Projet INFSI 350 - RayMini

Arnaud Douceur
Daniel Ross

Avril 2011

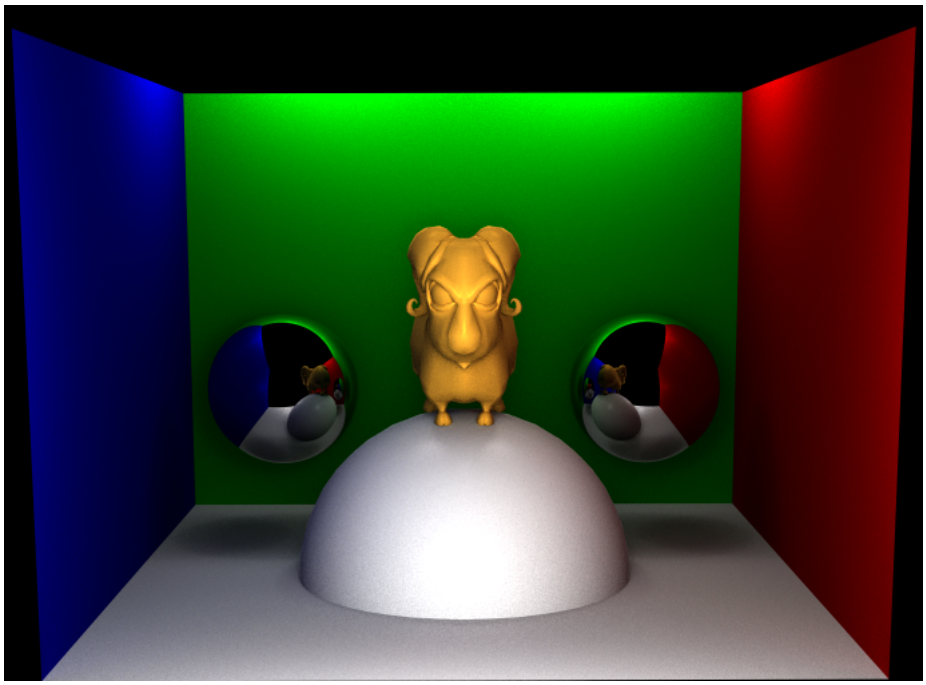

Table des matières

1 Fonctionnalités implémentées

1. Intersection rayon-triangle
2. BRDF de Phong
3. Rendu multi-threads
4. Kd-tree
5. Ombres dures
6. Ombres douces (source de lumière étendu en forme de disque)
7. Ambient occlusion
8. Anti-aliasing
9. Mirroir
10. Profondeur de champ
11. Pathtracing

2 Introduction

Le projet de 3D du module INFSI350 proposait d'implémenter en C++ un moteur de rendu en lancé de rayons (raytracing).

Partant d'une interface graphique en Qt et d'une fenêtre de pré-visualisation avec une scène 3D rendue via OpenGL, le but de ce projet était d'afficher notre propre rendu de la scène.

Pour cela, nous avons dans un premier temps implémenté l'éclairage direct avec des effets tels que la BRDF de Phong, les ombres dures et douces, l'ambient occlusion et l'anti-aliasing. Dans un second temps, nous avons choisi d'implémenter le pathtracing ainsi que des effets supplémentaires comme la profondeur de champ et les matériaux mirroirs.

3 Éclairage Direct

3.1 BRDF de Phong

Afin d'implémenter une BRDF de Phong, nous avons ajouté un attribut *shininess* à la classe Material pour satisfaire l'équation vu en cours :
$f(\omega_i, \omega_o) = k_d (n \wedge \omega_i) + k_s (r \wedge \omega_o)^S$ avec $r = 2n(\omega_i \wedge n) - \omega_i$ et k_d, k_s et S les coefficients diffus, spéculaire et brillance.

