Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Computer Graphics Lab Report 01

on

‘Graphics Rendering - Lab 01 Task’

Submitted By:

Reewaj Khanal (61)

Submitted to:

Mr. Dhiraj Shrestha

Assistant Professor

Department of Computer Science and Engineering

School of Engineering

Kathmandu University

Dhulikhel, Kavre

Submission Date: Thursday 25 April 2024

Question No. 1 Mention the name of the Programming language and Graphics Library you are using this semester for performing your Computer Graphics Lab and Project.

Answer:

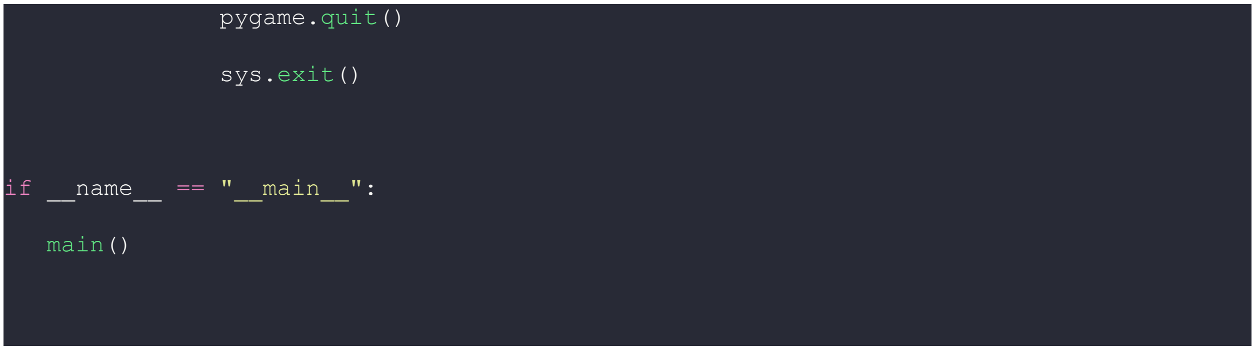
The programming language I'm using for my Computer Graphics Lab and Project this semester is **Python**. For graphics rendering, I'm using the following set of libraries:

- OpenGL (PyOpenGL) [Open Graphics Library]: For graphics rendering.
- Pygame [Python Game Development]: For creating games and multimedia applications.
- GLFW [Graphics Library Framework]: For handling windows, inputs, and OpenGL contexts.
- GLUT [OpenGL Utility Toolkit]: For creating graphical user interfaces.
- Cmath [Complex Math]: For mathematical functions in the complex domain.

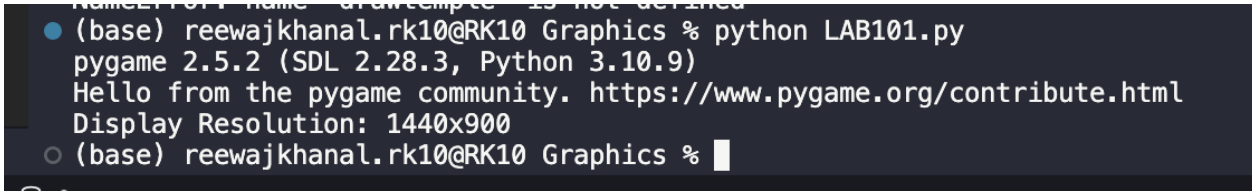
Question No. 2 Write the code snippets for setting the graphics environment in your chosen graphics library and display the resolution of your display system through functions/classes provided by your graphics library.

