COIN Alix
LEGRAND Léo-Paul
15/11/2021

Rapport de TP3 – Représentation visuelle d'objets

I. Introduction

Le but de ce TP est d'afficher une fenêtre graphique interactive avec python contenant des objets 3D.

II. Travail Préparatoire

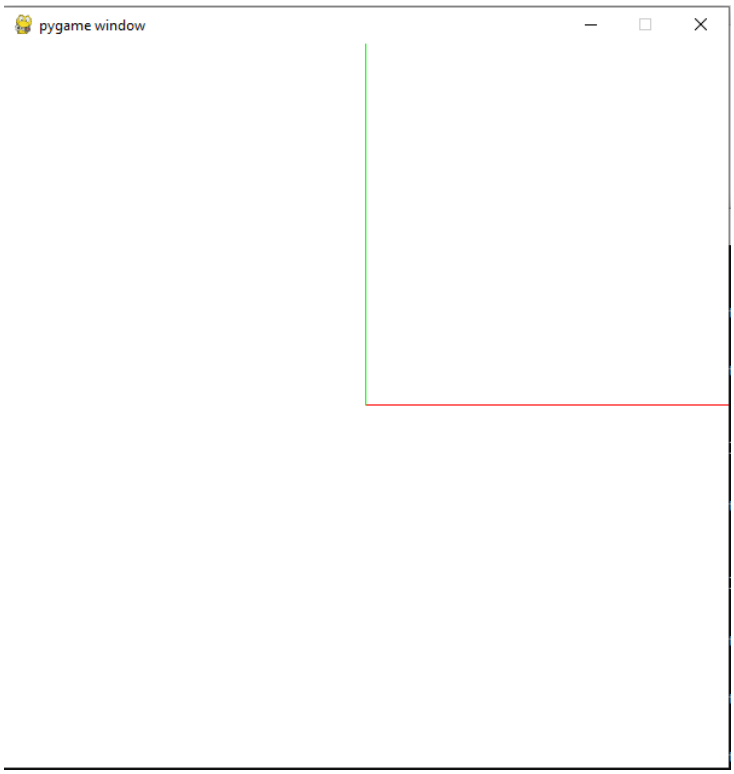
1. Question (1).

On importe la librairie pygame, on initialise le module, on crée une fenêtre de taille (200, 300) et on ferme cette fenêtre et quitte.

2. Question (2).

On fait la même chose que précédemment, mais au lieu de quitter juste après avoir créé la fenêtre, on attend que l'utilisateur appuie sur une touche du clavier pour quitter le programme.

3. Question (3)

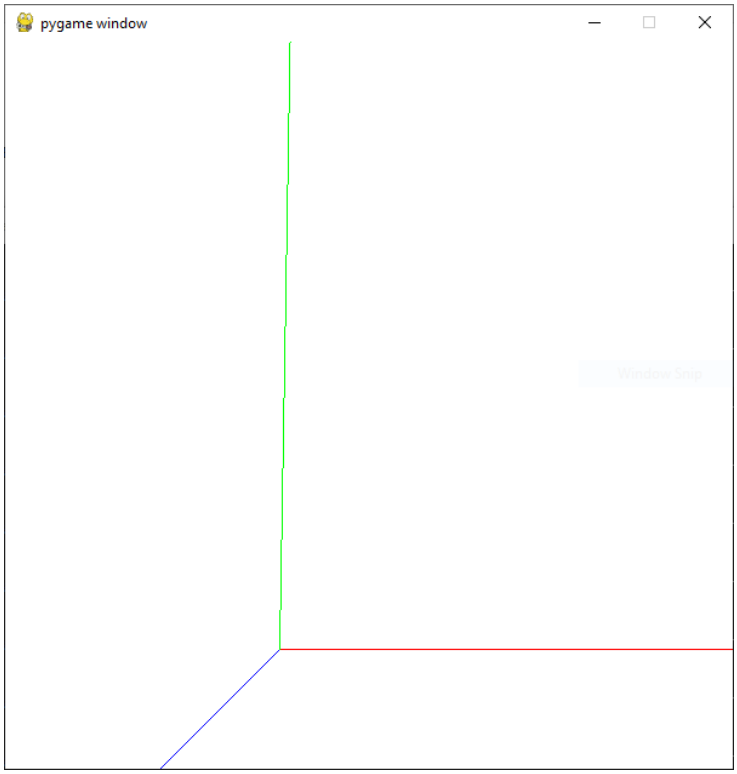


4. Question (4)

En utilisant :

gl.glTranslatef(-0.2, -0.2, 0)

gl.glRotatef(-10, 1, 0, 0)



5- Question (1)

Le fichier main.py contient une multitude de fonctions (*Qx…*) qui génèrent des configurations différentes d'affichage avec plus ou moins d'objets et les affichent. Une *configuration* est un objet qui initialise Pygame et OpenGL selon les paramètres passés dans son constructeur, et auquel on peut ajouter des objets. On obtient la représentation des axes x y z au centre de l'écran.

Le chaînage de *setParameter* et *display* est possible car *setParameter* retourne un objet *configuration*.

Le traitement particulier de *screenPosition* est effectué car sa modification implique de modifier la matrice de projection.

On ajoute *gl.glRotatef(-90, 1, 0, 0)* à la fin de la méthode.

III. Mise en place des interactions avec l'utilisateur avec Pygame

1. Question (1)

Si la touche « Page Up » ou « Page Down » est pressée, on choisit de modifier la matrice ModelView qu'on multiplie par 1,1 ou 1/1,1 grace à la fonction gl.scalef

Pour que le code fonctionne, il est nécessaire d'utiliser les codes des touches « Page Up » et « Page Down » (1073741899 et 1073741902), obtenus grâce à la fonction print(self.event.key)

2. Question (2)

De la même manière, si (IF) la molette est actionnée vers le haut (mousebutton == 4) ou (ELIF) vers le bas (mousebutton == 5)

3. Question (3)

La methode processmouseMotionEvent gère le clic gauche pour la rotation et le droit pour la translation.

La rotation pouvant s'effectuer autour de deux axes, il est nécessaire d'implémenter deux lignes, une pour x et une pour z. On utilise la fonction gl.glRotatef.

Gl.glRotatef(self.event.rel[1]*0.5, 1, 0, 0) -> si la souris est déplacée le long de l'axe y (1), la fonction effectue une rotation de la moitié de la valeur du déplacement autour de l'axe x.

Gl.glRotatef(self.event.rel[0]*0.5, 0, 0, 1) -> si la souris est si la souris est déplacée le long de l'axe x (0), la fonction effectue une rotation de la moitié de la valeur du déplacement autour de l'axe z.

Pour la translation, gl.Translatef permet de translater sur trois axes en même temps : une seule ligne est donc nécesaire :

gl.glTranslatef(self.event.rel[0]*0.1, 0, self.event.rel[1]*0,1) -> si la souris est déplacée le long de l'axe x, la fonction effectue une translation sur x. Si elle est déplacée sur y, la fonction est déplacée sur z.

IV. Création d'une section

1. Question (1)

La méthode generate (self) contient deux fonctions :

- La première (self.vertices) permet de créer une liste de points dont les coordonnées sont définies grâces aux paramètres width, height, thickness. Huit points sont nécessaires à la définition d'un parallélépipède rectangle.
- La seconde (self.faces) créé les faces en associant quatre points dans les sens anti horaire.

2. Question (2)

La fonction Q2b() permet, grâce à Configuration().add(section), d'initialiser OpenGL et PyGames, avant de créer une instance de la classe section (donc un mur)