École Centrale Lyon

MSO 2.2

Informatique Graphique

Rapport

Rapport

Élèves :
Fabien Duranson

Enseignant :
Nicolas Bonneel

Introduction

Dans ce cours, nous avons pu créer un raytracer pas à pas en ajoutant des fonctionnalités de plus en plus avancées.

Un raytracer est un système de synthèse d'image et de rendu qui fonctionne en envoyant des rayons depuis une caméra et en calculant la couleur du pixel correspondant à ce rayon en fonction de plusieurs règles physiques liées à la géométrie, à la texture et/ou au matériau.

Dans ce cours nous avons choisi de nous intéresser à un raytracer plutôt qu'à des techniques de rendu moderne pour rester dans un cadre purement physique. Un raytracer est plus simple à implémenter que des techniques modernes mais beaucoup plus long. Nous donnerons toujours une référence de temps de calcul pour chaque image créée.

I. Rendu de sphères

I.1 Création des premières sphères et éclairage

La première étape a été d'ajouter des sphères colorées à une scène et une source de lumière. La contribution de la lumière dépend de la réflexion sur la sphère. Dans la figure 1, la lumière est placée en haut à droite de la scène.

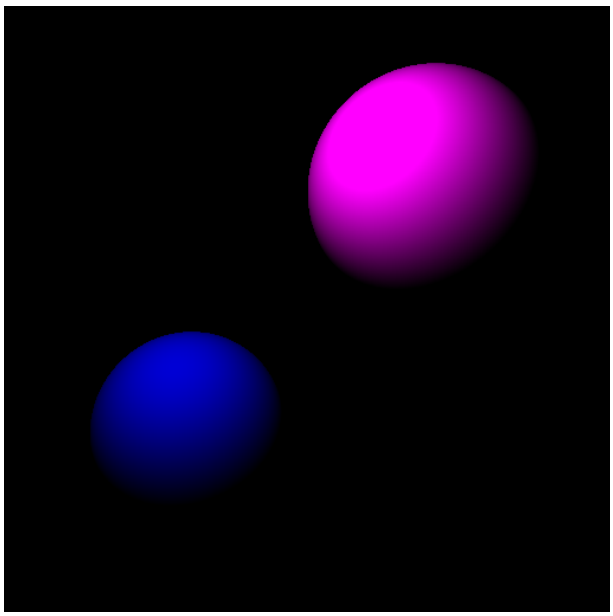

FIGURE 1 – 2 sphères éclairées par une source de lumière ($<$ 1ms)

I.2 Miroirs

Nous avons alors créé une scène complète avec des murs (des sphères "infinies"). La scène contient 9 sphères (5 murs + 4 objets) et une lampe.

J'ai défini le caractère réfléchissant d'une sphère comme une valeur entre 0 et 1 donnant la proportion de réflexion par rapport à l'absorption. J'ai mis les 3 plus petites sphères à un taux de réflexion de 1 et le sol à un taux de réflexion de 0.3. Voici le résultat :

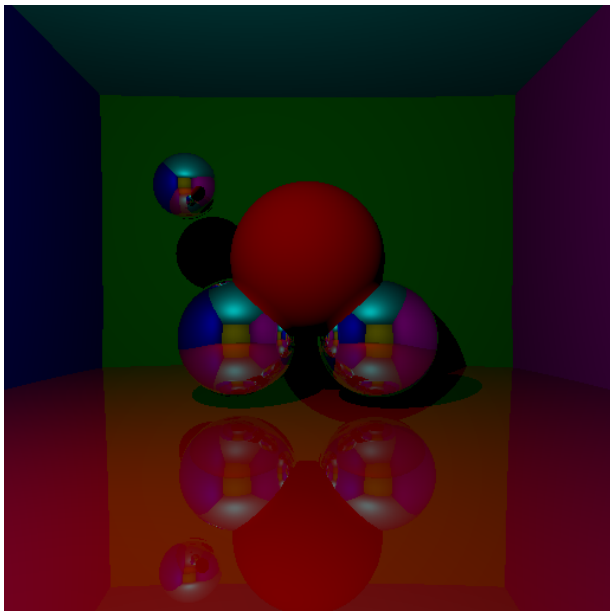

FIGURE 2 – Sol et objets réfléchissant (1,93 s)

I.3 Transparence

J'ai défini de la même manière un taux de réfraction par rapport à l'absorption :

