



ÉCOLE CENTRALE LYON

MOS 2.2
INFORMATIQUE GRAPHIQUE

Raytracing

Élève :
Jean WOLFF

Enseignant :
Nicolas BONNEEL

24 mars 2020

Dans le cadre du cours d'informatique graphique en dernière année à l'École Centrale de Lyon, on implémente un raytracer. Les objets visualisés seront initialement uniquement des sphères, puis nous ajouterons les triangles et maillages, avec de nombreux effets visuels, pour obtenir un rendu de plus en plus réaliste.

Le projet a été réalisé avec XCode (sur macOS). La compilation s'effectuait dans le terminal, avec les commandes :

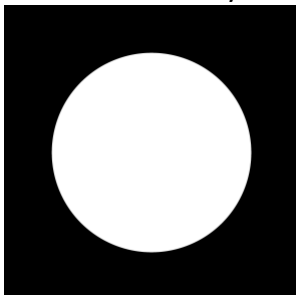
- Compilation : `clang++ -Xpreprocessor -fopenmp main.cpp Vector.cpp -std=c++11 -lomp -O3`
- Exécution : `./a.out 5 image11.png 1024 1024`, où les paramètres sont successivement : le nombre de rayons par pixel, le nom de l'image, sa largeur et sa hauteur.

La compilation durait environ 5 secondes. L'exécution, durait généralement entre 0.1 s et quelques secondes. Toutes les durées d'exécutions étaient mesurées, mais seules celles supérieures à une dizaine de secondes sont précisées dans ce rapport. Les calculs pour chaque pixel sont parallélisés grâce à openmp.

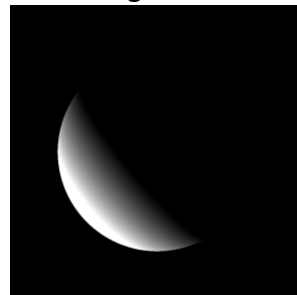
Fonctionnalités implémentées :

- Intersection rayon - sphère
- Éclairage d'un matériau diffus
- Ombres portées
- Surfaces spéculaires
- Surfaces transparentes
- Sphère creuse
- Éclairage indirect avec une intégration de Monte Carlo
- Anti-aliasing
- Lumières étendues
- Ombres douces
- Profondeur de champ
- Intersection rayon – triangle

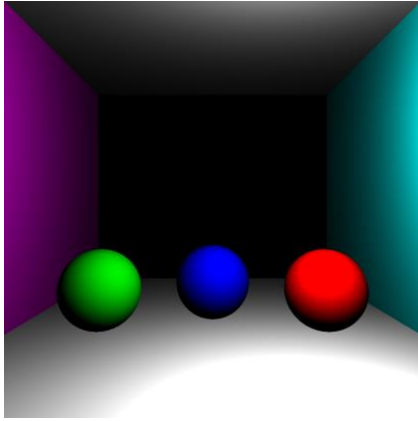
1. Intersection rayon - sphère



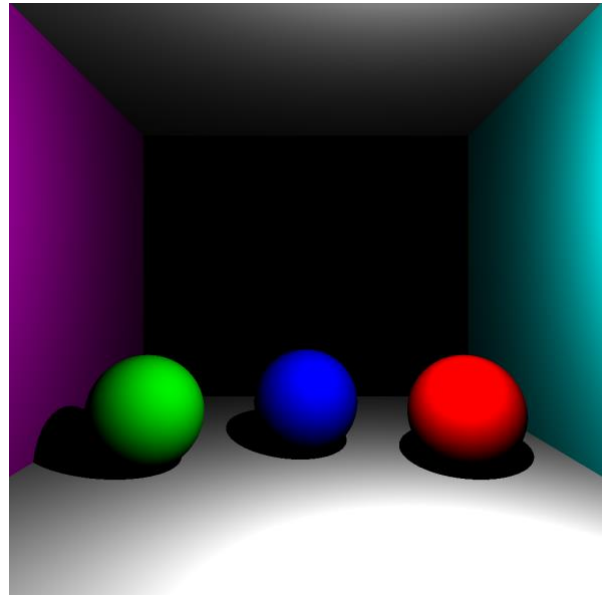
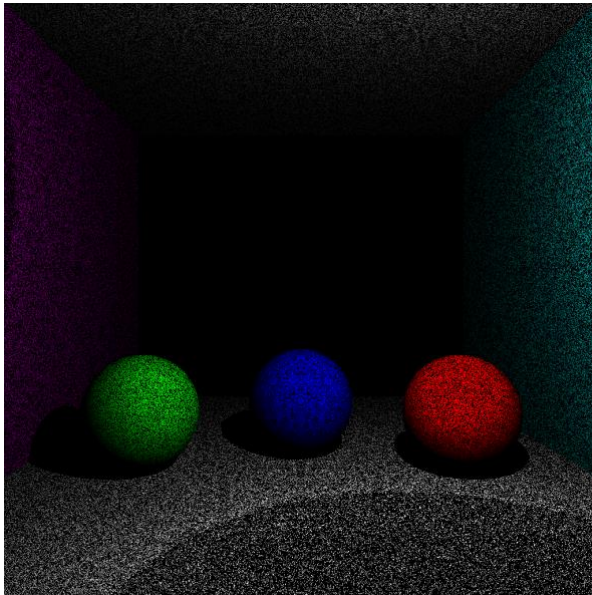
2. Éclairage d'un matériau diffus



