

Rapport IGTAI

Charlie Grand

Mars 2021

Voici le rapport demandé venant détailler la réalisation du TP de l'UE IGTAI. Le code source de ce TP se trouve dans le dossier raytracer/ et les images générées durant ce TP dans le dossier rendus/.

1 Progression et Difficultés

Les principales difficultés que j'ai rencontré durant ce TP sont liées au calcul de normale d'un triangle. Même avec l'aide de l'enseignante je n'ai pas pu le mettre en place et je l'ai remplacé par l'utilisation des normales stockées dans le fichier .obj.

Il y a eu quelques ralentissements dus à la mauvaise compréhension du sujet et à la confusion possible causée par les noms de certains arguments de fonction, mais rien de remarquable.

2 Système de maillage

Après avoir implémenter les fonctionnalités demandées dans le sujet de TP, je me suis mis en tête d'avoir un système de maillage afin de pouvoir rendre des objets plus complexes.

Cet ajout est donc composé de six classes et d'une fonction. Cinq d'entre elles sont destinées à représenter le maillage d'un objet. Deux de ces classes (MeshEdge et MeshHalfEdge) ne sont pas vraiment utilisées dans ce TP mais peuvent être nécessaires pour d'autres rendus et pour avoir une représentation complète du maillage. La dernière classe sert à lire un fichier .obj utilisé pour stocker un modèle 3D et compatible avec la plupart des logiciels 3D, afin d'importer un modèle dans le raytracer.

Les modèles 3D étant représentés sous forme de polygones dans l'espace, il me fallait une fonction pour tester l'intersection d'un rayon avec ces polygones. J'ai donc une fonction qui converti ces polygones en triangles à la création de l'objet et une fonction intersectTriangle(). Cette fonction, afin d'obtenir la normale d'un triangle, utilise la valeur stockée dans le fichier .obj ce qui simplifie les calculs.

Voici quelques explications afin d'ajouter un modèle 3D dans une scène du raytracer.

Le modèle est représenté par un objet de type Mesh qui prend un argument un matériel. Afin d'importer un fichier .obj, un objet de type Importer doit être déclaré et un Mesh doit lui être fourni en paramètre. La fonction loadObjFile() de l'objet Importer doit être appelée sur le chemin vers le .obj. Enfin la fonction addMesh() prenant un argument une Scene et un Mesh sert à ajouter l'objet 3D à la scène.

```
Scene *scene = initScene();  
//on ajoute aussi une camera et des lumieres  
Material mat;  
//on regle le material comme on le souhaite  
  
Mesh *mesh = new Mesh(mat);  
Importer *importer = new Importer(mesh);  
importer->loadObjFile("mesh/mesh.obj");  
addMesh(scene, *mesh);
```

La classe Importer est un peu stricte et les fichiers .obj doivent inclure les coordonnées de normales mais pas les UV.

Une scène d'exemple utilisant un maillage est dans le code. Il est possible de la rendre en tapant après avoir compilé le projet :

```
make mrt  
./mrt image 5
```

Le programme peut mettre un peu de temps pour importer le fichier .obj et le calcul de l'image dépend évidemment du nombre de triangles de l'objet.