

Introduction à la modélisation 3D

Travaux pratiques

Au début de ce TP/TD, vous recevrez une archive zip contenant une base de code. Ce code permet d'afficher un *maillage triangulaire* à l'aide d'OpenGL.

1. Nous commencerons par l'analyser ensemble pour vous familiariser avec.
2. Vous devez faire évoluer ce code au fur et à mesure du TP, pour répondre aux questions.

1 Base de code

Installation

Téléchargez l'archive sur le moodle <https://moodle.umontpellier.fr/course/view.php?id=22845>. Pour compiler le code et l'exécuter :

```
1 $ make
  $ ./tp
```

Interactions utilisateur

```
void key (unsigned char keyPressed, int x, int y)
```

La fonction `key` permet de d'interpréter les entrées clavier utilisateur. Les options de visualisation activées par des touches sont les suivantes, en appuyant sur la touche :

- `n` : activation/désactivation de l'affichage des normales,
- `1` : activation/désactivation de l'affichage du modèle chargé
- `2` : activation/désactivation de l'affichage du modèle généré,
- `w` : changement du mode d'affichage (fil de fer/éclairé/non-éclairé/fil de fer + éclairé).

Vous pouvez interagir avec le modèle avec la souris :

- Bouton du milieu appuyé : zoomer ou reculer la caméra,
- Clic gauche appuyé : faire tourner le modèle.

Rendu de maillages

Le fichier `tp.cpp` contient une méthode `draw` : c'est une fonction appelée pour rafraîchir l'affichage dès que nécessaire. Elle permet de définir les éléments à afficher, leurs couleurs (`glColor3f(r,g,b)` en float de 0 à 1 donnant la couleur RGB = (r*255, g*255, b*255)). La fonction `drawMesh()` permet d'afficher un maillage triangulaire. Celle-ci fait appel à deux fonctions contenant du code **OpenGL** basique pour afficher des maillages. La première permet l'affichage des sommets :

```
5 void drawVertices( Mesh const & i_mesh ) {
    glBegin(GL_POINTS); //Fonction OpenGL de dessin de points
    for(unsigned int vIt = 0 ; vIt < i_mesh.vertices.size(); ++vIt) {
        Vec3 p = i_mesh.vertices[vIt];

        glVertex3f( p[0] , p[1] , p[2] );
    }
    glEnd();
}
```

La deuxième permet un affichage du maillage sans normales :

```
5 //Fonction de dessin en utilisant sans normales
  glBegin(GL_TRIANGLES); //Fonction OpenGL de dessin de triangles
  for(unsigned int tIt = 0 ; tIt < i_mesh.triangles.size(); ++tIt) {
      Vec3 p0 = i_mesh.vertices[i_mesh.triangles[tIt][0]];
      Vec3 p1 = i_mesh.vertices[i_mesh.triangles[tIt][1]];
      Vec3 p2 = i_mesh.vertices[i_mesh.triangles[tIt][2]];

      //Dessin des trois sommets formant le triangle
      glVertex3f( p0[0] , p0[1] , p0[2] );
  }
```

```

10         glVertex3f( p1[0] , p1[1] , p1[2] );
           glVertex3f( p2[0] , p2[1] , p2[2] );
       }
       glEnd();

```

ou en utilisant les normales au triangles :

```

2   void drawTriangleMesh( Mesh const & i_mesh ) {
       glBegin(GL_TRIANGLES);
       for(unsigned int tIt = 0 ; tIt < i_mesh.triangles.size(); ++tIt) {
           Vec3 p0 = i_mesh.vertices[i_mesh.triangles[tIt][0]];
           Vec3 n0 = i_mesh.normals[i_mesh.triangles[tIt][0]];

7           Vec3 p1 = i_mesh.vertices[i_mesh.triangles[tIt][1]];
           Vec3 n1 = i_mesh.normals[i_mesh.triangles[tIt][1]];

           Vec3 p2 = i_mesh.vertices[i_mesh.triangles[tIt][2]];
           Vec3 n2 = i_mesh.normals[i_mesh.triangles[tIt][2]];

12          glNormal3f( n0[0] , n0[1] , n0[2] );
           glVertex3f( p0[0] , p0[1] , p0[2] );
           glNormal3f( n1[0] , n1[1] , n1[2] );
           glVertex3f( p1[0] , p1[1] , p1[2] );
           glNormal3f( n2[0] , n2[1] , n2[2] );
           glVertex3f( p2[0] , p2[1] , p2[2] );

17      }
       glEnd();
   }

```

2 Exercice : création d'un maillage triangulaire de sphère 3D

Dans le fichier `tp.cpp`, compléter la fonction `void setUnitSphere(Mesh & o_mesh(int nX, int nY)`, qui créera un maillage triangulaire de sphère 3D.

- Créer des sommets discrétisant la sphère avec un nombre `nX` de méridiens et `nY` de parallèles.
Indications : un point 3D sur la sphère peut être obtenue à l'aide de la paramétrisation sphérique fonction de deux angles $(\theta, \phi) \in [0, 2\pi] \times [-\pi/2, \pi/2]$:
 - $x = \cos(\theta) * \cos(\phi)$
 - $y = \sin(\theta) * \cos(\phi)$
 - $z = \sin(\phi)$
- Générer la topologie : créer la liste des triangles du maillage.
- Ajouter la fonctionnalité suivante : l'appui de la touche “-” augmente le nombre de méridiens et celui de parallèles de 1. De la même manière l'appui sur la touche “+” diminue de 1 le nombre de méridiens et celui de parallèle.
- Ajouter les normales aux sommets.