

Projet INFSI 350 - RayMini

Arnaud Douceur
Daniel Ross

Télécom ParisTech

Avril 2011

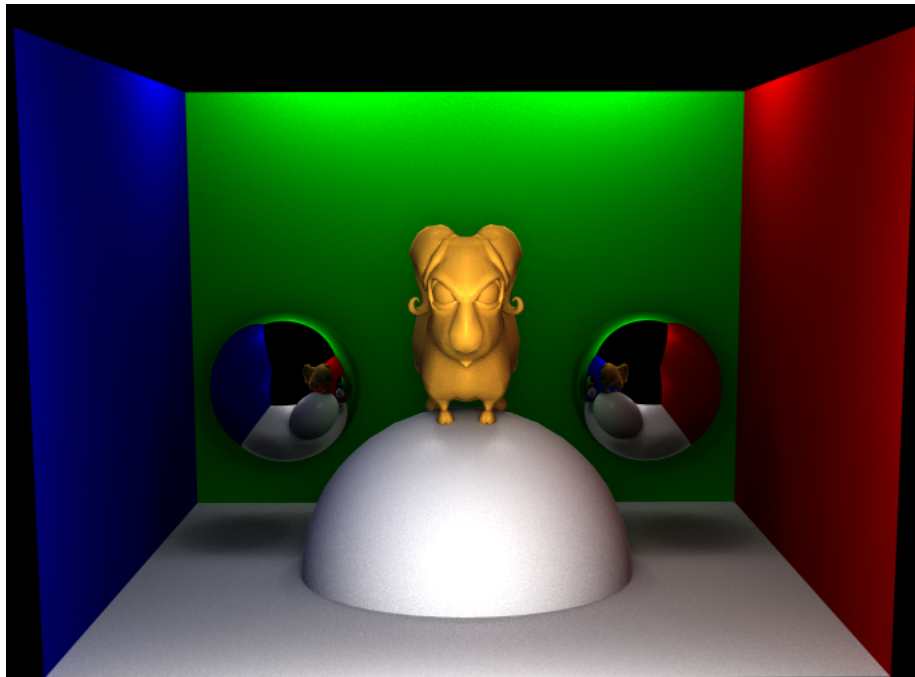


Table des matières

1	Introduction	3
2	Fonctionnalités implémentées	4
3	Éclairage Direct	5
3.1	BRDF de Phong	5
3.2	Ombres Dures	5
3.3	Ombres Douces	6
3.4	Ambient Occlusion	7
3.5	Anti-aliasing	8
4	Éclairage Indirect	9
4.1	Path Tracing	9
4.2	Miroir	10
4.3	Profondeur de champ	11
5	Limites	13

1 Introduction

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

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 path tracing ainsi que des effets supplémentaires comme la profondeur de champ et les matériaux miroirs.

2 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 étendue en forme de disque)
7. Ambient occlusion
8. Anti-aliasing
9. Mirroir
10. Profondeur de champ
11. Path Tracing

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.

