



ÉCOLE
CENTRALE LYON

ÉCOLE CENTRALE LYON

INFORMATIQUE GRAPHIQUE

Rapport



Ugo Insalaco - 20 mars 2023

Table des matières

1	Introduction	2
2	Sphères diffuses et miroir	2
3	Sphères transparentes	3
4	Fresnel et éclairage indirect	4
5	Maillages	5
6	Antialiasing et éclairage sphérique	7
7	Textures	8
8	Profondeur de champs	8
9	Rendu final	9

1 Introduction

Ce rapport présente les différentes étapes dans l'écriture d'un modèle de rendu 3D en C++. Dans cette méthode de tracé de rayon, nous commençons par afficher des sphères ayant une certaine position dans l'espace et une texture diffuse (et en appliquant une correction gamma), puis sont développés les modèles des sphères miroirs et des sphères transparentes en faisant faire des rebonds à chaque rayon. Le cas des sphères transparentes et du calcul des coefficients de Fresnel nous amène à créer une boucle en envoyant plusieurs rayons par pixel de notre image, ce qui est réutilisé par la suite pour la création d'éclairage indirect et la profondeur de champs. Enfin nous nous attaquons à l'affichage de maillages en implémentant une structure d'accélération de calcul : le BVH, puis la gestion des textures des maillages. Plusieurs images (notamment les dernières) ont été réalisées sur un PC d'emprunt disposant d'un processeur i5 avec 2 coeurs et 4 processeurs logiques, ce qui explique en partie les longs temps de calcul

2 Sphères diffuses et miroir

Première étape : lancer des rayons de notre caméra, passant par une grille de pixel et observer si le rayon atteint un objet et si cet objet est en éclairage direct avec la lumière. Dans le premier cas, on peut renvoyer l'albedo associée à l'objet et dans le cas contraire, le pixel affiché est noir.

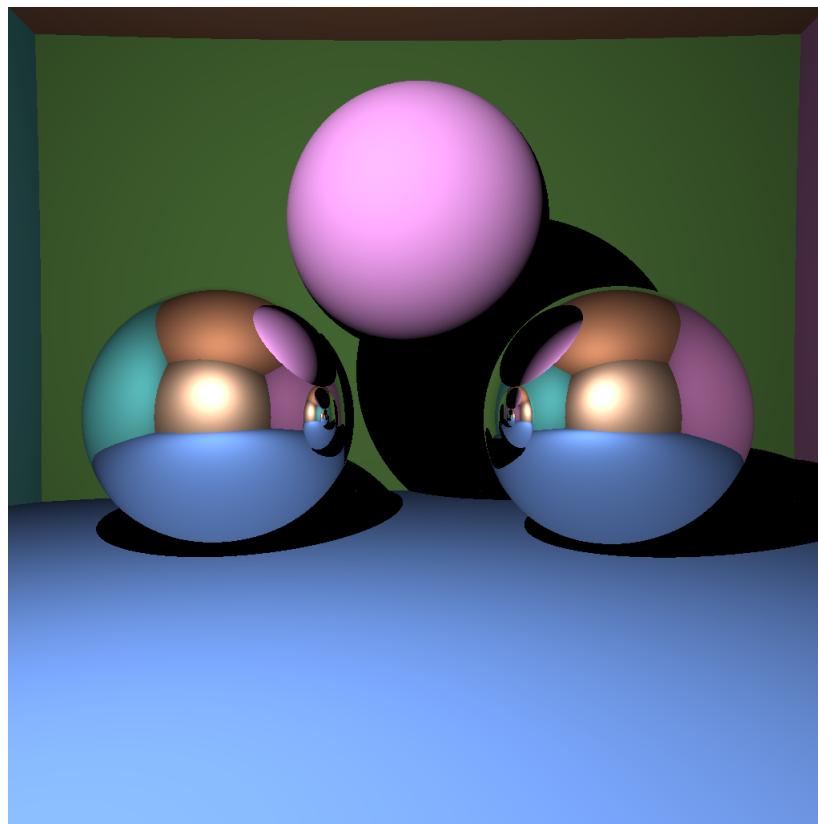


FIGURE 1 – Sphères diffuses, miroirs et éclairage direct

Si l'objet intersecté est un miroir, on lance alors un nouveau rayon dans la direction

symétrique au rayon incident par rapport à la normale de l'objet. Cette création récursive de rayon nous invite donc à créer une fonction *getColor* elle même récursive, en lui fournissant un nombre maximal de rebond (dans la suite gardé entre 4 et 6)

