

# Ray Tracing

Leonardo Falabella  
Mateus Galvão



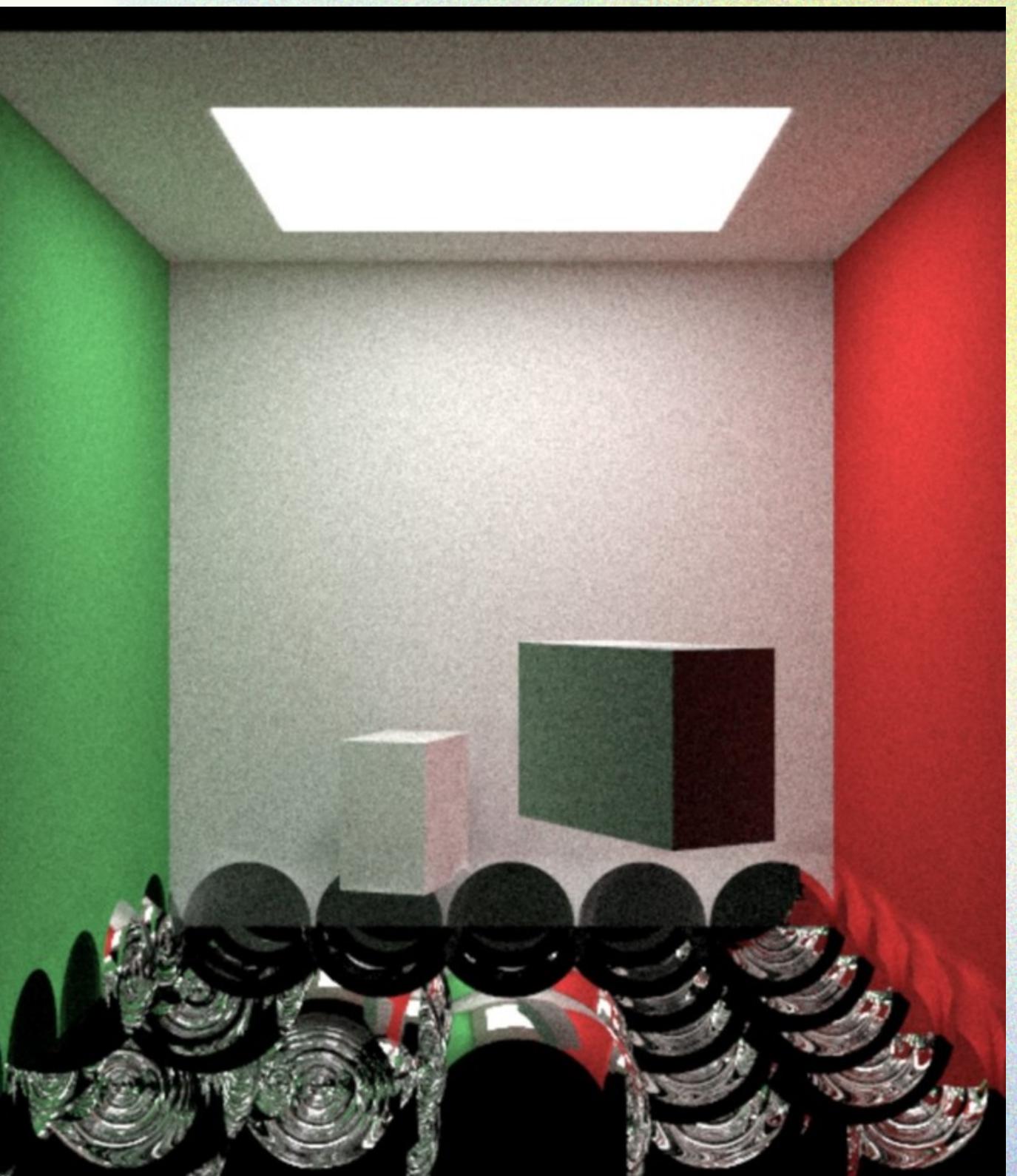
# Sommaire

- Introduction
- Principes du Ray Tracing
- Implementation
- Organization des classes
- Interface
- Classes Interface et affichage
- Améliorations possibles
- Optimization

