

# Raycasting

Une bibliothèque C++ pour générer et explorer des niveaux 3D,  
basée sur un moteur de raycasting

**Amine Maazizi**

Année 2024-2025

## But du Projet

Ce projet consiste à :

- Le développement d'une bibliothèque d'objets permettant de décrire les éléments composant un niveau de jeu, avec une architecture modulaire et extensible.
- Le développement d'un moteur de raycasting basé sur un paradigme orienté objet.

Pour tester l'intégration de ces fonctionnalités :

- Un environnement permettant d'afficher les résultats du raycasting.
- Un éditeur de niveaux intuitif, facilitant la création et la configuration des scènes par l'utilisateur.
- L'implémentation d'une génération procédurale pour fournir une méthode simple et efficace de création automatique de niveaux variés.
- Une démo interactive inspirée de *Wolfenstein 3D*, permettant de tester et mettre en valeur les performances et fonctionnalités du moteur.

## Présentation de la Technique Raycasting

### Présentation Théorique de l'Algorithme

Le raycasting est une méthode de rendu graphique qui simule la perspective 3D dans un environnement 2D. Pour chaque colonne de l'écran, un rayon est projeté dans la direction du joueur, calculant les intersections avec les murs de la scène.

Les étapes principales de l'algorithme sont :

1. Calcul de l'angle du rayon basé sur la position et la direction du joueur.
2. Détection de l'intersection la plus proche avec un mur.
3. Calcul de la hauteur de la colonne affichée en fonction de la distance.
4. Application de textures pour un rendu réaliste.

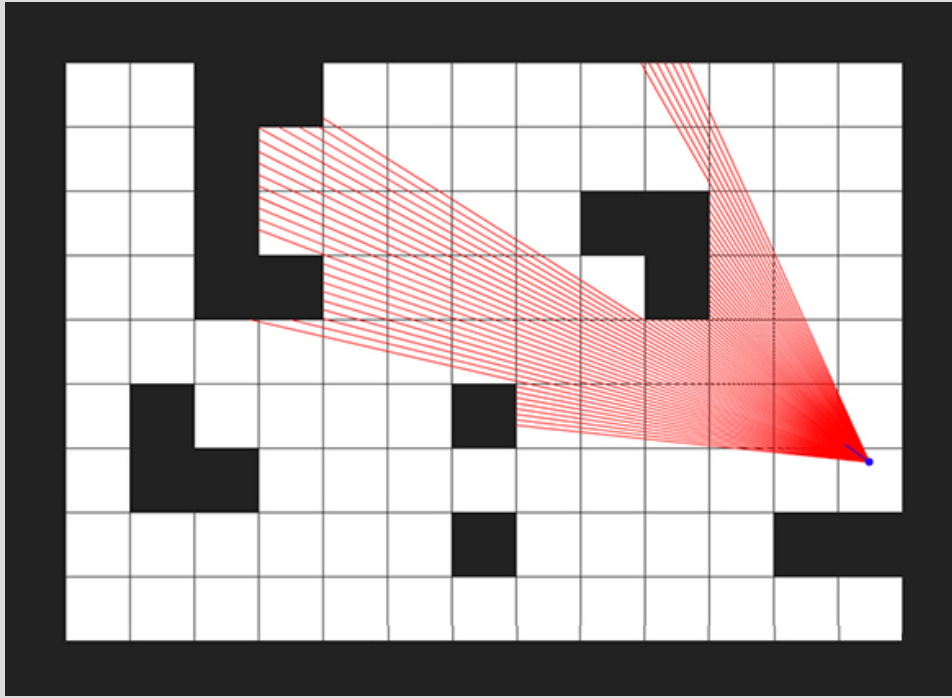


Figure 1: Diagramme des étapes de l'algorithme de raycasting.

