

Nom:

Prénom:

TSI

Partie synthèse d'images

(50 min. maximum)

Une feuille A4 manuscrite autorisée par personne.

Tout autre document interdit.

Calculatrices non autorisées.

Répondre directement sur l'énoncé d'examen dans les cases prévues (si vous manquez de place, écrivez au dos des pages).

Question 1.

- Qu'appelle t'on méthode d'interpolation de **Gouraud** et de **Phong** ? Quelle différence existe t-il entre ces deux approches ?

Question 2.

- On considère le code GLSL suivant:

```
#version 120
void main (void)
{
    gl_Position = gl_Vertex;
}
```

Comment appelle-t-on le shader pour lequel ce code s'applique ?

Ce code réalise d'une certaine manière la projection d'un sommet de l'espace 3D vers une position dans l'espace de la caméra. Comment pourrait-on qualifier cette projection ?

Question 3.

Soit les parties de code suivantes:

Dans la **fonction d'initialisation du programme principal**:

```
float sommets[]={0.0f,0.0f,0.0f,
                 1.0f,0.0f,0.0f,
                 0.0f,1.0f,0.0f,
                 1.0f,1.0f,0.0f};
int index[]={0,1,2 , 1,3,2};

glGenBuffers(1,&vbo);
glBindBuffer(GL_ARRAY_BUFFER, vbo);
glBufferData(GL_ARRAY_BUFFER, sizeof(sommets), sommets, GL_STATIC_DRAW);

glEnableClientState(GL_VERTEX_ARRAY);
glVertexPointer(3, GL_FLOAT, 0, 0);

glGenBuffers(1,&vboi); PRINT_OPENGL_ERROR();
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, vboi);
glBufferData(GL_ELEMENT_ARRAY_BUFFER, sizeof(index), index, GL_STATIC_DRAW);
```

Dans la **fonction d'affichage du programme principal**:

```
glUniform4f(get_uni_loc(shader_program_id, "tr"), -0.5f, -0.5f, 0.99f, 0.0f);
glDrawElements(GL_TRIANGLES, 3*2, GL_UNSIGNED_INT, 0);
```

Dans le **vertex shader**:

```
#version 120
uniform vec4 tr;
varying vec4 p;
void main (void)
{
    p=gl_Vertex+tr;
    p.x*=2.0;
    p.y*=2.0;
    gl_Position = p;
}
```

Dans le **fragment shader**:

```
#version 120
varying vec4 p;
void main (void)
{
    gl_FragColor = vec4(abs(p.x), abs(p.y), abs(p.z), 1.0);
}
```

On supposera que l'ensemble des variables utilisées dans le programme et le contexte d'affichage en OpenGL a correctement été initialisé et appelé. Le programme compile et s'exécute sans erreurs.

- Décrivez le plus précisément possible ce que l'on **observe** sur l'écran en expliquant les raisons. En particulier, décrivez précisément les **données** envoyées sur la carte graphique pour l'affichage, l'**action des shaders** sur le résultat visuel. Donnez le plus de précisions possibles sur les **valeurs** associées. Aidez vous d'un **schéma** en indiquant les **valeurs spécifiques**.

Remarque. Si vous ne connaissez pas le nom d'une couleur, décrivez la par ses composantes la constituant.

Rappel: Par défaut, OpenGL affichage le point $(-1,-1)$ en bas à gauche de la fenêtre, et le point $(+1,+1)$ en haut à droite de la fenêtre.

