

TP N°3

Les matrices de visualisation et de projection (rendu direct)

Afin d'utiliser les fonctionnalités suivantes, il faut ajouter le lien vers libglu32.a dans le linker settings. Si les fonctions commençant par « glu » ne fonctionnent pas, il faut ajouter l'option `GLFW_INCLUDE_GLU` dans Compiler settings > #define.

La matrice de visualisation (*modelview*)

On prévient OpenGL que la matrice courante est MODELVIEW et on l'initialise. Elle est utilisée pour les transformations de modélisation ainsi que celles de la visualisation.

Afin de positionner la caméra et l'orienter, on utilise la fonction suivante :

```
gluLookAt(0,0,1, 0,0,0, 0,1,0);
```

Les 3 premiers paramètres sont les coordonnées de la caméra (origine du repère de visualisation), les 3 suivants sont les coordonnées du point de référence et les 3 derniers montrent la direction du UP vector (Y positive dans l'exemple).

La matrice de projection

Afin d'appliquer la projection sur notre scène, on insère le code suivant :

```
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
gluPerspective( 45, 1, 0.1, 100 );
```

On remarque que le type de projection ici est perspective. Elle est utilisée en changeant le mode de la matrice utilisée à la matrice de projection.

Les paramètres de `gluPerspective` sont dans l'ordre: l'angle du champ de vision selon l'axe verticale des Y, un ratio x/y pour le champ de vision selon l'axe horizontale des X, le plan le plus proche et celui le plus éloigné qui délimitent le volume de visualisation.

Pour la projection orthogonale, on utilise `glOrtho(gauche, droite, bas, haut, plan_proche, plan_loin)`.

Exercices:

- Appliquer une projection perspective sur le dessin du TP précédent. Est-ce que ce dernier est visible ?
- Positionner la caméra sur l'axe des Z afin de rendre la scène visible.
- Changer les paramètres de `gluLookAt` pour s'éloigner et se rapprocher, déplacer le point de référence puis pour faire des rotations de la caméra.

Annexe:

Afin que la taille des objets suit le redimensionnement de la fenêtre, il faut faire appel à une fonction dès qu'il y a réaffichage de la fenêtre:

```
void Reshape(GLFWwindow* window, int width, int height)  
{  
    glViewport(0, 0, width, height);  
}
```

Les paramètres de `glViewport` sont les coordonnées du coin bas gauche du viewport suivies de ses dimensions. L'appel se fait dans la fonction main via l'évènement:

```
glfwSetFramebufferSizeCallback(window, Reshape);
```