



# Projet Synthèse D'images

---

Fait par :

**ABED Nada Fatima-Zohra  
REBAI Mohamed Younes**

## 1. Résumé :

Dans ce projet nous a été demandé de réaliser une scène 3D avec OpenGL3.

Nous avons eu l'idée de reproduire les pyramides de « gizeh » (jour/nuit), en utilisant les techniques vues en TP.



Figure 1.1 : « les pyramides de gizeh »

## 2. Modélisation des pyramides :

On a dessiné plusieurs pyramides (une base carrée placée sur le plan XZ en définissant les 18 sommets) et leurs vecteurs Normal afin d'appliquer la texture.



Figure 2.1 : « Pyramide avec texture »

```

]STRVertex vertices[] = {

    vec3(-0.5f, -0.5f, -0.5f), vec3(0.0f, 0.0f, -1.0f),vec2(0.0f,0.0f),
    vec3(0.5f, -0.5f, -0.5f), vec3(0.0f, 0.0f, -1.0f),vec2(1.0f,0.0f),
    vec3(0.0f, 0.5f, 0.0f), vec3(0.0f, 0.0f, -1.0f),vec2(0.5f,1.0f),

    vec3(-0.5f, -0.5f, 0.5f), vec3(0.0f, 0.0f, 1.0f),vec2(0.0f,0.0f),
    vec3(0.5f, -0.5f, 0.5f), vec3(0.0f, 0.0f, 1.0f),vec2(1.0f,0.0f),
    vec3(0.0f, 0.5f, 0.0f), vec3(0.0f, 0.0f, 1.0f),vec2(0.5f,1.0f),

    vec3(-0.5f, -0.5f, -0.5f),vec3(-1.0f, 0.0f, 0.0f),vec2(0.0f,0.0f),
    vec3(-0.5f, -0.5f, 0.5f),vec3(-1.0f, 0.0f, 0.0f),vec2(1.0f,0.0f),
    vec3(0.0f, 0.5f, 0.0f),vec3(-1.0f, 0.0f, 0.0f),vec2(0.5f,1.0f),

    vec3(0.5f, -0.5f, -0.5f),vec3(1.0f, 0.0f, 0.0f),vec2(0.0f,0.0f),
    vec3(0.5f, -0.5f, 0.5f),vec3(1.0f, 0.0f, 0.0f),vec2(1.0f,0.0f),
    vec3(0.0f, 0.5f, 0.0f),vec3(1.0f, 0.0f, 0.0f),vec2(0.5f,1.0f),

    vec3(0.5f, -0.5f, -0.5f),vec3(0.0f, -1.0f, 0.0f),vec2(0.0f,0.0f),
    vec3(0.5f, -0.5f, 0.5f),vec3(0.0f, -1.0f, 0.0f),vec2(1.0f,0.0f),
    vec3(0.0f, 0.5f, 0.0f),vec3(0.0f, -1.0f, 0.0f),vec2(0.5f,1.0f),

    vec3(-0.5f, -0.5f, 0.5f),vec3(0.0f, 1.0f, 0.0f),vec2(0.0f,0.0f),
    vec3(-0.5f, -0.5f, -0.5f),vec3(0.0f, 1.0f, 0.0f),vec2(1.0f,0.0f),
    vec3(0.0f, 0.5f, 0.0f),vec3(0.0f, 1.0f, 0.0f),vec2(0.5f,1.0f),

};

```

Figure 2.2 : «les coordonnées des sommets»

Afin de modéliser plusieurs pyramides on a repris la même structure de la pyramide centrale puis on lui applique une série de transformation (scale, translation...).



Figure 2.3 : «les 3 pyramides»



Pour la reproduction du ciel on a opté pour une « skybox » donc 5 plans (en haut, à droite, à gauche, avant et arrière), pour qu'à l'intérieur on a l'impression d'être dans un espace fermé.

