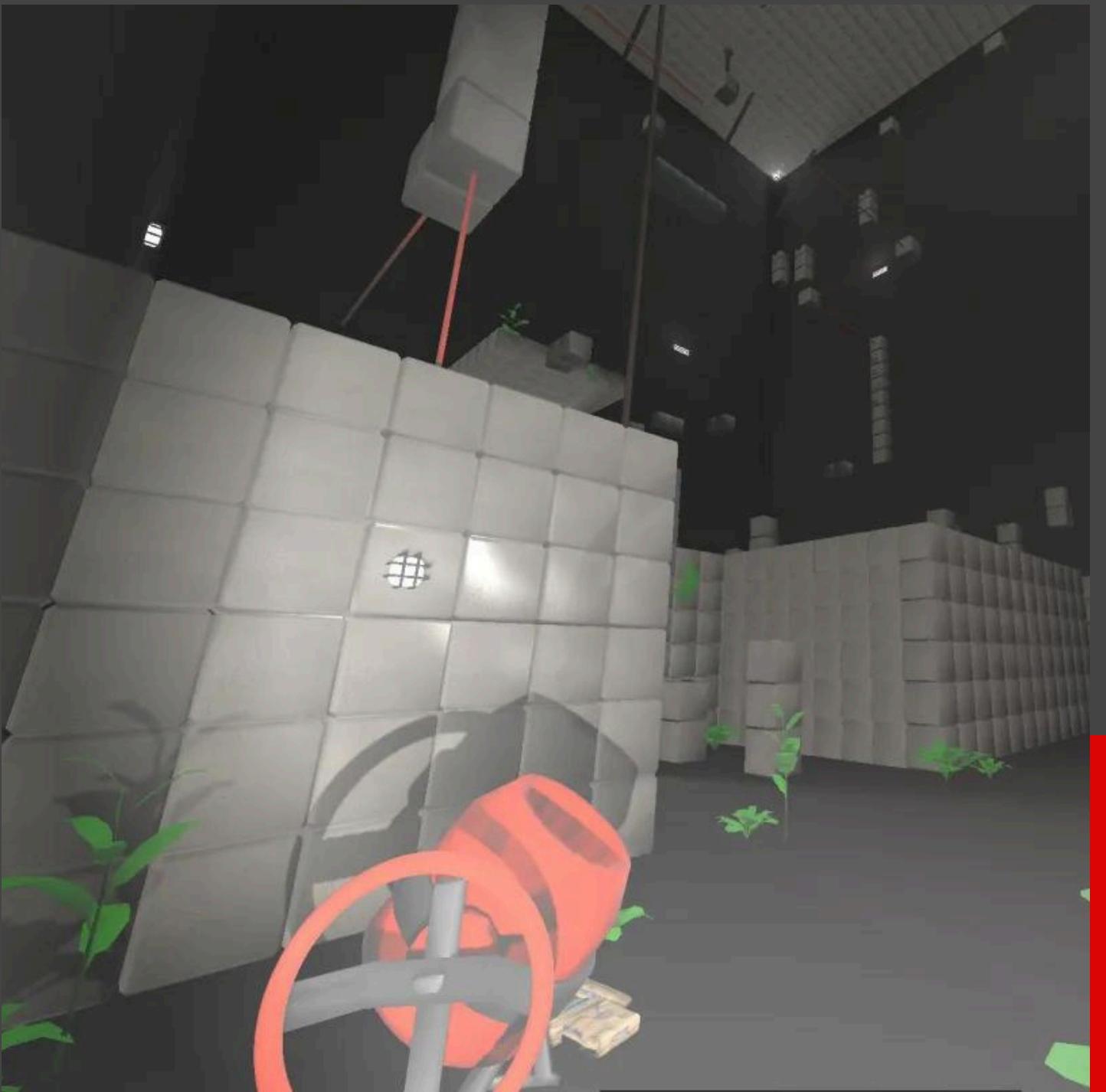


Table des matières

01	02	03	04
Introduction	Architecture du moteur	Moteur physique	Moteur de rendu
05	06	06	07
Notre jeu	Level design, UI & Audio	Démonstration	Conclusion
08			
	Références		



Introduction

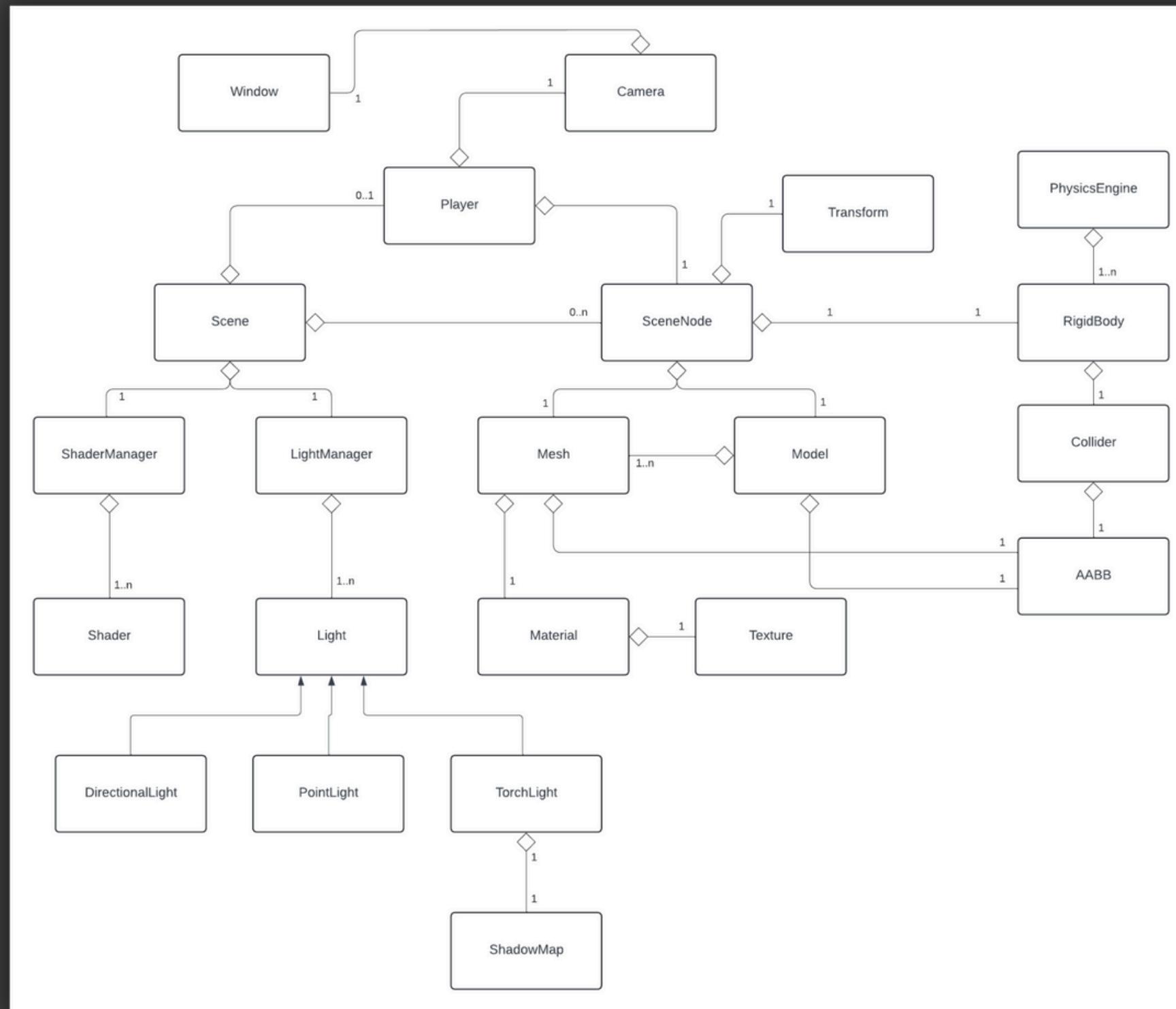
Qu'est-ce qu'un moteur de jeu ?



