

TP N°1

Introduction à OpenGL

Définitions :

OpenGL (Open Graphics Library) est une bibliothèque graphique qui permet de développer des applications 2D ou 3D.

Il est possible d'utiliser OpenGL en Python, Delphi, Java, et les commandes y sont similaires. Ici nous utiliserons le C/C++.

Les différents utilitaires utilisés avec OpenGL :

- OpenGL-ES

Open Graphics Library for Embedded System (OGLES ou GLES) est une dérivée d'OpenGL adaptée aux plateformes embarquées telles que les téléphones mobiles, les tablettes, les consoles de jeux vidéo portables, ... etc

- WebGL

Web Graphics Library, est une bibliothèque JavaScript pour le rendu graphique 2D et 3D. Elle est basée sur OpenGL-ES.

- GLU

OpenGL Utility Library, est une bibliothèque d'OpenGL qui contient des fonctions de rendu et de mappage entre l'écran et les coordonnées réelles.

- GLEW

OpenGL Extension Wrangler Library, est une bibliothèque OpenGL qui gère le chargement des extensions OpenGL.

- GLUT

OpenGL Utility Toolkit, est une bibliothèque OpenGL qui contrôle la fenêtre et récupère les entrées du clavier et de la souris.

- SDL

Simple DirectMedia Layer, est une bibliothèque destinée à gérer les fenêtres, clavier et souris. Elle peut être utilisée avec OpenGL mais aussi avec d'autres API graphiques telles que DirectX.

- GLFW : est une bibliothèque multi-plateforme destinée au développement d'applications OpenGL. Elle permet de créer des fenêtres, récupérer les entrées et gérer les événements.

Elle est Open Source et son développement est très actif.

Configuration de GLFW avec codeblocks :

Dans ce TP, nous allons utiliser la bibliothèque GLFW avec codeblocks.

Les dll nécessaires à opengl sont déjà présente dans le système. Quant à GLFW, il faut télécharger les fichiers binaires précompilés du site : glfw.org/download.

Ensuite, suivre les étapes suivantes :

- Créer un dossier d'installation 'GLFW' dans l'emplacement de votre choix.
- Copier les dossiers « include » et « lib » (lib-mingw-w64 ou lib-mingw, selon votre système ou autre, selon votre compilateur) à partir des fichiers téléchargés et décompressés vers votre dossier d'installation.

- Copier « glfw3.dll » dans votre dossier système.
- Créer un nouveau projet application console dans codeblocks et aller dans ses propriétés, « Project's build options » puis dans « Linker setting » ajouter libopengl32.a et libgdi32.a (qui se trouvent dans le dossier de votre compilateur) ainsi que libglfw3.a et libglfw3dll.a (qui sont dans votre dossier GLFW).
- Aller dans « Search directories » et ajouter les chemins vers votre dossier « include » et votre dossier « lib » dans les anglets « compiler » et « Linker » respectivement.

Ouverture d'une fenêtre :

Dans le source de votre projet, ajouter la directive :
`#include <GLFW/glfw3.h>`

Dans la fonction main :

- Initialiser glfw :

```
if (!glfwInit()) {  
    printf("Could not initialize glfw.\n");  
    return -1;  
}
```

- Créer la fenêtre :

```
GLFWwindow* window;  
window = glfwCreateWindow(640, 480, "OpenGL TP 1", NULL, NULL);  
if (!window)  
{  
    glfwTerminate();  
    return -1;  
}  
glfwMakeContextCurrent(window);
```

- Définir la couleur de l'arrière plan :
`glClearColor(0.5f, 0.5f, 0.5f, 1.0f);`

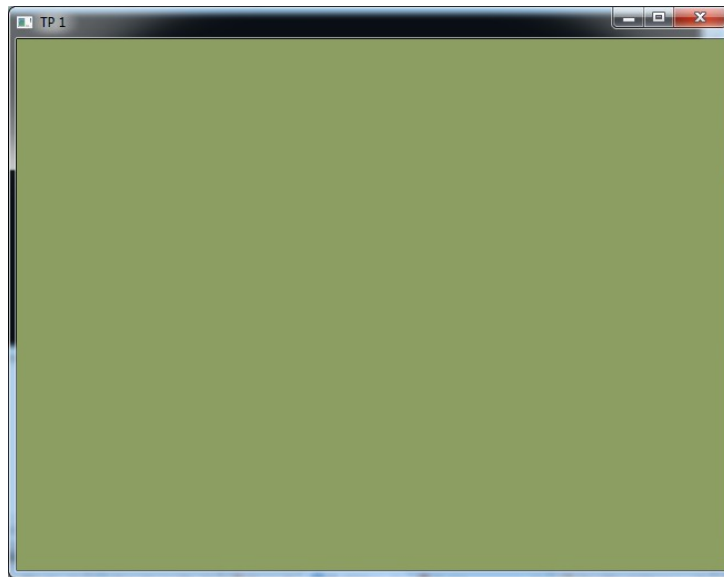
- Définir la boucle d'affichage. Elle sera appelée à chaque fois que la fenêtre sera rafraîchie.
`while (!glfwWindowShouldClose(window))`

```
{  
    glClear(GL_COLOR_BUFFER_BIT);  
    glfwSwapBuffers(window);  
    glfwPollEvents();  
}
```

A la sortie de la boucle, on détruit la fenêtre:

```
glfwDestroyWindow(window);  
glfwTerminate();
```

A ce niveau, nous avons une fenêtre qui s'affiche.



Dessin 2D (rendu direct) :

On utilise `glBegin()` pour dessiner les éléments primitifs avec comme paramètre l'élément, on lui envoie les coordonnées des vecteurs-clés via `glVertex()`, et on termine avec `glEnd()`. `glVertex` représente un sommet en OpenGL et s'écrit de cette manière : `glVertexXT`, où :
X définit le nombre de dimension, entre 2 et 4.
T définit le type des coordonnées : 'i' pour int, 's' pour short, 'f' pour float et 'd' pour double.

- Ajouter le code suivant dans la fonction d'affichage afin de dessiner un triangle avant `glfwSwapBuffers` :

```
glBegin(GL_TRIANGLES);  
    glVertex2i(0,1);  
    glVertex2i(-1,0);  
    glVertex2i(1,0);  
glEnd();
```

Par défaut, les coordonnées sont dans l'intervalle $[-1,1]$.