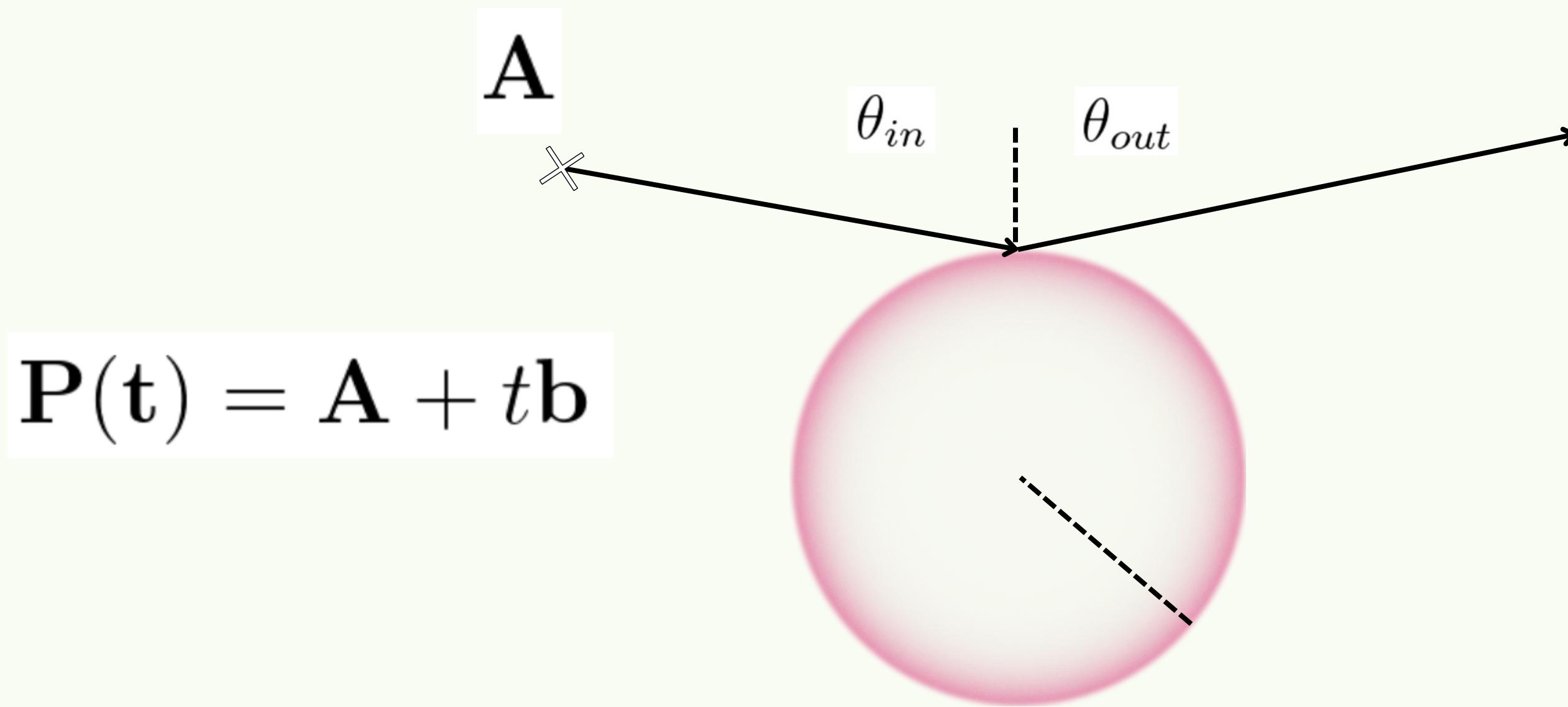
# Introduction

---

- Technique graphique avancée : Simulation réaliste de la lumière.
- Calcul des rayons de lumière pour des effets visuels précis.
- Amélioration des graphismes dans les jeux et la production cinématographique.
- Création d'ombres, de réflexions et de reflets plus réalistes.