3 Sphères transparentes

La création des sphères transparentes est semblable à celle des sphères miroirs et s'appuie sur les lois de Descartes sur la réflexion et la réfraction. Chaque sphère a donc un indice de réfraction qui lui permet de plus ou moins dévier la lumière. Dans cette implémentation ici, on utilise une pile de sphère qui permet de comptabiliser quelles sphères on a traversé et l'on fait sortir la première sphère de la pile dès que l'on en sort (ce que l'on évalue avec la direction de la normale sortante de la sphère au moment de l'impact du rayon).

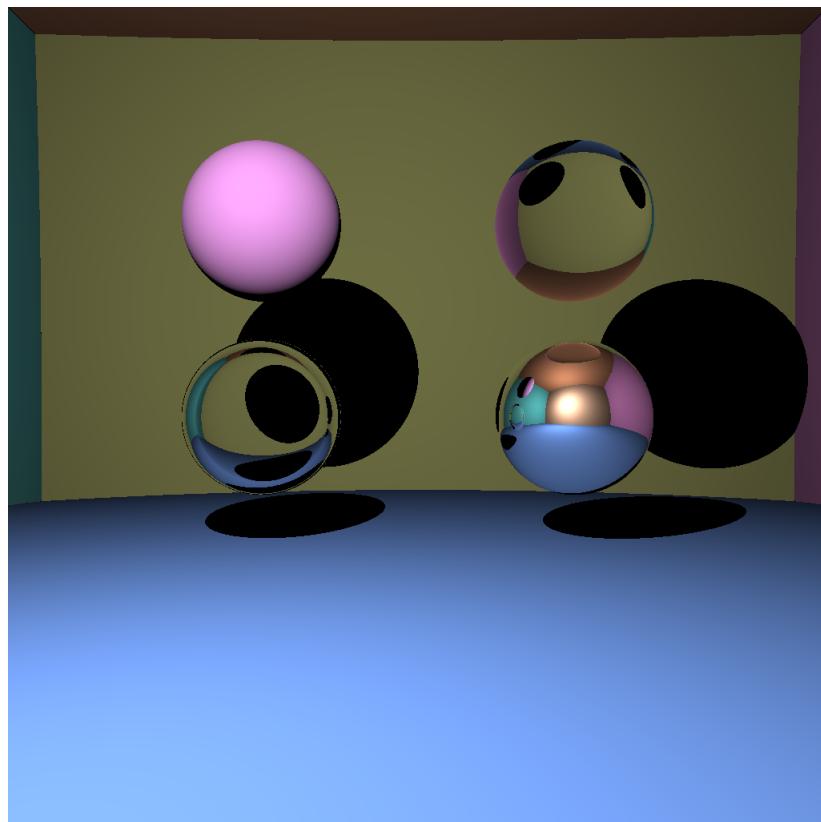


FIGURE 2 – Sphères transparentes, miroires et diffuses

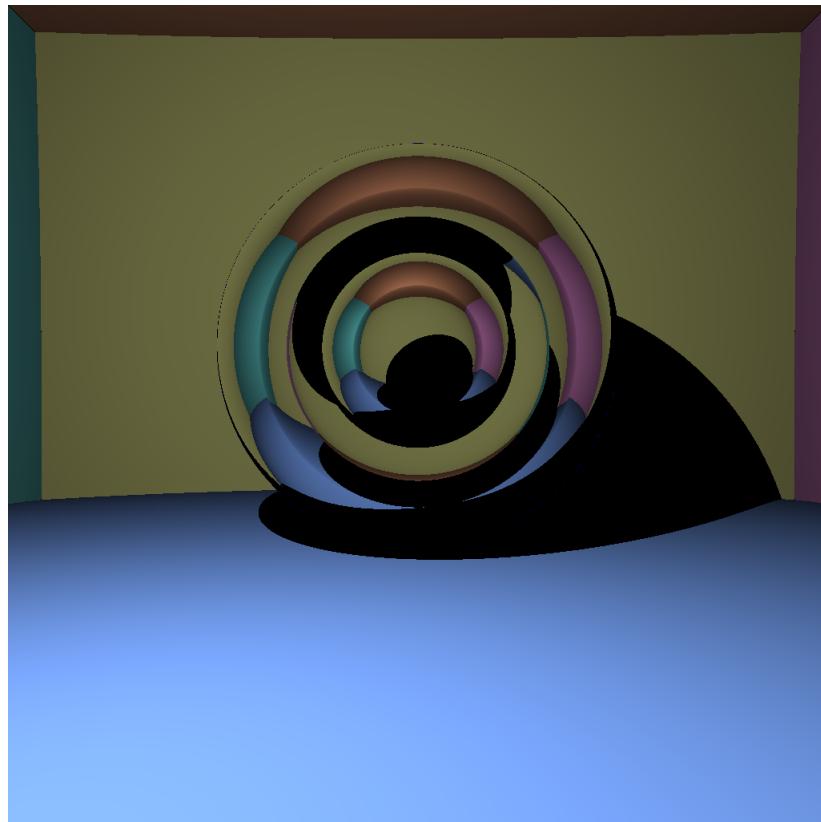