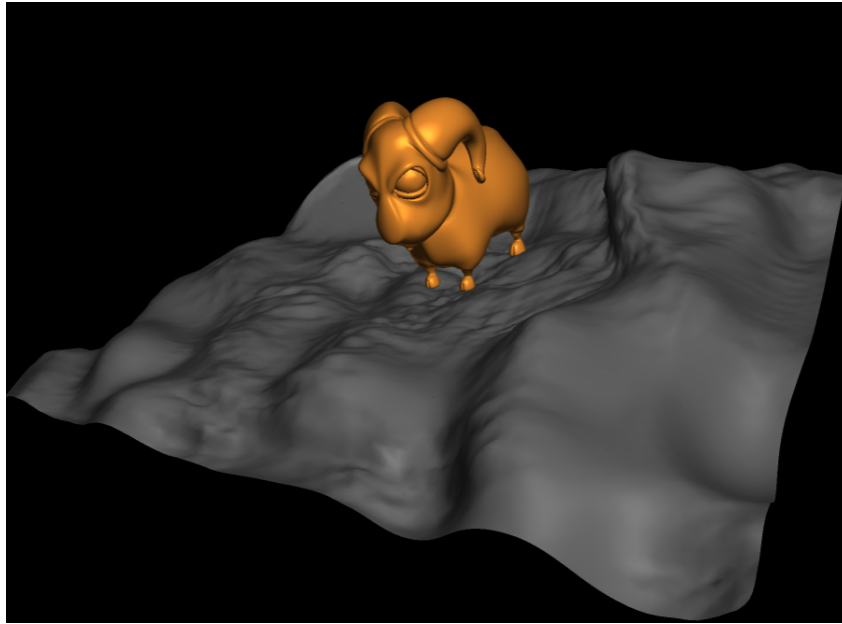


FIG. 1 – BRDF de Phong

3.2 Ombres Dures

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

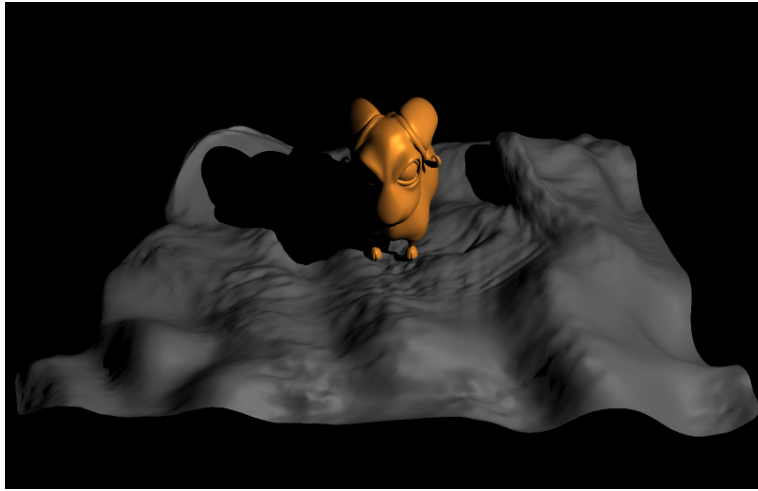


FIG. 2 – Ombres dures

3.3 Ombres Douces

Afin d'obtenir des ombres douces nous avons implémenté une classe Area-Light 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 non obstrués pondère le résultat de la BRDF de Phong afin d'obtenir l'effet escompté.

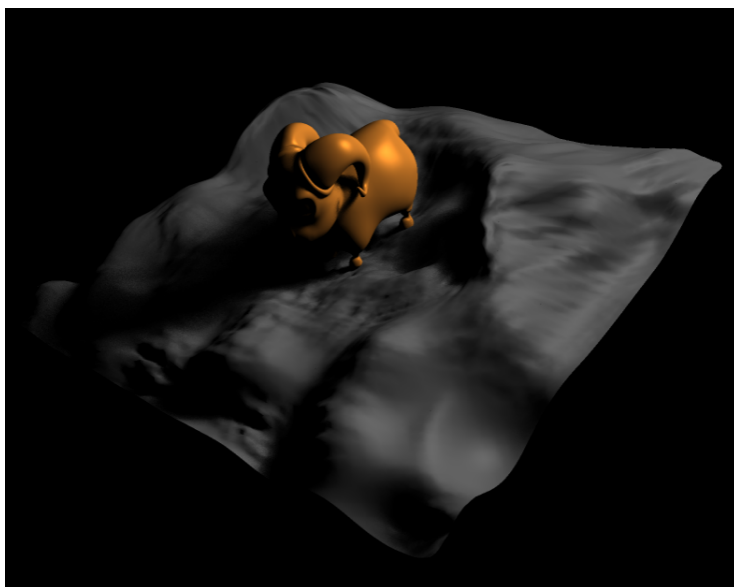


FIG. 3 – Ombres douces

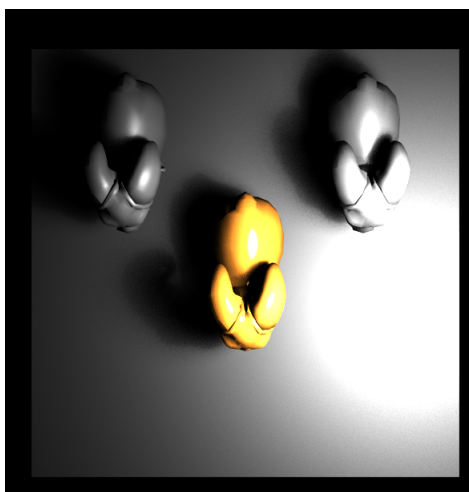


FIG. 4 – Source de lumière étendue

3.4 Ambient Occlusion

L'ambient occlusion (mis en avant sur la Figure 5 sans BRDF de Phong) 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 ($N = 64$).



FIG. 5 – Ambient Occlusion

3.5 Anti-aliasing

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



FIG. 6 – Anti-aliasing

4 Éclairage Indirect

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

4.1 Path Tracing

Voici un petit extrait de scènes que nous trouvons sympathiques.