Answer:

```
import pygame
import sys

def main():
    # Initialize Pygame
    pygame.init()

    # Set display resolution
    screen_info = pygame.display.Info()
    screen_width = screen_info.current_w
    screen_height = screen_info.current_h

    # Set display mode
    screen = pygame.display.set_mode((screen_width, screen_height))
    pygame.display.set_caption("Graphics Environment Setup")

    # Display resolution
    print("Display Resolution: {}x{}".format(screen_width, screen_height))

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

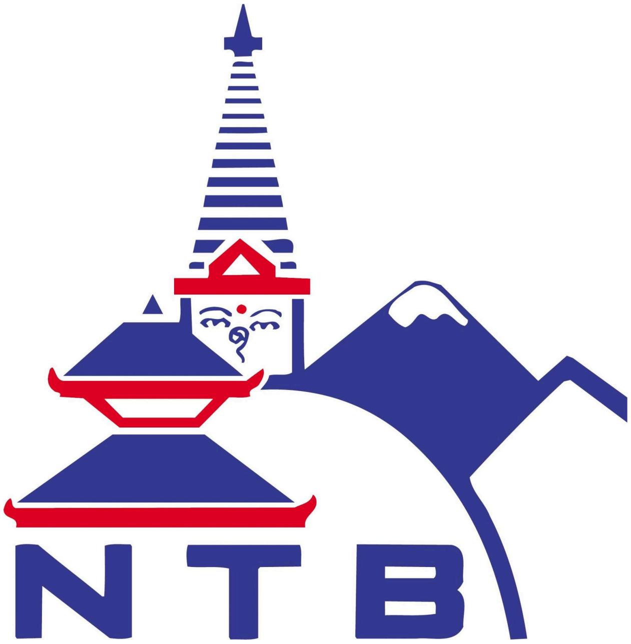


```
               pygame.quit()

               sys.exit()


if __name__ == "__main__":

   main()
```



```
NameError: name 'drawtemple' is not defined
● (base) reewajkhanal.rk10@RK10 Graphics % python LAB101.py
  pygame 2.5.2 (SDL 2.28.3, Python 3.10.9)
  Hello from the pygame community. https://www.pygame.org/contribute.html
  Display Resolution: 1440x900
○ (base) reewajkhanal.rk10@RK10 Graphics % █
```

Question No. 3 Get Familiar with the coordinate system and Draw a Logo of Nepal Tourism Board given below using the chosen Graphics geometric functions/ classes provided by the your chosen graphics library and also color the Logo accordingly.



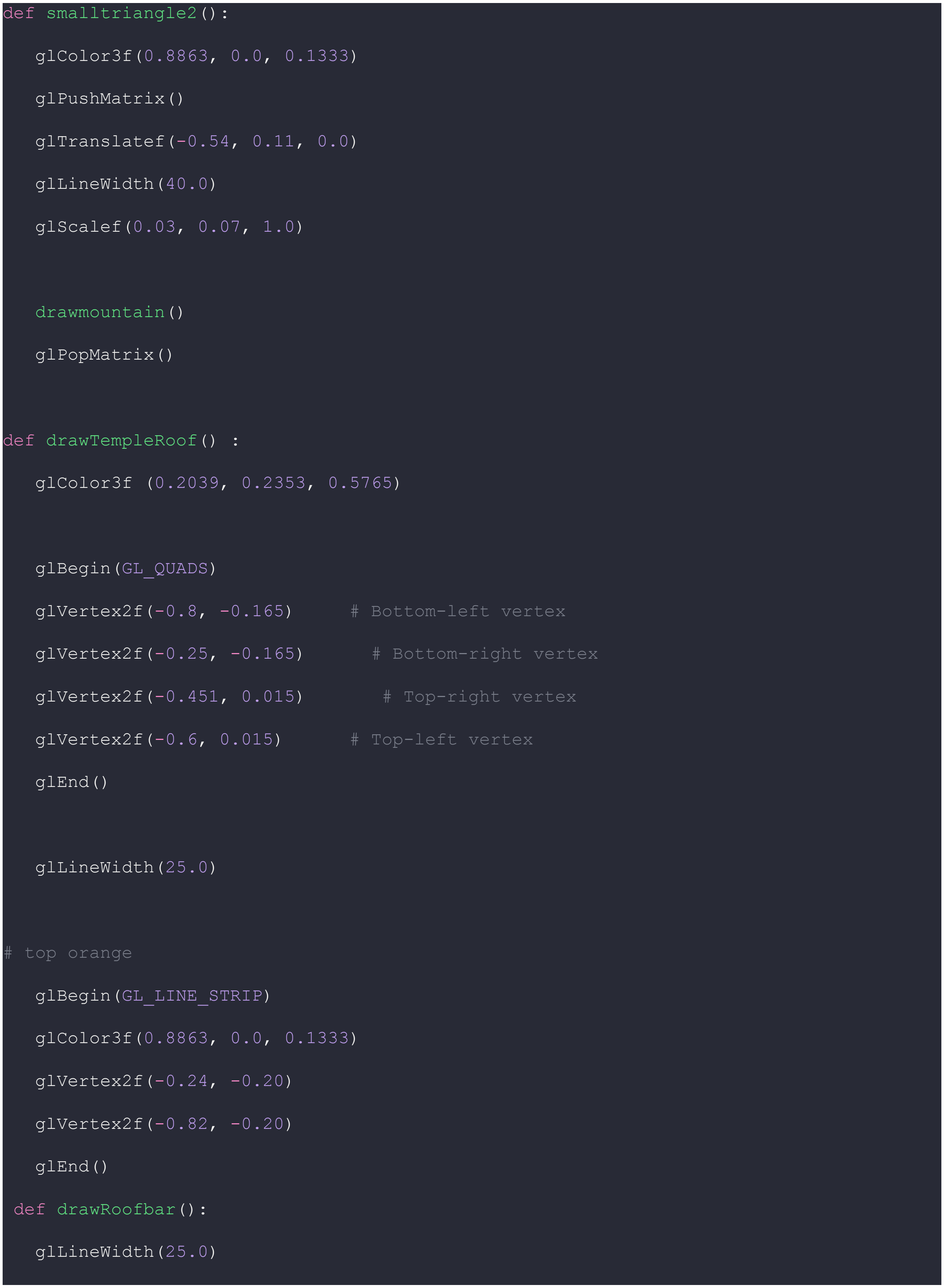
Answer:

```
import pygame

import math

from pygame.locals import *

from OpenGL.GL import *

from OpenGL.GLU import *


def main():

   pygame.init()

   display=(1440,900)
   pygame.display.set_mode(display, DOUBLEBUF|OPENGL|RESIZABLE)

   gluPerspective(40, (display[0]/display[1]), 0.1, 50.0)

   glTranslatef(0.0,0.0, -5)


   while True:

       for event in pygame.event.get():

           if event.type == pygame.QUIT:

               pygame.quit()

               quit()


       drawtoppart()

       drawmountain()

       drawsecondmountain()

       drawtempledome()

       drawRedpoly()

       drawtemple()

       ntb_text()


       pygame.display.flip()




def drawtoppart():

   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

   glLoadIdentity()

  # Set background color to white


   glClearColor(1.0, 1.0, 1.0, 1.0)


   # #Top Traingle
   # glBegin(GL_TRIANGLES)
   # glVertex2f(0,0)
   # glVertex2f(-0.05,-0.2)
   # glVertex2f(0.05,-0.2)``
   # glEnd()
   # #lines
   # glLineWidth(3)
   # glBegin(GL_LINES)
   # glVertex2f(-0.07,-0.2)
   # glVertex2f(0.07,-0.2)
   # glEnd()


def drawmountain(color=(0.2039, 0.2353, 0.5765)):
   glBegin(GL_POLYGON)
   glColor3f(color[0], color[1], color[2])
   glVertex2f(0.75, -0.1)
   glVertex2f(-0.25, -1.0)
   glVertex2f(1.75, -1.0)
   glVertex2f(0.75, -0.1)
   glEnd()

def drawsecondmountain():
   glPushMatrix()
   glTranslatef(-0.4, 0.25, 0.0)
   glLineWidth(40.0)
   glColor3f(0.2039, 0.2353, 0.5765)
   #color filling inside mountain