FIGURE 3 – Sphère transparente d'indice 1.5 dans une sphère transparente d'indice 1 dans une sphère transparente d'indice 1.5

4 Fresnel et éclairage indirect

On peut enfin implémenter les coefficient de Fresnel pour le sphères transparentes afin de faire rebondir une partie des rayons sur la surface de la sphère. Pour cela on réalise donc une boucle pour envoyer plusieurs rayons par pixel. On utilise ensuite cette boucle pour réaliser l'éclairage indirect. L'image est de ce fait de moins bonne qualité puisque l'on choisit aléatoirement une direction pour l'éclairage indirect, selon la méthode de Monte-Carlo, mais le rendu est bien plus réaliste.

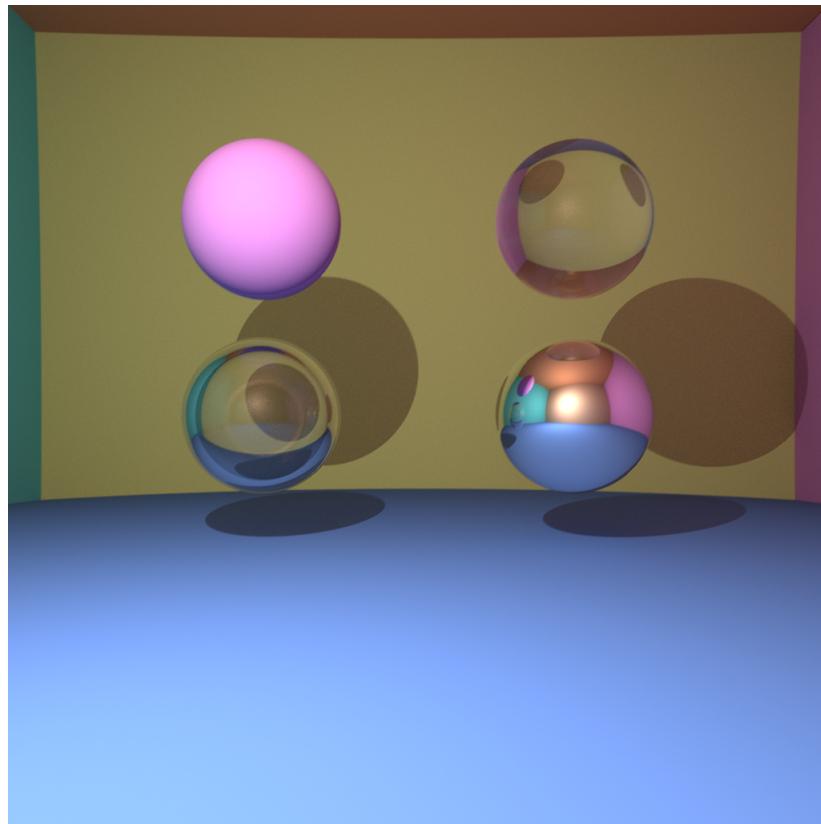
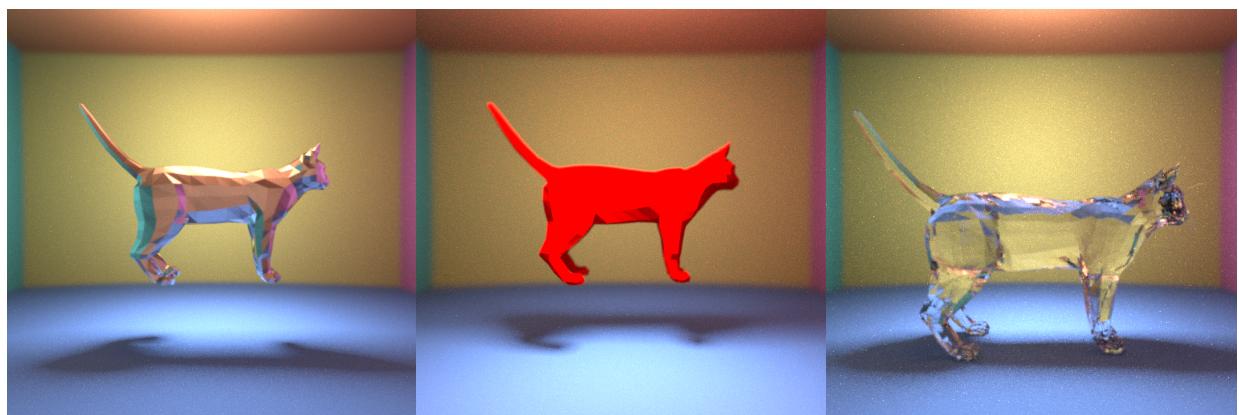


