Nom:

Prénom:

TSI
Synthèse d'images

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

Question 1.

- Définissez en quelques mots ce qu'est **GLUT**, et ce qu'est **OpenGL**.

Question 2.

- En **GLSL**, expliquez la différence entre une variable qualifiée de ***uniform*** par rapport à une variable qualifiée de ***varying***.

Question 3.

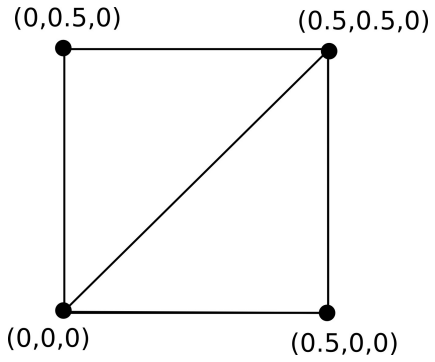
Considérons le code C suivant:

```
GLuint vbo=0;
float T[]={0,0,0 , 1,0,0 , 0,1,0 , 0,0,1};

glGenBuffers(1,&vbo);
glBindBuffer(GL_ARRAY_BUFFER,vbo);
glBufferData(GL_ARRAY_BUFFER,12*sizeof(float),T,GL_STATIC_DRAW);
```

- **Expliquez** précisément ce que **réalise** ce code (en particulier par rapport à la **carte graphique**). Ce code **affiche-il** quelque chose à l'écran, si oui qu'affiche t-il?

Question 4.

On affiche deux triangles à l'aide d'appels OpenGL. Ces deux triangles mis bout à bout forment un carré. Les coordonnées des sommets sont indiquées sur la figure suivante



On suppose que l'on place une caméra en face de ces triangles. On utilise les shaders suivants lors de l'affichage de ces deux triangles:

Vertex Shader

```
#version 120

varying vec4 p3d;

void main (void)
{
    gl_Position = ftransform();

    p3d=gl_Vertex;
    p3d.x *= 2.0f;
    p3d.y *= 2.0f;
}
```

Fragment Shader

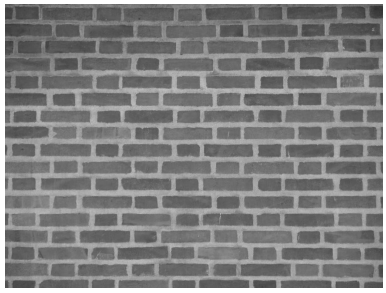
```
#version 120

varying vec4 p3d;

void main (void)
{
    float x=p3d.x; float y=p3d.y; float z=p3d.z;
    float r=0.0f; float g=0.0f; float b=0.0f;
    if(abs(x-0.5f)<0.25f && abs(y-0.5)<0.25f)
    {
        r=1.0f; g=1.0f;
    }
    else
    {
        r=x; g=y; b=z;
    }
    gl_FragColor = vec4(r,g,b,1.0f);
}
```

***Note**: Un nombre à virgule terminant par "f" indique un nombre flottant simple précision.*

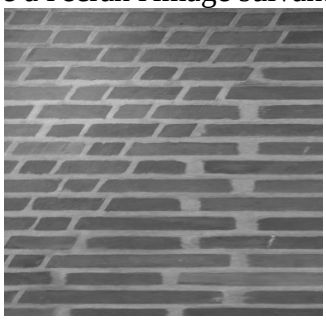
- Indiquez ce que l'on **observe** sur l'écran. En particulier, décrivez précisément l'**action des shaders** sur le résultat visuel. Donnez le plus de précision possibles sur les **valeurs** associées. Aidez vous d'un **schéma**.

Question 5.

Désormais, on utilise un shader d'affichage standard permettant de gérer les textures (identique à votre shader de TP qui affiche des textures).
On dispose d'un fichier contenant l'image suivante:



On affiche les deux triangles précédents en plaçant à la main les coordonnées de textures pour chacun des sommets. On observe à l'écran l'image suivante:



- Quelles pourraient être les **coordonnées de textures** associées à chaque sommet des triangles? Faites un **schéma** illustrant la **correspondance** entre les sommets du triangles et les coordonnées de texture sur l'image originale.

On supposera que les coordonnées de textures **(0,0)** correspondent au **coté bas à gauche** de l'image, et **(1,0)** au **coté bas à droite**.

***Note**: la valeur précise numérique des coordonnées de texture n'est pas importante.*