   drawmountain()
   glBegin(GL_POLYGON)
   glColor3f (0.2039, 0.2353, 0.5765)
   glVertex2f(0.75, -0.1)
   glVertex2f(-0.4, -1.0)
   glVertex2f(1.7, -1.0)
   glEnd()
   glPopMatrix()

   filloutsideMountain()
   drawSquare()
   drawDome()

def filloutsideMountain():
  # filling white inside outside mountain
   glPushMatrix()
   glTranslatef(0.0, -0.07, 0.0)
   glLineWidth(40.0)
   glColor3f(0.2039, 0.2353, 0.5765)

   drawmountain()
   glBegin(GL_POLYGON)
   glColor3f (1.0, 1.0, 1.0)
   glVertex2f(0.75, -0.1)
   glVertex2f(-0.4, -1.0)
   glVertex2f(1.7, -1.0)
   glEnd()
   glPopMatrix()

def drawtempledome():
    glPushMatrix()
```

```
 glTranslatef(-0.37, 0.94, 0.0)
 glLineWidth(40.0)
 glScalef(0.16, 0.74, 1.0)
 glColor3f(0.2039, 0.2353, 0.5765)

 drawmountain()
 glPopMatrix()

# # for white strips
 x_coord1 = -0.035
 x_coord2= -0.465
 y_coord=0.24
 line_width=1.0
 for  i in range (0,12) :
        glLineWidth(line_width)
        glBegin(GL_LINE_STRIP)
        glColor3f (1.0, 1.0, 1.0)
        glVertex2f(x_coord1, y_coord)
        glVertex2f(x_coord2, y_coord)
        glEnd()
        y_coord=y_coord+0.05
        line_width=line_width+0.4


# # little triangle
 glPushMatrix()
 glTranslatef(-0.268, 0.95, 0.0)
 glLineWidth(40.0)
 glScalef(0.025, 0.13, 1.0)
 glColor3f(0.2039, 0.2353, 0.5765)
```

```
    drawmountain()

    glPopMatrix()

    glColor3f (0.2039, 0.2353, 0.5765)


#little blue line


    glLineWidth(18.0)

    glBegin(GL_LINE_STRIP)

    glVertex2f(-0.23, 0.81)

    glVertex2f(-0.265, 0.81)

    glEnd()


    glLineWidth(18.0)

    glBegin(GL_LINE_STRIP)

    glVertex2f(-0.207, 0.82)

    glVertex2f(-0.290, 0.82)

    glEnd()


def drawRedpoly():

#outerwhite


   glColor3f(1.0, 1.0, 1.0)

   glBegin(GL_POLYGON)

   glVertex2f(-0.155, 0.15)   # #Botton-right

   glVertex2f(-0.35, 0.15)    # Bottom-left

   glVertex2f(-0.35, 0.23)     # Top-left

   glVertex2f(-0.255, 0.30)    # Top-middle

   glVertex2f(-0.155, 0.23)    # Top-right

   glEnd()
```

```
#red
   glColor3f(0.8863, 0.0, 0.1333)
   glBegin(GL_POLYGON)
   glVertex2f(-0.17, 0.15)    #Botton-right
   glVertex2f(-0.34, 0.15)    # Bottom-left
   glVertex2f(-0.34, 0.2)     # Top-left
   glVertex2f(-0.255, 0.28)    # Top-middle
   glVertex2f(-0.17, 0.2)    # Top-right
   glEnd()