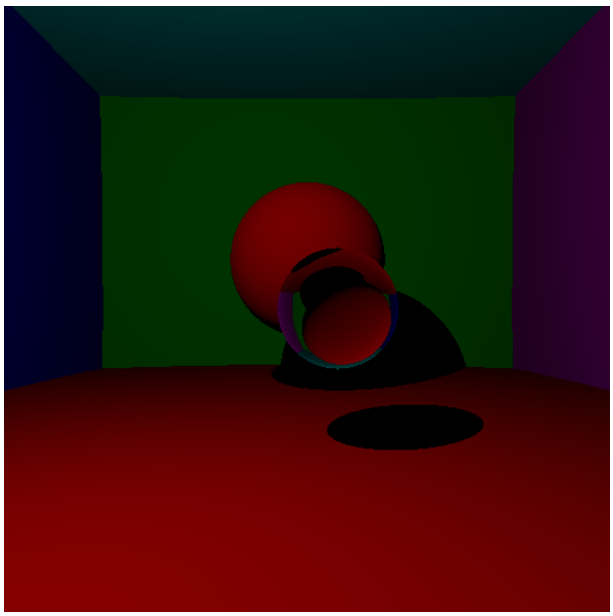

FIGURE 3 – Réfraction dans une sphère (1,27 s)

J'ai pu obtenir des scènes assez complexes en un temps correct :

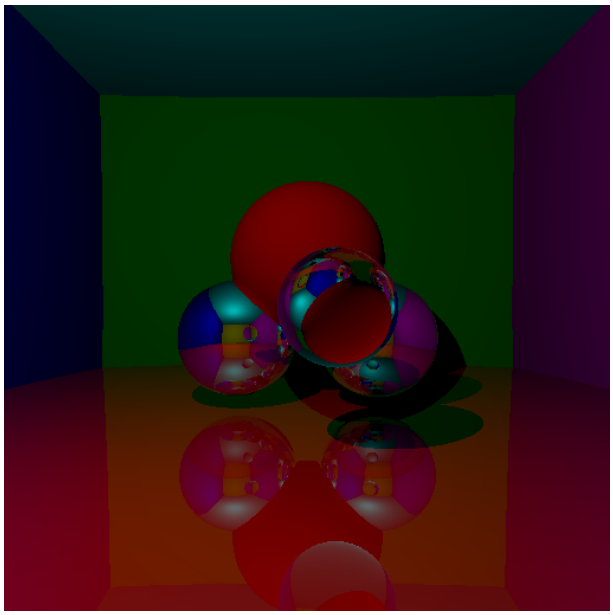

FIGURE 4 – Scène optique complexe (2,52 s)

Ici les deux sphères du bas sont des miroirs et la sphère en avant-scène est transparente.

I.4 Antialiasing

En zoomant sur une image, on se rend compte que la différence entre 2 pixels d'un bord d'objet est très marquée, ce qui choque l'oeil habitué à plus de résolution. On va donc créer un dégradé à cet endroit en déviant de manière aléatoire le rayon censé calculer la couleur du pixel puis faire la moyenne sur plusieurs rayons envoyés pour le même pixel.

On obtient ce genre de résultat :

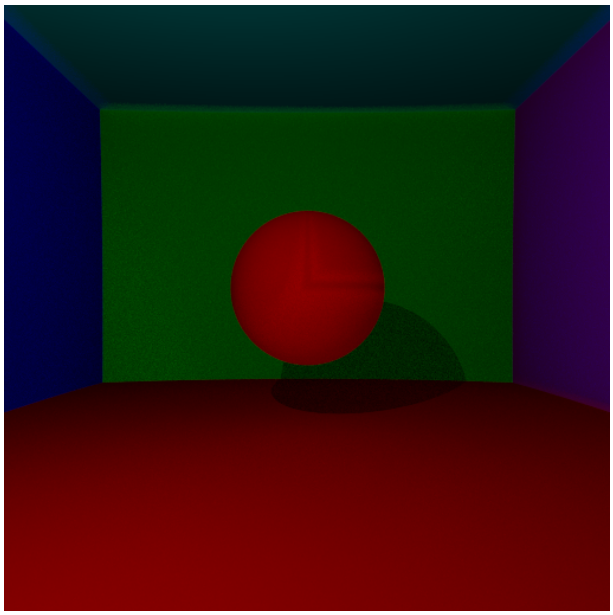

FIGURE 5 – Antialiasing avec 100 rayons (105s)

Cette fonctionnalité multiplie le temps de calcul par le nombre de rayons que l'on veut envoyer, on obtient une image très bruitée si il y a peu de rayons.

n.b. Il y a ici un bug dans le calcul qui forme un "L". Il a été corrigé par la suite.

