



Informatique Graphique

Rapport Informatique Graphique

Auteur :

M. Rémi GIRARD

Encadrants :

M. Nicolas BONNEEL

Version du
29 mars 2023

Table des matières

1	Introduction	2
2	Résultats en image	3
3	Benchmark Comparaison	7
4	Retour sur le Cours	8

1 Introduction

Dans ce rapport se trouve les résultats que j'ai obtenu dans le cadre du cours d'informatique graphique. Bien que je ne sois pas arrivé au bout du cours, vous trouverez dans ce rapport des images de sphères normales, miroir, transparente et creuse, des implémentations de l'anti-aliasing et de profondeur de champ, ainsi que des visuels de meshes, sans texture cependant.

Le code des différentes séances est sur GitHub, de la 1ère à la sixième, sous les noms test1 à test6.

Au cas où vous trouveriez que mon travail ressemble à d'autres élèves, j'ai surtout travaillé avec Paul Graydon, et dans une moindre mesure avec Antoine Lavandier et Axel Weibert. J'ai échangé des bouts de codes (jamais plus d'une ligne) avec eux quelques rares fois au cours des séances. J'ai surtout été aidé, donc si des lignes de codes sont identiques, c'est sans doute moi qui suis fautif. Je me rappelle avoir donné mon code dans l'une des premières séances à un autre élève (hors des 3 cités précédemment) dont je ne me rappelle cependant plus le nom. C'était censé lui servir de comparaison pour trouver les erreurs dans son programme et d'un moyen de suivre le cours.

2 Résultats en image

Dans cette section, vous trouverez les images montrant les scènes que j'ai réussi à implémenter. Les résultats ont été calculés avec 120 rayons par pixels sur des images de 512*512.

Voici une image montrant des sphères respectivement miroir, creuse et transparente, avec de l'anti-aliasing et de l'éclairage direct :

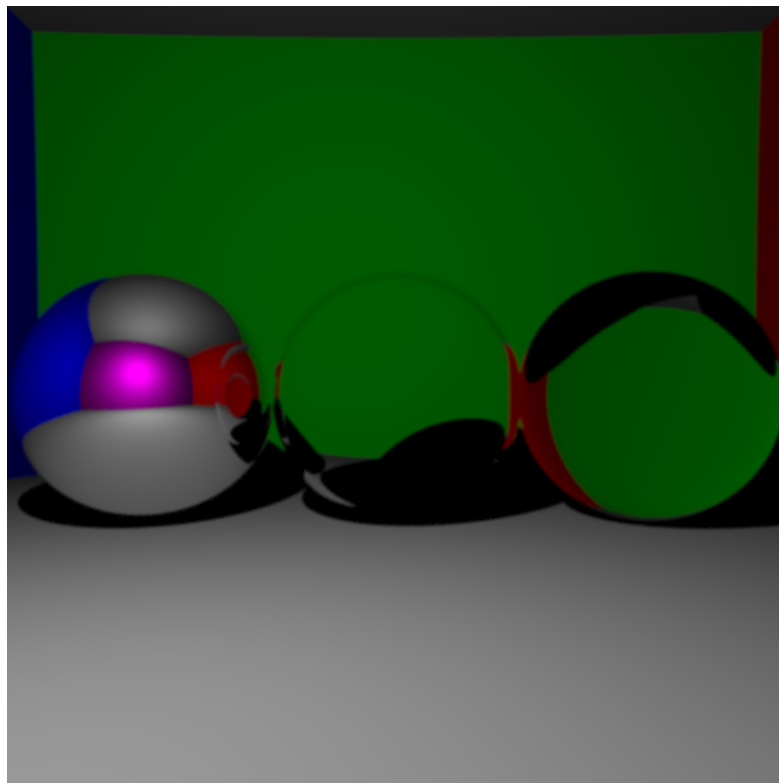


FIGURE 1

Même image, mais avec de l'éclairage indirect ajouté :

