

# Moteur de lancer de rayons

Coiffier Guillaume - Béthune Louis - Felderhoff Joël

Janvier 2017

## Présentation

Le *Raytracing* (ou lancer de rayons en québécois supérieur) est une méthode de rendu de scènes 3D. Il permet d'obtenir des images d'excellente qualité en simulant le comportement naturel de la lumière : réflexions, ombres, changement d'indice, diffusion, éclairage, flou de profondeur...

La théorie sur le domaine est assez ancienne (années 60) mais l'exigence de cette méthode en terme de charge de calcul ont longtemps limité son usage. Cette technologie a donc connu une renaissance dans les années 2000, notamment dans le cinéma d'animation ou la simulation scientifique.

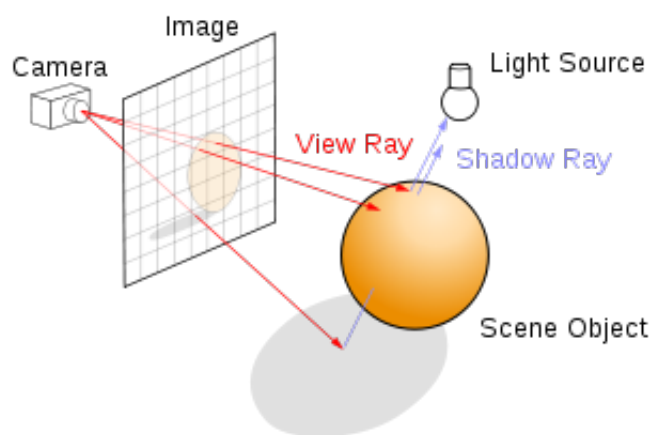
C'est une technique assez simple à mettre en œuvre et on obtient rapidement des résultats, même si un moteur complet (comme celui ayant produit l'image ci-dessous) nécessite un travail titanesque.



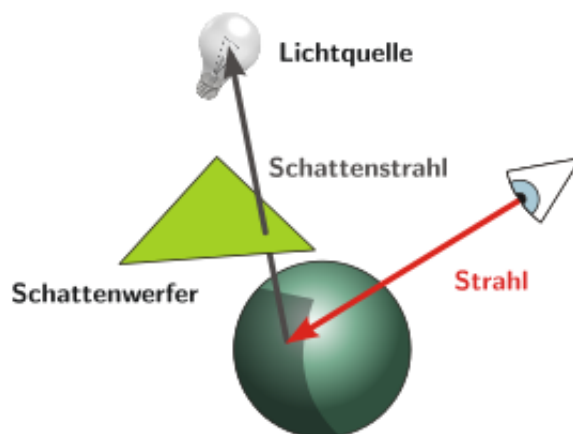
## 1 Principe

L'idée est de simuler le comportement d'un rayon lumineux, en utilisant le principe du parcours inverse de la lumière.

Ainsi, pour chaque pixel de la caméra, on lance un rayon et on cherche le premier objet qu'il intersecte. On peut alors afficher ce pixel de la couleur de l'objet.



On peut prendre en compte l'éclairage de façon très simple : on calcule les coordonnées du point d'intersection, puis on "tire" un rayon imaginaire en direction de la source de lumière. Si ce rayon atteint la source de lumière, le pixel apparaît éclairé. Dans le cas contraire, si on intersecte un autre objet de la scène, alors il y a occlusion et le pixel apparaît en sombre.



## 2 Votre mission

### Niveau 1 : le z-buffer

Implémentez un raytracer élémentaire. Pour cela vous devez supporter :

**Une caméra** simple unidirectionnelle, de normale orientée selon l'axe z et centrée en

**Des objets simples** comme des sphères monochromatique. Pensez à une architecture assez souple pour pouvoir gérer d'autres formes plus tard.

Équation décrivant un rayon :  $t \in \mathbb{R}, d \in \mathbb{R}^3, O + dt$  où  $d$  est la direction

Équation décrivant une sphère :  $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$   
de centre  $(x_0, y_0, z_0)$  et de rayon  $R$

**Un z-buffer** pour prendre en compte la profondeur. Les objets proches doivent être affichés avec des couleurs vives, et les objets lointains avec des couleurs sombres.

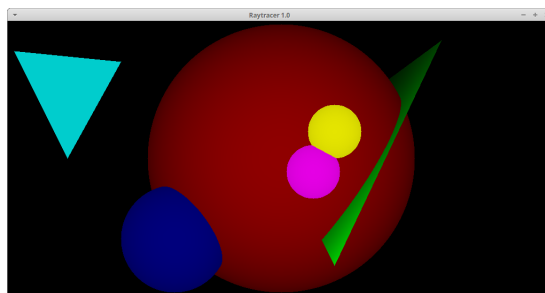
### Niveau 2 : d'autres formes et des couleurs

Rajoutez :

**D'autres formes** comme des triangles par exemple. Intéressez vous à l'algorithme de *Moller Trumbore*.

**De la couleur** sur vos objets. Prévoyez une architecture assez souple pour gérer des matériaux plus compliqués plus tard (surfaces réfléchissantes, textures).

**Exemple :**



## Conseils

### 2.1 Architecture

blablabla

## 2.2 Bibliothèque et langage

La bibliothèque standard du C++ ne propose pas nativement de moyens de faire du fenêtrage ou des applications graphiques. Il vous faudra donc une bibliothèque extérieure, dont il faudra apprendre la documentation et le fonctionnement.

Il en existe des dizaines mais je suggère à titre personnel la SFML(<https://www.sfml-dev.org/index-fr.php>) qui est :

- simple à utiliser et très intuitive, peu volumineuse
- suffisamment rapide pour l'usage que l'on va en faire
- entièrement orientée objet
- exploitant les mécanismes du C++ idiomatique comme la RAI
- développée par un français, avec des tutoriels en français et une communauté francophone active
- multi-plateforme