I.5 Éclairage indirect et softlight

Jusqu'à présent, les ombres sont étaient très marquées car on affectait "noir" au pixel s'il n'était pas directement éclairé. On ajoute ici un éclairage indirect en rendant toute surface diffuse à la lumière. Ceci nous permet de rendre les ombres moins noires.

On a également élargit la lumière d'une source ponctuelle à une sphère ce qui permet d'adoucir les contours de la lumière.

On obtient alors des résultats bien plus réalistes :

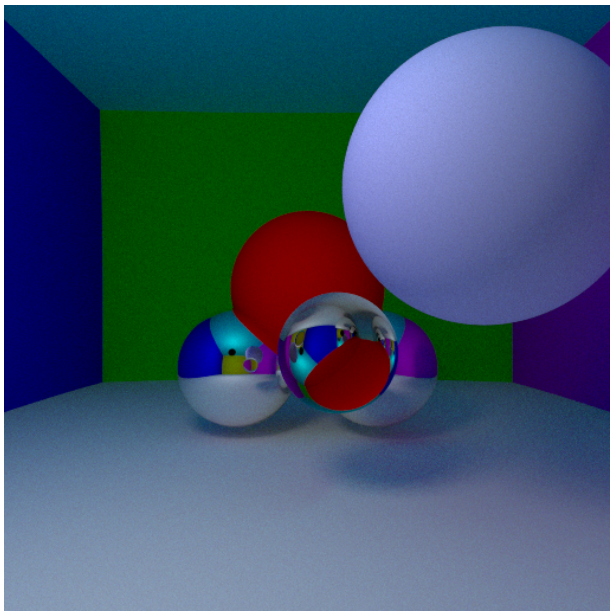

FIGURE 6 – Éclairage indirect et softlight avec 100 rayons (410s)

Les temps de calculs commencent à être très long (environ 7 minutes pour un rendu), j'ai essayé de mettre en place une structure de calcul parallèle en utilisant OpenMP mais n'ai pas réussi à l'utiliser avec XCode qui utilise le compilateur clang, donc j'ai fait avec.

I.6 Focus

Nous avons ensuite ajouté une ouverture d'objectif pour donner une impression de mise au point d'un appareil photo réel. Il suffisait de rendre aléatoire le point de départ du rayon et d'utiliser la profondeur de champs pour calculer la direction du rayon. Avec une grande ouverture et une profondeur de champs en avant-cène, on obtient un résultat comme celui-ci :

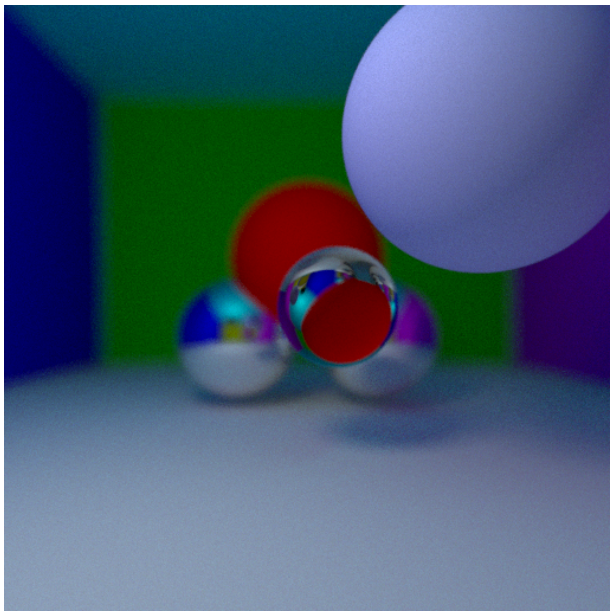

FIGURE 7 – Mise au point au premier plan avec grande ouverture avec 100 rayons (401s)

n.b. Le temps de calcul est sensiblement le même ici, car on ne rajouter pas de boucle algorithmique.

II. Rendu d'objets 3D

Nous nous sommes ensuite intéressé aux rendus d'objets 3D (mesh). Nous avions déjà la plupart des logiques implémentées et n'avions qu'à redéfinir certaines fonctions dans le cas d'un mesh (l'intersection rayon-objet notamment).