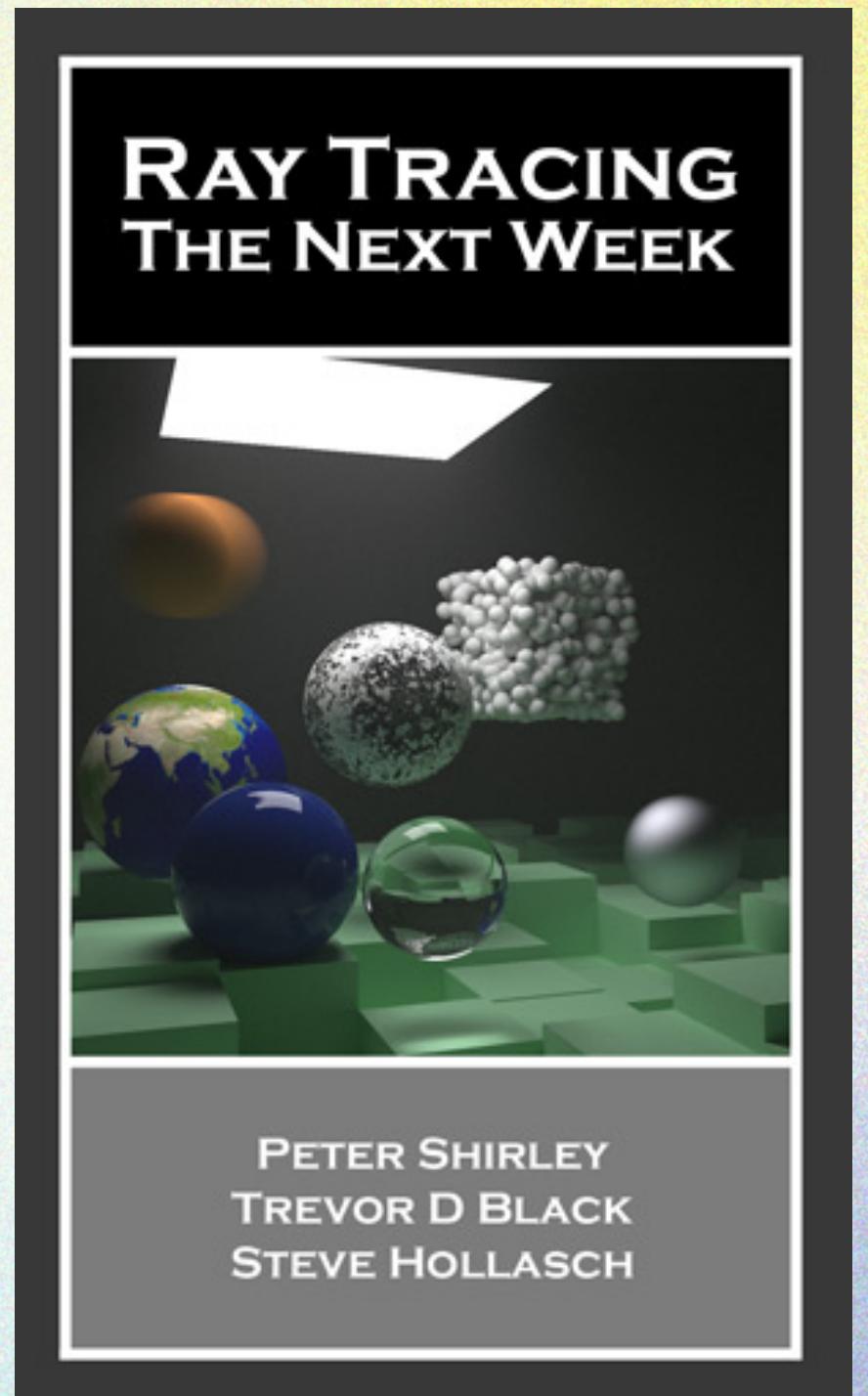


# Principes du Ray Tracing

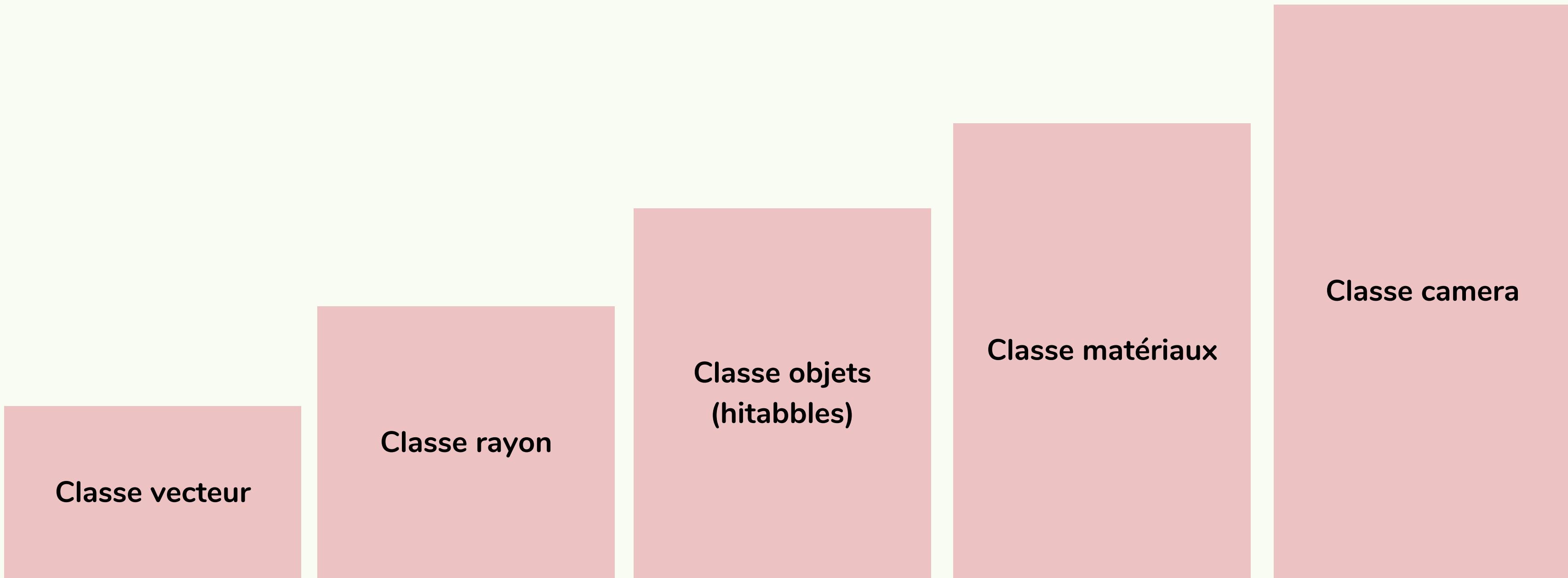