#inner white
   glBegin(GL_POLYGON)
   glColor3f(1.0, 1.0, 1.0)
   glVertex2f(-0.19, 0.15)    #Botton-right
   glVertex2f(-0.32, 0.15)    # Bottom-left
   glVertex2f(-0.255, 0.24)    # Top-middle
   glEnd()


def drawSquare():
   glLineWidth(25.0)
 # top line part
   glBegin(GL_LINE_STRIP)
   glColor3f(0.8863, 0.0, 0.1333)
   glVertex2f(-0.035, 0.15)
   glVertex2f(-0.465, 0.15)
   glEnd()


   glLineWidth(20.0)


   glBegin(GL_LINE_STRIP)
   glColor3f (0.2039, 0.2353, 0.5765)

   glVertex2f(-0.08, 0.115)

   glVertex2f(-0.08, -0.4)

   glEnd()


   glBegin(GL_LINE_STRIP)

   glColor3f (0.2039, 0.2353, 0.5765)

   glVertex2f(-0.425, 0.115)

   glVertex2f(-0.425, -0.165)

   glEnd()

   drawtemple()


 #inside of square

   drawFace()




# def drawFace():

#     PI=3.14159265358979323846264338327950


#     glColor3f(0.8863, 0.0, 0.1333)


#     glBegin(GL_TRIANGLE_FAN)

#     centrex=-0.25

#     centrey=0.075

#     for  i in range(0,100):

#         angle = 2.0 * PI * (i) / (100)

#         x = centrex+0.015 * math.cos(angle)

#         y = centrey+0.015 * math.sin(angle)

#         glVertex2f(x, y)
#     glEnd()




def drawtemple():

   drawTempleRoof()

   glPushMatrix()


   glScalef(1.5, 1.2, 1.0)

   glTranslatef(0.17, -0.3, 0.0)


   drawTempleRoof()

   glPopMatrix()


   drawRoofbar()


# smaller triangle

   glColor3f(0.2039, 0.2353, 0.5765)


   smalltriangle()


def smalltriangle(color=(0.2039, 0.2353, 0.5765)):

   glPushMatrix()

   glTranslatef(-0.54, 0.11, 0.0)

   glLineWidth(40.0)

   glScalef(0.03, 0.07, 1.0)


   drawmountain(color)

   glPopMatrix()
```



```
def smalltriangle2():
   glColor3f(0.8863, 0.0, 0.1333)
   glPushMatrix()
   glTranslatef(-0.54, 0.11, 0.0)
   glLineWidth(40.0)
   glScalef(0.03, 0.07, 1.0)

   drawmountain()
   glPopMatrix()

def drawTempleRoof() :
   glColor3f (0.2039, 0.2353, 0.5765)

   glBegin(GL_QUADS)
   glVertex2f(-0.8, -0.165)     # Bottom-left vertex
   glVertex2f(-0.25, -0.165)      # Bottom-right vertex
   glVertex2f(-0.451, 0.015)       # Top-right vertex
   glVertex2f(-0.6, 0.015)      # Top-left vertex
   glEnd()

   glLineWidth(25.0)

