Université Pierre-et-Marie-Curie

Licence d'informatique

3I008 - Modèles de programmation et interopérabilité des langages

Rapport du Projet

Luong Binh Thanh, 3504859
Kheireddine Anissa, 3502069

1 mai 2017

La commande `make` va produire 3 fichiers exécutables : `raytracer-demo, sceneLoader, kdtree-demo`. Pour quitter le programme, taper `q` sur l'interface graphique.

Dans ce projet, nous considérons la lumière comme une *source lumineuse ponctuelle.*

1 Raytracer-demo

En lançant `./raytracer-demo`, la fenêtre affichera une démonstration du lancer de rayon d'une scène avec comme seuls objets : des sphères.

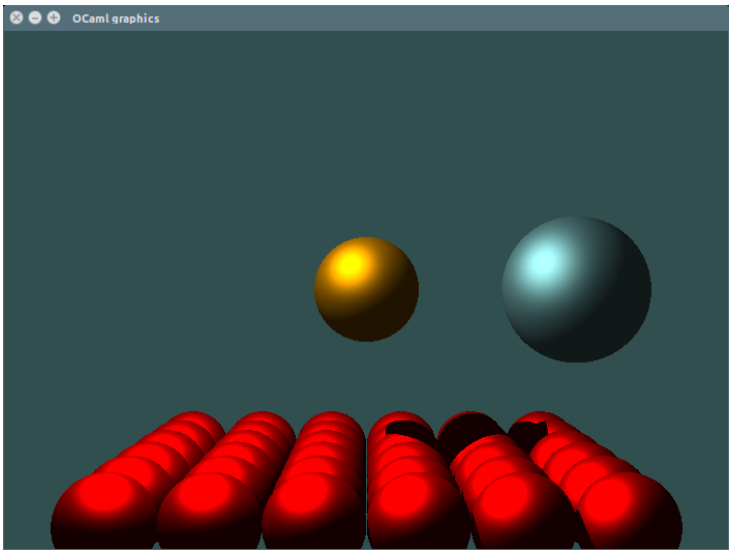

Figure 1 : Scène simple

Les sphères rouges sont crées automatiquement grâce à la fonction `create_list_sphere` donnée par le centre du carré de sphères, les vecteurs directeurs, le rayon et le matériau de chaque sphère.

2 SceneLoader

On peut charger une scène depuis un fichier avec la commande `./sceneLoader 'nom_du_fichier'`. La scène ne contient que des sphères. En remplaçant la fonction `parse_light` en commentaire, on peut avoir une source *lumineuse étendue* à la place de la source ponctuelle.

Voici la scène chargée depuis le fichier `test` donné dans le projet :

camera 0. 0. 0. 0. 0. 5. 2.0943951 700 500
material sphereMat 255 0 0 1. 0.6 0.7 10.
sphere 1. 1. 5. 0.4 sphereMat
light -10. -30. 20. 255 255 255 0.1

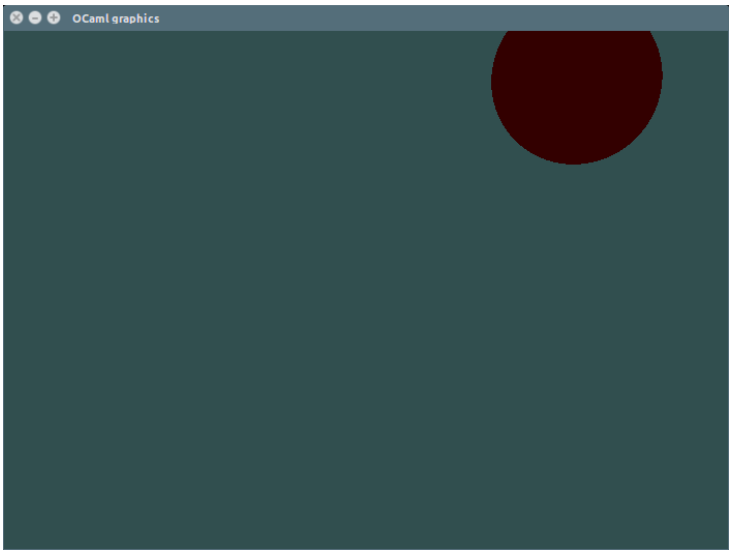

Figure 2 : Load scène

On peut aussi sauvegarder une scène dans un fichier avec la fonction `save_scene`.

3 Kdtree-demo

Dans cette partie on introduit de nouveaux objets : triangle et plan. On a encapsulé tous les objets considérés dans le type `objet` avec le patron de conception *(design pattern)*.

