UTBM Printemps 2007
UV IN55 – Synthèse d'images

# Examen Final

Mardi 26 Juin 2007

Coefficient : 35 %
Aucun document ni calculatrice autorisés.

**Chaque PARTIE devra être rédigée sur une FEUILLE SEPAREE.**

## Partie I (6 points) :

### Exercice 1 : Question de cours (6*0,5 points)

Pour chacune des questions suivantes, justifiez votre réponse.

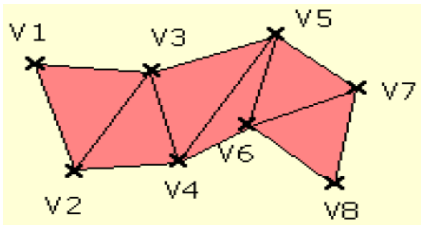
1. Parmi ces transformations, quelle(s) est/sont celle(s) qui n'affectent pas le caractère perpendiculaire d'une normale à une surface lorsqu'appliquée(s) à la fois sur l'ensemble des vertices composant la surface et sur la normale à la surface ?

**a) Translation b) Rotation c) Changement d'échelle uniforme**
**d) Changement d'échelle non uniforme**

2. Quelle technique permet de diviser les polygones composant un modèle 3D en polygones primitifs (triangles) pour pouvoir être affichés ?

**a) La rasterization b) La tesselation c) La technique de simplification d) Le billboarding**

3. On désire afficher le modèle suivant :




Quelle primitive OpenGL permet de réduire le nombre de sommet à utiliser ?

**a) GL_TRIANGLES b) GL_TRIANGLE_STRIPS c) GL_TRIANGLE_FAN**

4. On désire visualiser un objet pivotant sur lui-même et pivotant également à une certaine distance d'un axe. Quelle combinaison de transformations permet d'obtenir ce résultat ?

**a) Rotation Rotation Translation b) Rotation Translation Rotation c) Translation Rotation Rotation**

5. Un objet possédant une couleur émissive peut-il être considéré comme une source de lumière ?

**a) Oui b) Non**

6. Soient un matériau M caractérisé par une couleur diffuse C(1.0 0.5 0.0) et une source de lumière diffuse L( 0.0 0.0 1.0). De quelle couleur apparaît un point illuminé possédant le matériau M ?

### Exercice 2 : Programme OpenGL (1+2 points)

1. Décrivez en quelques lignes le procédé de manipulation de la pile de matrices en OpenGL (fonctionnement en mémoire, fonction utiles).

2. Décrivez ce qui se passe dans la fenêtre OpenGL lorsqu'on exécute le programme suivant. Vous pouvez étayer votre réponse par une illustration.

```
static GLfloat day=0, year=0, yearmoon=0;
void Affichage(void)
{
            glMatrixMode(GL_MODELVIEW);
            glLoadIdentity();
            gluLookAt(0.0,0.0,5.0,0.0,0.0,0.0,0.0,1.0,0.0);
            glClear(GL_COLOR_BUFFER_BIT);
            glColor3f(1.0,1.0,1.0);
            glPushMatrix();
            glutWireSphere(1.0,20,16);
            day=(day+1);
            year=(year+0.5f);
            yearmoon=(yearmoon+4);
            glTranslatef(2.0,0.0,0.0);
            glRotatef(year, 0.0,1.0,0.0);
            glColor3f(1.0,0.0,0.0);
            glutWireSphere(0.2,10,8);
            glTranslatef(0.0,0.0,1.0);
            glRotatef(year, 0.0,1.0,0.0);
            glColor3f(1.0,1.0,0.0);
            glutWireSphere(0.1,10,8);
            glColor3f(0.0,0.0,1.0);
            glutWireSphere(0.3,10,8);
            glPushMatrix();
            glRotatef(year, 0.0,1.0,0.0);
            glTranslatef(0.0,0.0,1.0);
            glColor3f(0.0,1.0,0.0);
            glutWireSphere(0.3,10,8);
            glPopMatrix();
            glPopMatrix();
            glutSwapBuffers();
}
```

Cette fonction Affichage est appelée dans la fonction principale *main* du programme **planetarium.c** par la fonction glutDisplayFunc(Affichage).
La fonction glutWireSphere(GLdouble radius,GLint slices, GLint stacks) crée une sphere en mode filaire d'un rayon radius, d'un nombre slices de coupes en latitude et d'un nombre stacks de coupes en longitude.

