

Master Informatique 1ère année 2013-2014

Informatique Graphique

Lundi 27 janvier 2014 - 12h-12h30

Tous les documents sont autorisés.

On rappelle les prototypes des fonctions OpenGL usuelles:

- void glTranslatef(GLfloat tx,GLfloat ty,GLfloat tz);
- void glRotatef(GLfloat a,GLfloat ax,GLfloat ay,GLfloat az);
- void glScalef(GLfloat rx,GLfloat ry,GLfloat rz);
- void glPushMatrix();
- void glPopMatrix();
- void glutSolidCube(double c);
- void myCylindre(float h,float r,int ns);

La fonction GLUT permettant de modéliser une sphère est définie par le prototype suivant:

- void glutSolidSphere(double r,int n,int m);

où r est le rayon, n et m sont les valeurs de facettisation longitudinale et latitudinale.

Question 1

Développer une fonction C permettant de modéliser la scène composée de 8 sphères de rayon 2.0 et 12 cylindres de hauteur 10.0 et de rayon 1.0 placés de telle manière que les cylindres et les sphères matérialisent respectivement les arêtes et les sommets d'un cube virtuel de côté 10.0 centré sur l'origine du repère de modélisation.

Question 2

Un bras robot en forme de tentacule est composé de n segments successifs formés d'un cylindre de hauteur 1.0 et de rayon 1.0 et d'une sphère "articulation" de rayon 0.5 (voir dessin ci-contre pour $n = 3$). Les centres des sphères et des cylindres alternés sont distants de 0.8. Chaque segment possède 2 degrés de liberté par rapport au segment qui le précède: une rotation autour de l'axe y passant par le centre de la sphère articulation suivi d'une rotation autour de l'axe x passant par le centre de la sphère articulation. Deux tableaux tRy et tRx de $n-1$ réels définissent les valeurs de ces rotations. Le dessin ci-dessous présente un tentacule de 20 segments soit donc 2×19 angles de rotation qui sont responsables de sa courbure.

Développer une fonction C de modélisation d'un tel bras robot répondant au prototype suivant:

```
void tentacule(int n,float *tRy,float *tRx);
```

Le premier cylindre du tentacule est centré sur l'origine du repère de modélisation et est orienté en y dans ce repère.

