TP3 pt.1

PAULIN Maxime PRUVOST Jordan

November 24, 2021

2 Préparation

PyGames

2.1

On crée une nouvelle fenêtre de taille 300x200 qui disparait instantanément.

2.2

Cette fois ci la fenêtre n'affiche que des pixels noirs ou artefacts si l'on est (mal)chanceux. Elle disparait lors de l'appuie d'une touche.

OpenGL

```
import pygame
import OpenGL.GL as gl
import OpenGL.GLU as glu

if __name__ == '__main__':
    pygame.init()
    display=(600,600)
    pygame.display.set_mode(display, pygame.DOUBLEBUF | pygame.OPENGL)

    # Sets the screen color (white)
    gl.glClearColor(1, 1, 1, 1)
    # Clears the buffers and sets DEPTH_TEST to remove hidden surfaces
    gl.glClear(gl.GL_COLOR_BUFFER_BIT | gl.GL_DEPTH_BUFFER_BIT)
    gl.glEnable(gl.GL_DEPTH_TEST)

    glu.gluPerspective(45, 1, .1, 50)

    while True:
        for event in pygame.event.get():
          if event.type == pygame.QUIT:
            pygame.quit()
                exit()
```

2.3

```python
import pygame
import OpenGL.GL as gl
import OpenGL.GLU as glu
from time import sleep

if __name__ == '__main__':
    pygame.init()
    display=(600,600)
    pygame.display.set_mode(display, pygame.DOUBLEBUF | pygame.OPENGL)

    # Sets the screen color (white)
    gl.glClearColor(1, 1, 1, 1)
    # Clears the buffers and sets DEPTH_TEST to remove hidden surfaces
    gl.glClear(gl.GL_COLOR_BUFFER_BIT | gl.GL_DEPTH_BUFFER_BIT)
    gl.glEnable(gl.GL_DEPTH_TEST)

    # Placer ici l'utilisation de gluPerspective.
    fovy, aspect, zNear, zFar = 45, 1, .1, 50
    glu.gluPerspective (fovy, aspect, zNear, zFar)

    ### Pour la question 3
    gl.glTranslatef(0,2,-5)
    gl.glRotatef(-90,1,0,0)


    gl.glBegin(gl.GL_LINES) # Indique que l'on va commencer un trace en mode lignes (
    segments)
    gl.glColor3fv([255, 0, 0]) # Indique la couleur du prochian segment en RGB
    gl.glVertex3fv((0, 0, -2)) # Premier vertice : depart de la ligne
    gl.glVertex3fv((0, 1, -2)) # Deuxieme vertice : fin de la ligne
    gl.glColor3fv([0, 0, 255]) # 2e trais
    gl.glVertex3fv((0, 0, -2))
    gl.glVertex3fv((1, 0, -2))
    gl.glColor3fv([0, 255, 0]) # 3e trais
    gl.glVertex3fv((0, 0, -2))
    gl.glVertex3fv((0, 0, -3)) # on ne le voie pas sans rotation
    gl.glEnd() # Find du trace
    pygame.display.flip() # Met a jour l'affichage de la fenetre graphique

    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                exit()
```

Découverte de l'environnement du TP

Le fichier Configuration contient une classe Configuration dont le constructeur prends comme argument un dictionnaire comprenant : axes, un booléen; xAxisColor, un triplet de float, idem pour yAxisColor zAxisColor; et finalement screenPosition, la distance (float) au root de la scène.
Le constructeur initialise un écran pygame de 800x600, et active openGL de manière à avoir de la perspective. Il initialise aussi une matrice de transformation 4x4. Cette classe comprend les fonction getParameter et setParameter, dont le nom est explicite. Elle comprend aussi la fonction draw qui dessine les 3 axes sur le root de la scène, avec les couleurs définies dans le constructeur. La fonction display met à jour l'écran avec les données des matrices représentant les objets. Les autres fonctions prennent en charge les input de la souris et du clavier.

Quant au fichier Main, il import tout les constructeur des autres fichier de son dossier et nous permet de faire une fonction par question.

1.1

```python
Configuration({'screenPosition': -5, 'xAxisColor': [1, 1, 0]}).display()
```

Cette ligne recule la position de l'écran de projection de 5 unités et change la couleur de l'axe X vers un jaune, et l'affiche. Cependant, comme écrite dans l'énoncé, la Configuration retournée présente déjà

un display, que l'on ne peut pas re-display. Il faut donc l'enlever au sein de la fonction. La chaine était cependant bien possible car Configuration(...).setParameter(...) est bien un objet de type Configuration, donc il est possible de le .display(). On a besoin de reculer le plan de projection pour qu'un axe ne clip pas dedans.

1.2

Les touches pageUp et pageDown sont respectivement représentées dans pygame avec : pygame.K_PAGEUP et pygame.K_PAGEDOWN

2

2.2

Configuration().add(Section(...)).display() ajoute à un nouvel objet Configuration un nouvel objet Section et le render.

3

3.1

Le constructeur de Wall est assez similaire à celui de Section, mais Wall ne prends pas d'edge en parametre. De plus, Wall stocke une liste d'objets. Cette liste à l'air d'être faite pour ne contenir que des Section liées à ce mur.