II.1 Rendu en couleur unie et boite englobante

Les premiers résultats étaient très décevants, les temps de calcul étaient énormes, la résolution était très mauvaise et l'image était très bruitée. Nous avons tout de suite implémenté une boite englobante qui nous a permis de réduire drastiquement le temps de calcul quand le rayon était sûr de n'intersecter aucun triangle de l'objet.

Nous avons tout d'abord fait des rendus avec un chien de couleur unie (jaune ici par incontestable goût artistique). Voici le résultat :

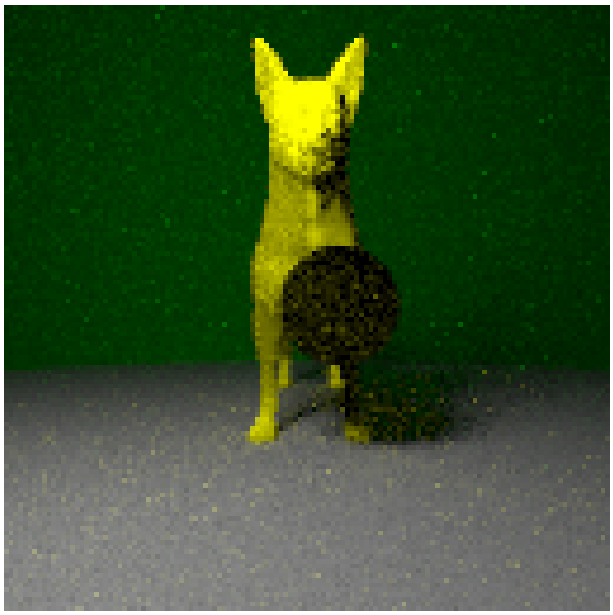

FIGURE 8 – Chien jaune avec boule noire devant avec boite englobante (18s)

Pour cette image la boite englobante accélère d'un facteur 8 le calcul : de 102s à 18s.

II.2 Arbre de boites englobantes

Nous avons encore poussé l'optimisation par boite englobante en créant un arbre de boite englobantes afin de s'y déplacer et de ne finir qu'avec une dizaine de triangles à tester au maximum.

On peut alors créer des images de meilleure qualité :

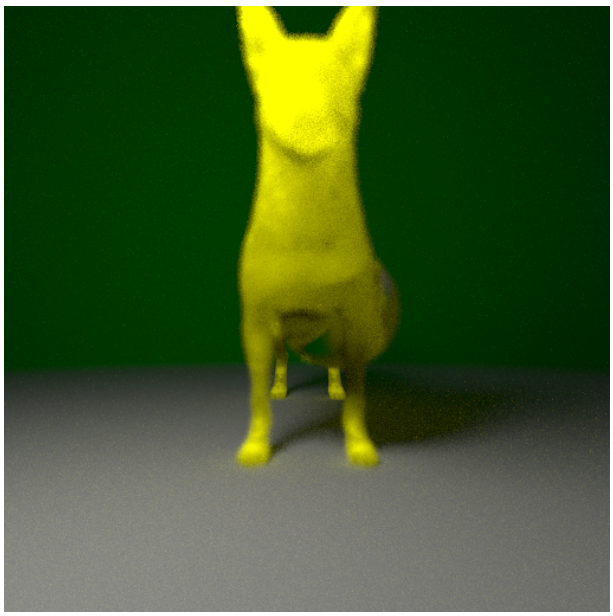

FIGURE 9 – Chien jaune avec boule transparente avec arbre de boites englobantes (18 min)

J'ai ensuite pu ajouté plusieurs objets 3D dans la même scène et le temps de calcul était grand mais pas déraisonnable :

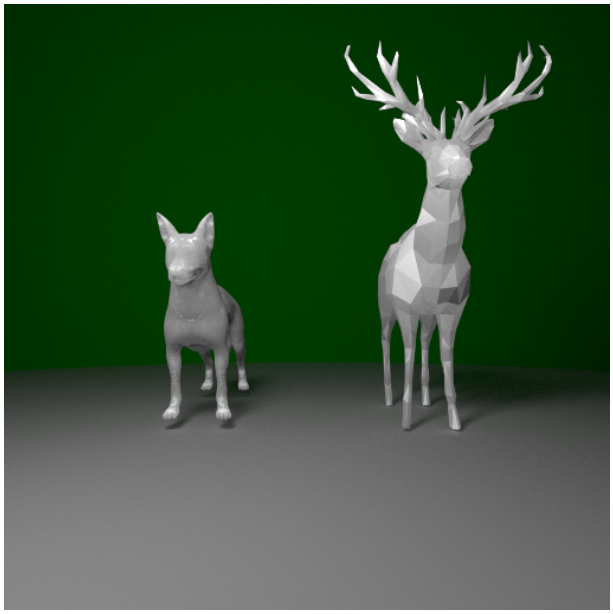

FIGURE 10 – Cerf et chien en haute résolution (43 min)

II.3 Ajout de la texture et changement de l'angle de la caméra

Nous avons ensuite utilisé le fichier de texture fourni avec le modèle 3D pour utiliser les vraies couleurs sur chaque point d'intersection pour le calcul de la couleur du pixel.

J'ai également défini une transformation de la caméra qui m'a permi de la positionner où je voulais.

Voici le résultat final :

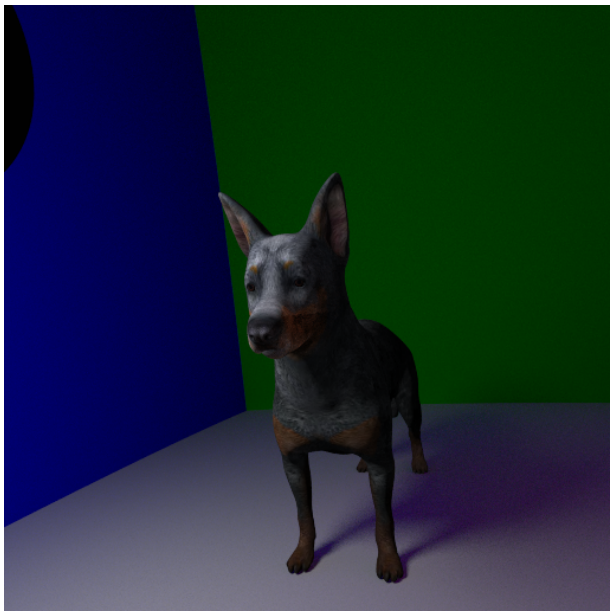

FIGURE 11 – Chien avec textures d'artiste (20 min)

Conclusion et retours

Dans ce cours, je me suis familiarisé avec le C++ et beaucoup de fonctionnalités du C++11 et de la STL. Ce qui me sera très utile pour mon TFE et les métiers que je vise. J'ai aussi beaucoup appris sur la synthèse d'images et refait un peu d'optique et de maths ce qui ne fais jamais de mal.

J'ai trouvé ce cours très interactif et je vous ai trouvé très à l'écoute et compréhensif sur les problèmes que nous avions ce qui faisait avancer le cours et nous faisait progresser dans un bon climat. J'aime les langages de programmations bas niveaue et j'ai beaucoup aimé m'y entraîner.

Cependant, j'ai trouvé ce cours assez compliqué et décevant pour trois raisons :

— La mise en place était très compliquée car nous n'avions jamais utilisé de langage de programmation compilée en cours à Centrale. La plupart des retards que l'on accumulait en cours viennent de ce faux départ et de problèmes liés au fait que tout le monde code du Python et ne s'attend pas à des erreurs. Un tutoriel donné à l'avance et un cours de 1h de Hello World aurait sûrement empêché la plupart de ces problèmes.

— Nous n'avions pas le temps de nous poser des questions architecturales ni même toutes les informations pour pouvoir faire des choix pertinents. Je suis parti plusieurs fois dans des choix qui différaient des vôtres pour ne pas faire "de la dactylographie" mais le cours suivant mon cours m'a énormément complexifié l'implémentation d'une fonctionnalité qui était évidente à implémenter pour vous. J'ai donc pris du retard car vous avez directement enchaîné sur la feature suivante. J'ai le sentiment que nous sommes obligés de recopier, au moins en partie, ce que vous faites vu le rythme auquel vous avancez et vu que nous n'avons pas de vision d'avenir sur ce que nous faisons.

— Nous ne respections pas les bonnes pratiques du C++. Vous ne nous poussiez pas à faire des fichiers séparés, des headers et des cpp, les variables étaient nommées en notation mathématique pour être plus compactes. Je trouve ça assez dommage car nous n'avons pas d'autres occasions que les cours universitaires pour travailler ce genre de pratiques et dans quelques semaines nous entrerons en entreprise.

Malgré ces retours, c'était vraiment un très bon cours, un des meilleurs que j'ai eu à Centrale.

Merci beaucoup.