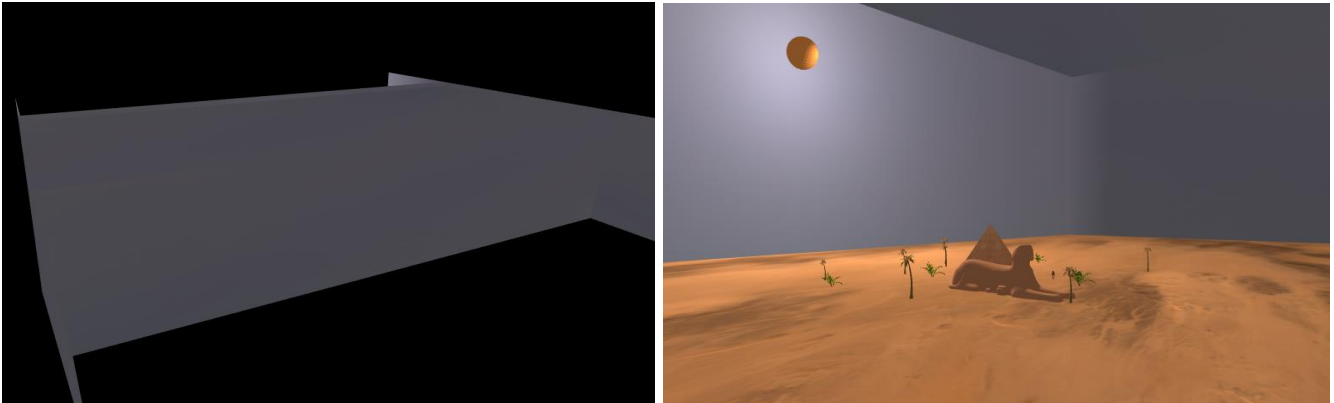


Figure 3.2 : «Skybox»

#### 4. Importation des objets :

Afin d'importer des modèles 3D déjà prêts on a utilisé la fonction « loadOBJ » qui se trouve dans le fichier « objLoad.cpp ».

##### i. Le sphinx :

```
glBindVertexArray(VAO8);

//up date the view
View = lookAt(camPos, camDirection, camUp);
Model = mat4(1.0f);
Model = translate(Model, vec3(-4.0f, -1.15f, 4.0f));
Model = scale(Model, vec3(0.05f, 0.05f, 0.05f));

MVP = Projection * View * Model;

ModelID = glGetUniformLocation(ShaderProgram, "Model");
glUniformMatrix4fv(ModelID, 1, GL_FALSE, &Model[0][0]);

MatrixID = glGetUniformLocation(ShaderProgram, "MVP");

glUniformMatrix4fv(MatrixID, 1, GL_FALSE, &MVP[0][0]);
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, text7);
glUniform1i(TextureID, 0);
glDrawArrays(GL_TRIANGLES, 0, 186336);
```

Figure 4.1 : «Code de l'affichage du sphinx»



Figure 4.2 : «le sphinx»

## ii. Les palmiers : on a affiché plusieurs types palmiers avec différentes positions, tailles et rotations.

```
//up date the view
View = lookAt(camPos, camDirection, camUp);
Model = mat4(1.0f);
Model = translate(Model, vec3(0.0f, -1.0f, 5.0f));
Model = scale(Model, vec3(0.05f, 0.05f, 0.05f));

MVP = Projection * View * Model;

ModelID = glGetUniformLocation(ShaderProgram, "Model");
glUniformMatrix4fv(ModelID, 1, GL_FALSE, &Model[0][0]);

MatrixID = glGetUniformLocation(ShaderProgram, "MVP");

glUniformMatrix4fv(MatrixID, 1, GL_FALSE, &MVP[0][0]);
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, text4);
glUniform1i(TextureID, 0);
glDrawArrays(GL_TRIANGLES, 0, 18561);

View = lookAt(camPos, camDirection, camUp);
Model = mat4(1.0f);
Model = translate(Model, vec3(5.0f, -1.0f, -3.0f));
Model = scale(Model, vec3(0.03f, 0.03f, 0.03f));

ModelID = glGetUniformLocation(ShaderProgram, "Model");
glUniformMatrix4fv(ModelID, 1, GL_FALSE, &Model[0][0]);

MVP = Projection * View * Model;

MatrixID = glGetUniformLocation(ShaderProgram, "MVP");

glUniformMatrix4fv(MatrixID, 1, GL_FALSE, &MVP[0][0]);
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, text4);
glUniform1i(TextureID, 0);
glDrawArrays(GL_TRIANGLES, 0, 18561);
```

Figure 4.3 : «un exemple sur l’affichage de deux palmiers »

