UNIVERSITE DE BOURGOGNE
U.F.R. Sciences et Techniques
Filière : *Licence 3 Informatique*

2019-2020

Session 1

EPREUVE :

Examen Synthèse d'Image janvier 2020

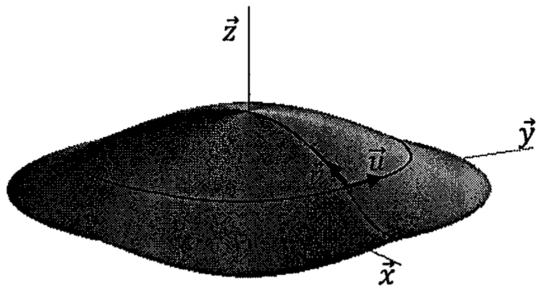
Durée : 1h30

Seul document autorisé : une feuille A4 recto-verso ***manuscrite.***
Les exercices peuvent être traités indépendamment les uns des autres.
Le barème est donné à titre indicatif.

N° d'anonymat :

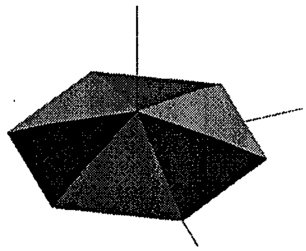
Partie 1 : *Modélisation d'une soucoupe volante à partir de sa représentation paramétrique (environ 10 points)*
But : Modéliser sous forme de facettes une soucoupe de coefficient a=1.5 et b=0.5.

Le nombre de discrétisation de la soucoupe dans la direction u est Nu et dans la direction v est Nv. Toutes les faces de la soucoupe sont quadrilatérales.

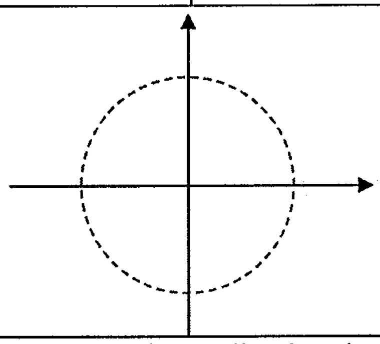
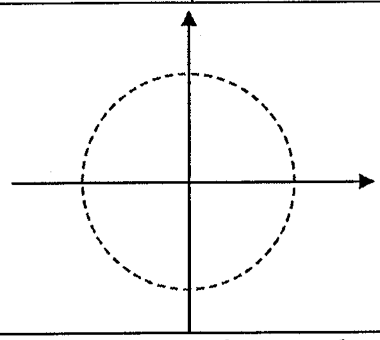


$$\begin{cases} x(u,v) = a \times cos(u) \times cos(v) \\ y(u,v) = a \times sin(u) \times cos(v) \\ z(u,v) = b.sin^3(v) \end{cases} \quad avec \quad \begin{matrix} u \in [0,2\pi[\\ v \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right] \end{matrix}$$

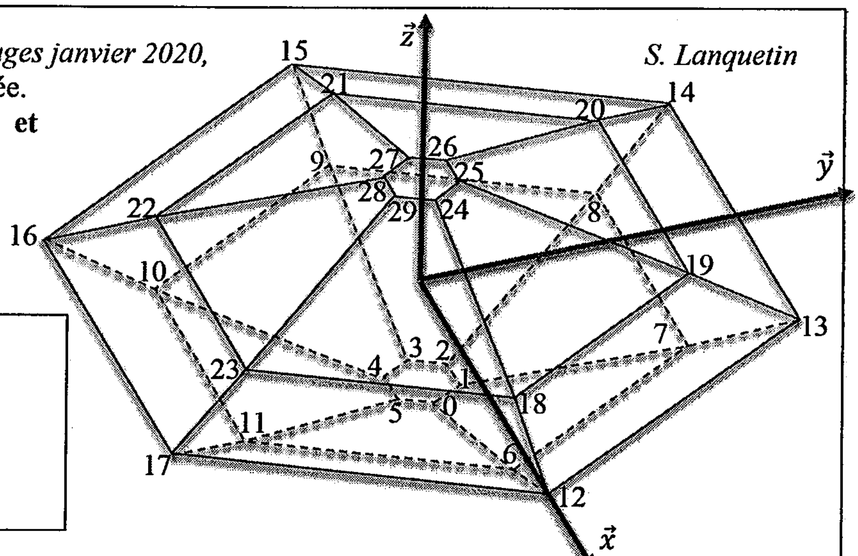
✓ Donner la longueur des intervalles de u et v.



1. Discrétisation de la soucoupe avec $Nu = 6$ et $Nv = 5$.
   ✓ Compléter les dessins et les parties grisées dans le tableau ci-dessous.

<table>
<tr><td>$u \in$</td><td>$Nu =$</td><td>$v \in$</td><td>$Nv =$</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="4">Placer les bornes de l'intervalles de u (respectivement v) sur le cercle trigonométrique gauche (respectivement droit) et dessiner l'arc de cercle correspondant pour u (respectivement pour v).</td></tr>
<tr><td colspan="4">Placer les points de discrétisation sur l'arc de cercle trigonométrique correspondant pour u et v.</td></tr>
<tr><td colspan="2">L'intervalle est découpé en   parties.</td><td colspan="2">L'intervalle est découpé en   parties.</td></tr>
<tr><td colspan="4">Donner ci-dessous le nombres de parties de chaque intervalle en fonction de Nu et Nv.</td></tr>
<tr><td colspan="2">Nombre de parties de u =</td><td colspan="2">Nombre de parties de v =</td></tr>
</table>

Pour $Nu = 6$ et $Nv = 5$ on obtient le maillage ci-contre.



La numérotation des sommets est donnée.
L'indice de boucle sur u est noté i et celui sur v est noté j.

- ✓ Donner le nombre de sommets et de faces de la soucoupe en fonction de Nu et Nv.

- ✓ En déduire les formules des déplacements du et dv de u et de v en fonction de Nu et de Nv.

2. Donner la liste des indices de sommets par face dans le tableau ci-après. En déduire une formule des indices de points qui forment une face pour chaque j en fonction de i, Nu et Nv.

	Indice face	Indice des sommets par face – Indice 1[er] sommet	Indice 2[nd] sommet	Indice 3[ème] sommet	Indice 4[ème] sommet	Indices des sommets d'une face en fonction de i, Nu et Nv
j=0	0	0	1	7	6	
	1					
	2					
	3					
	4					
	5					
j=1						
j=2						
j=3						

✓ En déduire une formule générale pour les indices de sommets par face en fonction de Nu, Nv, i (indice de boucle sur u) et j (indice de boucle sur v).

✓ Donner l'indice d'une face en fonction de Nu, Nv, i (indice de boucle sur u) et j (indice de boucle sur v).

3. Écrire une fonction `coord(...)` ayant pour paramètres a, b, et u et v et qui retourne un sommet de la soucoupe.

```
class Point{
public:
    float x;
    float y;
    float z;
};
```

4. Écrire l'algorithme pour remplir la liste pSoucoupe des coordonnées en fonction de Nu et Nv.

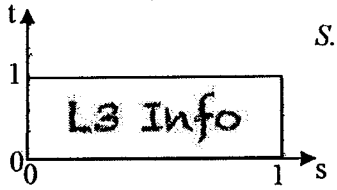
5. Écrire l'algorithme pour remplir la liste `fSoucoupe` des indices de sommets en fonction de Nu et Nv.

6. Compléter la fonction `soucoupe(...)` permettant de dessiner une soucoupe de paramètres a, b en précisant Nu et Nv.

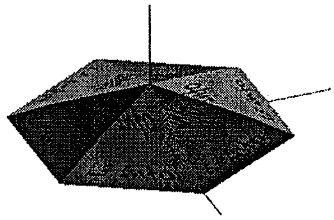
```
void soucoupe(float a, float b, int Nu, int Nv){

}
```

Exercice 3 : *Textures (environ 2 points)*

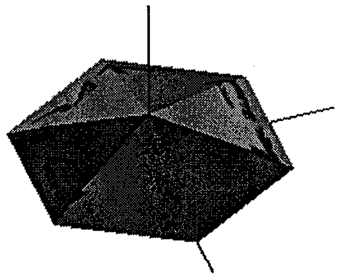
On utilise la texture ci-contre.



1. Modifier la fonction `soucoupe(...)` pour plaquer la texture L3Info sur chaque facette.



2. Modifier la fonction `soucoupe(...)` pour découper la texture L3Info afin de l'enrouler sur la soucoupe.



Partie 2 : *Transformations (environ 5 points)*

Soit une transformation M composée d'une translation T de vecteur $(-1,0,0)$ suivie d'une rotation R d'axe y et d'angle 180°.