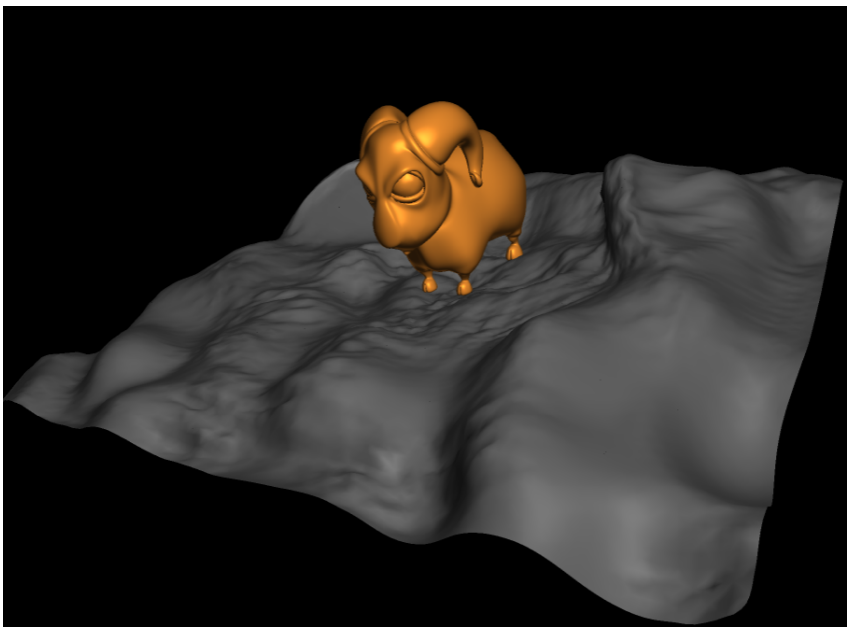

FIG. 1 – BRDF de Phong

3.2 Ombres Dures

À partir d'une source de lumière ponctuelle, nous avons défini une méthode binaire qui colorie les pixels qui sont visibles en effectuant une BRDF de Phong.

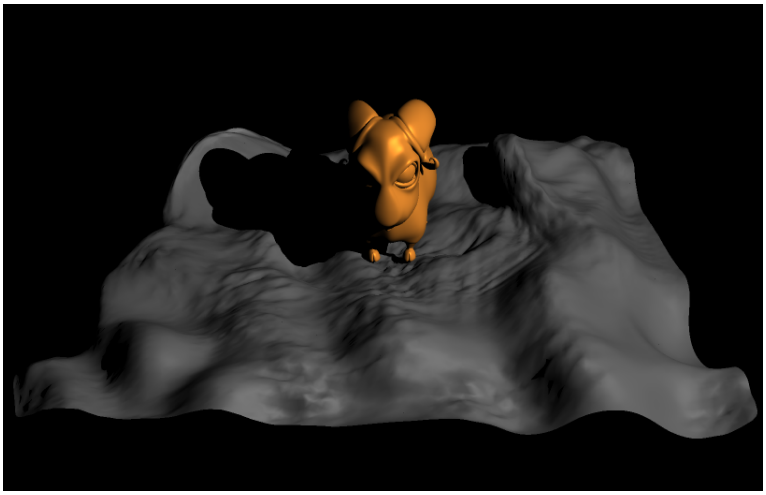

FIG. 2 – Ombres dures

3.3 Ombres Douces

Afin d'obtenir des ombres douces nous avons implémenté une classe AreaLight qui étend la classe Light. La lampe a une forme de disque. Nous simulons ses effets, en tirant aléatoirement N rayons dans la direction de la lumière. La proportion de rayons visible pondère le résultat de la BRDF de Phong afin d'obtenir l'effet escompté.

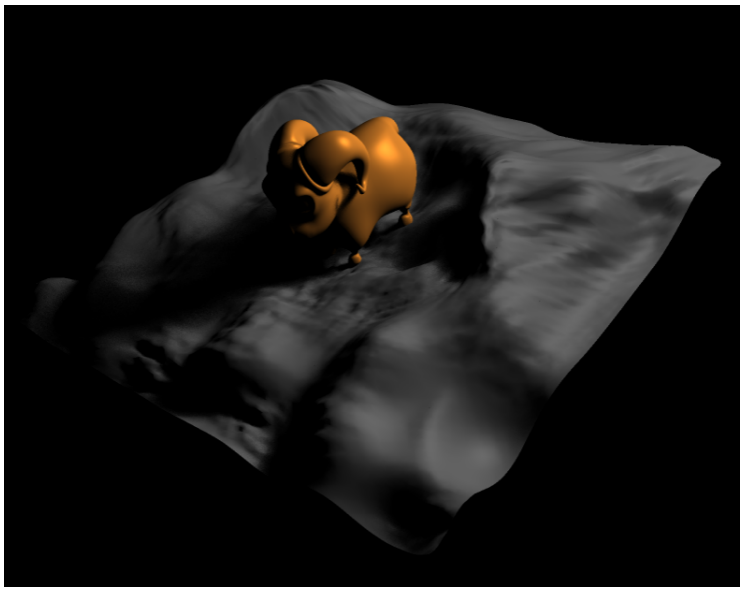

FIG. 3 – Ombres douces

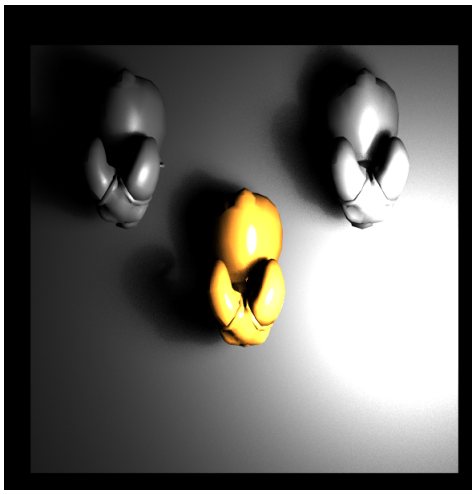

FIG. 4 – Source de lumière étendue

3.4 Ambient Occlusion

L'ambient occlusion (ici sur un modèle rendu sans BRDF de Phong afin de mettre en évidence l'apport de l'ambient occlusion) à été réalisé en considérant la demi sphère, d'un rayon de 5% de la taille de la scène 3D, le long de la normale avec un nombre de rayons N.

FIG. 5 – Ambient Occlusion

3.5 Anti-aliasing

Afin d'éviter le problème classique de crénelage, nous calculons chaque pixel en le découpant selon une grille de sous pixels, puis en moyennant ces derniers.

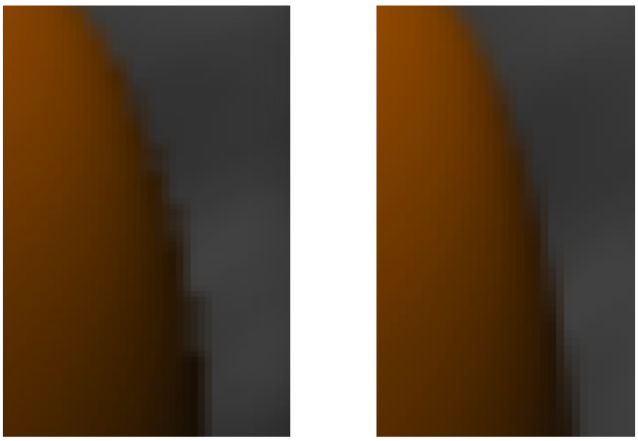

FIG. 6 – Anti-aliasing

3.6 Mirroir

Bien que le projet ne le demandait pas explicitement, nous avons choisi d'ajouter des matériaux mirroirs.

