

Lancer de rayon

Laura GENEAU, Safidy RAKOTOBE

4 janvier 2026

Table des matières

1	Contexte	3
2	Mise en scène	3
3	Concepts physiques et mathématiques	3
3.1	Intersection d'un rayon et d'une forme	3
3.1.1	Intersection d'un rayon et d'une sphère	3
3.1.2	Intersection d'un rayon et d'un plan infini	4
3.2	Réflexion d'un rayon sur une surface	4
3.3	Calcul des ombres portées	5
4	Conception Logicielle : Diagramme UML	5
5	Nos choix d'implémentation	6
6	Problèmes rencontrés et solutions	6
6.1	Précision des nombres flottants	6
6.1.1	Réflexion et points noirs	6
6.1.2	Shadow Acne	7
7	Interactivité et contrôles	8
7.1	Contrôle de la lumière	8
7.2	Contrôle de la caméra	8

1 Contexte

La synthèse d'images 3D repose principalement sur deux approches : la rasterisation et le lancer de rayon. Contrairement à la rasterisation qui projette des objets sur un plan, le lancer de rayon simule le trajet inverse de la lumière en suivant des rayons partant de l'œil du spectateur à travers chaque pixel de l'écran.

Ce projet vise à implémenter un algorithme de base permettant de calculer l'intersection de ces rayons avec des objets géométriques (sphères, cubes, quadrillatères) et de gérer les phénomènes d'illumination, d'ombre portée et de réflexion. La scène finale consiste en une boîte composée de cinq quadrillatères, contenant divers volumes et une source de lumière, offrant ainsi un rendu réaliste par simulation physique des rebonds lumineux.

2 Mise en scène

La scène est composée d'une caméra orientée vers la face ouverte d'une boîte verte mate. Une source de lumière est disposée sur la face supérieure de cette boîte. À l'intérieur, nous avons intégré plusieurs volumes : une sphère rouge réfléchissante, un cube bleu réfléchissant et un quadrillatère jaune mat.

3 Concepts physiques et mathématiques

3.1 Intersection d'un rayon et d'une forme

3.1.1 Intersection d'un rayon et d'une sphère

Le calcul d'intersection repose sur la résolution d'un système à deux équations :

- **Le Rayon :** Un point P appartenant au rayon est défini par :

$$P = P_0 + t\vec{d}$$

Où P_0 est l'origine du rayon, \vec{d} sa direction normalisée, et t la distance parcourue ($t > 0$).

- **La sphère :** Un point P appartient à la sphère si :

$$\|P - C\|^2 = R^2$$

Où C est le centre de la sphère et R son rayon.

En substituant l'équation du rayon dans celle de la sphère, on obtient une équation du second degré : $at^2 + bt + c = 0$, avec :

- $a = \vec{d} \cdot \vec{d}$
- $b = 2\vec{d} \cdot (P_0 - C)$
- $c = (P_0 \cdot P_0) + (C \cdot C) - 2(C \cdot P_0) - R^2$

Analyse du discriminant $\Delta = b^2 - 4ac$:

- Si $\Delta < 0$: Pas d'intersection.
- Si $\Delta = 0$: Le rayon est tangent (un point de contact).
- Si $\Delta > 0$: Le rayon traverse la sphère. On retient la plus petite valeur de t positive.

3.1.2 Intersection d'un rayon et d'un plan infini

- **Le Rayon** : $P = P_0 + t\vec{d}$
- **Le Plan** : Défini par son origine et sa normale \vec{N} :

$$(P - \text{origine}) \cdot \vec{N} = 0$$

La distance d'intersection t est alors donnée par :

$$t = \frac{(\text{origine} - P_0) \cdot \vec{N}}{\vec{d} \cdot \vec{N}}$$

Note : Si $\vec{d} \cdot \vec{N} = 0$, le rayon est parallèle au plan.

3.2 Réflexion d'un rayon sur une surface

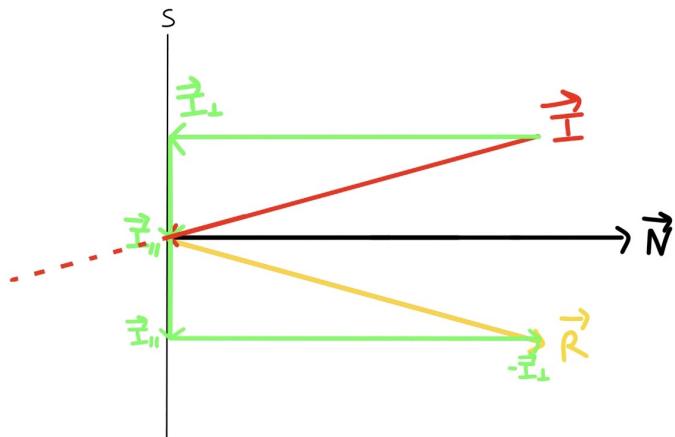


FIGURE 1 – Schéma de réflexion avec coefficient de shininess

Légende et calculs :

