

Nom:

Prénom:

## TSI (session 2) Synthèse d'images

*Aucun document autorisé  
Calculatrices numériques autorisées  
Répondre directement sur l'énoncé d'examen*

On affiche une grille de  $N \times N$  sommets ( $N > 10$ ). Cette grille couvre uniformément le carré de coordonnées  $(0,0),(1,0),(1,1),(0,1)$ .

L'affichage de cette grille est réalisé en OpenGL, et on utilise les shaders suivants:

### Vertex Shader

```
#version 120

varying vec3 normal;
varying vec4 vertex_3d_original;
varying vec4 vertex_3d_modelview;

void main (void)
{
    const float pi=3.14159;
    const float R=0.2;
    const float h=0.4;

    float u=gl_Vertex.x;
    float v=gl_Vertex.y;
    float x=R*cos(2*pi*u);
    float y=R*sin(2*pi*u);
    float z=h*v;
    vec4 p=vec4(x,y,z,1.0);

    vec3 n=vec3(cos(2*pi*u),sin(2*pi*u),0);

    gl_Position = gl_ModelViewProjectionMatrix * p;
    normal = n;
    vertex_3d_original=p;
    vertex_3d_modelview=gl_ModelViewMatrix*p;
}
```

### Fragment Shader

```
#version 120

varying vec3 normal;
varying vec4 vertex_3d_original;
varying vec4 vertex_3d_modelview;

vec3 light=vec3(1.3,1.2,5);

void main (void)
{
    vec3 n=normalize(normal);

    vec3 p=vertex_3d_modelview.xyz;
    vec3 vertex_to_light=normalize(light-p);
    vec3 reflected_light=reflect(-vertex_to_light,n);
    vec3 user_to_vertex=normalize(-p);

    float diffuse_term=0.7*clamp(dot(n,vertex_to_light),0,1);
    float specular_term=0.2*pow(clamp(dot(reflected_light,
                                          user_to_vertex),0,1)
                                ,128);

    float ambient_term=0.2;
    vec4 white=vec4(1,1,1,0);
    vec4 color=vec4(0.3,0.4,0.9,1.0);

    gl_FragColor = (ambient_term+diffuse_term)*color+
                   specular_term*white;
}
```

- Décrivez le plus précisément possible l'objet observé à l'écran.
- Quel type d'illumination est réalisé par le Fragment shader?

Réponse: