

# INGI2325 : Computer Graphics

## Readme

### 2008 - 2009

Dessart Charles-Eric, SINF21

March 13, 2009

## 1 Exécuter le projet

Les sources du projet se trouvent dans le répertoire *src/*.

Pour exécuter le projet, il faut lancer le fichier java */bin/Test.class* à partir du répertoire racine du projet. Le programme prend des arguments en ligne de commande. Tous les arguments sont obligatoires et doivent être définis dans l'ordre. Soit:

- **sdl\_file:** Chemin du fichier SDL contenant la scène.
- **width:** Longueur de la fenêtre où sera dessiné la scène.
- **height:** Hauteur de la fenêtre où sera dessiné la scène.
- **cr:** Valeur de réflectance des matériaux. La valeur doit être comprise entre 0 et 1.
- **technique:** Technique utilisée pour dessiner la scène. Valeurs: *rasterization* ou *raytracing*.
- **save\_image:** Sauver l'image. L'image est sauvée à la racine du projet. Valeurs: *true* ou *false*.

**Attention:** Sur des machines plus lentes tels que les ordinateurs de la salle Intel, il est important de laisser quelques secondes entre la création de la fenêtre et le début la rasterisation/raytracing. Ce problème n'est pas spécifique à ce projet.

Exemple:

- Ouverture de la console
- Aller dans le répertoire racine du projet
- Taper: `java -classpath bin Test XML/test.sdl 512 512 0.8 rasterization true`
- Attendre 5 secondes avant de voir apparaitre les premiers résultats (voir paragraphe **Attention**)

## 2 Description du projet

La première partie de ce projet permet d’afficher à l’écran une scène dont le contenu est défini dans un fichier SDL soit par rasterisation, soit par raytracing. Le programme affiche une seule forme, le scenegraph n’étant pas encore implémenté. Il est possible de disposer plusieurs points de lumière dans la scène. La lumière est diffusée selon la loi de Lambert. Le projet a été testé en salle Intel avec succès avec la VM par défaut de java (version 1.6). Le projet est aussi compatible avec la version 1.5.

### 2.1 Difficultés rencontrées

- Se replonger dans le calcul matriciel.
- Définir les points l b n r t pour le raytracing. J’ai eu difficile de comprendre l’utilité du field of view.
- Gérer les conversions float-int notamment pour la rasterisation.
- Gérer l’influence d’un point de lumière sur la couleur d’un pixel.
- Trouver le vecteur normal d’un point en interpolant les vecteurs normaux des sommets pour le raytracing.

### 2.2 Choix faits

- La structure des scènes est la même que celle que l’on retrouve dans les fichiers SDL pour une question de facilité de chargement.
- Le programme n’utilise pas des processus concurrents pour effectuer la rasterisation ou le raytracing car cela posera peut être problème par la suite. Je ne connais pas encore assez le sujet de la 2ème partie du projet.
- Les points l b n r t f pour la rasterisation sont définis en fonction de la position de la caméra. Je ne suis pas arrivé à obtenir le même effet de distance pour la rasterisation et le raytracing pour une même position de la caméra.
- Les formes définies dans les fichiers SDL sont chargées d’un fichier OBJ puis modifiées selon les paramètres. Cela est facile et peu couteux en ressources.
- L’axe vertical des cones et des cylindres est l’axe des Y et non des Z car dans les fichiers OBJ, c’est l’axe des Y qui est utilisé.

### 2.3 Propriétés supplémentaires du projet

- Il est possible de placer plusieurs points de lumière dans la scène.
- Un type de forme géométrique a été ajouté à la sémantique des fichiers SDL. La forme *OBJModel* dont l’attribut *model* définit un fichier OBJ à charger. Il est possible par exemple de définir un tricératops dans le fichier SDL (voir XML/test.sdl).