

Cazin Florent  
Readme projet Rasterizer M2 ILJ

### **Implémenté:**

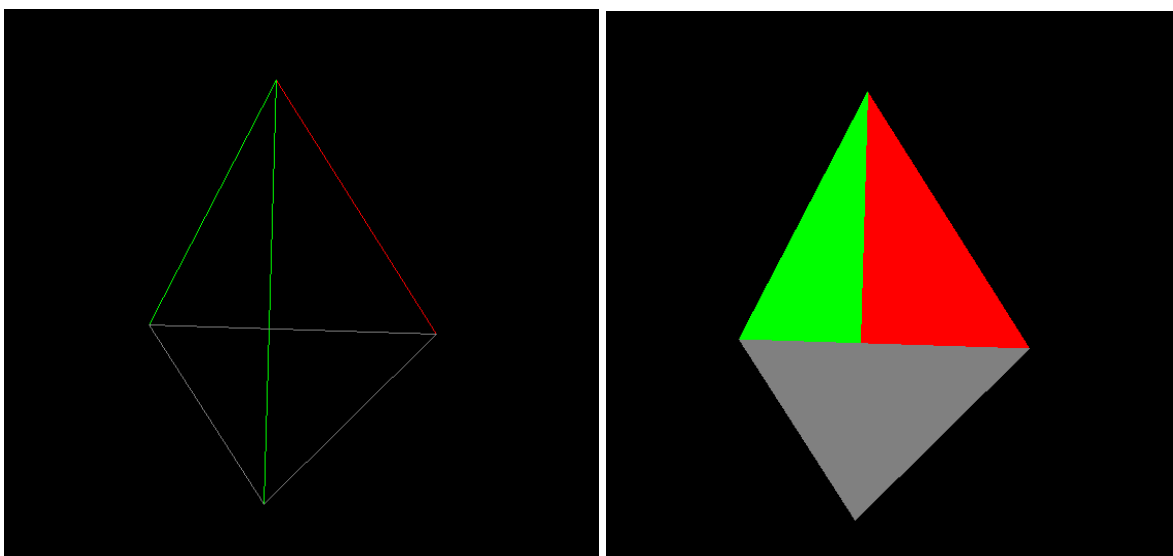
Aline library  
Lines and tris  
Perspective proj  
--  
face.h  
matrix.h  
object.h  
object.cpp  
scene.h  
scene.cpp  
shape.h  
vector.h  
vertex.h  
--

**Résumé:** Chargement d'un fichier ou de plusieurs fichiers .obj pour afficher un objet 3D (lignes/remplissage).

Utilisation des flèches pour pivoter à gauche, à droite, vers le haut et vers le bas.

### **Soucis que j'ai en ma connaissance:**

-Lorsque l'on remplit (ou "filled") un objet, certaines faces qui devraient être cachées par d'autres ne le sont pas, et sont donc affichées par-dessus (exemple sur la photo 2).

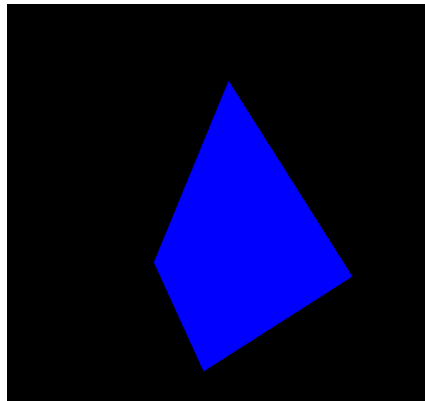
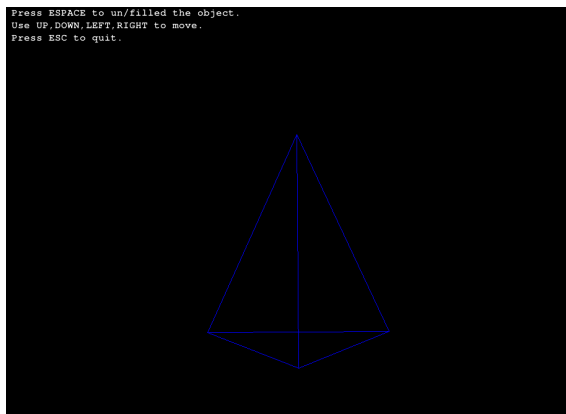