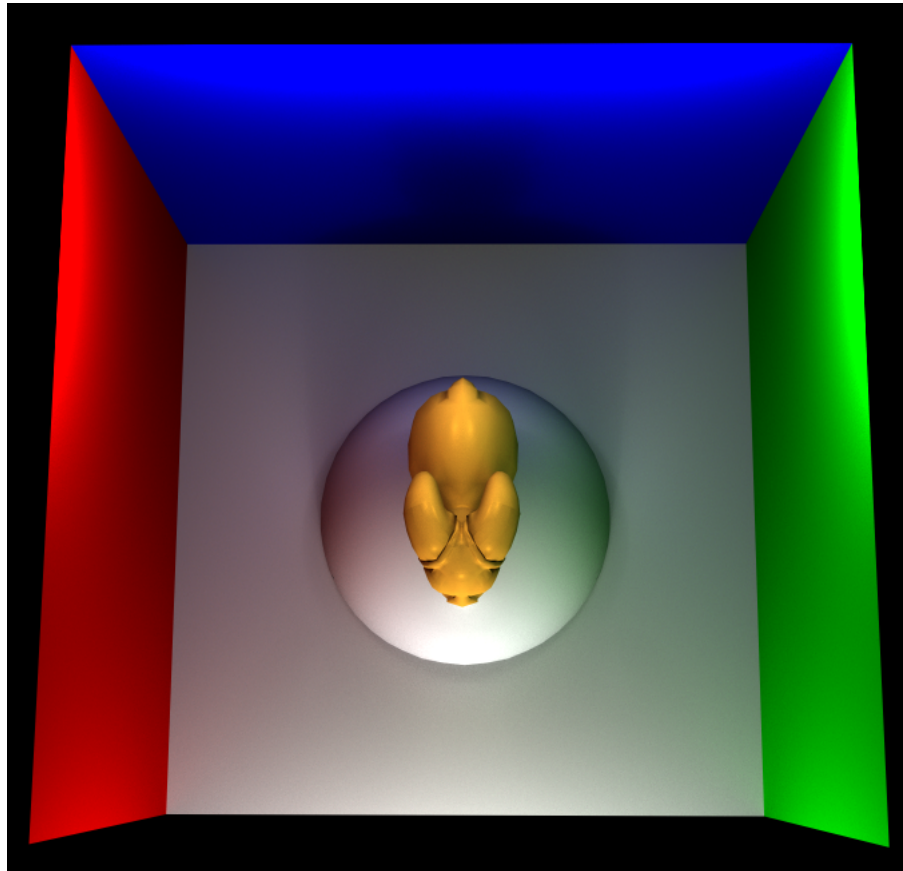


FIG. 7 – Éclairage Global 1

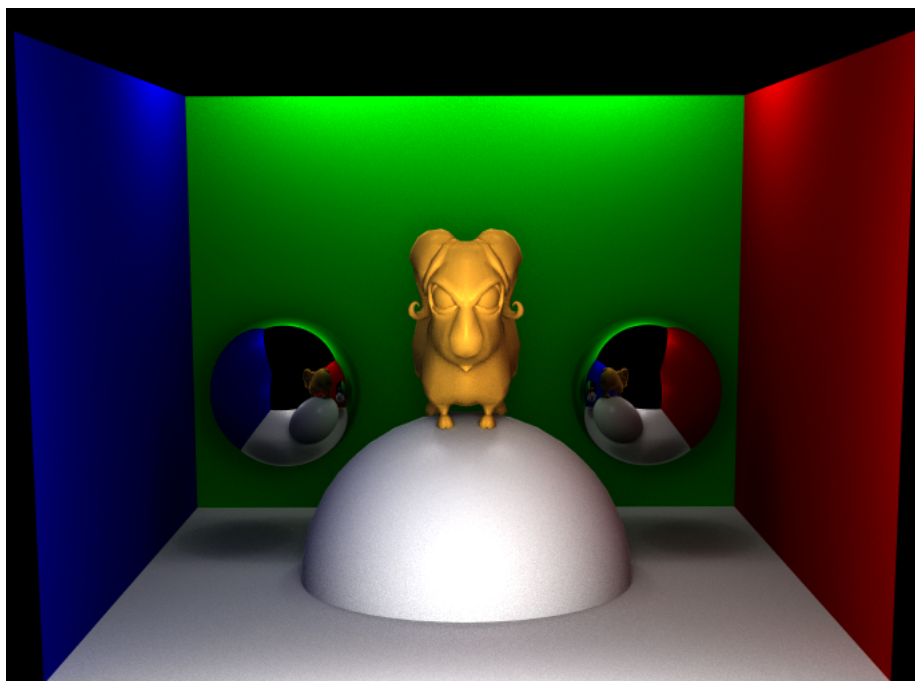


FIG. 8 – Éclairage Global 2

4.2 Mirroir

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

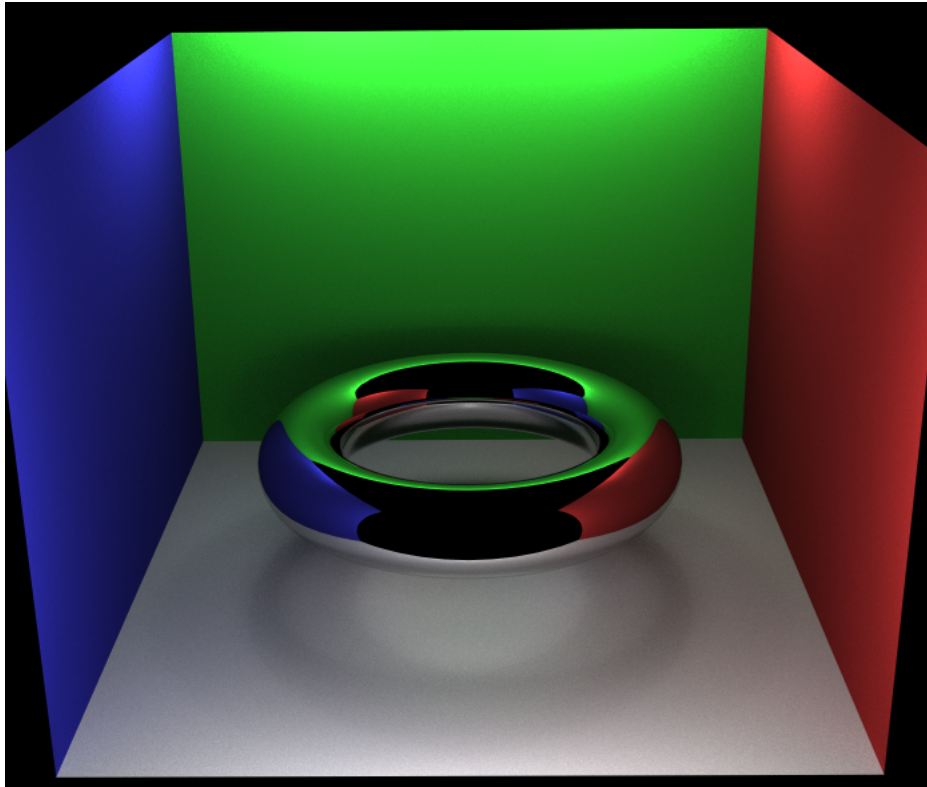


FIG. 9 – Tore Mirroir

4.3 Profondeur de champ

Nous avons également implémenté un effet flou de profondeur.

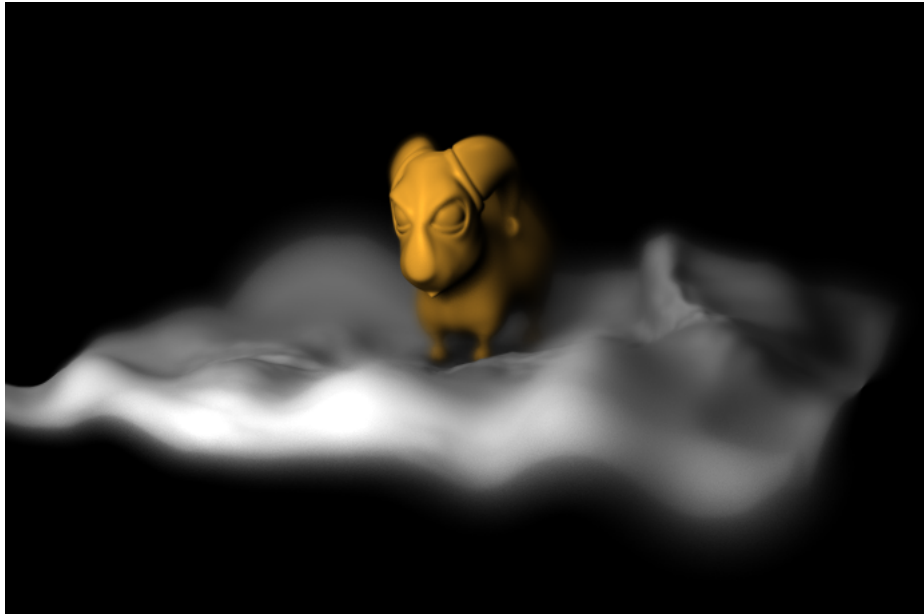


FIG. 10 – Armée de béliers

5 Limites

Sans parler d'une liste de fonctionnalités quasi-infinie que nous pourrions ajouter au moteur de rendu, ce paragraphe liste certains points qui pourraient être améliorés, si nous avons eu plus de temps.

- 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 réhaussé pour palier à ce problème; ce qui n'est pas idéal!
 - Nous avons commencé 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. Notamment le chargement de scènes à la volée ou encore activer des options telles que l'effet de profondeur.
- Chargement de fichier
 - Nous ne pouvons que charger 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. Nous pourrions envisager la gestion des tétraèdres.
- Matériaux
 - Ajouter les matériaux de type verre avec l'implémentation des effets de réfraction.