- \vec{I} : Rayon incident
- \vec{R} : Rayon réfléchi
- \vec{N} : Normale à la surface

Le vecteur réfléchi est calculé selon la formule :

$$\vec{R} = \vec{I} - 2(\vec{I} \cdot \vec{N})\vec{N}$$

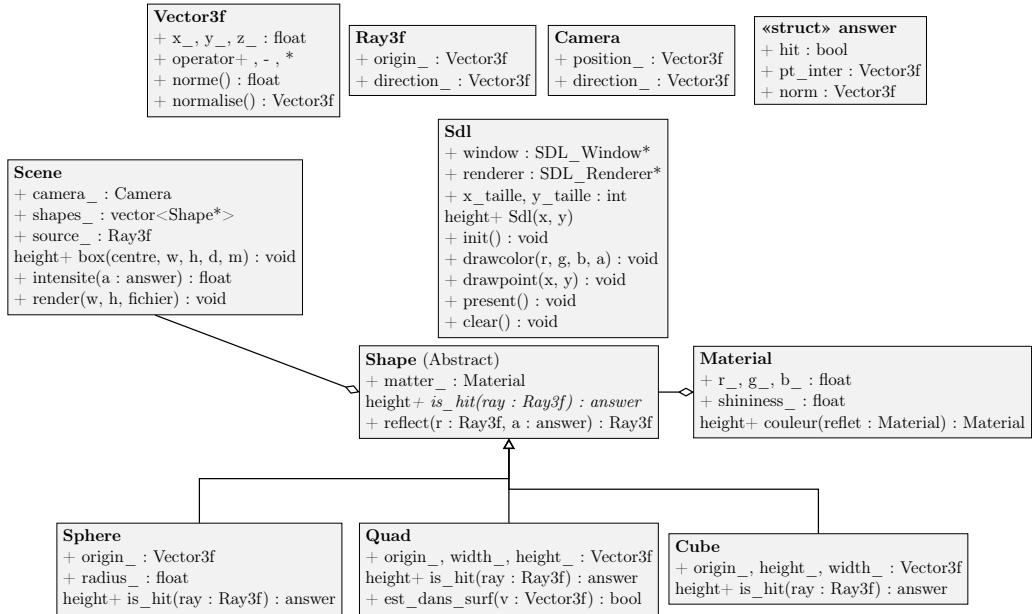
3.3 Calcul des ombres portées

Pour déterminer si un point d'impact P est ombragé, un rayon est lancé depuis P vers la source de lumière L .

- **Direction :** $\vec{d}_{ombre} = \frac{\vec{L}-\vec{P}}{\|\vec{L}-\vec{P}\|}$
- **Correction d'origine :** $P_{start} = P + \epsilon \vec{N}$ pour éviter l'auto-intersection.

Le point est à l'ombre si un obstacle est détecté entre l'origine et la lumière ($0 < t < \text{distance_lumiere}$). L'intensité du pixel est alors limitée à la lumière ambiante.

4 Conception Logicielle : Diagramme UML



Note : Le système est complété par les fonctions globales de calcul récursif, de gestion des couleurs et les opérateurs mathématiques vectoriels.

5 Nos choix d'implémentation

La méthode `is_hit` a été conçue pour retourner une structure `answer` contenant, outre le booléen de succès, le point d'intersection précis et la normale associée. Ce choix évite la multiplication des méthodes virtuelles complexes et centralise le calcul de réflexion dans la classe mère `Shape`.

Les types de données simples (`Ray3f`, `Vector3f`) sont passés par valeur pour plus de lisibilité, tandis que la structure `answer` est passée par référence pour optimiser les performances.

La fonction récursive gère simultanément les ombres et les réflexions. Un compteur de profondeur limite les rebonds récursifs (éitant les boucles infinies entre miroirs) et assure que le test d'ombre n'est effectué qu'une seule fois par rayon primaire.

6 Problèmes rencontrés et solutions

6.1 Précision des nombres flottants

L'utilisation de types `float` induit des erreurs d'arrondi impactant le rendu.

6.1.1 Réflexion et points noirs

Sans correction, le point d'origine du rayon réfléchi peut se retrouver par erreur à l'intérieur de la forme. Le rayon s'auto-intersecte alors, créant des points noirs sur la surface.