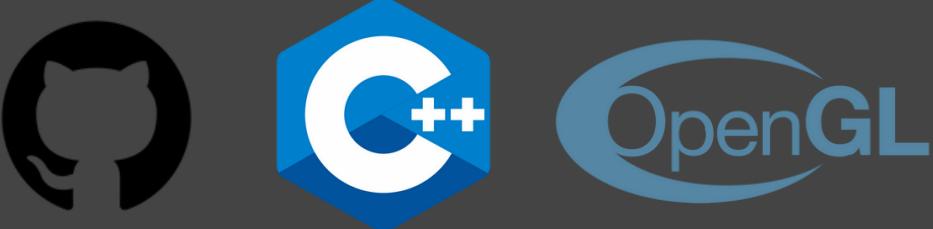
Les objectifs :

- Réaliser un moteur de jeu maison
- Création d'un jeu de plateforme pour illustrer le moteur de jeu

Architecture du moteur



Outils utilisés



Librairies utilisées



Statistiques

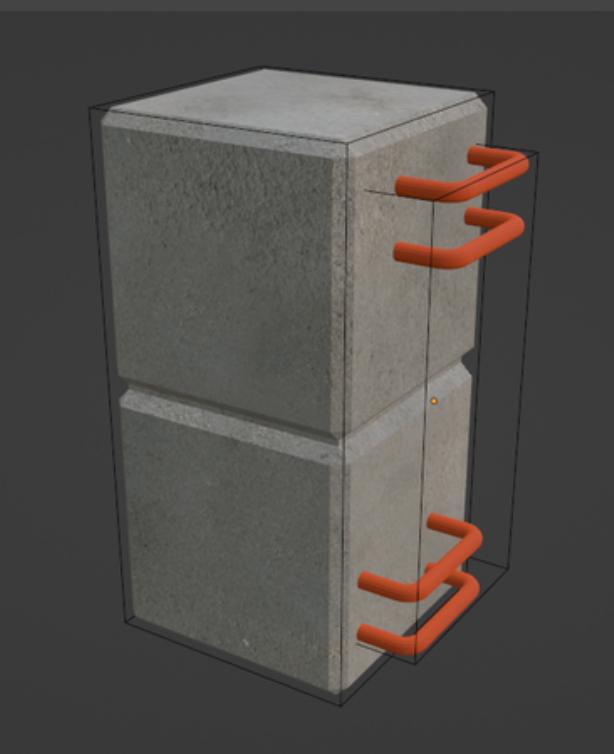
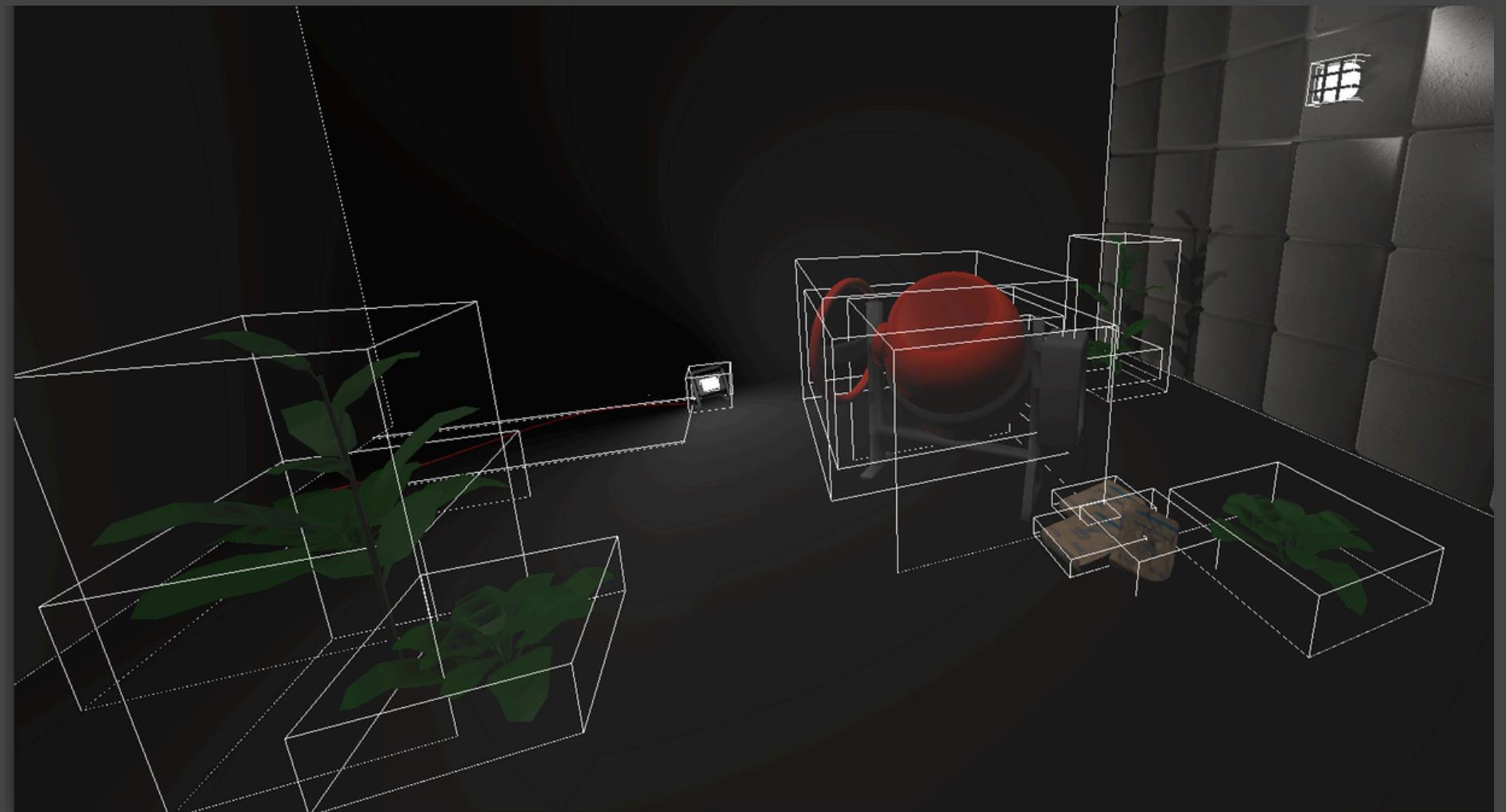
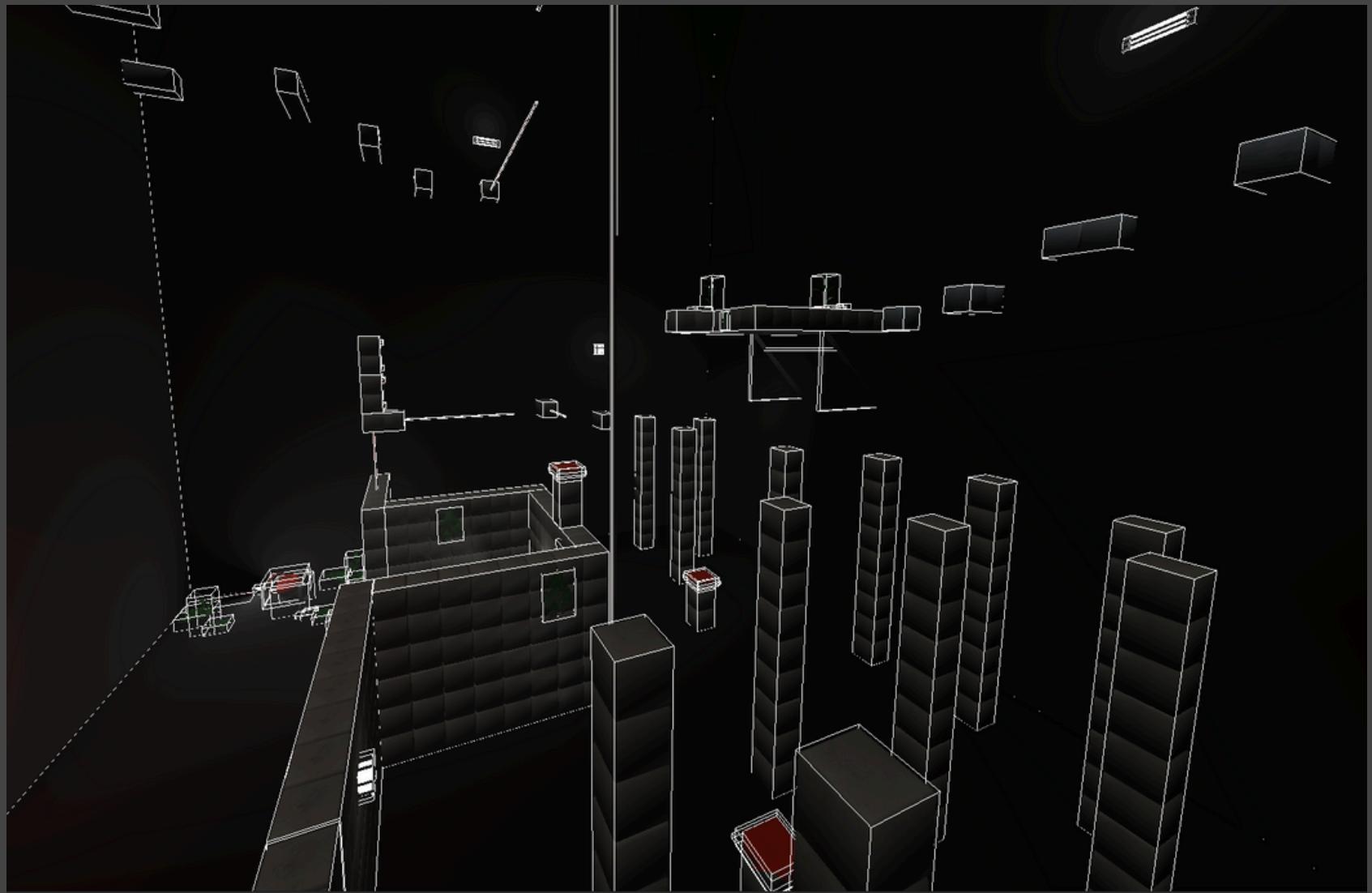
- 57 fichiers écrits
- 26 classes
- 6 shaders
- 5381 lignes
- 174 commits