## Ressources Complémentaires

Pour approfondir, consultez les ressources suivantes :

- Tutoriel sur le Raycasting (Lodev)
- Raycasting sur Wikipédia

## Quelques Éléments de Réflexion sur les Différentes Étapes

### Phase 1 : Fondation de l'Engin

La première étape du projet consiste à poser les bases du moteur de raycasting. Cette phase vise à établir une architecture solide et modulaire pour le développement futur.

Elle repose sur trois piliers principaux : la définition des objets, le moteur de raycasting lui-même, et un format flexible pour décrire les niveaux.

### Objectifs de cette phase :

- Développement d'une bibliothèque d'objets permettant de décrire les éléments constituant un niveau de jeu. Chaque objet sera défini avec des propriétés géométriques simples, comme des murs, des portes ou des objets interactifs, dans une structure orientée objet modulaire.
- Conception d'un moteur de raycasting pour gérer le calcul des interactions entre les rayons et les éléments de la scène. Ce moteur devra :
  - Détecter efficacement les intersections entre les rayons et les objets.
  - Déterminer les surfaces visibles et leurs propriétés.
  - Restituer les perspectives adaptées en temps réel.
- Mise en place d'un format de description simple et flexible pour configurer les scènes.

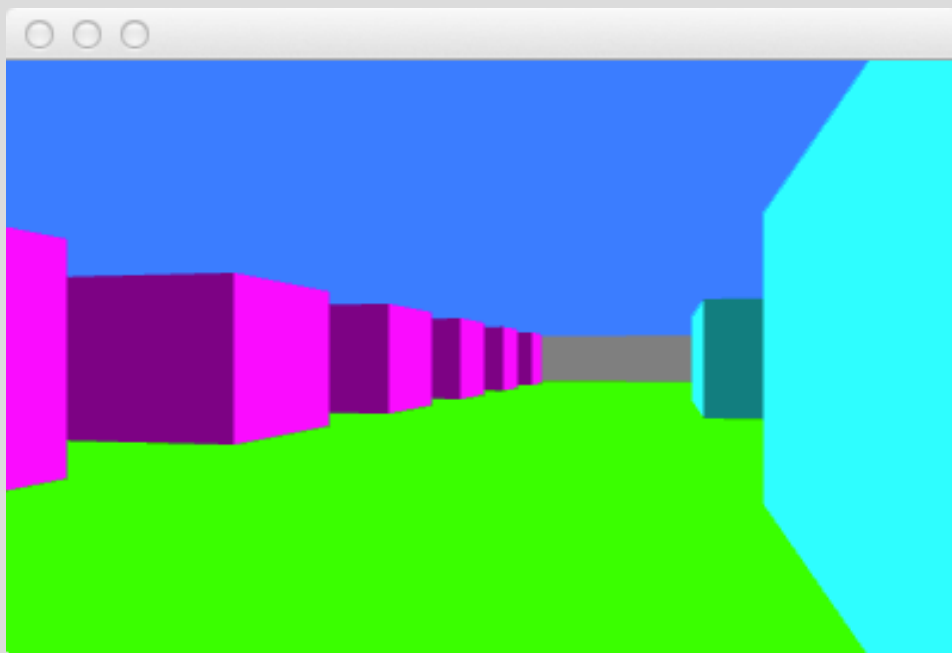


Figure 2: Résultat du Raycasting.

## Phase 2 : Réalisation du Démonstrateur

Une fois les fondations posées, le projet se concentrera sur la réalisation d'un démonstrateur qui mettra en valeur les fonctionnalités du moteur. Cette étape vise à fournir des outils interactifs et à démontrer les capacités du moteur dans des conditions réalistes, notamment à travers un jeu inspiré de *Wolfenstein 3D*.

### Objectifs de cette phase :

- **Développement d'un éditeur de niveaux avec une interface graphique (GUI) :** Cet éditeur offrira aux utilisateurs la possibilité de créer, agencer et modifier des scènes interactives de manière intuitive.
- **Intégration d'une fonctionnalité de génération procédurale :** Grâce à des algorithmes configurables, cette fonctionnalité permettra de créer automatiquement des niveaux variés, réduisant ainsi le travail manuel et augmentant la diversité des environnements.
- **Création d'une démo interactive inspirée de *Wolfenstein 3D* :** Cette démo servira à tester l'intégration des différentes fonctions, tout en illustrant les capacités du moteur à gérer :
  - Des environnements complexes composés d'éléments comme des ennemis, des portes, et des textures dynamiques.
  - Une fluidité de gameplay et une performance optimisée dans des scènes interactives.