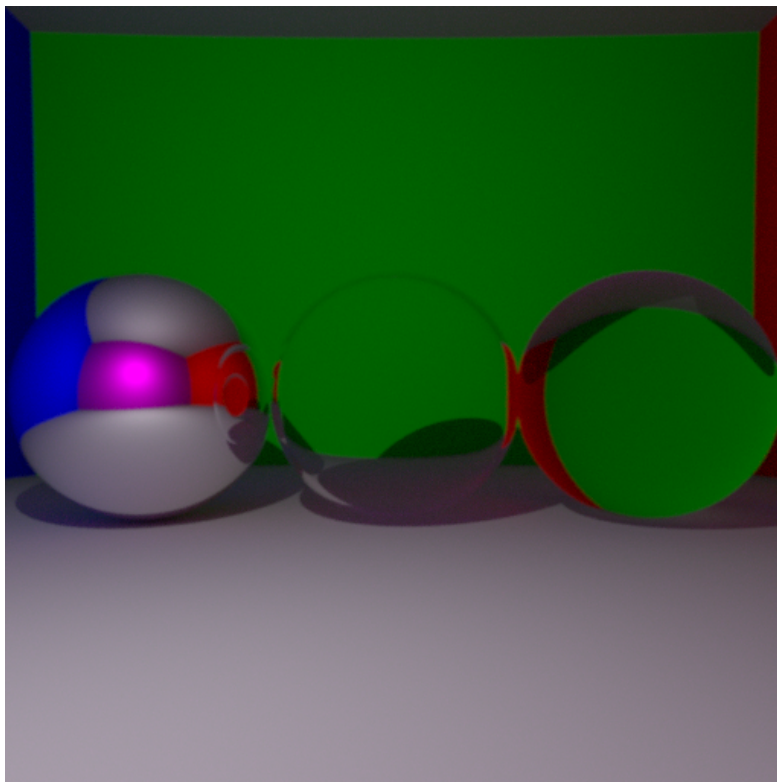


FIGURE 2

Voilà une image représentant le flou induit par la profondeur de champ (anti-aliasing désactivé pour plus de lisibilité), en éclairage par source non ponctuelle :

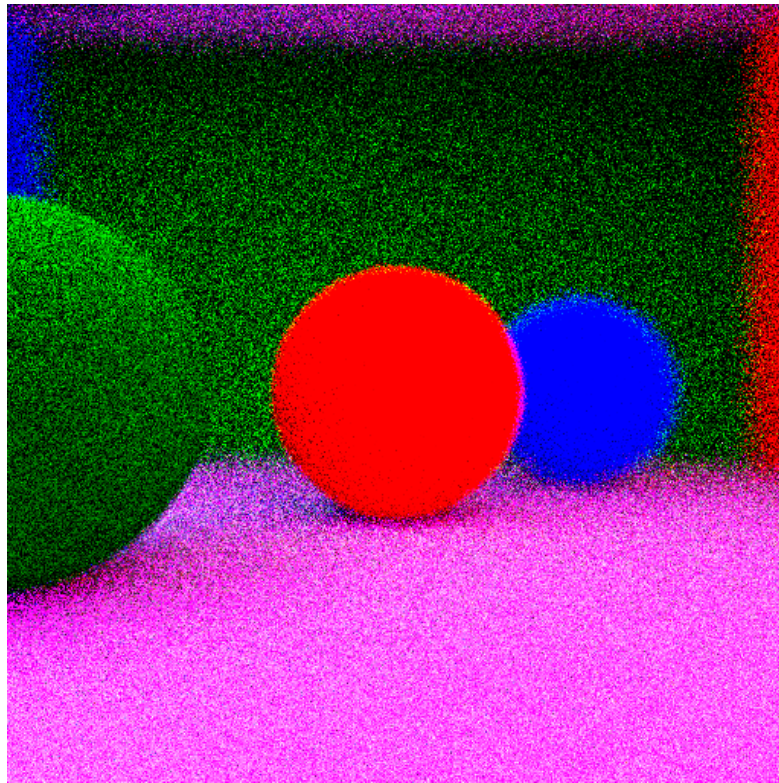


FIGURE 3

Voilà une image de mesh représenté à taille réduite, éclairé par une source non-ponctuelle :

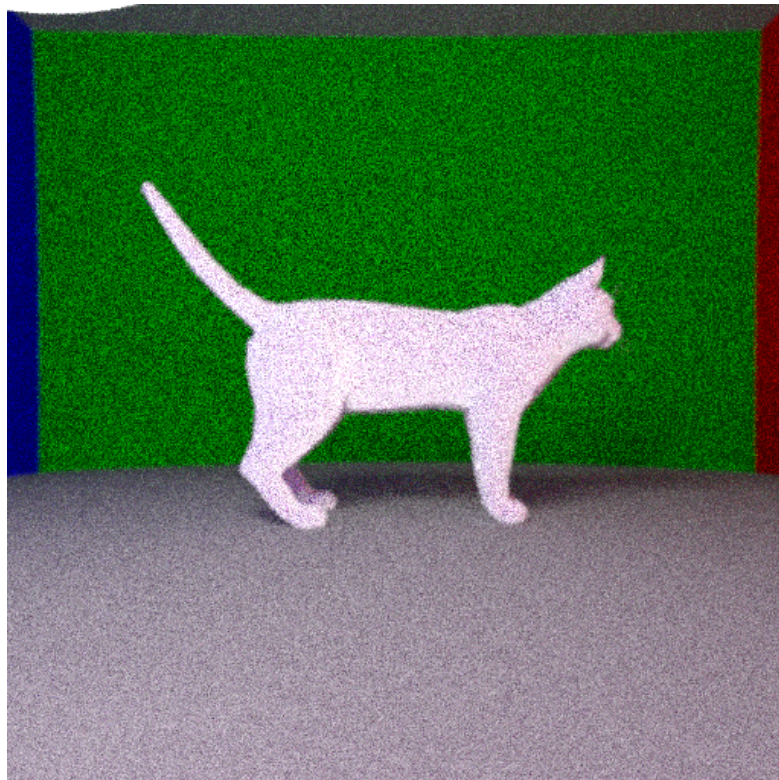


FIGURE 4

3 Benchmark Comparaison

En reprenant l'image du chat, on obtient en temps d'exécution :

Taille	Nombre de rebond	Nombre de rays par pixel	Bounding box	Temps d'exécution
512*512	20	1	Oui	18s
512*512	20	10	Oui	182s
512*512	20	50	Oui	950s
512*512	20	1	Non	127s
512*512	40	1	Oui	30s
128*128	20	1	Oui	1s
256*256	20	1	Oui	5s

Ma bounding box est basique et sans optimisation; c'est une simple boîte, sans l'arbre qui est censé l'améliorer.

4 Retour sur le Cours

Tout d'abord, il s'agit d'un des cours qui m'a le plus plu ce semestre -voire à Centrale-, malgré le fait que je ne soit pas allé jusqu'au bout. Le fait que vous codiez le projet vous-même en cours, et le fait que vous aidiez de façon efficace les élèves en debugging aide beaucoup à suivre le cours et à s'investir.

Si je dois faire une critique au cours, c'est que la charge de travail m'a semblé beaucoup plus lourde aux premiers et second cours qu'au reste. Puisque certains d'entre nous n'ont jamais fait de C++, c'est assez dur de suivre votre rythme au début, et c'est facile d'être perdu. Je pense que vous pourriez peut-être prendre plus de temps sur les premiers cours, décaler la séance d'autonomie vers le milieu-fin du cours, et rattraper l'éventuel retard avant la séance d'autonomie, pour permettre aux élèves de servir de l'autonomie pour finir de coder après une grosse séance.

J'ai trouvé le partiel vraiment bien fait et bien dosé niveau difficulté. La partie savoir représente bien le cours, et la partie debugging était presque ludique.