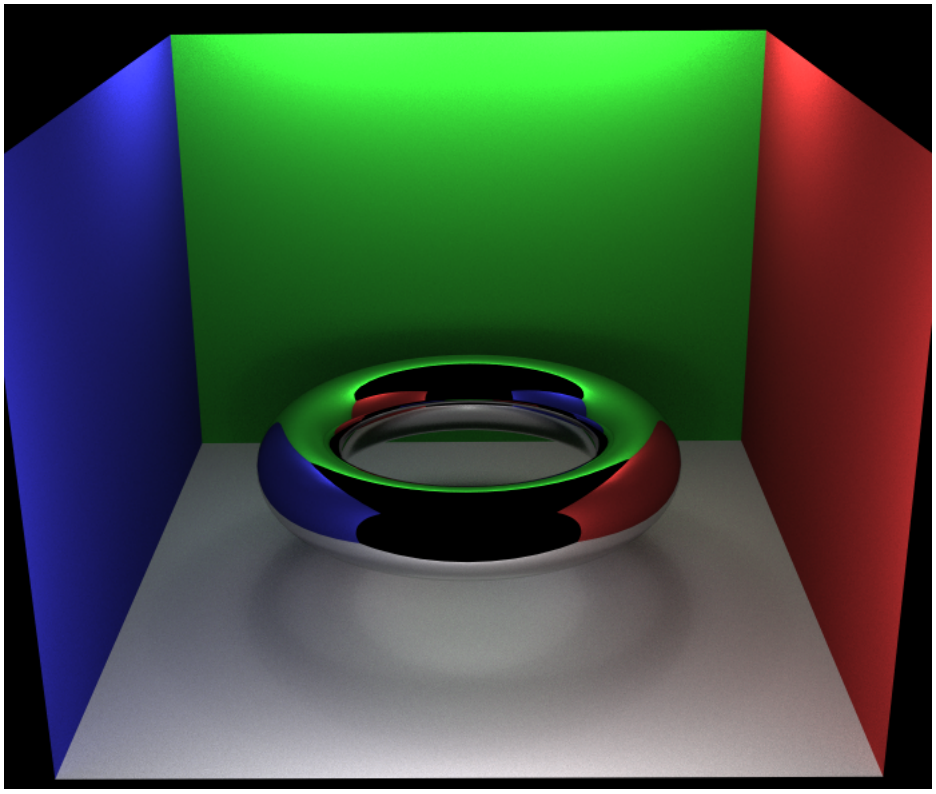

FIG. 7 – Tore Mirroir

3.7 Profondeur de champ

Nous avons également implémenté un effet flou de profondeur.

FIG. 8 – Armée de béliers

4 Éclairage Indirect

Nous avons fait le choix de la méthode pathtracing pour l'éclairage global de la scène.

4.1 Pathtracing

Voici un petit extrait de scènes que nous trouvons élégantes et sympathiques.