Figure 3: Capture de Wolfenstein 3D.

## Phase 3 : Amélioration de l'Outil

La dernière phase du projet est dédiée à l'amélioration des performances et à l'enrichissement des fonctionnalités du moteur. Elle vise à transformer l'outil en une plateforme robuste et polyvalente, capable de répondre aux exigences de jeux ou de simulations complexes.

### Objectifs de cette phase :

- **Optimisation des performances avec le multithreading :** Le moteur sera conçu pour tirer parti des architectures multicœurs, permettant d'accélérer les calculs de raycasting et de gérer efficacement les scènes complexes.
- **Ajout d'un système de particules :** Ce système enrichira les visuels en simulant des effets comme des flammes, de la fumée ou des impacts de projectiles, renforçant ainsi le réalisme des environnements.

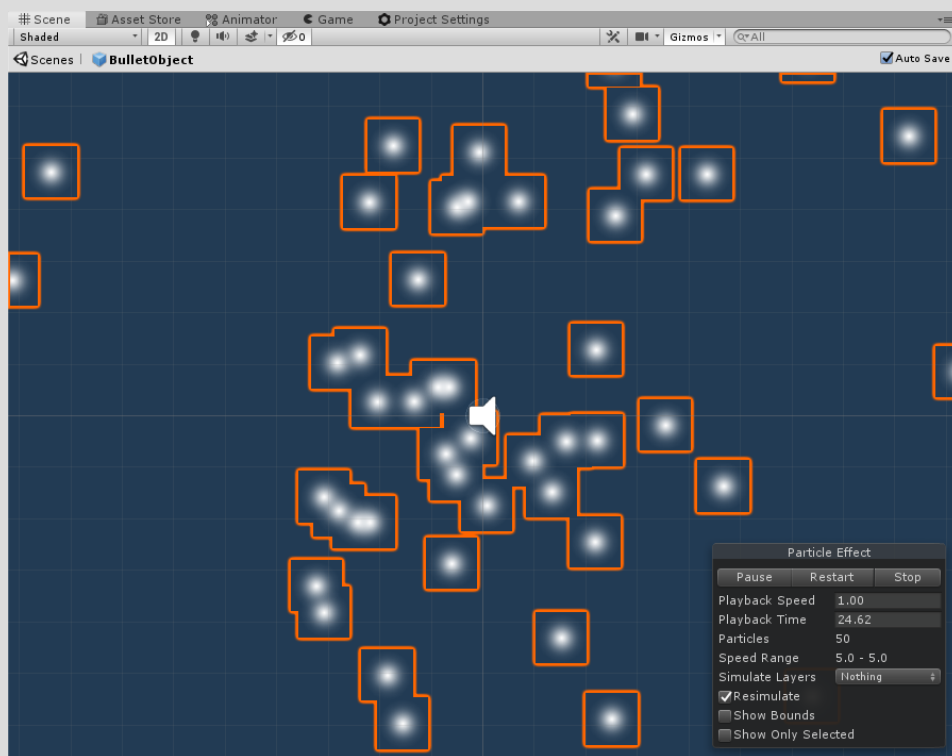


Figure 4: Effets visuels générés par le système de particules.

- **Intégration d'effets sonores :** Des sons spécifiques seront associés à des actions et objets, créant une immersion sonore qui complète l'expérience visuelle.
- **Réalisation de tests de résistance :** Ces tests permettront d'évaluer la capacité du moteur à gérer des niveaux de plus en plus complexes, incluant un grand nombre d'objets, de sources lumineuses et d'interactions dynamiques.