Si jamais vous souhaitez rendre un objet "plus visible", si je peux m'exprimer ainsi, lorsque vous le remplissez, vous pouvez simplement supprimer la couleur à la ligne

432. En effet, un constructeur avec une couleur par défaut (bleu) a été réalisé, mais je n'ai pas pensé à cela immédiatement et donc je n'ai pas eu le temps de mettre en place la possibilité de changer directement la couleur depuis le programme.

```
432 Face face(i1-1,i2-1,i3-1, colorArray[ite++]);
```

```
432 Face face([i1-1,i2-1,i3-1]);
```



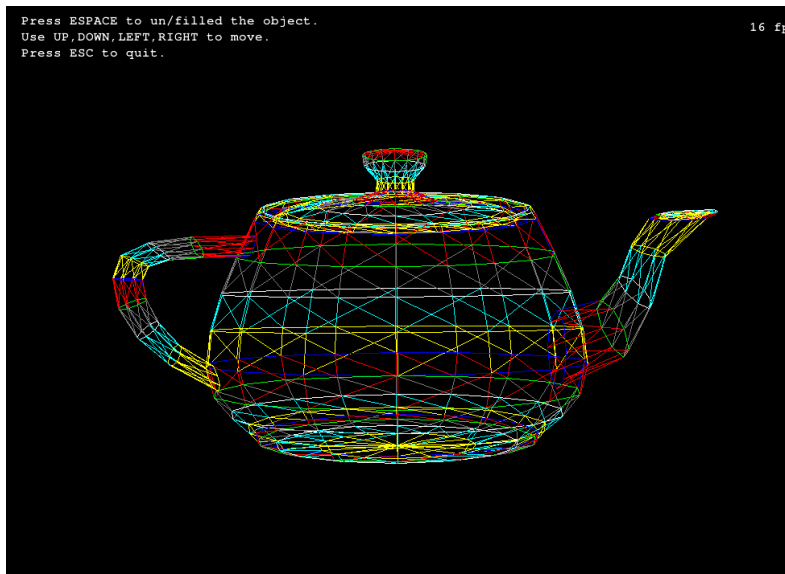
Cela permet de moins perturber l'affichage car on ne distingue plus les faces arrière. Cela aurait pu être résolu avec la dernière feuille de route (hidden surface removal, etc.), mais cela n'a pas été implémenté.

-Lorsqu'un objet se trouve derrière la caméra, cela provoque un crash du programme. Il est parfois nécessaire d'ajuster manuellement le vecteur de translation, en fonction de l'objet chargé, afin d'éviter d'avoir l'objet derrière la caméra ou trop près/loin. Cette modification doit être effectuée dans le fichier scene.cpp, à la ligne 437.

```
437 Object object(shape,Vec3r{0.0,0.0,-8.0},Vec3r{0.0,0.0,3.15},Vec3r{1.0,1.0,1.0});
```

J'ai mis la valeur à -8 ici, cela convient pour les formes basiques (triangle, carré, etc.). Pour le fichier teapot.obj, mon programme crash car l'objet est derrière. Cependant, lorsque je mets la valeur à -250 afin que l'objet ne soit pas derrière, ça fonctionne. Ainsi, selon l'objet que vous voulez charger, il est nécessaire d'essayer manuellement de trouver une distance convenable afin de l'afficher correctement sans provoquer de crash.

```
437 Object object(shape,Vec3r{0.0,0.0,-250.0},Vec3r{0.0,0.0,3.15},Vec3r{1.0,1.0,1.0});
```



-Je n'ai pas implémenté de tests de mon côté.

(quand on exécute le programme sans obj, cela lance simplement le programme avec 2 objets 2D prédéfinies, cela m'avait servi lors des tests donc je l'ai laissé.)

**//! Mon programme est donc fourni avec une distance de -8 par défaut, il faut donc modifier cela si nécessaire.**

**-Après chaque modification, il suffit de faire un make.**

**-Pour exécuter mon programme, il faut faire:**

**export LD\_LIBRARY\_PATH=\${LD\_LIBRARY\_PATH}:/bin**

**puis:**

**./scene obj1.obj obj2.obj ...**

**J'ai voulu faire un script mais cela ne fonctionne pas, donc on est obligé ici de faire ces lignes à la main.**

**Si un obj.obj ne s'ouvre pas correctement ou une typo etc, une erreur dans la console va s'afficher mais le programme ne va logiquement pas crash.**