Université de Sherbrooke
Département d'informatique
Hiver 2014

IMN428 – Infographie

Travail pratique 2

Objectifs

1. Introduction à la modélisation
2. Utiliser les fonctions d'OpenGL pour gérer l'illumination locale.
3. Gérer plus d'une fenêtre sous OpenGL.

Description

On appelle communément «*modeler* 3D» le programme que vous serez appelé à compléter pour ce travail. L'objectif d'un tel programme est de fournir une interface simple pour créer des formes géométriques complexes.

Pour ce faire, vous devez implémenter deux fonctions fréquemment utilisées à savoir les **surfaces de révolution** (*surfaces of revolution*) et les **surfaces de translation** (*sweeping surfaces*). Pour chacune de ces fonctions, l'objet 3D est généré à partir d'une silhouette 2D créée par l'utilisateur. Pour générer une surface de révolution, le programme prend la silhouette et la fait tourner autour d'un axe (ici l'axe y). C'est la forme géométrique résultant de cette rotation qui donne l'objet 3D. En ce qui concerne les surfaces de translation, elles sont obtenues en faisant «glisser»la silhouette le long d'un chemin unidimensionnel. Pour ce travail, vous devez la faire glisser le long de l'axe Z (voir tp3Solution.exe pour un exemple).

Pour faire ce travail, partant des fichiers fournis (tp3.cpp et structures.h) vous devez ajouter du code aux endroits indiqués par l'étiquette **«AJOUTER DU CODE ICI !»** .

Description de l'interface

L'interface actuelle possède deux fenêtres. Une fenêtre qu'on appelle le «modeler» et une seconde qu'on appelle le «viewer». La première fenêtre sert à créer et/ou modifier la silhouette 2D. Les points de contrôle de la silhouette sont contenus dans le tableau global : **point2D silhouettePointArray[NB_MAX_POINTS]**. Vous devez ajouter le code afin que les points de contrôle apparaissent de couleur verte et de grosseur 6 et que les segments de droite apparaissent de couleur rouge et d'épaisseur 5. La caméra de cette fenêtre fonctionne en mode *orthographique*.

Note : la silhouette et l'axe central sont affichés à l'aide de la commande **glVertex2f(...)** et que les points de contrôle peuvent être déplacés à l'aide du bouton du milieu de la souris.

En ce qui concerne la fenêtre *viewer*, elle contient par défaut un **plancher** et **deux lumières**. L'utilisateur peut cependant afficher un autre objet à l'aide du menu déroulant. Ces objets, l'utilisateur peut les visualiser dans tous les sens à l'aide du bouton de gauche de la souris. La caméra

du *viewer* fonctionne en mode *perspective* et ses propriétés sont contenues dans la variable globale **gCam**.

Note : on peut modifier ces propriétés à l'aide des touches b,B,c,C,e et E du clavier.

Les objets 3D

Le plan

Le plan est rendu à l'aide de la fonction **drawPlane(int n)** qui affiche un plan de $n \times n$ *quads* (polygone à quatre sommets). Le plan est à la hauteur Y = -100 et s'étend de -100 à 100 le long des axes X et Z. Sa normale doit être colinénaire à l'axe Y. La résolution du plan peut être modifiée à l'aide des touches "a" et "A" du clavier.

Les deux lumières

Les deux lumieres sont des sphères centrées à la position **gLights[...].position** et qu'on affiche à l'aide de la fonction **glutSolidSphere(...)**. Vous devez afficher chaque lumière en utilisant sa composante diffuse comme couleur.

Les autres objets

Les objets modélisés peuvent être rendu sous la forme d'une surface de révolution ou d'une surface de translation dépendant du mode sélectionné. Le nombre de polygones avec lesquels vous afficherez les objets est fixé par la variable **objectResolution** (et contrôlée par les touches "a" et "A" du clavier). Vous devez aussi permettre de rendre le modèle en mode *wireframe* (fil de fer) à l'aide de la touche "w" du clavier. À noter que les surfaces de translation doivent avoir une profondeur de 400 (de Z = -200 à Z = +200).

Pour ce qui est du *teapot*, il est fortement recommandé d'utiliser la fonction **glutSolidTeapot**.

