

**Master01 : Vision Artificielle**

**Durée :01h 30mn**

**Date : 16/05/2024**

## **Contrôle**

### **Questions de cours (7.5 pts)**

Q1 : Expliquez les étapes principales de la production d'une image de synthèse. Discutez le rôle de chaque étape. **(5 pts)**

R1 : Les principales étapes de la production d'une image de synthèse sont :

- 1- La modélisation : consiste à créer le modèle des différents objets de la scène.
- 2- Les transformations : rotation, changement d'échelle, et rotation pour positionner les objets.
- 3- La projection : on utilise la projection perspective.
- 4- L'élimination des parties cachées : on élimine les parties non visibles par l'œil de l'être humain.
- 5- Le calcul du rendu : on utilise l'algorithme de Lancer de rayons pour calculer la transformation et la transmission et l'algorithme de radiosit  pour le calcul de la radiosit .

Q2 : Effectuez une comparaison entre les trois mod les suivants : B zier, B-Spline, NURBS. **(1.5 pts)**

R2 : Les trois courbes utilisent les points de contr le, mais la courbe de B zier souffre du probl me de contr le global et la pr cision. Les courbes de B-Spline ont r solu le probl me de contr le global par l'utilisation d'une fonction qui est nulle sur les intervalles. Et la courbe de NURBS utilise les cordonn es homog nes pour augmenter la pr cision.

Q3 : Quel est le r le de l'algorithme de Lancer de rayons ? Expliquez comment cet algorithme a  t  utilis  pour  liminer les parties cach es ? **(1pts)**

R3 : Le r le de cet algorithme est le calcul du rendu mais lorsqu'il y a l'effet de r flexion et de transmission.

Dans le d but, cet algorithme a  t  utilis  pour  liminer les parties cach es, il traite la 1<sup> re</sup> intersection du rayon observateur-pixel avec l'objet. Les autres objets sont cach s.

### **Exercice n 01 (7pts)**

Soit une sc ne avec plusieurs objets 3D r partis en profondeur.

- 1- D crivez le mod le de projection utilis  et quel est son r le ? **(2 pts)**
  - 1- Le mod le de projection utilis  est la perspective
  - 2- Son r le est le passage   partir de 3D vers 2D

Pour  liminer les parties cach es, vous avez utilis  un algorithme :

- 2- Quel est le nom de cet algorithme ? **(1 pts)** cet algorithme est Z-Buffer
- 3- Expliquez son fonctionnement ? **(2pts)**

Cet algorithme utilise deux buffers (un buffer de couleur initialis    la couleur de fond et de profondeur initialis     $\infty$ ),   chaque fois il compare la profondeur de tous les pixels des objets de la sc ne avec la profondeur enregistr e dans le buffer. Si la

profondeur est inférieure alors la couleur de ce pixel sera enregistrée dans le buffer de couleur

- 4- Comment pouvez-vous implémenter cet algorithme en utilisant la bibliothèque OpenGL ? (2pts)

En OpenGL, il suffit d'activer la fonction Z-Buffer par la fonction suivante :

`glEnable(GL_DEPTH_TEST);`

Et au lieu d'effacer uniquement le buffer de couleur, on efface les deux buffers (de couleur et de profondeur)

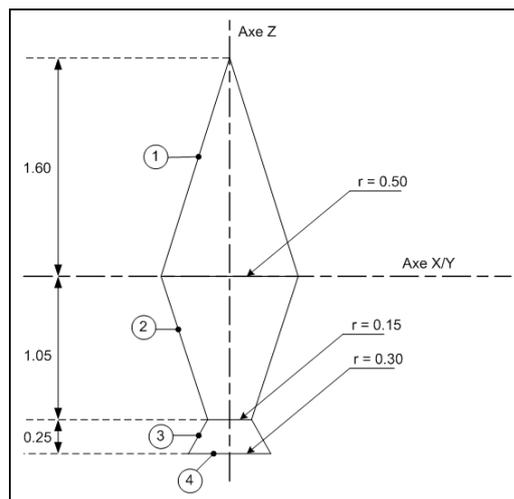
`glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);`

### Exercice n°02 (5.5 pts)

Soit la fonction OpenGL suivante :

```
{ glPushMatrix();
  GLUquadric* params = gluNewQuadric();
  gluCylinder(params,0.5,0,1.6,20,1);
  glTranslated(0,0,-1.05);
  gluCylinder(params,0.15,0.5,1.05,20,1);
  glTranslated(0,0,-0.25);
  gluCylinder(params,0.3,0.15,0.25,20,1);
  gluDisk(params,0,0.3,20,1);
  gluDeleteQuadric(params);
  glPopMatrix(); }
```

- Quel est le rôle des fonctions `glPushMatrix()` et `glPopMatrix()` ? (2pts)  
`glPushMatrix()` : sauvegarde la matrice actuelle ;  
`glPopMatrix()` : restitue la matrice sauvegardée.
- Donnez le résultat final de cette fonction (3.5 pts)
- Le résultat est une roquette



**Bonne chance**