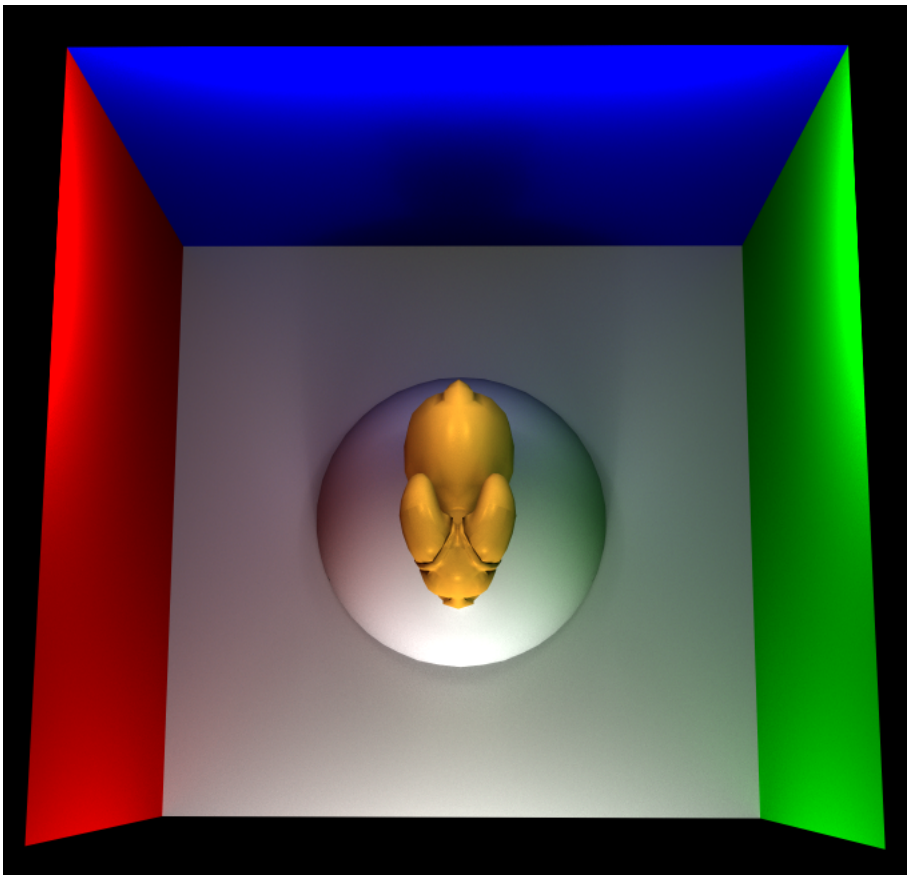

FIG. 9 – Éclairage Global 1

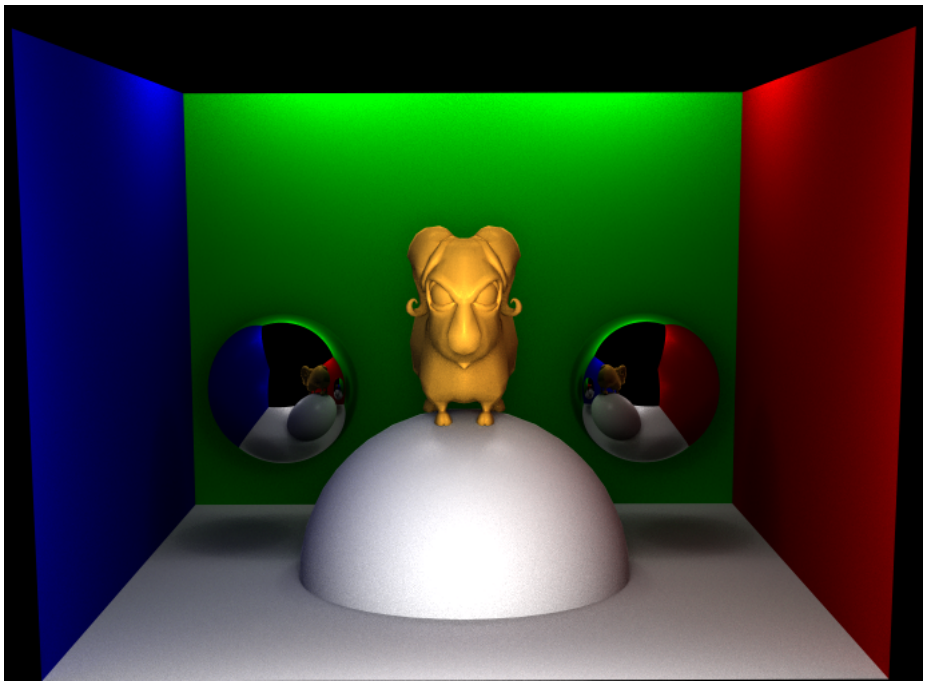

FIG. 10 – Éclairage Global 2

5 Limites

Sans parler des fonctionnalités quasi-infinies que nous pourrions ajouter au moteur de rendu, ce paragraphe liste certains points qui pourraient être améliorés.

- Kd-tree :
  - Actuellement nous utilisons une heuristique naïve pour la construction et la traverse
  - Notre condition d'arrêt n'est pas satisfaisante (i.e : s'arrêter lorsqu'un noeud contient un certain nombre de triangles ou moins) et peut provoquer un bug sur la construction de grosses scènes. Le nombre de triangles par noeud doit être réhausser pour palier à ce problème ; ce qui n'est pas idéal !
  - Nous avons commencer l'implémentation de SAH (Surface Area Heuristic), mais n'avons pas eu le temps de finir proprement.
- Interface graphique
  - La barre de progression du rendu : ne fait pas ce qu'elle est censé faire parce que la classe RayTracer n'est pas un QThread et il faudrait modifier pas mal de choses.
  - Il faudrait ajouter plus d'options pertinentes à l'interface Qt.
- Chargement de fichier
  - Nous ne pouvons charger que des fichiers au format OFF. Il pourrait être intéressant de créer un parseur pour les autres types de fichiers 3D.
  - Les primitives des maillages doivent exclusivement être des triangles.
- Chargement de la scène
  - Le choix de la scène 3D est précompilé.