# Mise en œuvre

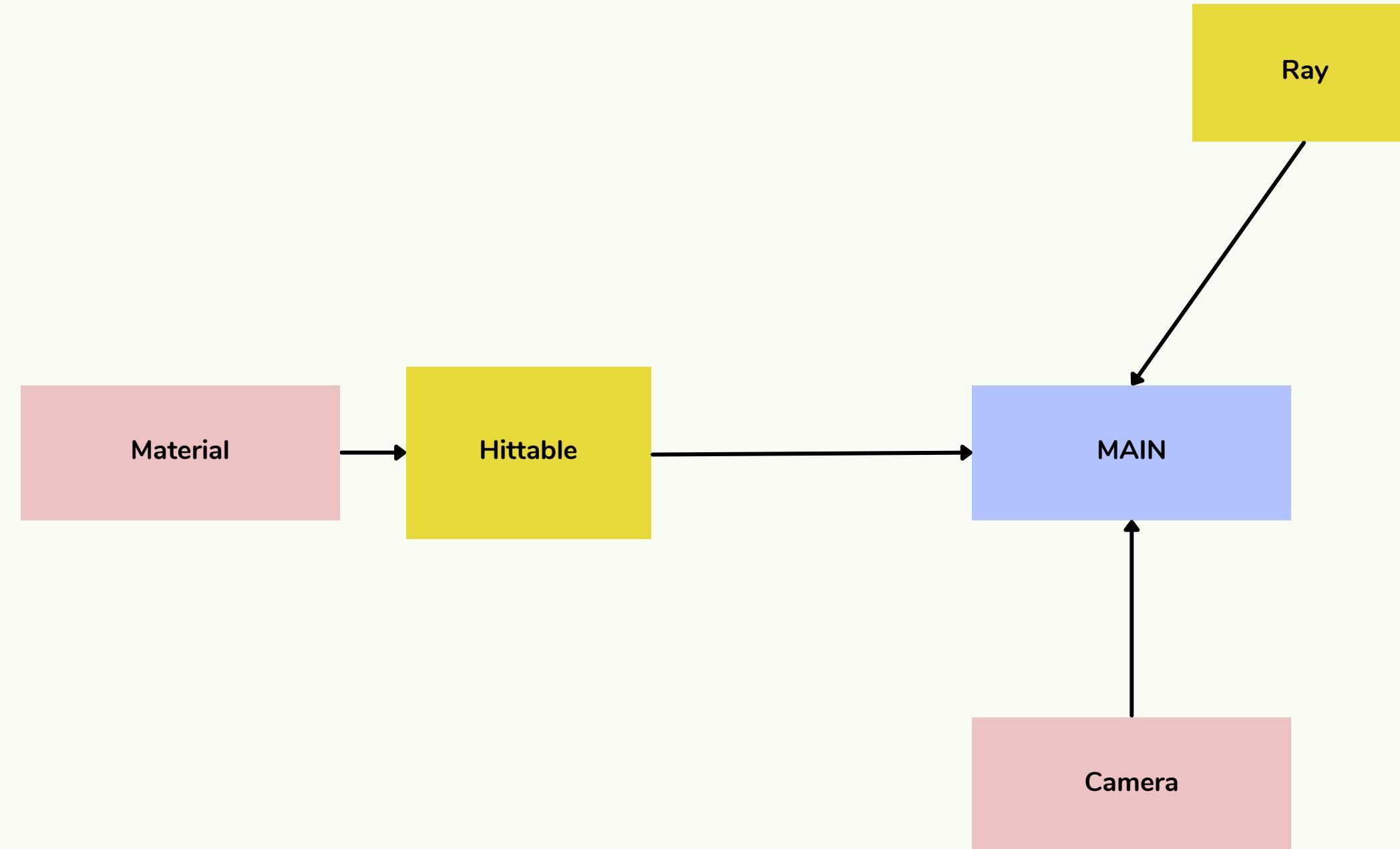
- Didactique
- Utilisation des aspects modernes du C++
- Peu optimisé