FIGURE 4 – Implémentation des coefficients de Fresnel, visibles sur la sphère en bas à droite et en haut à gauche, se comportant en partie comme un miroir (image réalisée en 100 rayons et 10 rebonds, temps non calculé)

5 Maillages

Grâce au parser de fichier obj et au calcul d'intersection de chaque rayon avec un triangle, il est ensuite possible d'afficher des géométries plus complexes telles que le chat ci-dessous.



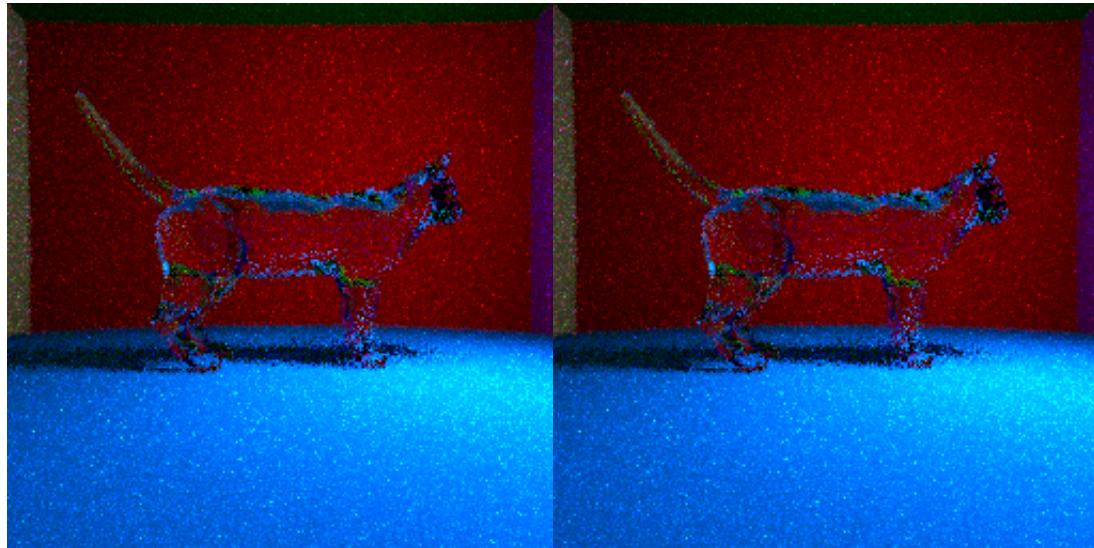
(a) chat miroir

(b) chat diffus

(c) chat transparent, 128 rayons,
5 rebonds, 55 min de calcul

FIGURE 5 – chats

Une structure d'accélération a été implémentée afin de ne pas avoir à calculer d'intersection pour chaque triangle du maillage à chaque rayon : le BVH. On peut donc comparer les temps de calcul pour deux images similaires, avec et sans BVH :



(a) chat avec BVH, $t = 12s$

(b) chat sans BVH, $t = 121s$

Pour une image de qualité strictement équivalente, on observe que le BVH permet de réduire le temps de calcul d'un facteur 10.

Certain·es artistes choisissent pour leurs modèles de fournir des normales en chaque sommet différentes de celles que l'on calcul "à la main" à partir de la position des sommets, ce qui permet au moment du rendu d'avoir un modèle plus lisse.

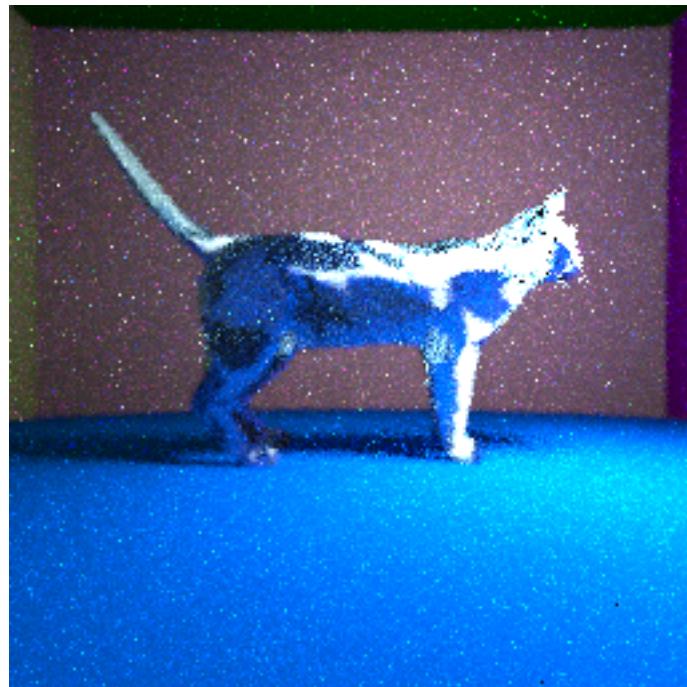
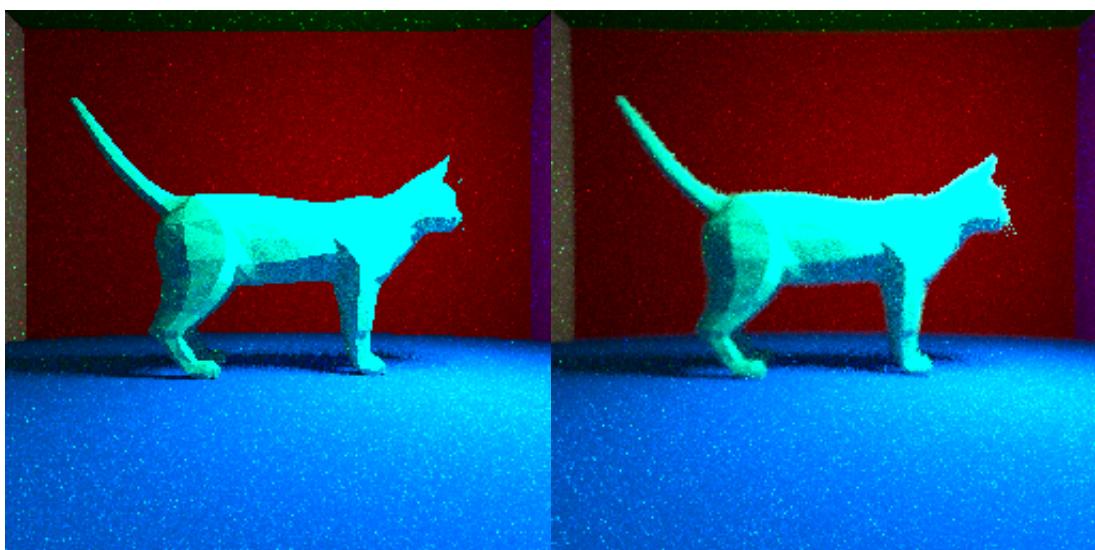


FIGURE 7 – Interpolation de normales pour notre chat

6 Antialiasing et éclairage sphérique

La boucle réalisée précédemment permet aussi d'envoyer des rayons ne passant plus par le centre de chaque pixel de la caméra, mais de réaliser une nouvelle intégration par monte carlo en choisissant aléatoirement une nouvelle origine à notre rayon et ainsi "lisser" les contours de chaque forme grâce à de l'antialiasing.



(a) chat sans antialiasing

(b) chat avec antialiasing

Afin d'adoucir les contours des ombres, de l'éclairage sphérique a été implémenté, ce qui permet pour l'éclairage indirect de ne pas avoir de démarcation précise de l'ombre mais plutôt une couleur qui s'assombrit peu à peu. On peut par exemple comparer les ombres des figures 4 et 5 pour se convaincre du réalisme d'un tel éclairage.