Voici un exemple :

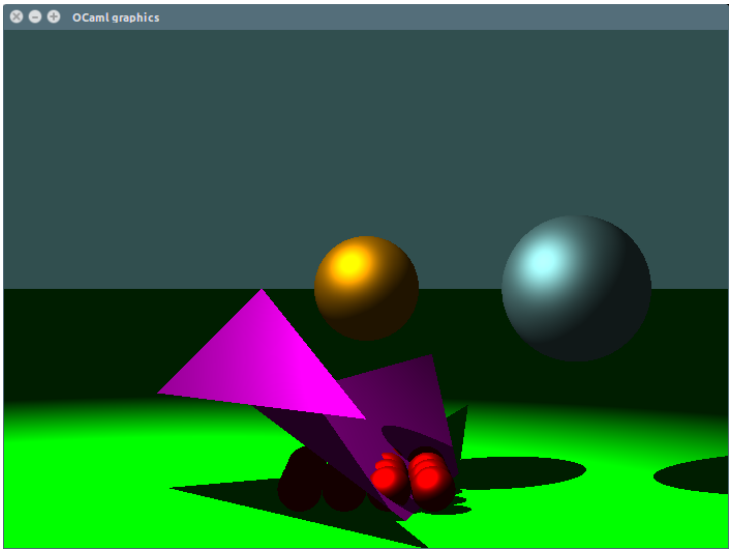

Figure 3 : Scène complexe

La complexité de la scène (nombre d'objets) rend le temps d'exécution plus long. De ce fait, on introduit dans cette partie, une nouvelle structure de données *kd-tree*, dans le but de diminuer le temps de calcul en factoriser les calculs inutiles (*par exemple : lancer de rayon sur une partie ou il n'y a aucun objet*) .

Comme on l'a vu, le nombre d'objets est important, pour une scène qui a moins de points d'intersections le calcul est rapide. En revanche, le rendu d'une scène complexe est lent.

En réalité, cette structure de données n'est pas très efficace car pour chaque rayon sortant de la caméra, on parcourt tout l'arbre pour trouver l'intersection avec l'objet, puis le reparcourir encore une fois afin de vérifier si l'objet est illuminé par la source lumineuse.

Le parcours du kd-tree (à n nœuds) a une complexité de l'ordre de O(n log n). Ce qui nous donne, pour chaque pixel : O(n^2 $(logn)^2$).

De plus, pour un arbre d'une profondeur pas assez grande, les calculs sont mauvais. Prenons un exemple, pour `t = 10` on obtient :

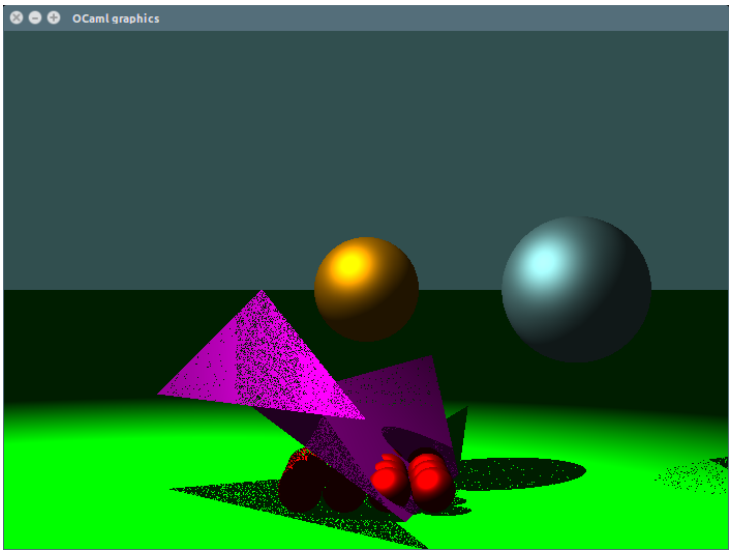

Figure 4 : Hauteur de l'arbre, un paramètre important dans la structure

Ce phénomène se produit lors de la procédure, on découpe la boîte contenant les objets plusieurs fois, ce qui réduit énormément la précision du calcul. Nous avons constaté aussi que pour une profondeur k assez grande (`k = 100`), le rendu est très lent.

4 Comparaison de temps calcul

4.1 Profondeur de l'arbre

Voici quelques exécution du lancer de rayon avec le kd-tree en variant la profondeur de l'arbre :

Structure	Temps d'exécution
Liste	1.675041
Kd-tree de hauteur 1	2.592809
Kd-tree de hauteur 4	9.000466
Kd-tree de hauteur 10	13.078327
Kd-tree de hauteur 20	13.239288
Kd-tree de hauteur 100	15.115611

L'augmentation de la profondeur du kd-tree est proportionnelle au temps d'exécution.

4.2 Objets de la scène

On essaie maintenant de comparer la stratégie avec et sans la structure kd-tree en variant le nombre d'objet dans la scène :

Sans kd-tree :

Nombre d'objets	Temps d'exécution
2 objets	0.900142
5 objets	2.004059
7 objets	2.330014
38 objets	3.504864

Avec kd-tree de hauteur 3 :

Nombre d'objets	Temps d'exécution
2 objets	1.266533
5 objets	2.040127
7 objets	2.396312
38 objets	4.492200

Conclusion :

On peut conclure que la méthode de lancer de rayon simple est plus performante en terme de temps de calcul et de précision qu'avec le kd-tree.