Moteur physique

- **Intégration d'Euler explicite**
- **Chaque objet a une vitesse définie par son accélération au cours du temps**
- **Forces appliquées aux objets**
 - **Gravité, friction**

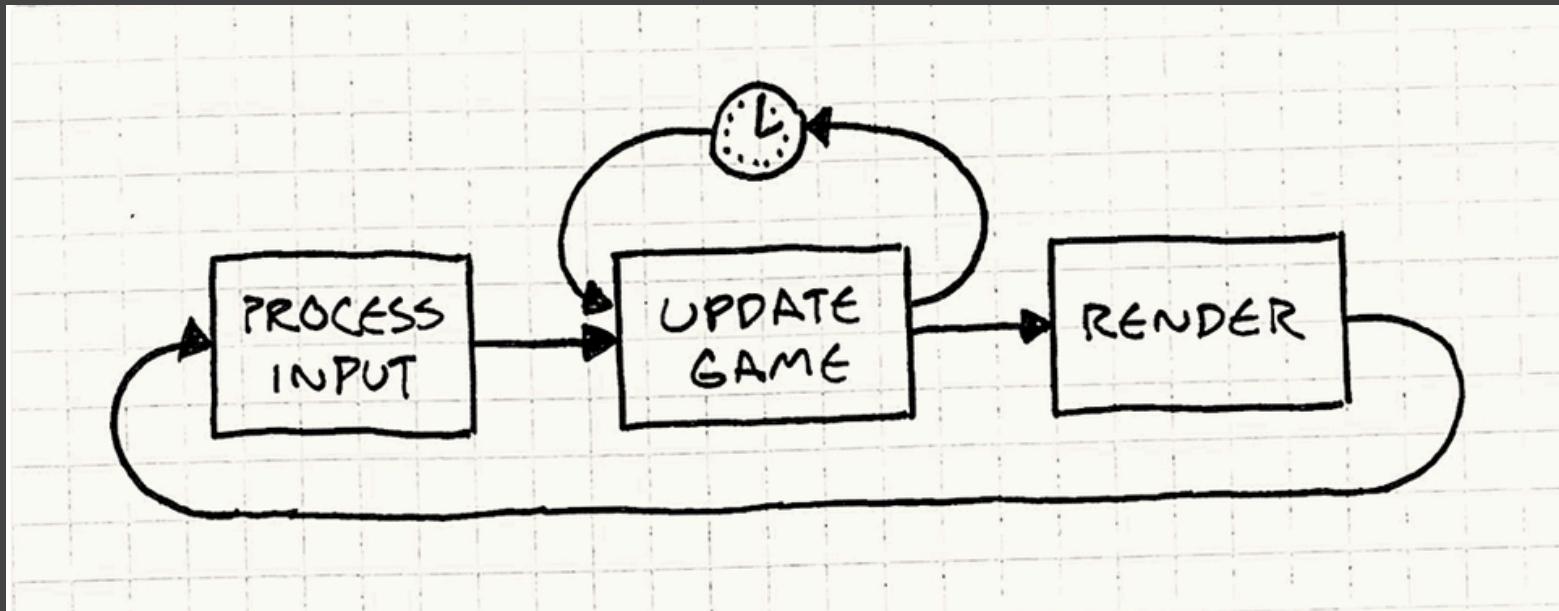
Détection et résolution des collisions

- Axis-Aligned Bounding Boxes (AABB)
- Détection de collision AABB-AABB grâce au Separating Axis Theorem
- Résolution de la collision en calculant la profondeur sur l'axe le moins pénétrant



Gameloop

- “Fixed timestep”
- Permet d'avoir un delta time fixe
- Limiter le nombre d'images loupées



```
double previous = getCurrentTime();
double lag = 0.0;
while (true)
{
    double current = getCurrentTime();
    double elapsed = current - previous;
    previous = current;
    lag += elapsed;

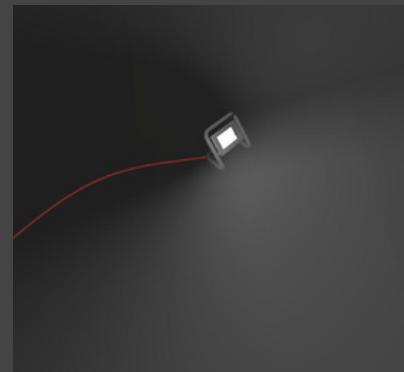
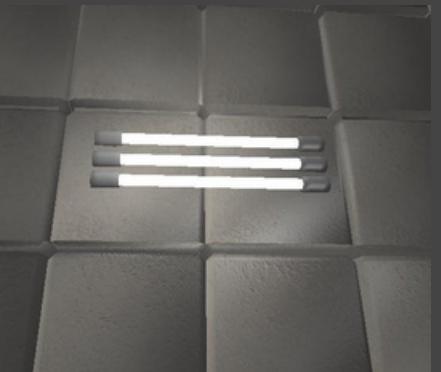
    processInput();

    while (lag >= MS_PER_UPDATE)
    {
        update();
        lag -= MS_PER_UPDATE;
    }

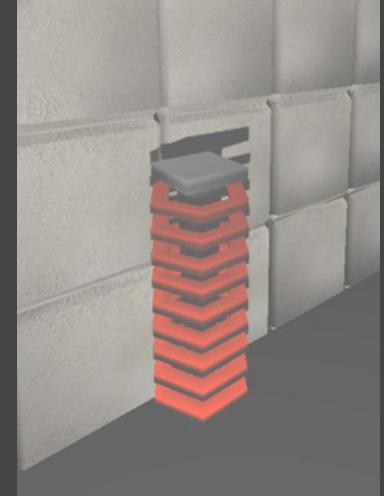
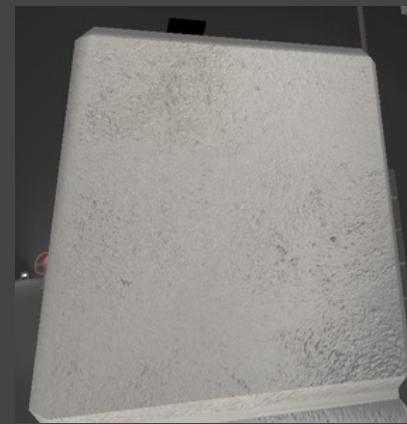
    render();
}
```