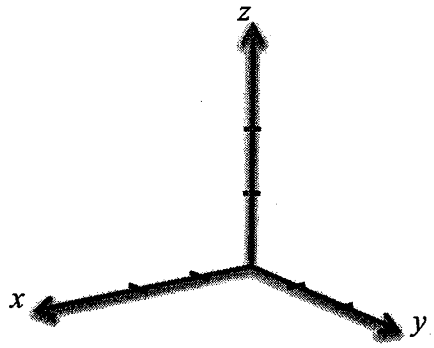
1. Donner l'expression de cette rotation et de cette translation sous la forme de matrices homogènes R et T.

2. Calculer R et T.

3. Donner l'expression de cette transformation sous la forme d'une matrice homogène M en fonction des matrices R et T.

4. Calculer M.

5. Soit P le point de coordonnées (0,1,0,1). Placer P et P' (image de P par la transformation M) dans le repère ci-contre :



6. Soit P le point de coordonnées (0,1,0,1). Donner les coordonnées du point P' image de P par la transformation M (toujours en coordonnées homogènes).

Partie 3 : *Cours (environ 5 points) Écrire la réponse dans les cadres.*

Question 1 :

A quelle transformation correspond la matrice ci-contre. Préciser ses paramètres.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Question 2 :

A quelle transformation correspond la matrice ci-contre. Préciser ses paramètres.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Question 3 :

A quelle transformation correspond la matrice ci-contre. Préciser ses paramètres.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

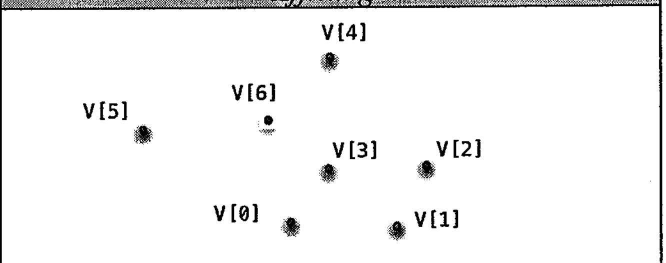
Question 4 :

Compléter l'affichage obtenu en exécutant le code suivant.

Code

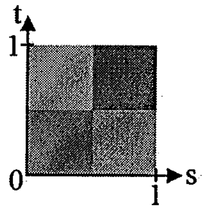
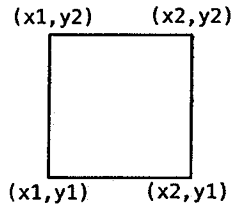
```
class Point{
public:    double x,y,z;
};
void dessin(){
    Point V[7];
    glColor3f(0.0,0.0,1.0);
    glBegin(GL_TRIANGLES);
       for(int i=0;i<7;i++)
           glVertex3f(V[i].x,V[i].y,V[i].z);
    glEnd();
}
```

Affichage

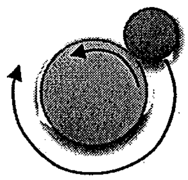


Question 5 :
Compléter l'affichage obtenu en exécutant le code suivant.

Code	*Affichage*

`glEnable(GL_TEXTURE_2D);`
`glBegin(GL_QUADS);`
`    glTexCoord2f(0, 1/4);`
`    glVertex2f(x1,y1);`
`    glTexCoord2f(1, 1/4);`
`    glVertex2f(x2,y1);`
`    glTexCoord2f(1,1);`
`    glVertex2f(x2,y2);`
`    glTexCoord2f(0,1);`
`    glVertex2f(x1,y2);`
`glEnd();`
	 

Question 6:
On considère une scène avec deux sphères : sphère1 de rayon 2 et sphère2 de rayon 1.



1. Compléter le code pour que sphère1 tourne sur elle-même
2. Compléter le code pour que sphère2 tourne autour de sphère1

```
int angleR=0 ;
//On suppose que l'angle est incrémenté dans la fonction animation appelée
avec glutIdleFunc(animation); dans le main.
void affichage()
{//Ne coder que ce qui concerne les sphères



//sphere1



    glutSolidSphere(                );


//sphere2




    glutSolidSphere(                );
```