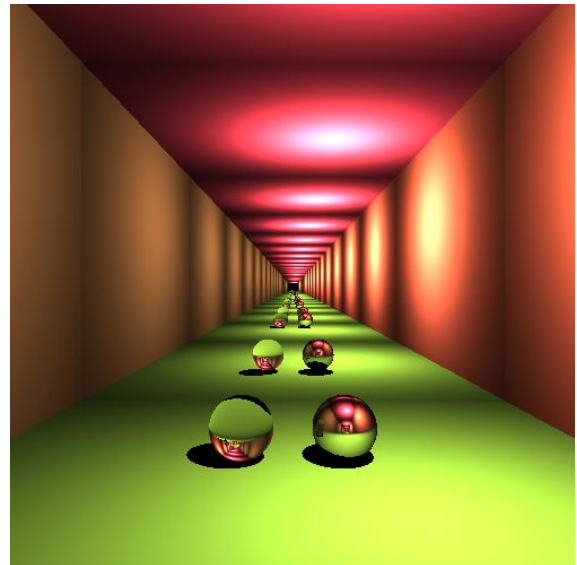
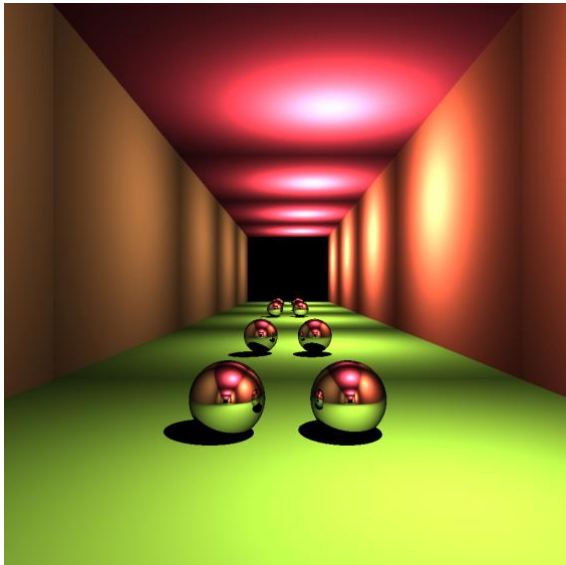
3. Intersection rayon - scène



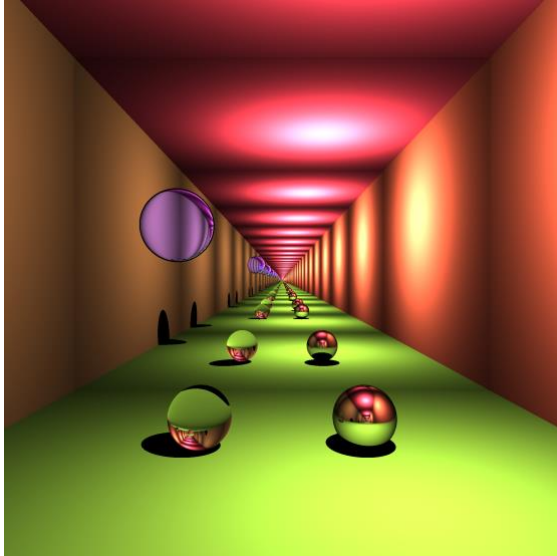
4. Ombres portées



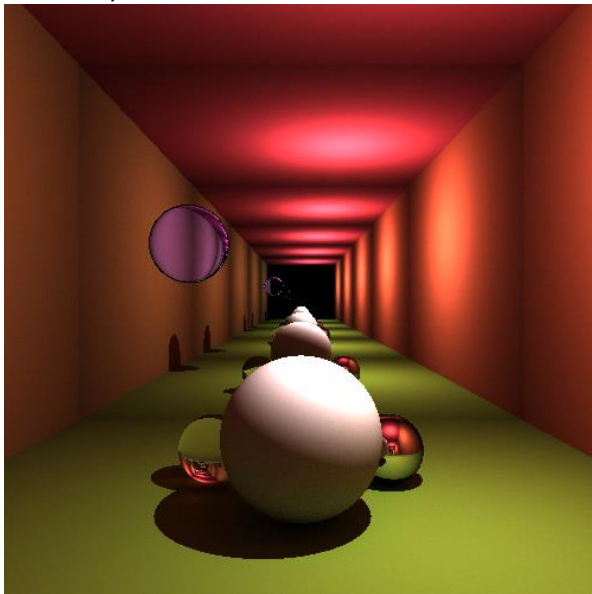
5. Surfaces spéculaires (5 rebonds au maximum) 6. Surfaces transparentes (10 rebonds max)