L'illumination locale

OpenGL peut calculer la contribution de chaque lumière sur une scène. Toutefois, pour y arriver, certains paramètres doivent être correctement initialisés. Lorsqu'un programme OpenGL est lancé, ses calculs d'illumination sont par défaut désactivés. Pour les activer, la fonction **glEnable(GL_LIGHTING)** doit être appelée. Une fois l'illumination activée, OpenGL donne accès à 8 lumières identifiées par **GL_LIGHT0, GL_LIGHT1, ... , GL_LIGHT7**. Chacune de ces lumières peut être allumée ou éteinte à l'aide des fonctions **glEnable(GL_LIGHTx) et glDisable(GL_LIGHTx)** (pour ce travail, deux lumières sont utilisées). Tous les paramètres des lumières peuvent être modifiés à l'aide de la fonction **glLight()** dont voici deux variantes :

```
void glLightf( GLenum light, GLenum pname, GLfloat param )
void glLightfv( GLenum light, GLenum pname, GLfloat* param )
    - light: l'identificateur de lumière (GL_LIGHT0, GL_LIGHT1, ...)
    - pname: le paramètre à modifier (voir plus bas...)
    - param: la valeur du paramètre
```

La variable param contient la valeur à assigner au paramètre pname. glLight permet de changer les coefficients d'atténuation (GL_CONSTANT_ATTENUATION, GL_LINEAR_ATTENUATION et GL_QUADRATIC_ATTENUATION) ainsi que les valeurs RGBA (4 floats) pour les composantes ambiante (GL_AMBIENT), diffuse (GL_DIFFUSE) et spéculaire (GL_SPECULAR). De la même façon, cette fonction permet de spécifier la position XYZ (avec GL_POSITION) des lumières.

Note : on peut déplacer les lumières à l'aide du bouton du milieu de la souris. On sélectionne la lumière blanche à l'aide de la touche "1" et la lumière bleu à l'aide de la touche "2".

Le modèle d'éclairage d'OpenGL permet aussi de spécifier les propriétés de réflectance (le «matériel») des objets de la scène. On peut modifier ces propriétés à l'aide des fonctions suivantes :

```
void glMaterialf ( GLenum face, GLenum pname, GLfloat param )
void glMaterialfv( GLenum face, GLenum pname, GLfloat* param )
    - face: la face (GL\_FRONT, GL\_BACK ou GL\_FRONT\_AND\_BACK)
    - pname: le paramètre à modifier (voir plus bas...)
    - param: la valeur du paramètre
```

Le paramètre face permet d'associer un matériau différent pour chaque coté d'un polygone (nous utiliserons GL_FRONT_AND_BACK). Le paramètre param contient la valeur (RGBA) à assigner au paramètre pname(GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR). On peut aussi spécifier GL_SHININESS afin de modifier l'exposant "n" de la composante spéculaire (entre 0 et 128).

Note : la composante spéculaire est contrôlée par les touches "x", "X", "z" et "Z" du clavier.

OpenGL permet aussi de modifier le *shading* à l'aide de la fonction suivante :

```
void glShadeModel( GLenum mode )
    - mode: le mode de shading (GL_FLAT ou GL_SMOOTH)
```

Cette fonction permet de rendre les objets dans un mode de *shading* uniforme (GL_FLAT) ou de Gouraud (GL_SMOOTH). Le *shading* est modifié par la touche "s" du clavier.

Note : puisque les calculs d'illumination dépendent de la normale à la surface, assurez-vous de bien la spécifier à l'aide de la fonction **glNormal()**. Dans votre code, on doit retrouver un **glNormal(.)** avant chaque **glVertex()**.

Suggestions

Pour ce travail, nous vous suggérons fortement de procéder suivant l'ordre que voici :

1. afficher la courbe et ses points de contrôle (fonction **displayModelerWindow(.)**) ;
2. afficher le plan (fonctions **drawPlane(.)** et **displayViewerWindow(.)**) ;
3. activer le rendu en fil de fer (*wireframe*) afin de mieux voir le plan (touche 'w' du clavier) ;
4. afficher le teapot ;
5. afficher les deux lumières (fonctions **drawLights(.)** et **displayViewerWindow(.)** ;
6. activer l'illumination ;
7. afficher la surface de translation (fonction **drawSweepObject(.)**) ;
8. afficher la surface de révolution (fonction **drawRevolutionObject(.)**) ;
9. afficher les normales (touche "n" du clavier).

Une autre recommandation : **ne tardez pas trop** avant de commencez ce travail.

Évaluation

Ce travail doit être fait par **équipe de TROIS ou QUATRE**. Au moment de soumettre votre travail, assurez-vous que votre code compile bien sous VisualStudio. Utilisez la commande **turnin** pour soumettre votre travail. La remise doit être faite au plus tard le **7 mars 2014** à minuit.