



Travaux dirigés d'*OpenGL* n°1

Synthèse d'image 2

— IMAC deuxième année —

Introduction à *OpenGL*

Ce TD a pour objectif de vous faire réviser certaines notions de géométrie et vous familiariser avec les fonction de base de l'*OpenGL*.

Durée de la séance : 2h.

Petits rappels avant de commencer

- Comment compiler mon programme écrit en C et utilisant *OpenGL*, *GLUT* et *GLU* ?
`gcc -Wall -I/usr/X11R6/include/ -L/usr/X11R6/lib/ -lglut -lGLU -lGL prog.c`
- Quels sont les entête à ajouter à mon programme ?
`GL/glut.h` et `GL/glu.h` (les entêtes *OpenGL* sont déjà incluse dans `glut` mais au cas où `GL/glu.h`).
- On utilise quel repère ?
Le repère *OpenGL* c'est à dire l'axe x dirigé vers la droite, l'axe y vers le haut et l'axe z pointant vers l'écran.
- Monsieur ? c'est quoi les paramètres de cette fonction *OpenGL/GLUT/GLU* ?
RTFM.
- C'est quoi *OpenGL*, *GLUT*, *GLU*, RTFM... ?
Google est ton ami !

► Exercice 1. Quelques transformations...

- Commencez par rappeler ce que donne la transformation d'un point de l'espace \mathbb{R}^3 définit par (x, y, z) en coordonnées homogènes. Exprimez les coordonnées homogènes suivantes (x, y, z, w) en coordonnées cartésiennes.
- Rappelez la forme des matrices homogènes permettant d'effectuer les transformations suivantes :
 - la translation suivant le vecteur (x, y, z) ;
 - l'homothétie dont les valeurs sont (a, b, c) ;
 - la rotation autour de l'axe z d'un angle de α degrés.

► **Exercice 2. *Glut ? Mais on connaît déjà !***

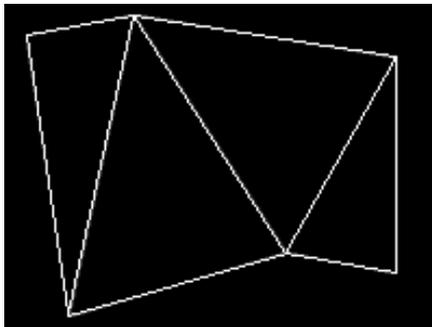
Le but de cet exercice est de vous rememorer le mode de fonctionnement de *GLUT*. A cette fin vous allez télécharger le petit programme se trouvant sur http://www-igm.univ-mlv.fr/~fdesorbi/web/ensfiles/si_td01_0.c.

Une fois le fichier récupéré, corrigez les quelques erreurs ou oublis volontaires qui s'y trouve (5 au total).

► **Exercice 3. *La primitive attitude en projection !***

Commencez par rappeler la différence entre la projection en perspective et la projection orthographique.

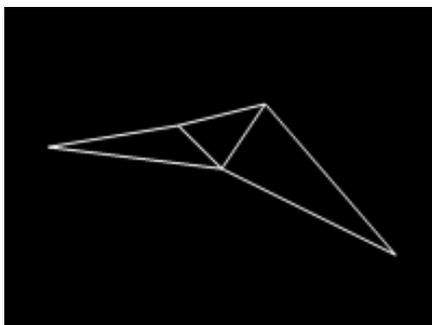
Pour chacun de ces ensembles de points (ils ne sont pas ordonnés) reproduire la figure associée en utilisant la bonne primitive en paramètre de la fonction *glBegin()*, la projection orthographique et le rendu en fil de fer (fonction *glPolygonMode()*) :



$\{-0.9f, 0.6f, 0.0f\}$
 $\{ 0.3f, -0.4f, 0.0f\}$
 $\{-0.4f, 0.7f, 0.0f\}$
 $\{-0.7f, -0.7f, 0.0f\}$
 $\{ 0.8f, 0.5f, 0.0f\}$
 $\{ 0.8f, -0.5f, 0.0f\}$



$\{ 0.8f, 0.6f, 0.0f\}$
 $\{ 0.8f, -0.6f, 0.0f\}$
 $\{-0.8f, 0.6f, 0.0f\}$
 $\{-0.8f, -0.6f, 0.0f\}$
 $\{ 0.8f, 0.0f, 0.0f\}$
 $\{-0.8f, 0.0f, 0.0f\}$



$\{-0.8f, 0.1f, 0.0f\}$
 $\{ 0.2f, 0.3f, 0.0f\}$
 $\{ 0.0f, 0.0f, 0.0f\}$
 $\{ 0.8f, -0.4f, 0.0f\}$
 $\{-0.2f, 0.2f, 0.0f\}$

Essayez de reproduire les deux images A et B en tenant compte de la fonction d'affichage suivante :

```
1 void drawFunc()
2 {
3   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
4   glMatrixMode(GL_MODELVIEW);
5   glLoadIdentity();
6   glTranslatef(-0.4f, -0.4f, -3.0f);
7   glutSolidTeapot(0.5f);
8   glTranslatef(1.0f, 1.0f, -15.0f);
9   glutSolidSphere(0.5f, 30, 30);
10  glutSwapBuffers();
11 }
```

Angle d'ouverture vertical du volume de vision : 45 degrés
Distance du plan near : 0.1
Distance du plan far : 10.0
Fonctions à utiliser : `glOrtho()` et `gluPerspective()`



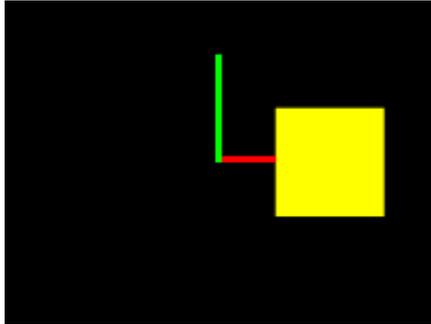
A



B

► **Exercice 4.** *OpenGL ça vous transforme la vie*

- Dessinez un repère dans votre scène et positionnez le point de vue en $(0.0, 0.0, 4.0)$. L'axe x sera rouge, l'axe y vert et l'axe z bleu.
- Dessinez un cube unitaire dont chaque face a une couleur différente et dont le centre est positionné en $(1.0, 0.0, 0.0)$. vous devez alors obtenir ceci :



- Appliquez séparément et sur l'ensemble de votre scène les transformations suivantes :
 - une rotation de 45 degrés autour de l'axe y ;
 - une rotation de 90 degrés autour de l'axe z ;
 - une translation de vecteur $(-1.0, 1.0, 0.0)$;
 - une translation de vecteur $(0.0, 0.0, -10.0)$;
 - une homothétie de valeurs $(2.0, 2.0, 2.0)$;
 - une homothétie de valeurs $(0.5, 1.0, 0.5)$.
- Donnez deux méthodes pour appliquer une rotation de 90 degrés autour de l'axe des x (indice : utilisez la fonction `glMultMatrix()`).

La cuisine de la semaine !

Retrouvez chaque semaine cette petite rubrique de l'infographie qui vous permettra d'améliorer ou diversifier vos connaissances dans le domaine.

Pour commencer simplement, un peu de publicité pour cette magnifique page :

<http://www-igm.univ-mlv.fr/~fdesorbi/web/openglut>

Et quelques liens qui vous seront très utiles ici :

<http://www-igm.univ-mlv.fr/~fdesorbi/web/liensinfo.html>