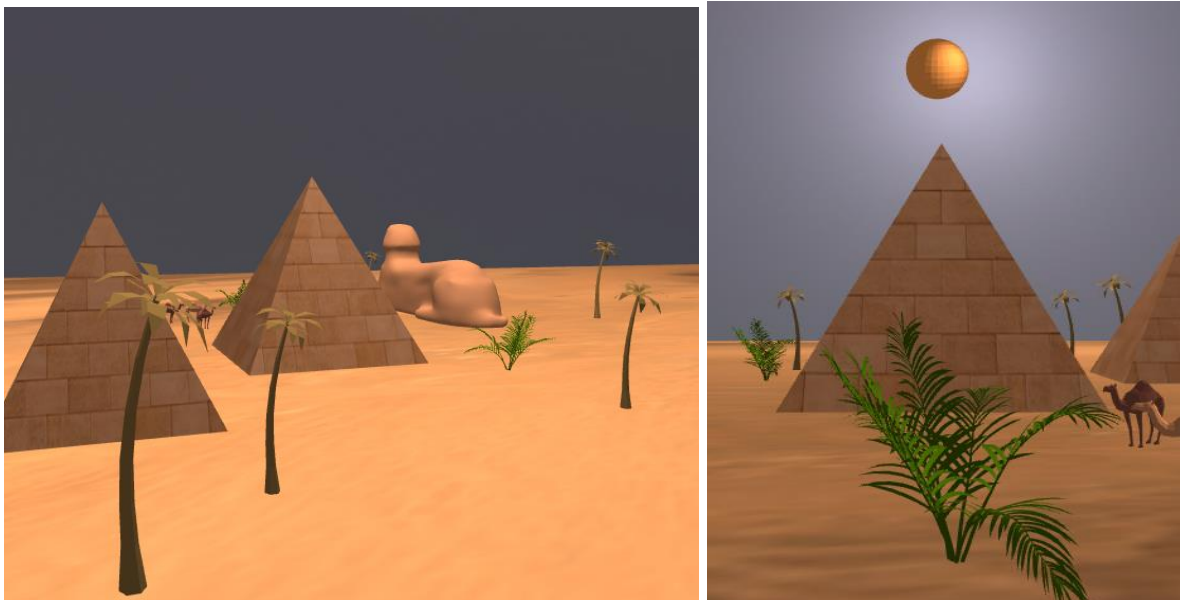


Figure 4.4 : « les deux types de palmiers importés »

## iii. Les chameaux :

```
glBindVertexArray(VAO9);

//up date the view
View = lookAt(camPos, camDirection, camUp);
Model = mat4(1.0f);

Model = translate(Model, vec3(1.0f, -0.8f, 3.0f));
Model = scale(Model, vec3(0.2f, 0.2f, 0.2f));
Model = rotate(Model, radians(-50.0f), vec3(0.0f, 1.0f, 0.0f));

MVP = Projection * View * Model;

ModelID = glGetUniformLocation(ShaderProgram, "Model");
glUniformMatrix4fv(ModelID, 1, GL_FALSE, &Model[0][0]);

MatrixID = glGetUniformLocation(ShaderProgram, "MVP");

glUniformMatrix4fv(MatrixID, 1, GL_FALSE, &MVP[0][0]);
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, text8);
glUniform1i(TextureID, 0);
glDrawArrays(GL_TRIANGLES, 0, 2466);
```

Figure 4.5 : «code de l’affichage des chameaux»

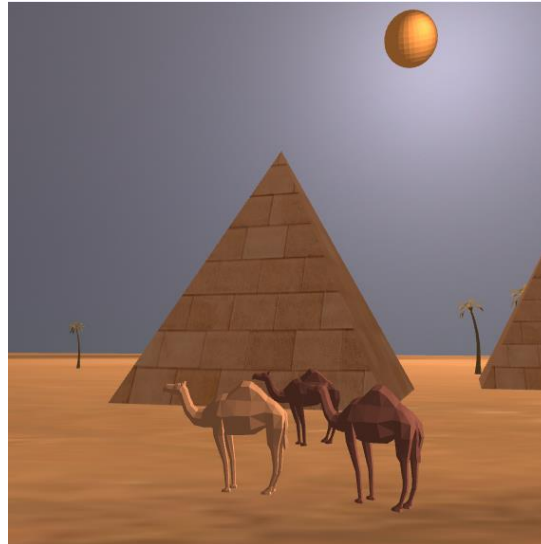


Figure 4.6 : «code de l’affichage des chameaux»

