

## IMN428 – Infographie

### Travail Pratique 2

## Objectifs

1. Utiliser quelques fonctions de base d'OpenGL
2. Utiliser les fonctions d'OpenGL liées aux transformations affines 3D.
3. Utiliser les piles de transformations

## Description

À l'aide du code C++ fourni (fichier tp2.cpp) vous devrez implanter les transformations affines suivantes : la translation, la rotation, le changement d'échelle et le cisaillement. Les transformations seront appliquées à un cube, un cône et un système d'axes. Les endroits où vous devez ajouter du code sont clairement indiqués par l'étiquette «AJOUTER DU CODE ICI!». Vous devez également fournir le code pour dessiner les trois axes du système. Chaque axe doit être une ligne OpenGL **d'épaisseur 4 et de longueur 20**, munie d'une pointe (flèche). Cette pointe est un tétraèdre et doit être créée à l'aide de la fonction *DrawTetrahedron()* fournie avec le code. L'axe X doit être **rouge**, l'axe Y **vert** et l'axe Z **bleu**. Il est à noter que la caméra est centrée à l'origine du monde (0,0,0) et que le système d'axes est situé à l'origine de l'objet sélectionné (cône ou cube).

## Description de l'interface

L'interface actuelle est très simple. Le bouton de droite permet de sélectionner le mode de transformation. Une fois le mode sélectionné, la transformation peut être appliquée en enfonçant le bouton de gauche et en déplaçant la souris verticalement (avant-arrière). Le terme "Rotation X" signifie une rotation autour de l'axe X.

Vous devez implanter les transformations suivantes : translation selon l'axe X, Y et Z, rotation autour de l'axe X, Y et Z, changement d'échelle en X, Y et Z, changement d'échelle uniforme (même facteur pour X, Y et Z) ainsi que le cisaillement suivant l'axe X, Y et Z. Vous devez également ajouter la possibilité de faire bouger indépendamment le cube et le cône (*Move Cube Vs Move Cone*).

Pour implanter les transformations, vous utiliserez les fonctions d'OpenGL suivante :

1. `glTranslate`
2. `glRotate`
3. `glScale`
4. `glMultMatrix`

À noter que la position et l'orientation des deux objets sont respectivement contenues dans les variables globales **cubeModelviewMatrix** et **coneModelviewMatrix**. Ces variables contiennent l'ensemble des transformations ayant été appliquées aux deux objets. Les fonctions OpenGL

1. `glMatrixMode`
2. `glPushMatrix`
3. `glPopMatrix`

vous permettront de sauvegarder et restaurer la matrice de transformation. Elles vous permettront également de manipuler la pile des matrices pour afficher les axes. De plus, les fonctions **glGetFloatv** et **glLoadMatrixf** vous permettront de manipuler le contenu de la matrice de transformation associée au cube (`cubeModelviewMatrix`) et au cône (`coneModelviewMatrix`). Finalement, les fonctions :

1. `glColor`
2. `glLineWidth`
3. `glVertex`
4. `glutPostRedisplay`
5. `glutSolidCone`

vous seront aussi très utiles pour ce travail. N'oubliez pas d'inscrire vos noms en commentaire au début du fichier `.cpp` remis !

## Évaluation

Ce travail doit être fait par **équipe de TROIS ou QUATRE**. Utilisez la commande **turnin** pour soumettre votre travail. La remise doit être faite au plus tard le **10 février 2012**. Au moment de soumettre votre travail, assurez-vous que votre code compile bien sous *Visual Studio*.