7 Textures

Les modèles 3D viennent en général avec une *Texture map* permettant d'appliquer une texture au modèle et d'un mapping entre les coordonnées UV de l'image et chaque sommet du maillage. En interpolant ce mapping, on peut générer un modèle texturé. Il arrive aussi souvent que des modèles possèdent plusieurs images de textures permettant de garder une organisation dans le design du modèle. C'est pourquoi un parser de fichier .mtl a été implémenté, afin de gérer plusieurs groupes de textures par modèle.

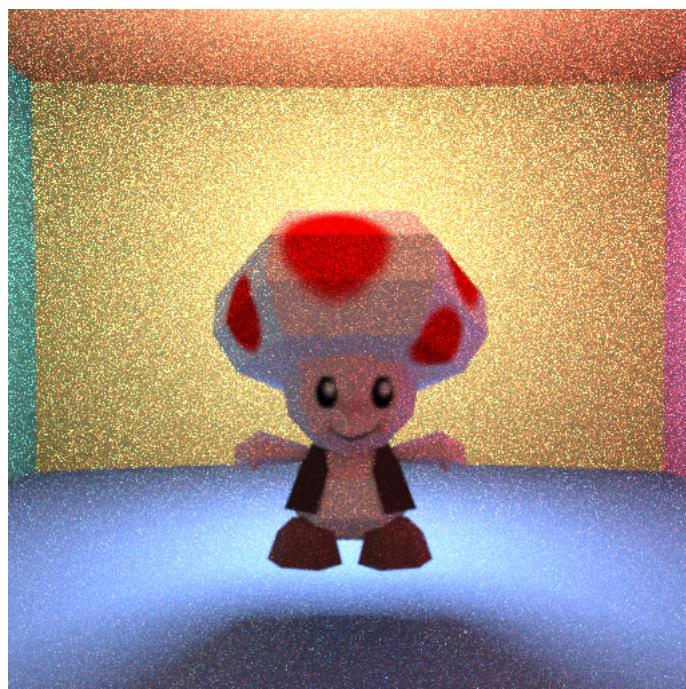


FIGURE 9 – Affichage d'un modèle avec texture

8 Profondeur de champs

En ajoutant un diamètre d'ouverture à notre caméra, il est possible d'ajouter un effet de profondeur de champs, faisant apparaître les objets hors de la zone de focus flous :

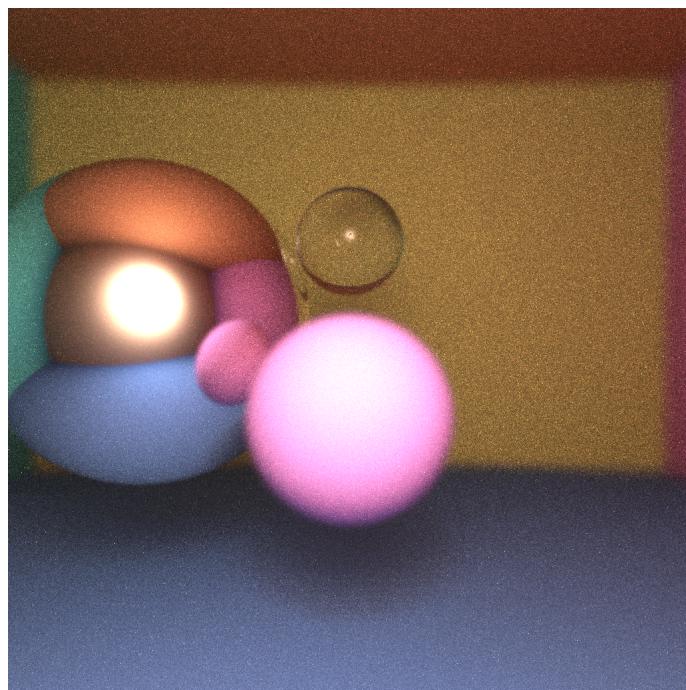


FIGURE 10 – Visualisation de la pronfondeur de champs

9 Rendu final

En transformant la position des maillages et de la caméra grâce à des translations, rotations et modification du fov, on peut créer des scènes complexes (pour peu que l'on mette le temps à placer correctement chaque objet)



FIGURE 11 – scène complexe, 128 rayons, 4 rebonds, 3h de placement des objets et 5h de temps de calcul