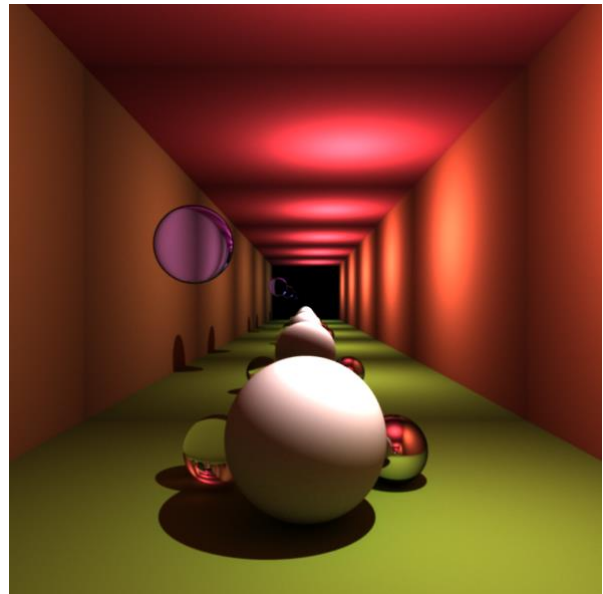
7. Sphère creuse (30 rebonds max)



8. Éclairage indirect avec une intégration de Monte Carlo (image 500x500, 100 rayons par pixel, durée d'exécution : environ 4 minutes)



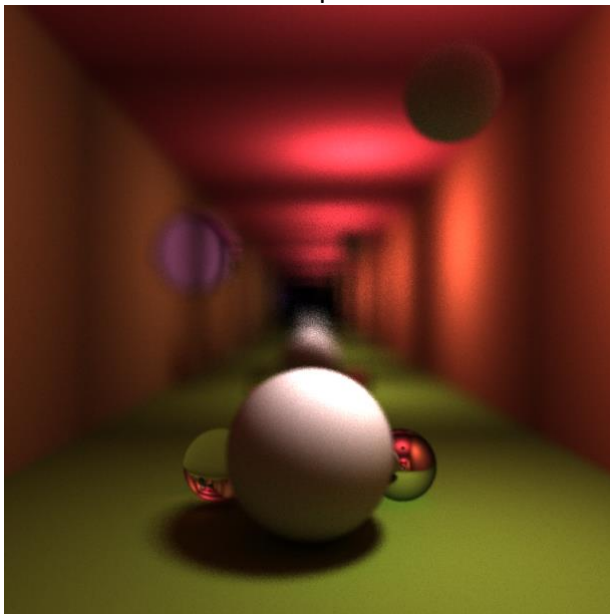
9. Anti-aliasing (image 1024x1024, 100 rayons par pixels, durée d'exécution : environ 4 minutes)



10. Lumières étendues, Ombres douces (image 1024x1024, 100 rayons par pixel, durée d'exécution : environ 5 minutes)

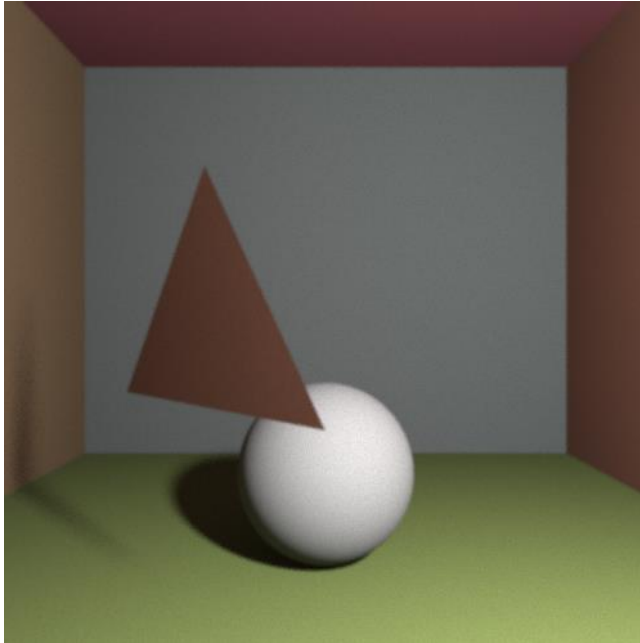


11. Profondeur de champ



12. Ajout d'un triangle (500x500, 100 rayons, 28s d'exécution)

A cette étape, la correction gamma a été ajoutée. Et on arrête de faire apparaître la sphère de lumière.



[Retour sur le cours](#)

L'un de mes meilleurs cours à Centrale : avoir un rendu visuel est très stimulant pour la programmation, on comprend bien ce qu'on fait, le professeur maîtrise très bien son sujet, les explications sont claires et précises.

Ce que j'en ai retiré : la compréhension de ce modèle de rendu réaliste. Je ne sais pas si cela me servira plus tard, mais c'était très intéressant. Et les bases de la programmation en C++.

A améliorer : envoyer un mail avant le premier cours pour dire aux étudiants d'installer un compilateur C++, et d'apprendre les bases du langage et du débogage. On est nombreux à avoir pris très vite beaucoup de retard car nous n'avions jamais fait de C++.