## Partie II (8 points) :

### Exercice 1 : Recherche optimisée d'objet (2+2 points)

Pour une application de picking, on désire retrouver l'objet 3D le plus proche d'un point P de l'espace. Est considéré objet le plus proche, l'objet possédant le vertex le plus proche du point P en question. Notre univers est composé d'objets 3D stockés dans une structure octree. Dans notre cas, les objets n'ont pas été subdivisés lors du stockage, c'est à dire que les objets sont stockés dans les feuilles de l'octree.

Les objets 3D sont composés d'une liste de vertices, chaque vertex disposant donc de trois coordonnées (X,Y,Z). Soit N un des noeuds de l'octree. $N.fils_n$ permettra alors d'accéder au $n^{ème}$ fils du noeud N et N.XMin, N.YMin, N.ZMin, N.XMax, N.YMax, N.ZMax aux coordonnées des points Haut et Bas de la boîte englobante.

1. Proposez l'algorithme de la fonction « parcoursOctree » permettant de retrouver la liste des objets appartenant à l'espace octree qui contient le point P :

   parcoursOctree(Point3D pt, Octree octree) -> Liste_Objet3D

2. Proposez l'algorithme de la fonction « trouvePlusProche » retournant l'objet 3D le plus proche du

point P parmi la liste construite précédemment.

trouvePlusProche(Point3D pt, Liste_Objet3D liste) -> Objet3D

### Exercice 2 : Implémentation du Z-Buffer (2+1+1 points)

Pour réaliser l'affichage de nos objets, nous désirons utiliser un Z-Buffer classique.

1. Donnez l'algorithme de la fonction « affichagePixel » implémentant le Z-Buffer. Cette fonction est appelée lors du rendu pour tous les points des objets de la scène. Les objets ne sont pas triés au préalable.

   affichagePixel(Entier u, Entier v, Entier profondeur, Couleur colPixel)

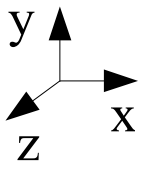
   Les paramètres de la fonction sont :
   - La position du pixel en u et v dans l'image finale
   - La profondeur du pixel
   - La couleur du pixel

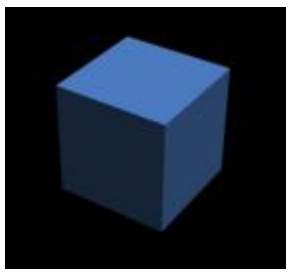
2. Nous souhaitons utiliser un buffer de profondeur 1 bit. Expliquez en détail les modifications à apporter pour pouvoir réaliser cette modification.

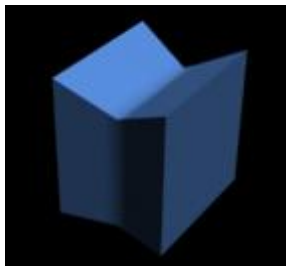
3. Quel problème peut survenir avec le Z-Buffer dans le cas d'univers de très grande dimension contenant des objets très petits et très proches les uns des autres ? Quelle modification faut-il entreprendre pour régler ce problème ?

## Partie III (6 points) :

### Exercice 1 – Déformations spatiales (2+1 points)





1. Expliquez la méthode pour obtenir le cube de droite déformé à partir du cube de gauche. Exprimer les coordonnées d'un sommet du cube déformé en fonction des coordonnées d'un sommet appartenant au cube initial de gauche.

2. Quelle particularité doit posséder le cube de gauche pour pouvoir obtenir une telle déformation ?

### Exercice 2 – Modèles d'éclairement (1,5+1,5 points)

Soient :
- A(1 1 0) un point de couleur $c_A$(0.5 1.0 0.0) et de normale $n_A$(1 0 1),
- B(-1 -1 0) un point de couleur $c_B$(0.8 0.2 1.0) et de normale $n_B$(0 0 1),
- P(0.5 0.5 0) le point appartenant à la droite AB,
- L(0 0 -1) le vecteur de direction d'une source de lumière directionnelle.

1. Donnez les composantes RVB du pixel situé au point P dans le cas du modèle d'ombrage de Gouraud.

2. Donnez les composantes RVB du pixel situé au point P dans le cas du modèle d'ombrage de Phong.

**Remarque :**

Soient C une des composantes RVB d'un point et C' la même composante RVB après illumination. C' peut être obtenu par l'équation suivante :

$$C' = C * \max( 0.0, -N.L )$$

N et L doivent être normalisés. N est la normale à la surface et L est le vecteur de lumière incident à la surface.