MOREL Rémi

DUMIRE Thibault

Date du TP (09/12/2021)

# Rapport de TP3 – Représentation visuelle d’objet 3D

## Introduction

Nous allons vous présenter notre compte rendu de notre TP qui a pour objectif la représentation visuelle d’objet 3D. Pour réaliser ceci, nous allons utiliser les différents fichiers mis à notre disposition. Qui à la fin, pourra nous permettre de voir une image 3D et de pouvoir réaliser des zooms, des translations et des rotations. Pour l’image 3D nous allons réaliser une maison qu’on pourra observer. Pour cela on va réaliser à l’aide d’objets simples que l’on va construire comme des murs, des portes, des fenêtres, …

## Préparation avant TP

### Utilisation de Pygame

Nous avons copié le code, nous avons observé l’ouverture d’une fenêtre de dimension 200x300 et sa fermeture juste après car on lui demande de se fermer à l’aide de ligne 4.

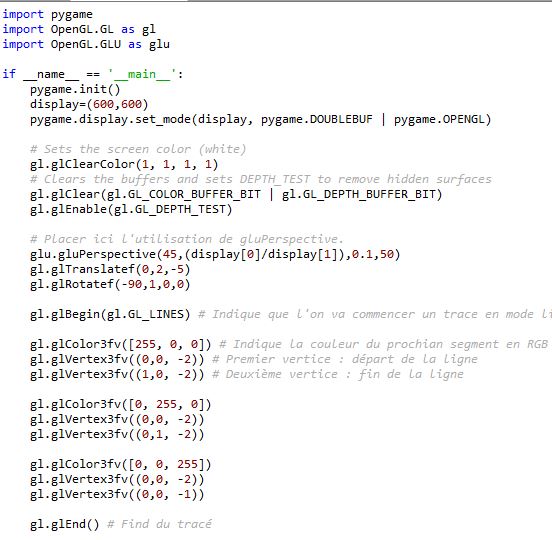
Nous avons ensuite copié le second code qui nous permet de fermer la fenêtre en appuyant sur une touche du clavier. Ce code enregistre les différents évènements et vérifie s’il y a un évènement où on a appuyé sur une touche de notre clavier à l’aide de la commande « pygame.KEYDOM » une fois validé on sort de la boucle d’enregistrement des évènement et il ferme la fenêtre.

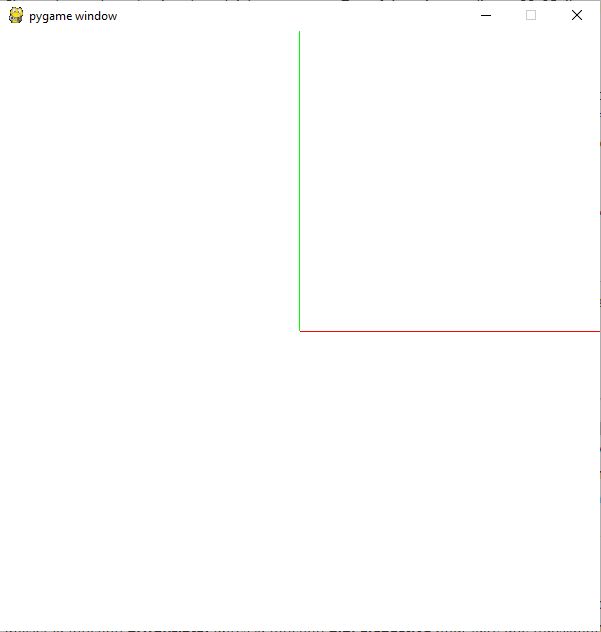
Une image contenant texte

Description générée automatiquement

### Utilisation de Pyopengl

Pour cette partie nous avons réalisé les différentes commandes pour tracer les différents axes représentant l’axe x, y et z pour pouvoir visualiser les différents axes à l’aide du code suivant :

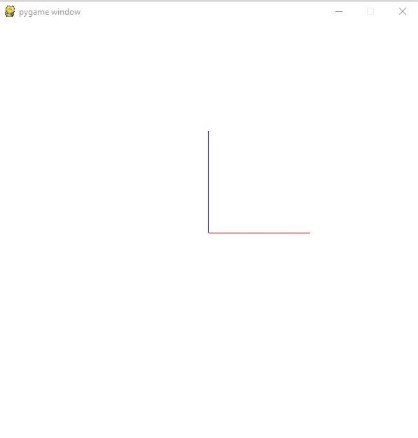




On obtient en premier cette photo où on peut observer l’axe x et y :

On a rajouté les commande gl.glTranslatef(0,2,-5) pour recentrer l’image et gl.glRotatef(-90,1,0,0) pour réaliser une rotation de -90° autour de l’axe x pour pouvoir observer l’axe z.

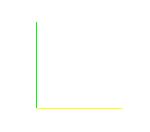
On obtient l’image suivante :



### Découverte de l’environnement de travail du TP

* 1a) On a ajouté la fonction Q1a() au fichier main celle-ci lance la classe configuration du fichier configuration.py. A l’intérieur de cette classe il y a plusieurs méthodes permettant l’affichage d’une fenêtre graphique avec tracé les axes. Dans cette classe il y a une méthode permettant de réaliser des rotations selon l’axe z en appuyant soit sur « z » soit sur « Z » dans un sens puis dans l’autre. On a aussi une méthode pour la touche « a » qui permet de mettre en pose la fenêtre graphique pendant un temps défini en paramètre. (affiche une fenêtre graphique blanche)
* 1b) 

En Initialisant les paramètre d’un objet de classe Configuration on obtient une fenêtre graphique avec deux axes qui sont plus proche que la configuration par défaut et elle change la couleur de l’axe x passant de la couleur rouge à la couleur jaune.



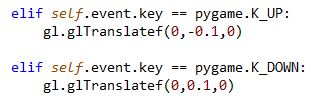
Le chainage entre setParameter() et display() est possible car la méthode setParameter() renvoie self un objet de la classe configuration.

Le traitement particulier dans le setter pour le paramètre screenPosition est due au fait que lorsque l’on change la position de l’écran il faut changer l’affichage dans la fenêtre graphique et garder le centrage.

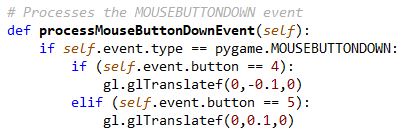
* 1c) On a rajouté une instruction dans la méthode initializeTransformationMatrix(). L’instruction est la suivante : gl.glRotatef(-90,1,0,0) après gl.glTranslatef pour avoir une rotation de -90° selon l’axe x pour avoir l’axe des z verticale et que l’axe des y soit notre profondeur.

## Mise en place des interactions avec l’utilisateur avec Pygame

* 1d) Nous avons ajouté des instructions dans la méthode processKeyDownEvent pour pouvoir assurer la gestion des touches « Page Up » et « Page Down » qui nous permettra par la suite un zoom ou un dézoom.



* 1e) Nous modifions maintenant la méthode processMouseButtonDownEvent pour pouvoir zoomer ou dézoomer à l’aide de la souris. Pour cela nous avons mis la méthode suivante :



Cette méthode vérifie que l’évènement de la molette est vérifié et vérifie le sens de rotation de la molette en vérifiant si le Button4 ou le Button5 est activer. On rajoute ainsi la commande demandée pour chaque condition.

* 1f) Pour cette question, nous allons modifier la méthode processMouseMotionEvent pour obtenir une rotation et une translation selon les touches de la souris et du déplacement de la souris.

## Conclusion