TD 2 - Shading

Xavier Décoret

Résumé

Le but de ce TD est d'apprendre à utiliser le shading OpenGL
- découverte de QGLViewer
- expérimentation de Flat et Gouraud shading
- découverte des textures

1 QGLViewer

La librairie libQGLViewer développé par Gilles Debunne étend le QGLWidget que nous avons utilisé au TD1 en permettant de manipuler une caméra OpenGL avec la souris.

Pour l'utiliser :
- rajouter `CONFIG *= qglviewer` dans votre `.pro`
- dérivez de `QGLViewer` au lieu `QGLWidget`
- utilisez `init()` au lieu `initializeGL()`
- utilisez `draw()` au lieu `paintGL()`
- ne cherchez pas à positionner ou définir la caméra en réglant les matrices modelview & projection dans `init()`, c'est fait pour vous (vous pouvez quand même modifier la modelview pour positionner vos objets)
- indiquez dans `init()` la "taille" de votre scène 2D en spécifiant une sphère englobante avec `setSceneCenter()` et `setSceneRadius()`

La librairie libQGLViewer vient avec une documentation exhaustive et de nombreux tutoriels. Utiliser google pour trouver cette documentation et gardez-là sous la main.

1.1 Matériaux OpenGL

Faire un programme qui affiche côte à côte un cylindre plein et cappé (avec des disques qui le ferme) éclairé par une lampe :
- en utilisant des IFS avec des nombres minimaux de sommets
- en utilisant du *flat shading* sur l'un et du *gouraud shading* sur l'autre
- en utilisant un `manipulatedFrame()` (voir la documentation QGLViewer) pour pouvoir déplacer deux lampes.

Faites varier la tesselation du cylindre et construire 2 configurations (tesselation + position relative lampes/cylindre) qui montre qu'on peut rater un effet spéculaire. Sauver une image de ces configurations en utilisant Ctrl+S dans le QGLViewer.

2 Textures

Faire un application qui affiche un dé en 3D. La figure suivante donne les paramètres d'un IFS pour gagner du temps. Le but ici est de voir tout les

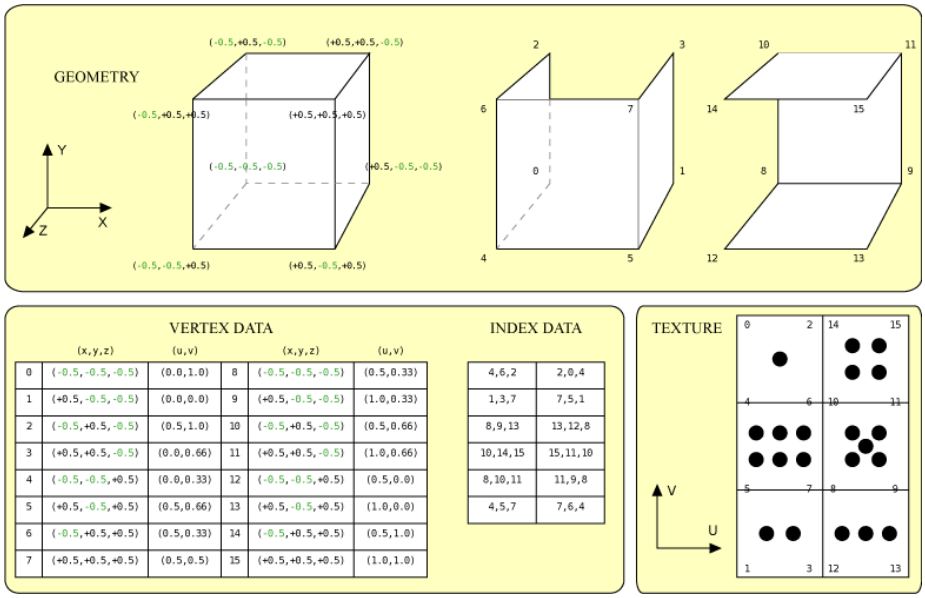

Fig. 1 – IFS pour un dé

états auquel il faut prêter attention avant d'avoir une texture. Voici des pistes vers les fonctions à considérer :

- `glEnable(GL_TEXTURE_2D)` ;
- `glTexImage2D()` ;
- `gluBuildMipmaps()` ;
- `glGenTextures()` ;
- `glBindTexture()` ;
- `glTexParameterf()` ;

Pour lire une image et récupérer ses pixels pour en fabriquer une texture, on utilisera la classe `QImage` de Qt. Consulter `assistant` pour en savoir plus et voir un example.

Regarder l'aide de la fonction `glTexEnvi` et trouver le moyen d'afficher le dé en combinant la texture et le Gouraud shading.