Computer Graphics Lab 1

Set up and Run First OpenGL with Python

1. Create new folder for project (recommend “pyopengl”)

2. Open VS Code and open folder as created in 1.

3. Open new Terminal

4. Create Python Virtual Environment:

    (python3 or pip3 : depends on previous installation of Python)

    ```
    python -m venv pyopengl_env
    source ./pyopengl_env/bin/activate
    ```

5. Install PyOpenGL, pygame and numpy

    ```
    pip install PyOpenGL pygame numpy
    ```

6. Key-in this code and run python file

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

pygame.init()
display = (800, 600)
pygame.display.set_mode(display, DOUBLEBUF|OPENGL)

gluPerspective(45, (display[0]/display[1]), 0.1, 50.0)
glTranslatef(0.0, 0.0, -5)

# Main loop
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            quit()

    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)

    glBegin(GL_LINES)
    glVertex3fv([0.0, 0.0, 0.0])
    glVertex3fv([-1.0, -1.0, 0.0])
```

```
    glEnd()

    pygame.display.flip()
    pygame.time.wait(10)
```

Run program and let TA check the result.

8. Key in this code and let TA check the result

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

pygame.init()
display = (800, 600)
pygame.display.set_mode(display, DOUBLEBUF|OPENGL)

gluPerspective(45, (display[0]/display[1]), 0.1, 50.0)
glTranslatef(0.0, 0.0, -5)

x_position = 0.0
move_direction = 0.01

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

    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)

    glBegin(GL_LINES)
    # Changed starting positions to center the line
    glVertex3fv([0.5 + x_position, 0.0, 0.0])    # Moved right by 0.5
    glVertex3fv([-0.5 + x_position, -1.0, 0.0])  # Moved right by 0.5
    glEnd()

    x_position += move_direction

    # Adjusted boundaries for centered line
```

```
        if x_position > 1.5 or x_position < -1.5:
            move_direction = -move_direction

        pygame.display.flip()
        pygame.time.wait(10)
```

TA Check result.

9. Create an rotation 3D cube size 1x1x1. Use the example code as hint:

```
your code here

#define vertices
vertices = [
    your code here
]

#define edge
edges = [
    your code here
]

# draw cube
def Cube():
    glBegin(GL_LINES)
        your code here
    glEnd()

def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF|OPENGL)

    gluPerspective(45, (display[0]/display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    while True:
        your code here
        glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
        glRotatef(1, 0, 1, 0)  # Rotate by 1 degree around Y axis

        Cube()
        pygame.display.flip()
        pygame.time.wait(10)

main()
```

TA Check result.