## 5. éclairage :

On a importé une sphère qui va faire office de soleil (lune en mode nuit) avec une source lumineuse (diffuse) qui bouge en fonction du soleil pour donner l’impression que c’est la sphère qui émet de la lumière .

```
float ambientIntens = 0.6;
vec3 normal = normalize(vNormal);
vec3 ambient= lightColor * ambientIntens;
vec3 lightDir = normalize(lightPos - vPos);
float diff = max(dot(normal, lightDir), 0.0);
vec3 diffuse = diff * lightColor;
vec4 res = texture(ourTexture, vTexture);
color = vec4(vec4(ambient + diffuse,1.0 ) * res);
```

Figure 5.1 : «Code du fragment shader»

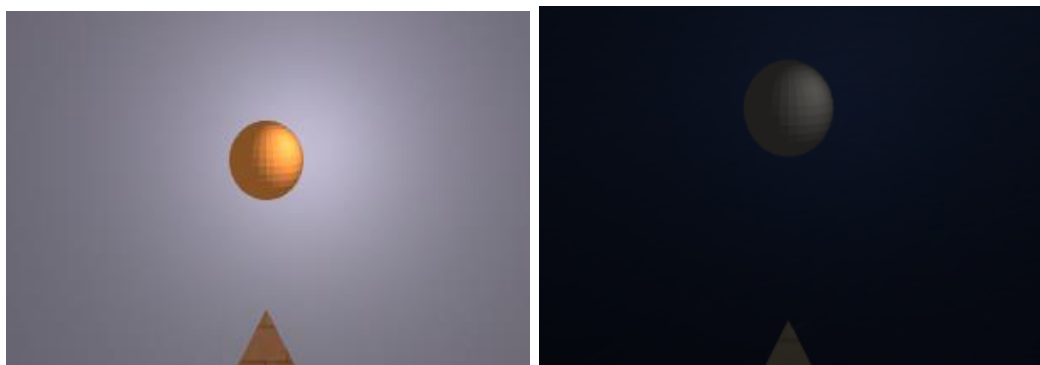


Figure 5.1 : « Sphère et source lumineuse »

## 6. Interactions et mouvements :

Pour les objets en mouvements on a le soleil (source lumineuse) qui tourne au tour de la pyramide centrale comme le montre la figure suivante :



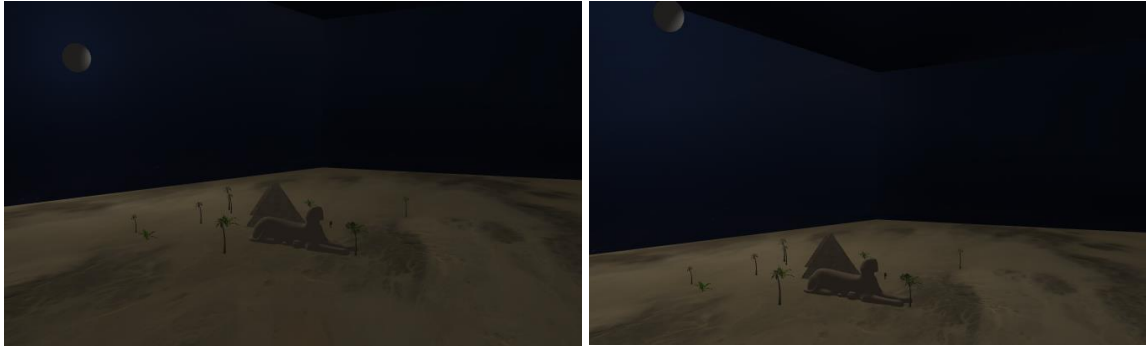


Figure 6.1 : «le mouvement de la lune et de la source lumineuse»

En ce qui concerne les interactions avec l'utilisateur, d'abord l'utilisateur a la possibilité de faire bouger la caméra pour découvrir la scène, en deuxième lieu basculer entre le mode jour et nuit en utilisant les touches « W » (mode jour) et « X » (mode nuit), enfin il peut contrôler la vitesse de rotation du soleil (lune) avec les touches « I » pour accélérer le mouvement « K » pour le ralentir.

## 7. Audio :

Au l' lancement du programme un audio se lancera en utilisant le code suivant :

```
//audio
PlaySound(TEXT("audio/desert1.wav"), NULL, SND_ASYNC);
```

Figure 6.1 : «Audio Player»

## 8. Conclusion :

A l'aide de ce projet on a pu revoir tous les aspects et techniques vu en TP et les appliquer tout le long du travail.