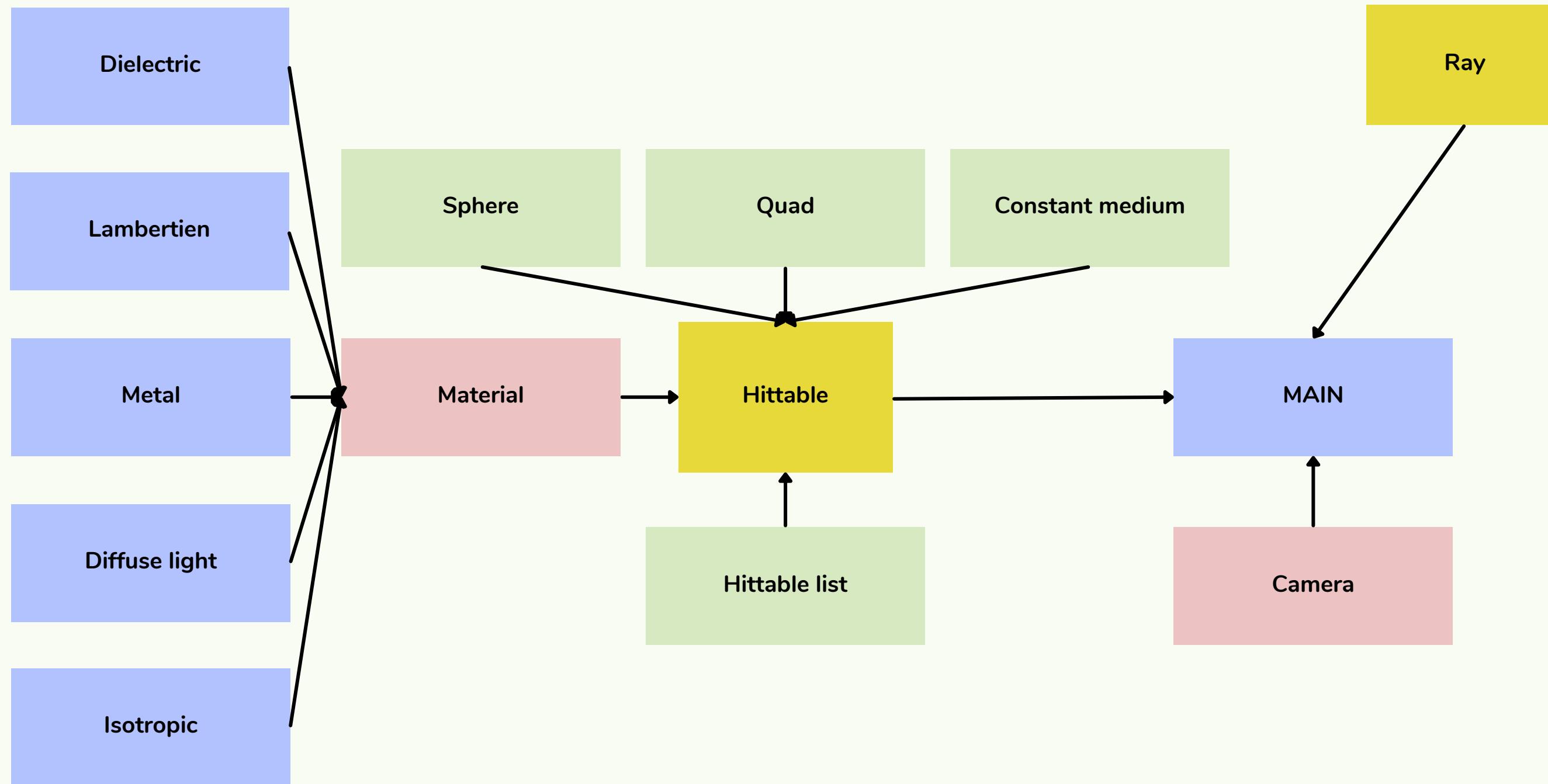
# Mise en œuvre



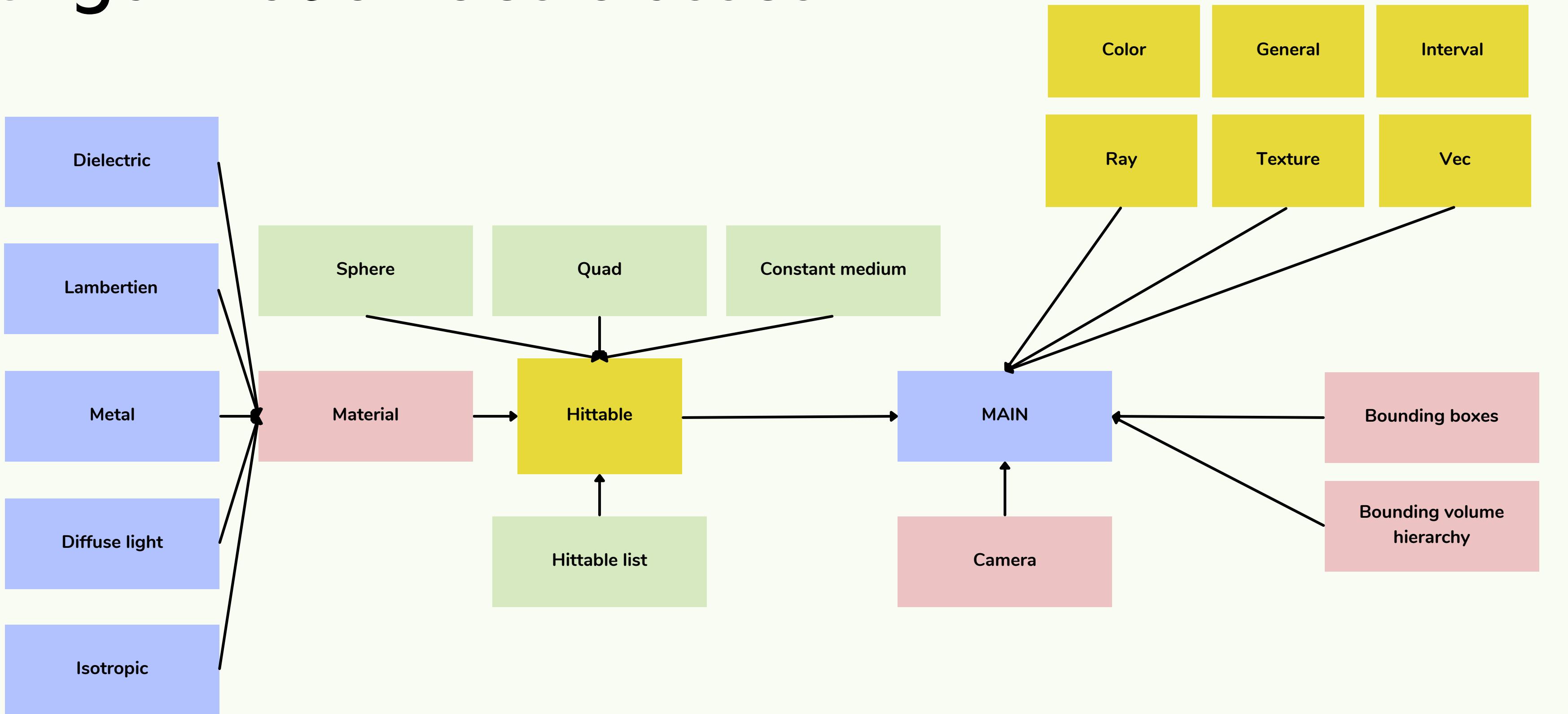
# Organization des classes



# Organization des classes



# Organization des classes



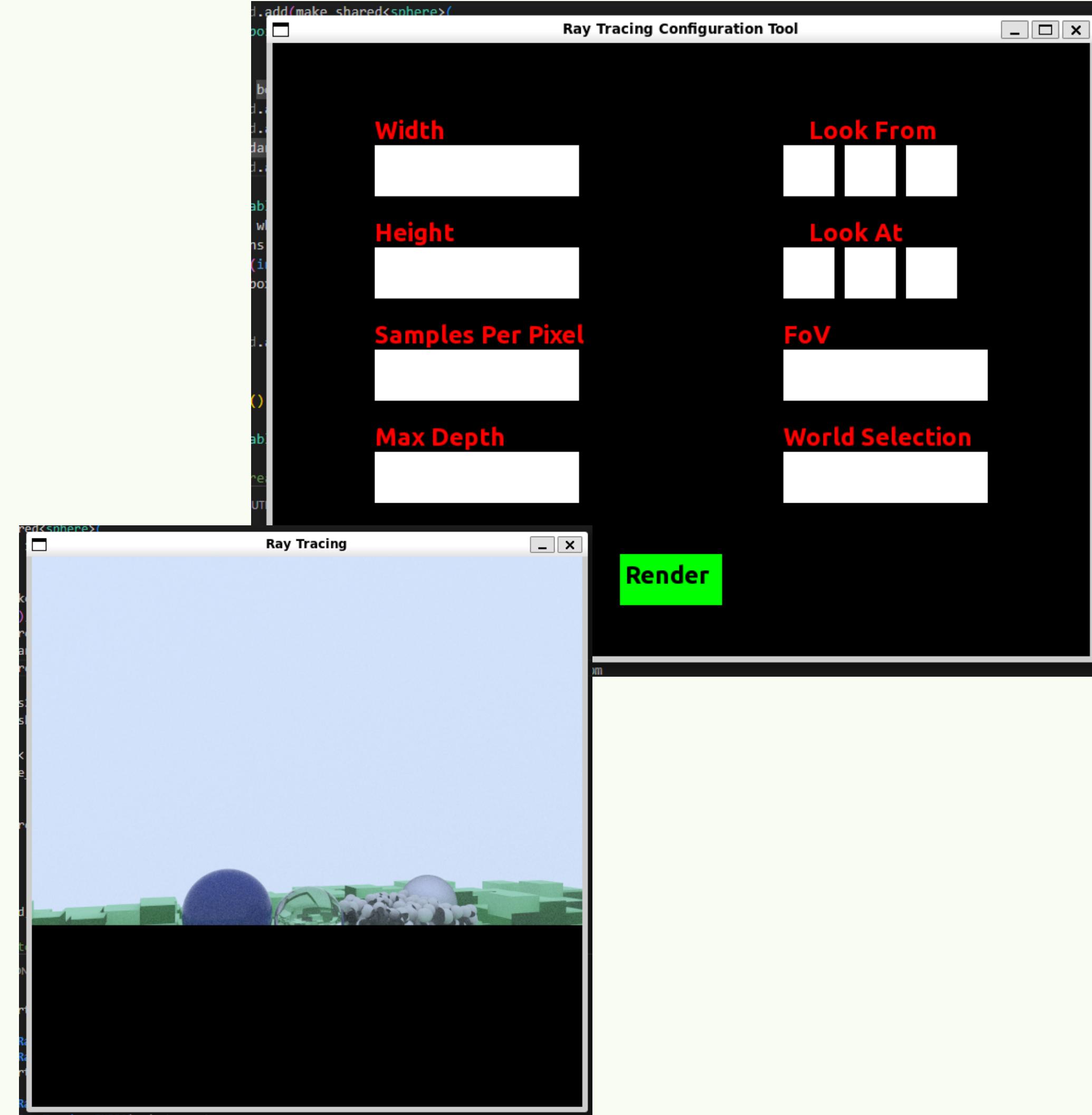
# Interface

## Avantages:

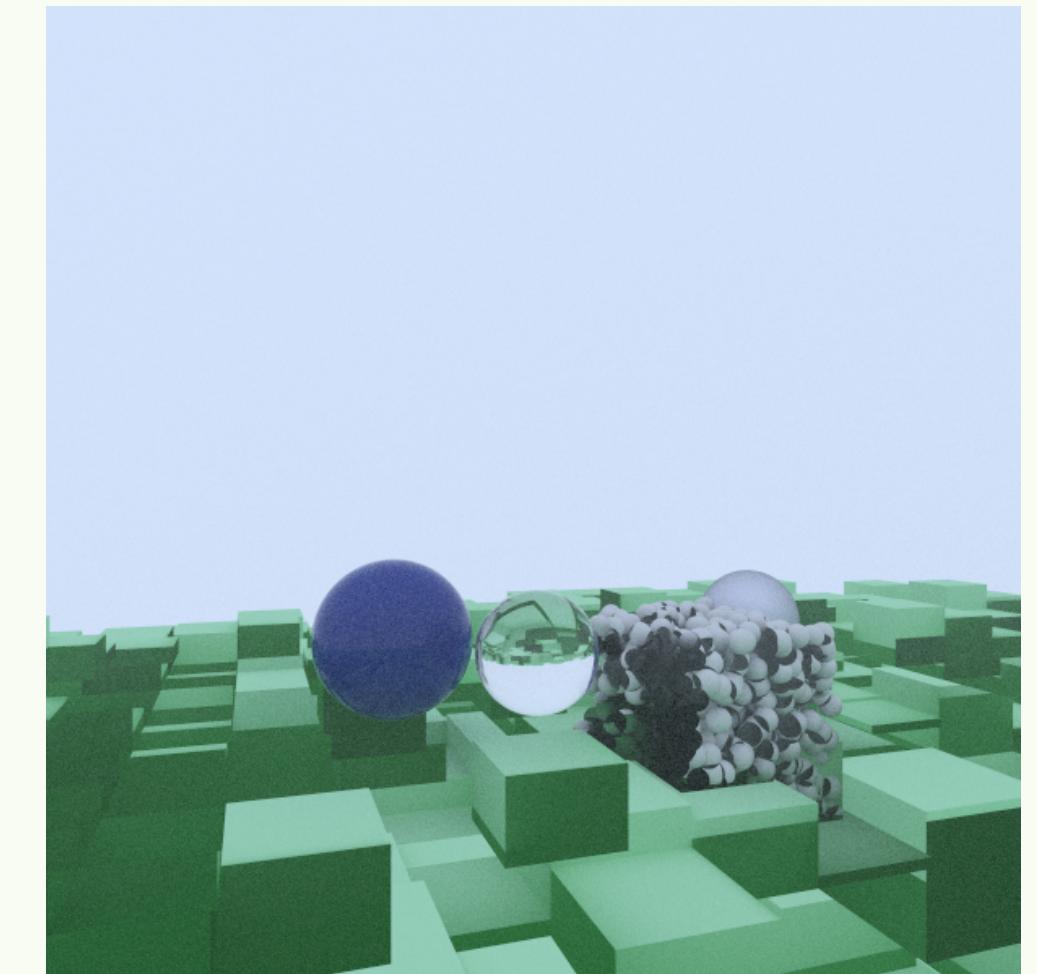
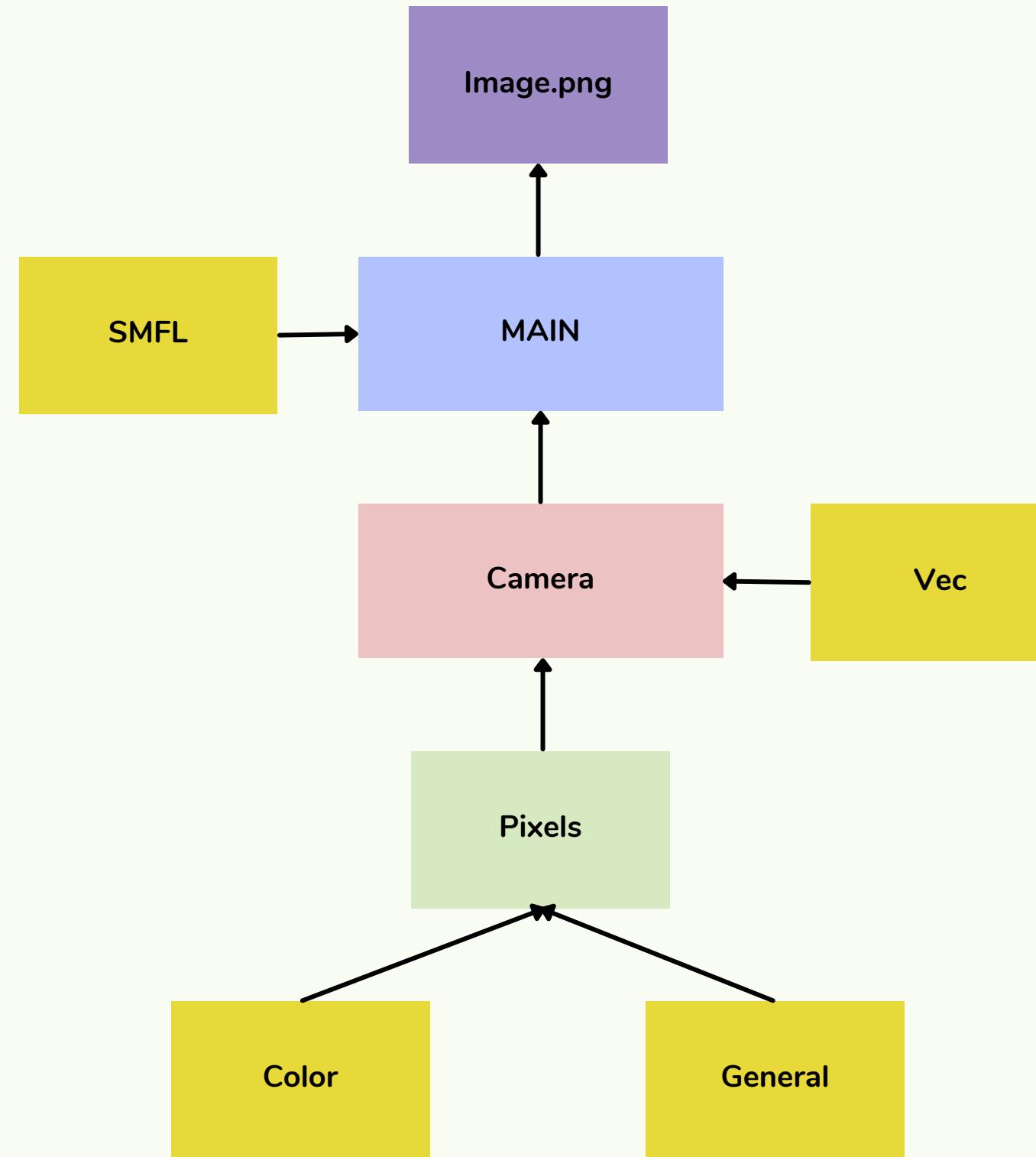
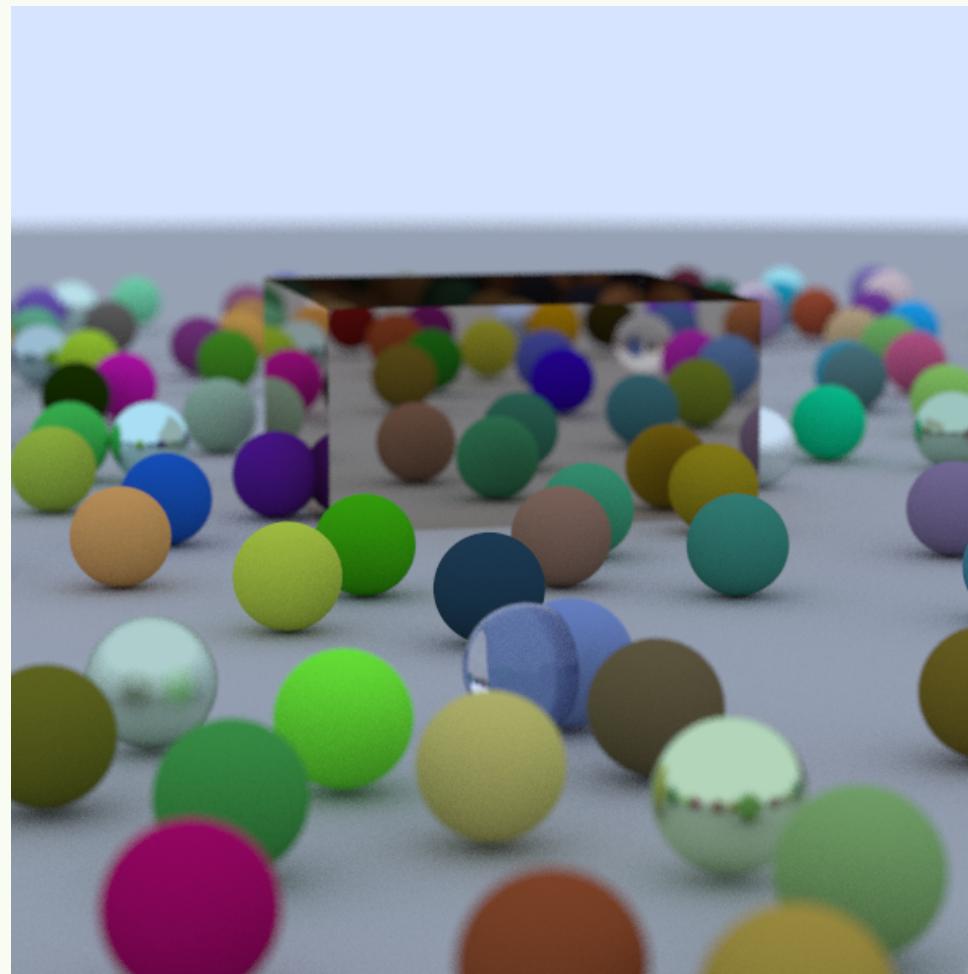
- Configuration des paramètres
- Affichage de l'image en temps réel
- Facilité d'utilisation par l'utilisateur
- Facile à mettre en œuvre
- Adaptabilité

## Inconvénients:

- La bibliothèque ne dispose pas de boutons et de boîtes de saisie de manière simplifiée.
- Difficulté dans le positionnement des objets.



# Classes Interface et affichage



# Améliorations possibles

## Parallélisation

- Paralléliser le calcul des rayons
- Idéalement en utilisant le GPU
- Séparer l'affichage du calcul dans différents threads

## L'interface

- Améliorer l'UX
- Permettre la modification d'objets sur scène

# Références

“Ray Tracing in One Weekend Series.” [raytracing.github.io/books/RayTracingInOneWeekend.html](https://raytracing.github.io/books/RayTracingInOneWeekend.html)  
(accessed 02. 01, 2024)

“SFML Documentation.” <https://www.sfml-dev.org/documentation/2.6.1/>  
(accessed 21.03, 2024)