Moteur de rendu

- Chargement de maillages avec Assimp
- Prise en charge des materials et des textures
- Modèle de Phong
- Trois types de lumières



- Support des normal maps
- Shadow mapping



Inspiration



Notre jeu

Crée par **Jan Malitschek**

Sorti le **31 mars 2023**

Jeu de parkour à la première personne



Types de collisions

Rebond



Classique



Echelle



Glissade



Player



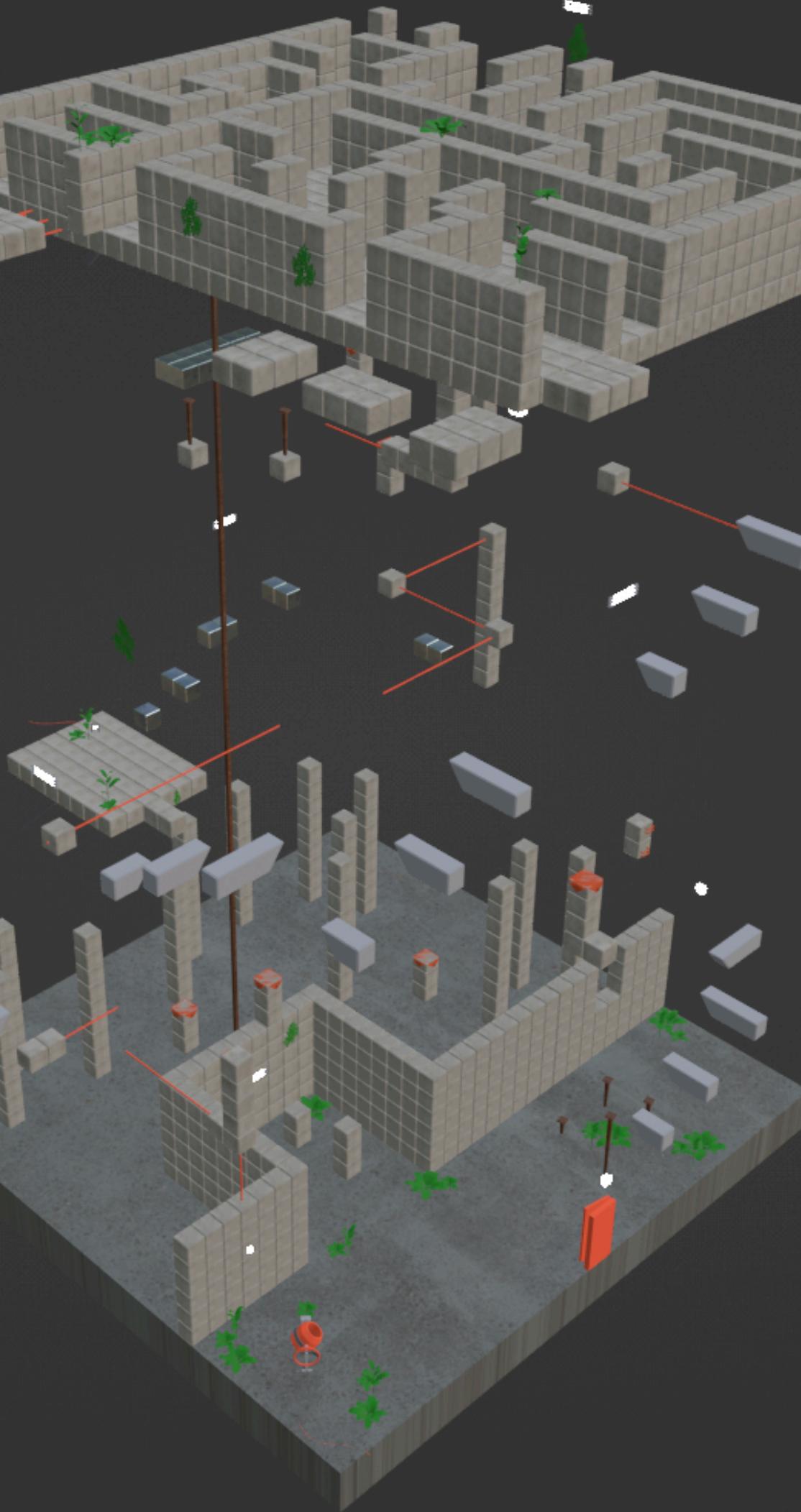
Level design, UI & Audio

Aidé par Jonas RONTEIX pour la
création des différentes plateformes

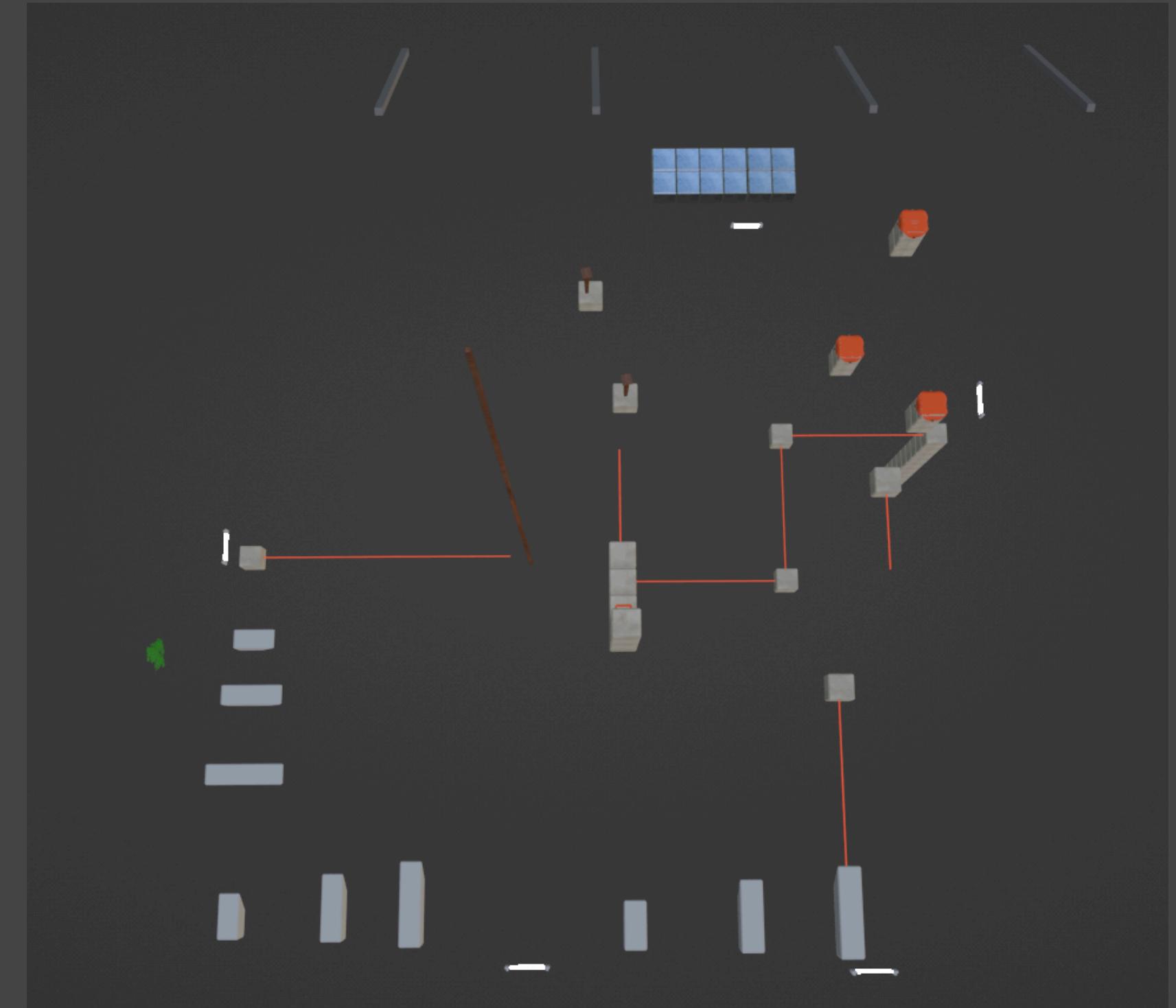
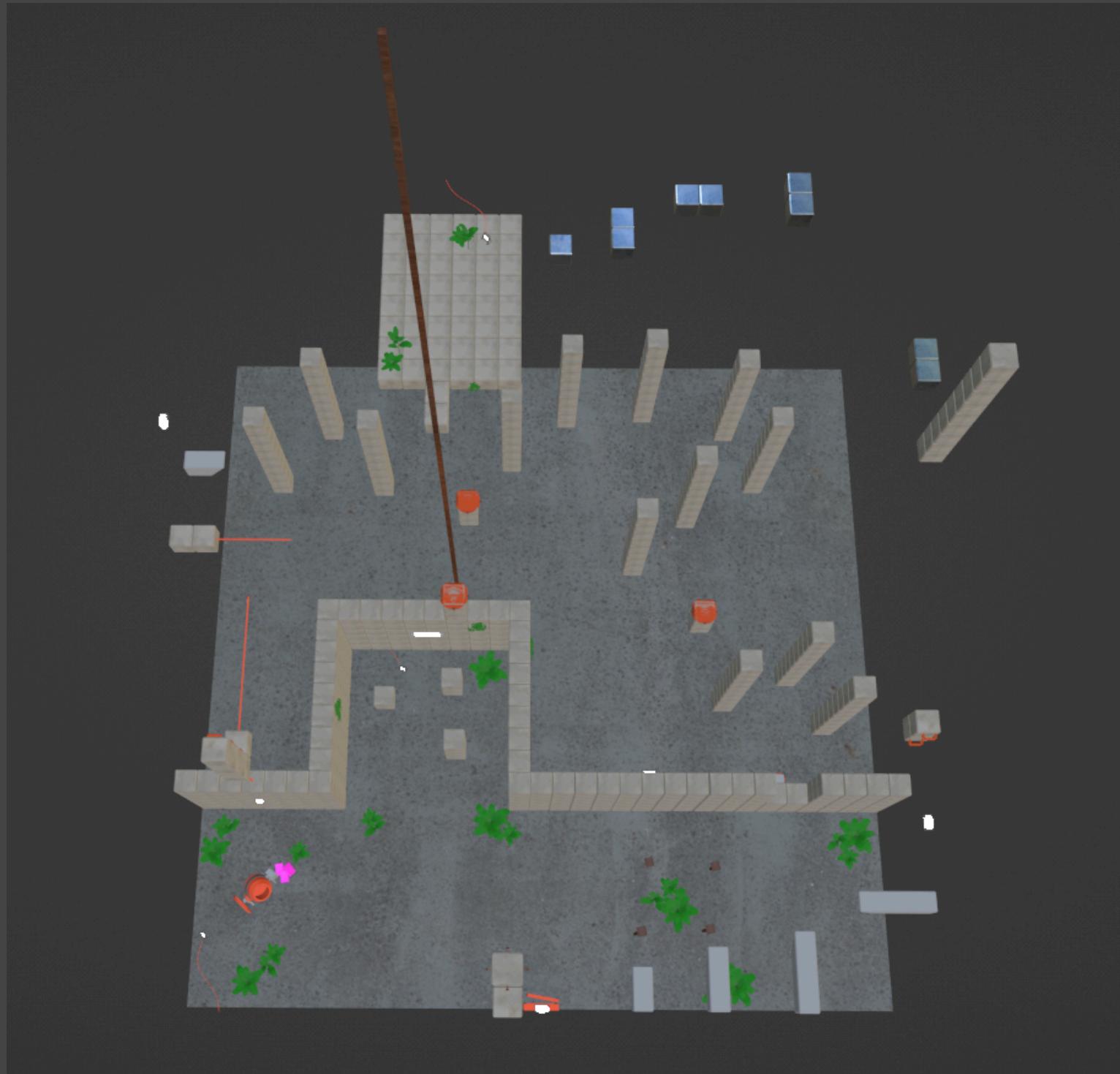