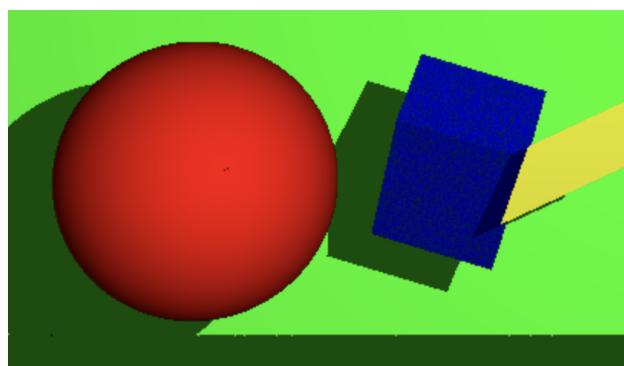


FIGURE 2 – Artefacts visuels sans décalage de l'origine

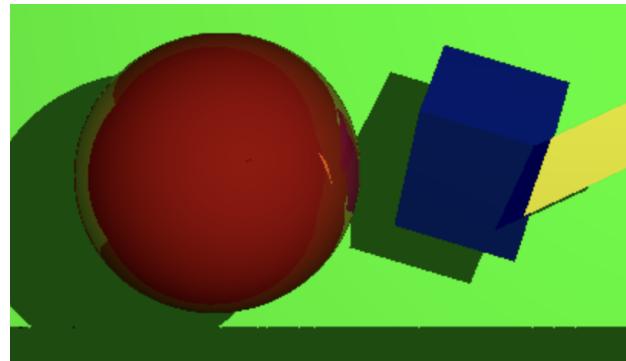


FIGURE 3 – Rendu corrigé par décalage du rayon réfléchi

6.1.2 Shadow Acne

Le même phénomène se produit pour les ombres : si l'origine du rayon vers la lumière n'est pas décalée selon la normale, l'objet projette une ombre sur lui-même (shadow acne).

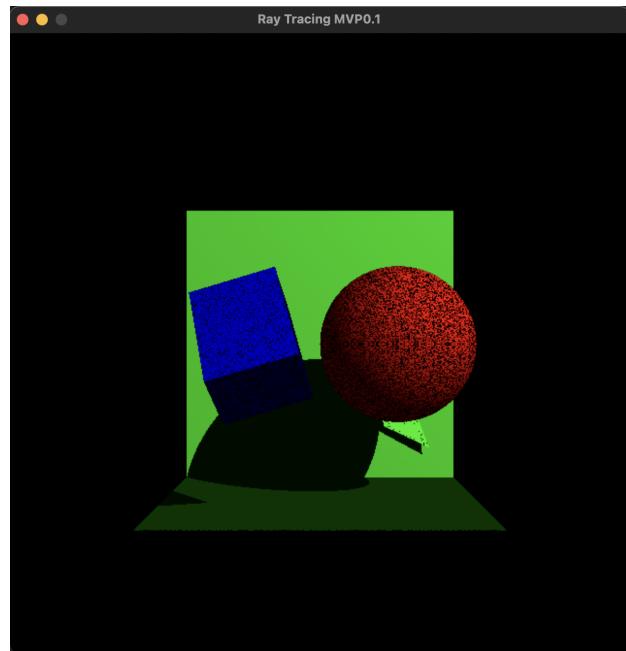


FIGURE 4 – Auto-intersection générant du shadow acne

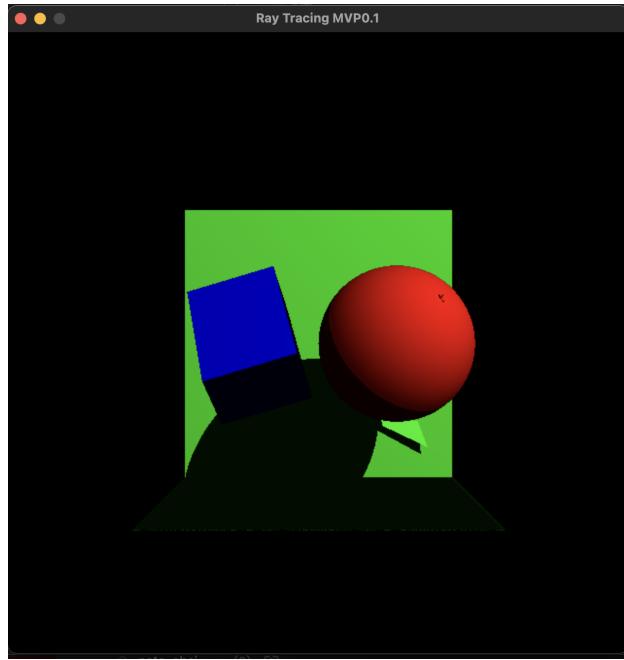


FIGURE 5 – Correction des ombres par décalage

7 Interactivité et contrôles

Nous avons implémenté le déplacement dynamique de la caméra et de la lumière via les entrées clavier pour explorer la scène.

7.1 Contrôle de la lumière

- **z / s** : Monter / Descendre la source.
- **q / d** : Déplacement latéral Gauche / Droite.
- **e / w** : Éloigner / Rapprocher (profondeur).

7.2 Contrôle de la caméra

- **Flèche Haut / Bas** : Monter / Descendre la caméra.
- **Flèche Droite / Gauche** : Déplacement latéral.