**Rapport du projet de synthèse d’image  
RAYTRACING**

Binôme Pape NDIAYE

Kevin MAUGE kmauge@etud.u-pem.fr

Pape NDIAYE pndiaye@etud.u-pem.fr

### Configuration globale du projet

Pour démarrer le projet il faut procéder de cette manière :

* Niveau 1 : lray -n 1 -i <mon\_fichier.format> -o image.bmp
  + i fichier d’entrée à parser
  + o fichier de sortie (Notre programme à la faculté de génerer des fichiers au format bmp)
* Niveau 2 : lray -n 2 -p 16 -i <mon\_fichier.format> -o image.bmp
  + p nombre de rayon par pixel
* Niveau 3 : lray -n 3 -i <mon\_fichier.format>

Format du fichier d’entrée:

**N:P:C:S**

* N = nom de la forme
* P = propriétés (séparé par une virgule)
  + Camera x**,**y**,**z
  + Rectangle x,y,z pour le centre:Longueur:Largeur
    - Exemple : **rectangle:0,0,0:34:20:5:0.3**
  + Sphère x,y,z pour le centre, rayon
    - Exemple : sphere:**0,0,0:30:5:0.7**
  + Triangle Ax1,Ay1,Az1, Bx2,By2B,Bz2,Cx3,Cy3,Cz3
    - Exemple : **triangle:1,2,0,3,2,0,4,0,3:3:0.7**
  + Cylindre x1,y1,z1 pour le sommet ,x2,y2,z2 pour la base:rayon
    - Exemple : **cylindre:10,0,5,10,50,5:50:2:0.6**
* C = entier de 1 à 7 pour la couleur dans l’ordre suivant : *white\_light, maroon, gold, gray, darkturquoise, crimson, firebrick, seagreen*
* S = valeur comprise entre 0 et 1 pour la composante spéciale de la couleur (luminescence)

Un exemple de fichier d’entrée est accessible via text/figure.txt

AMELIORATIONS (Extensions) effectuées:

* Calcul de l’ombrage
* Fichier image de sortie en bmp
* Texture sur des objets générés grâce à un algorithme
* Plusieurs  sources de lumière
* Mise en place d’un système d’intensité de lumière ambiante

# out.jpgscene (copy).jpgNIVEAU 1 :

# Niveau 2: Première partie : Génération automatique, procédurale d’objet

Figure 1 Avant ajout améliorations

Figure 2 Après ajout améliorations

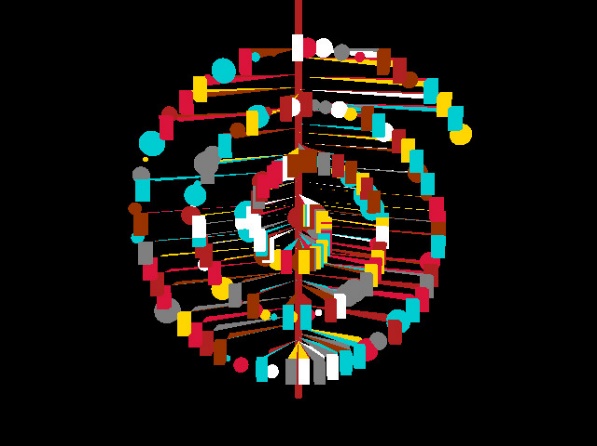
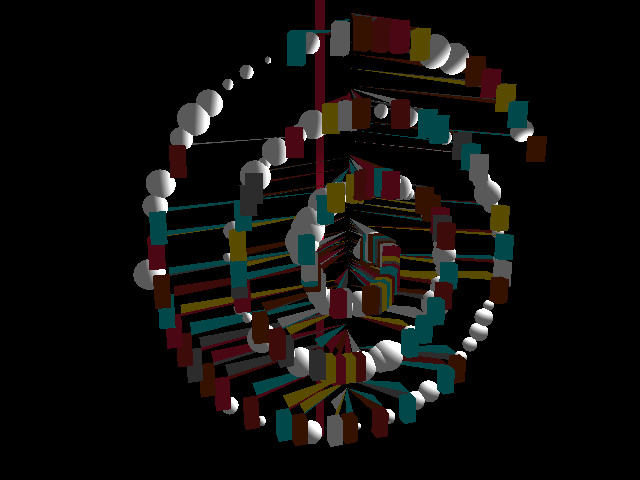
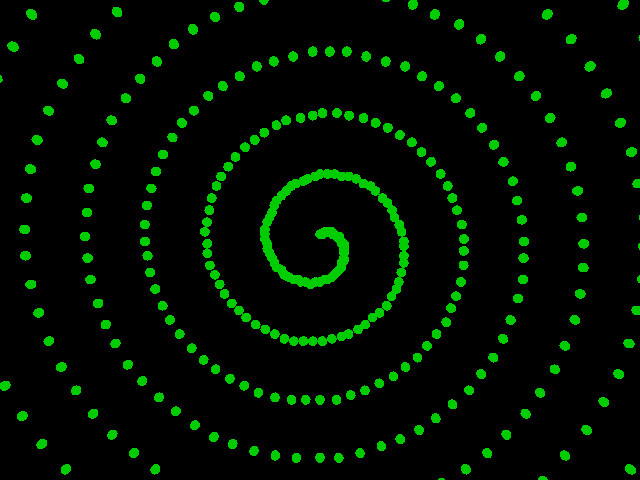
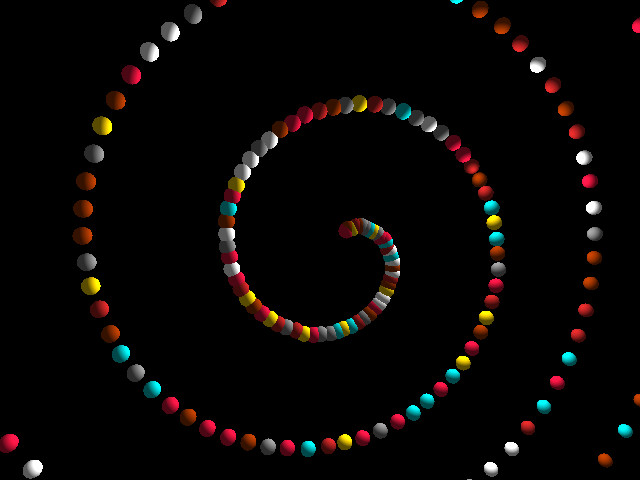
Génération automatique d’une spirale de 300 sphères   
Génération d’un ensemble composé de 1000 objets canoniques à l’aide de la suite de fibonacci:

Figure 3 Après ajout améliorations

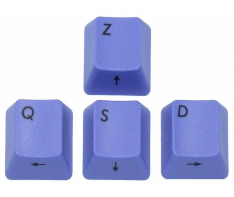
Figure 4 Avant ajout améliorations

Figure 5 Avant ajout améliorations

Figure 6 Après ajout améliorations

Deuxième partie : Remplissage de la structure de données

Nous avons la possibilité de calculer actuellement les sphères englobantes de tous les objets canoniques mais il reste actuellement à remplir en utilisant l’algorithme



# Niveau 3:

* On peut manipuler la caméra directement en utilisant les touches ou en utilisant les défileurs prévus à cet effet
* Appuyer sur la touche “i” nous permets d’enregistrer l’image courante dans la racine du projet sous le format out.bmp
* On peut modifier la lumière ambiante de la scène en déplaçant le défileur “am”
* On peut augmenter le nombre de lumières présentes dans la scène avec la molette “ps”

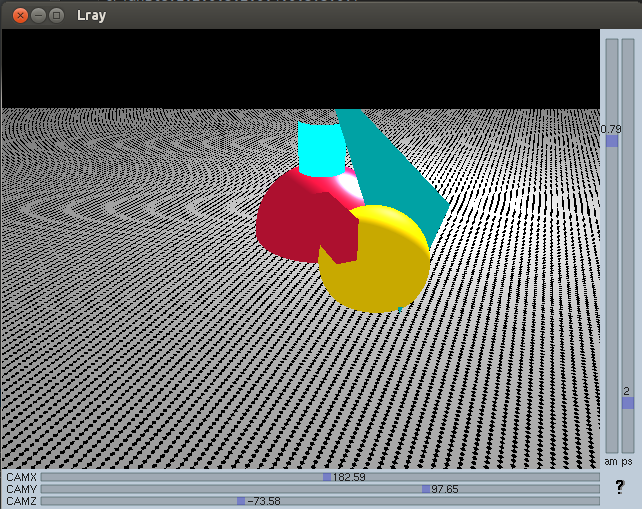
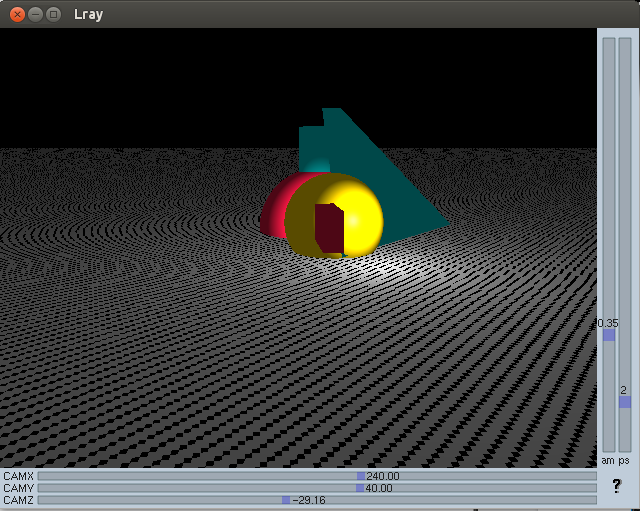
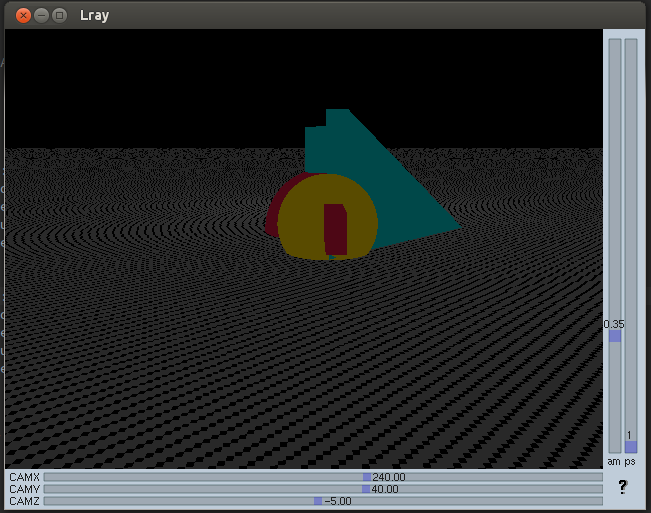


Figure 7 Fenêtre OpenGL avec augmentation progressif de la luminosité et déplacement de la caméra

# Perspectives :

* Utilisation des sphères englobantes (déjà codé) pour remplir l’arbre hiérarchique en utilisant l’algorithme
* Faire l’antialiasing
* Implémenter la transparence
* Implémenter les reflets