Utilisation de DearImGui

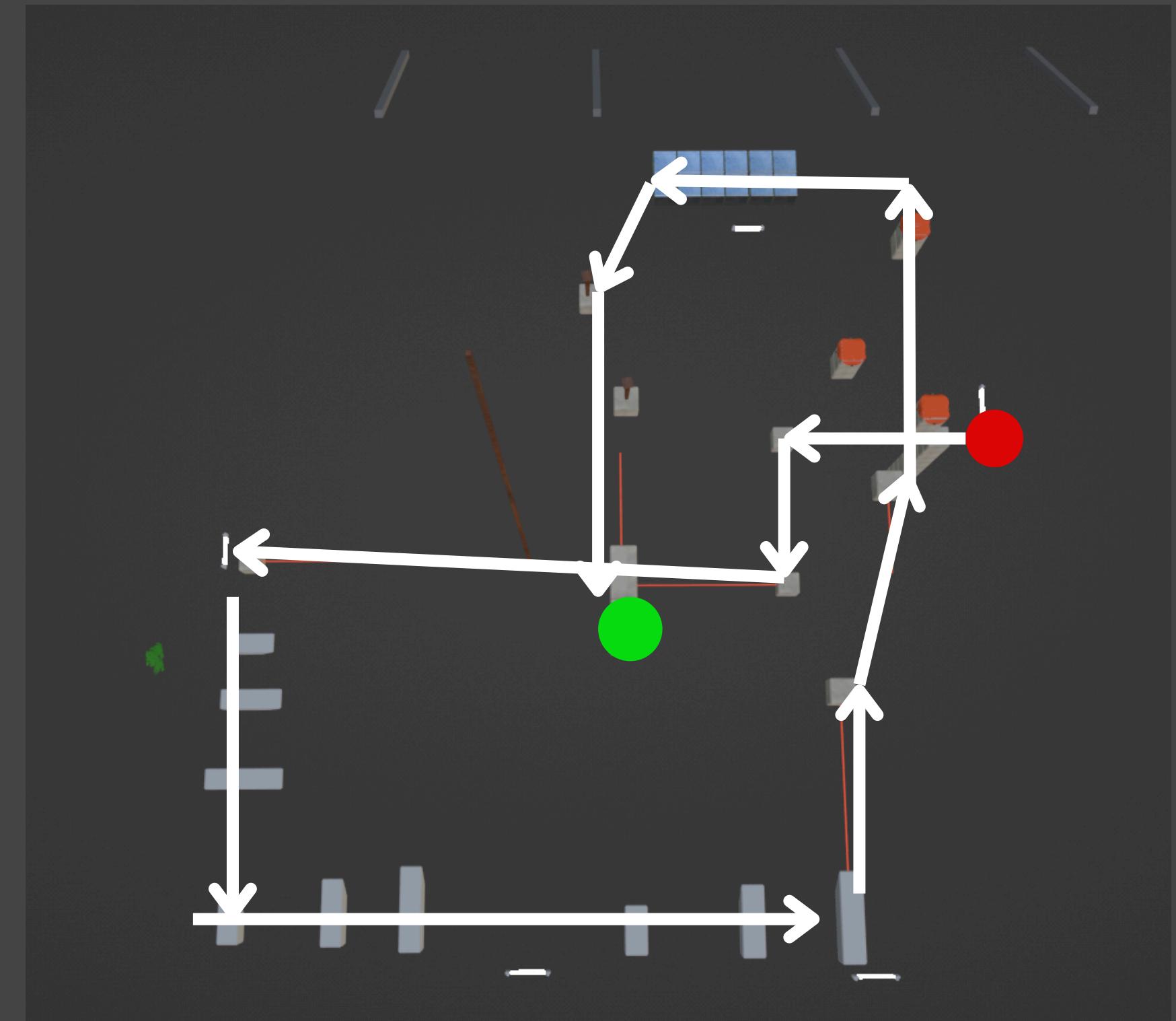
Utilisation de MiniAudio



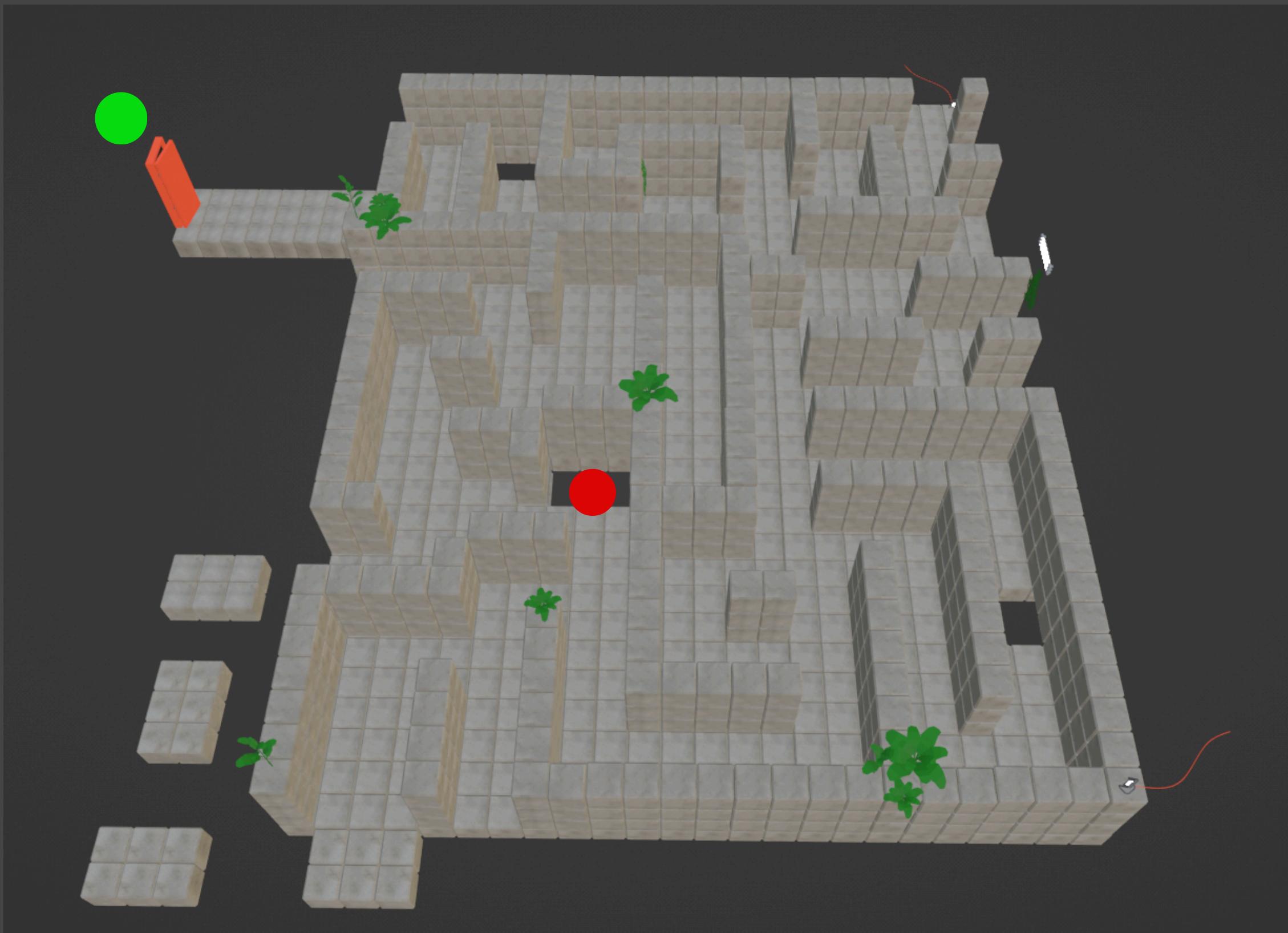
