WALLERAND Alex
SIMONET Théo
Date du TP (04/01/2022)

Rapport de TP3 Séance 2 – Représentation visuelle d'objets

I. Introduction

Nous allons faire la suite du TP3, nous devons continuer la création d'une section puis nous passerons au murs et tenterons de créer la maison.

II. Création d'une section

1. Question (2b).

Cette instruction permet d'initialiser la première face dans l'interface graphique avec des paramètres de taille.

```
def draw(self):
    gl.glPushMatrix()
    if self.parameters['edges'] == True:
        self.drawEdges()
    for faces in self.faces:
        gl.glPolygonMode(gl.GL_FRONT_AND_BACK, gl.GL_FILL)
        gl.glBegin(gl.GL_QUADS)
        gl.glColor3fv(self.parameters['color'])
        gl.glVertex3fv(self.vertices[faces[0]])
        gl.glVertex3fv(self.vertices[faces[1]])
        gl.glVertex3fv(self.vertices[faces[2]])
        gl.glVertex3fv(self.vertices[faces[3]])
        gl.glEnd()
    gl.glPopMatrix()
```

Pour construire chaque face, on utilise la même structure que celle de l'exemple mais on fait une boucle for pour chaque face que nous avons définie précédemment.

2. Question (2c).

```
def drawEdges(self):
    indice = [0, 1, 2, 3, 0]
    for faces in self.faces:
        for i in range(1, len(indice)):
            gl.glPolygonMode(gl.GL_FRONT_AND_BACK, gl.GL_LINE)
            gl.glBegin(gl.GL_LINES)
            gl.glColor3fv([self.parameters['color'][0]*0.8, self.parameters['color'][1]*0.8, self.parameters['color'][2]*0.8])
            gl.glVertex3fv(self.vertices[faces[indice[i-1]]])
            gl.glVertex3fv(self.vertices[faces[indice[i]]])
            gl.glEnd()
```

On écrit la méthode drawEdges de la même façon que la précédente.
Pour que la méthode drawEdges soit exécuté avant la méthode draw, on inclut dans cette dernière une condition via le setter qui quand elle est vraie exécute la méthode drawEdges.

III. Création des murs

1. Question (3a).

Le constructeur de la classe Wall vérifie si des paramètres ont été définies ou non. Si ce n'est pas le cas, alors des paramètres par défaut sont saisies.

```
def draw(self):
    gl.glPushMatrix()
    gl.glRotatef(self.parameters['orientation'], 0, 0, 1)
    self.parentSection.drawEdges()
    for objet in self.objects:
        objet.draw()
    gl.glPopMatrix()
```

On s'inspire de la méthode draw que l'on a créé pour une section, en utilisant les paramètres d'orientation données.

```
def Q3a():
    return Configuration().add(Wall({'width': 8, 'height': 2.5, 'orientation': 90, 'thickness': 0.3, 'color': [0.5, 0.5, 0.5]}))
```

Pour tracer le mur, on ajoute à la Configuration un objet de classe Wall dont on définit les paramètres ou non.

IV. Création d'une maison

1. Question(4a)

```
def draw(self):
    gl.glPushMatrix()
    gl.glTranslatef(self.parameters['position'][0],self.parameters['position'][1],0)
    gl.glRotatef(self.parameters['orientation'], 0, 0, 1)
    for objet in self.objects:
        objet.draw()
    gl.glPopMatrix()
```

La fonction draw est similaire à celle de la classe Wall.

```
def Q4a():
    # Ecriture en utilisant des variables : A compléter
    wall1 = Wall({'width': 8, 'height': 2.5})
    wall2 = Wall({'position' : [0,8,0], 'width': 8, 'height': 2.5})
    wall3 = Wall({'position' : [wall1.parameters["thickness"], -wall1.parameters["thickness"], 0], 'width': 8, 'height': 2.5, 'orientation': 90})
    wall4 = Wall({'position' : [wall1.parameters["thickness"], -wall1.parameters["width"], 0], 'width': 8, 'height': 2.5, 'orientation': 90})
    house = House({'position': [-3, 1, 0], 'orientation':0})
    house.add(wall1).add(wall3).add(wall4).add(wall2)
    return Configuration().add(house)
```

On remplace dans chaque variable wall, les paramètres pour l'objet Wall et on calcule la position de chaque mur en fonction des paramètres de taille et d'épaisseur que l'on a défini.

V. Conclusion

Nous avons pu construire la maison et ses 4 murs, mais nous n'avons pas eu le temps de continuer et faire les ouvertures de chaque mur. Nous avons donc quand même pu voir globalement les éléments les plus importants à comprendre pour les modules pyOpenGl et pyGame.
Nous avons perdu beaucoup de temps à se remettre dans le sujet du TP, car la 2ème séance s'est déroulé plus d'un mois après la 1ère avec les vacances entre deux.