Map



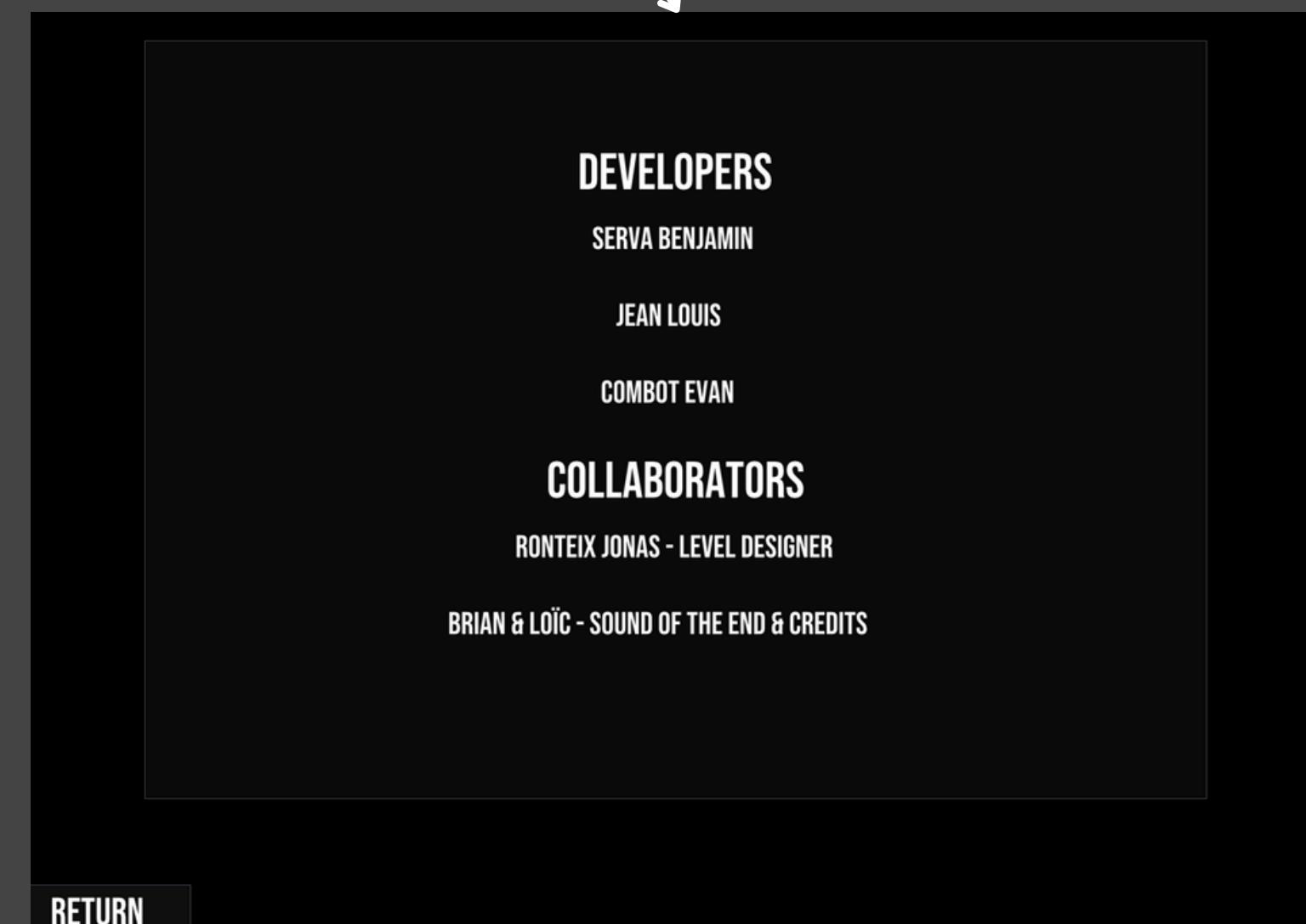
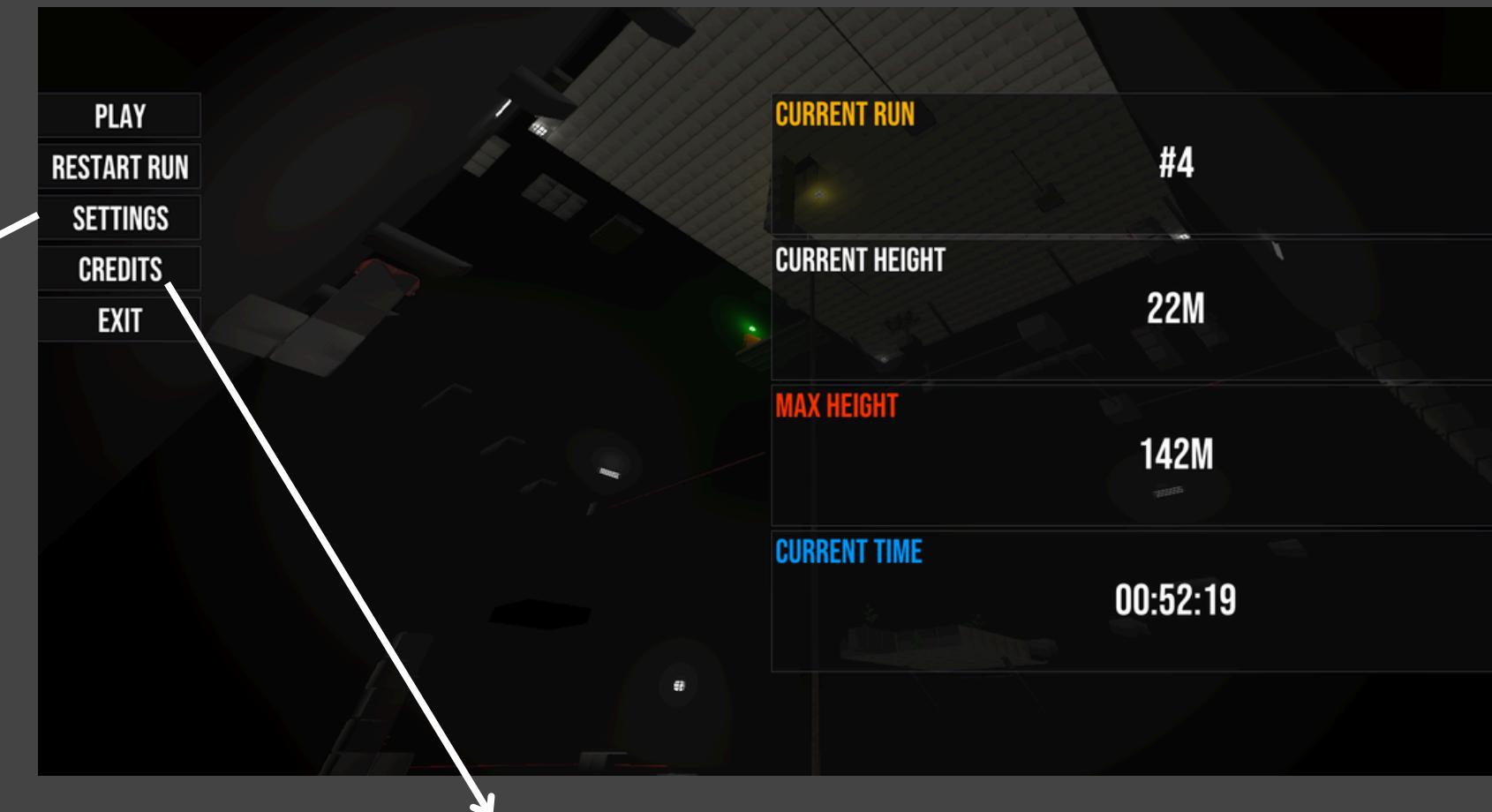
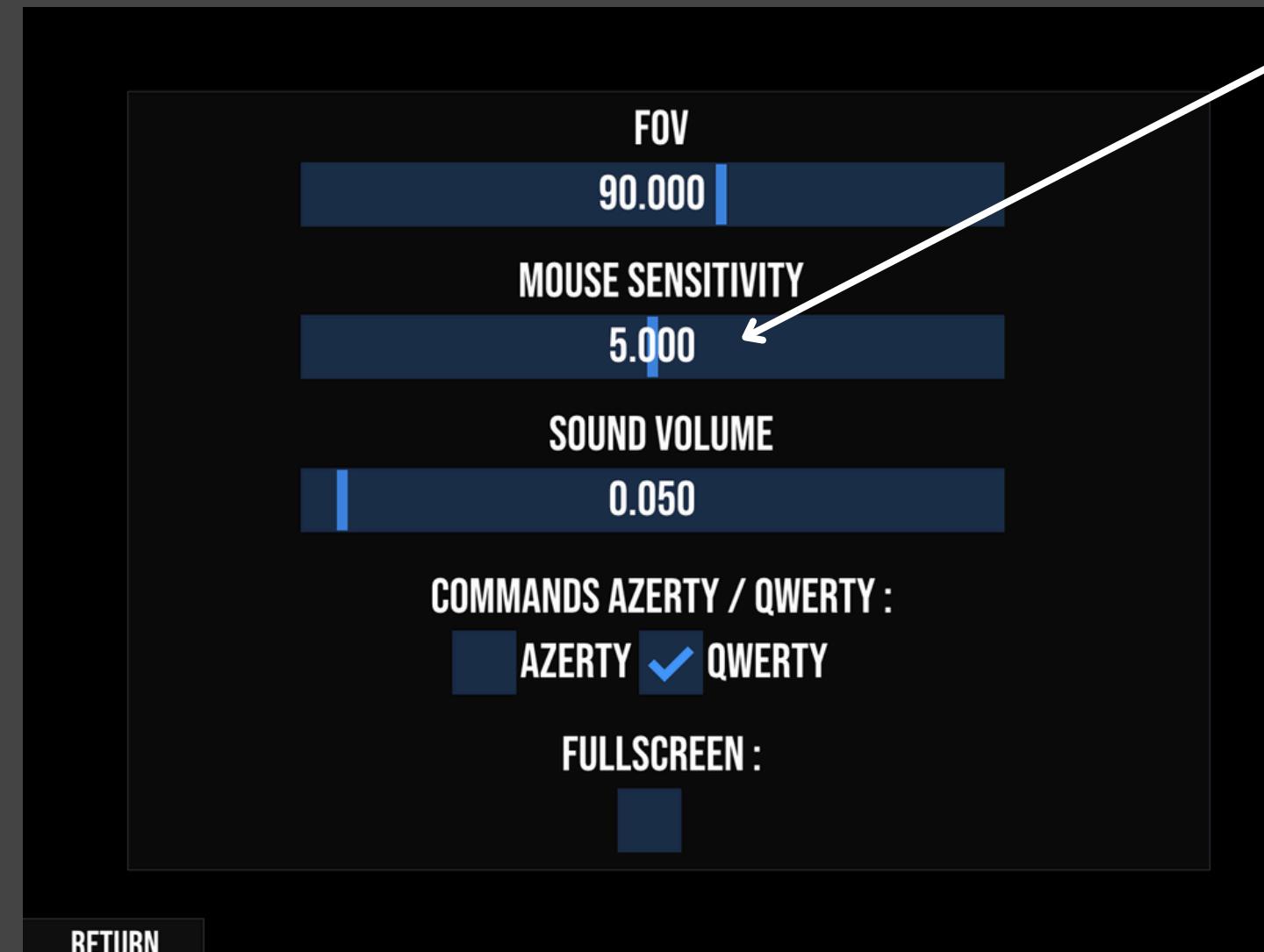
Map



Map



Menu



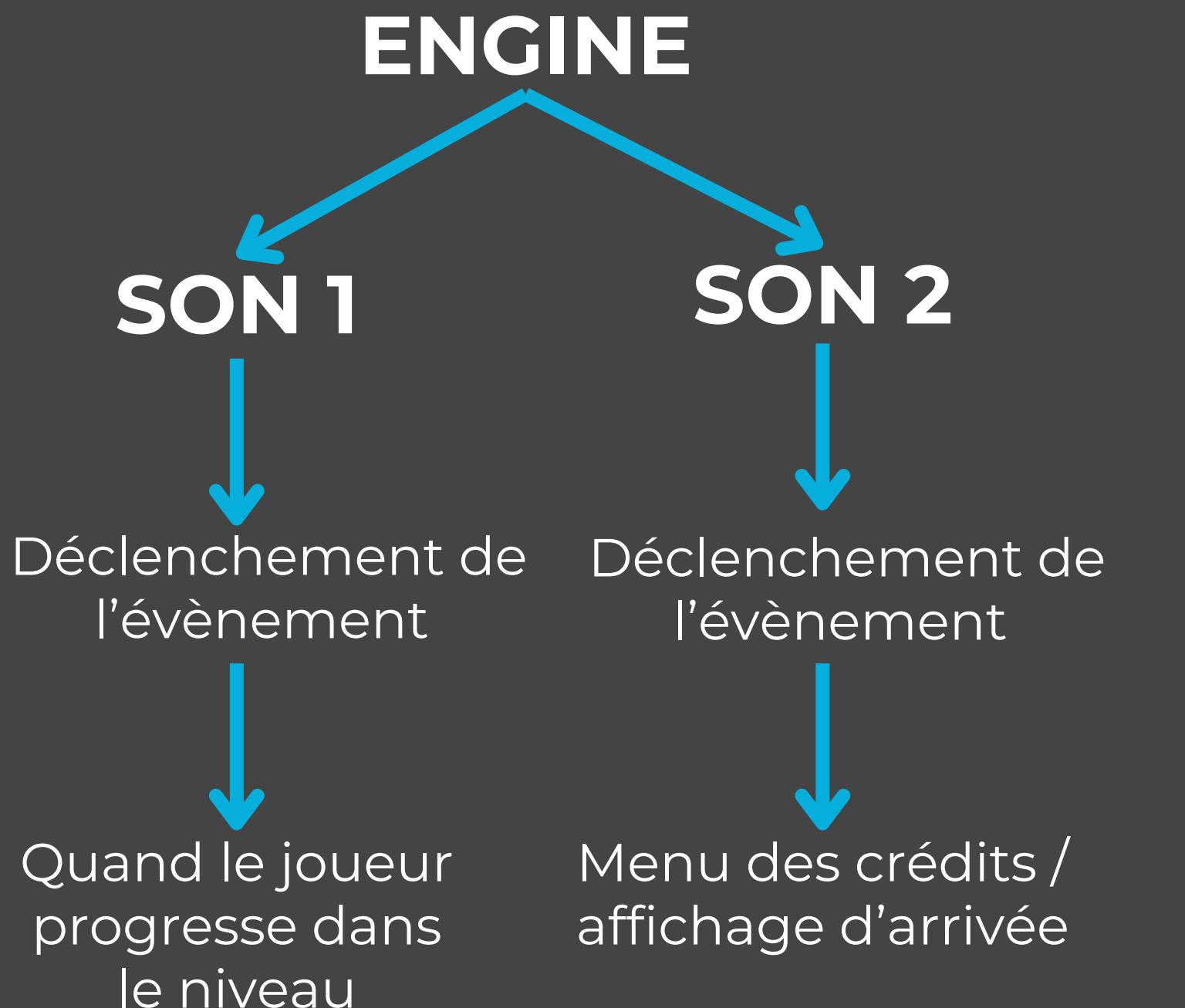
Menu



00:00:09

OM

Audio



Problèmes rencontrés

DéTECTé avec Valgrind



Fuites Mémoires



Réécriture du code à l'aide de smart pointers

Incohérences sur les déplacements / sauts



Utilisation du fixed timestep

- Cause
- Solution
- Erreur

Shake sur les échelles

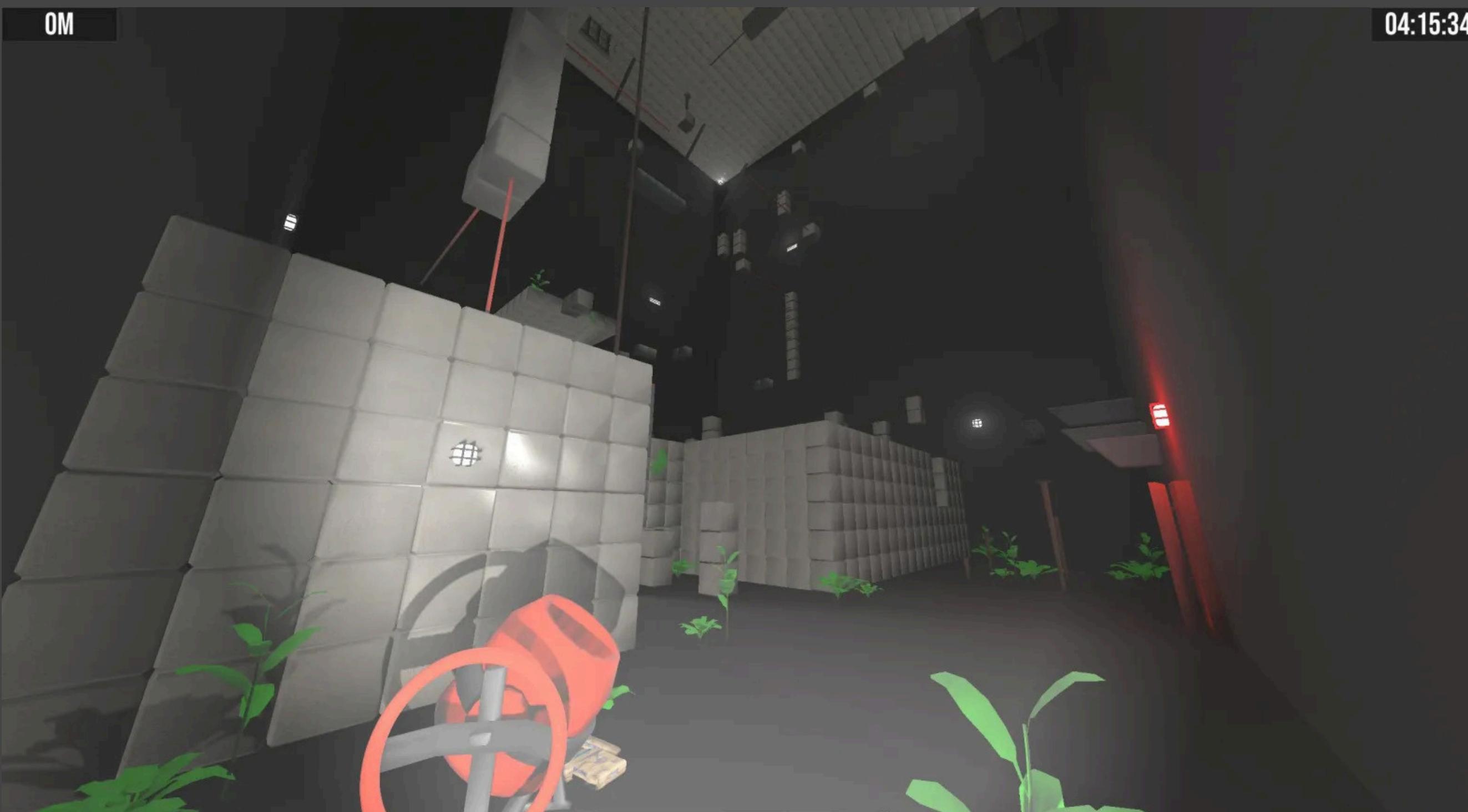


Incohérences sur la détection des collisions des échelles



Jointures entre les mesh pour ne créer qu'une seule AABB

Démonstration



Conclusion



Amélioration possibles :

- Shake lors de la collision après une grande chute
- Possibilité d'accéder à une caméra libre
- Amélioration de la map (en hauteur)
- Amélioration de l'ambiance (éclairage, ombre)

Objectifs atteints ?

Merci pour votre attention !

Avez-vous des questions ?

Remerciements

FARAJ Noura

RONTEIX Jonas

Brian DELVIGNE & Loïc KERBAUL

Références

- . <https://github.com/assimp/assimp> - **Assimp**
- . <https://github.com/g-truc/glm> - **GLM**
- . <https://www.glfw.org/> - **GLFW**
- . <https://glad.dav1d.de/> - **GLAD**
- . <https://github.com/ocornut/imgui> - **DearImGui**
- . <https://www.youtube.com/watch?v=1OCBBt-ZIww> - **Beton Brutal Theme 3**
- . <https://suno.com/> - **Musique**