# top orange
   glBegin(GL_LINE_STRIP)
   glColor3f(0.8863, 0.0, 0.1333)
   glVertex2f(-0.24, -0.20)
   glVertex2f(-0.82, -0.20)
   glEnd()
 def drawRoofbar():
   glLineWidth(25.0)
```

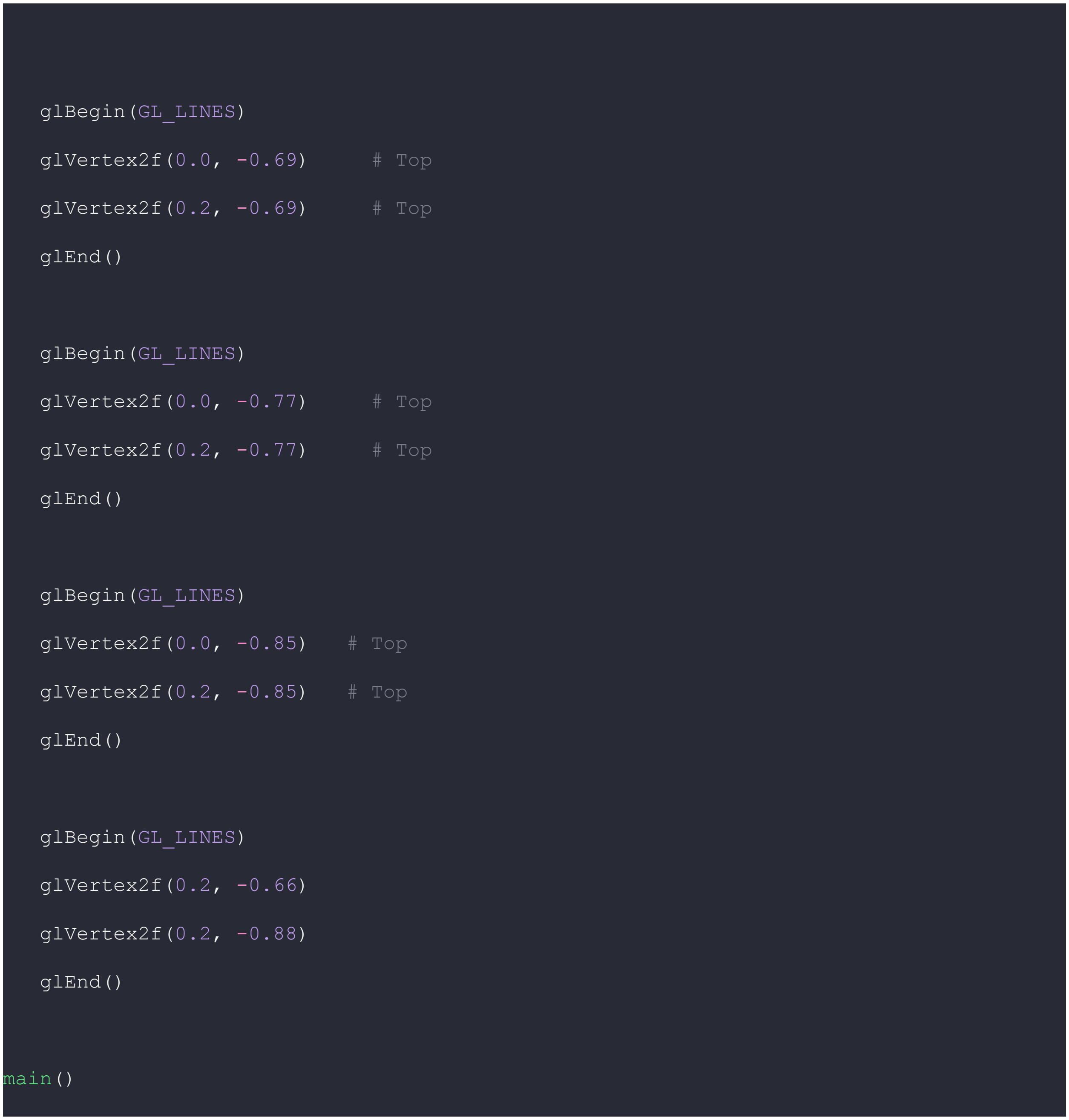
```
# top orange
   glColor3f(0.8863, 0.0, 0.1333)
   glBegin(GL_QUADS)
   glVertex2f(-0.63, -0.32)     # Bottom-left vertex
   glVertex2f(-0.42, -0.32)      # Bottom-right vertex
   glVertex2f(-0.3, -0.20)       # Top-right vertex
   glVertex2f(-0.75, -0.20)      # Top-left vertex
   glEnd()

   glColor3f(1.0, 1.0, 1.0)

   glBegin(GL_QUADS)
   glVertex2f(-0.61, -0.29)     # Bottom-left vertex
   glVertex2f(-0.44, -0.29)     # Bottom-right vertex
   glVertex2f(-0.36, -0.22)     # Top-right vertex
   glVertex2f(-0.69, -0.22)     # Top-left vertex
   glEnd()

# titled triangles
   glPushMatrix()
   glTranslatef(-0.4, 0.14, 0.0)
   glLineWidth(40.0)
   glRotatef(45.0, 0.0, 0.0, 1.0)
   smalltriangle(color=(0.8863, 0.0, 0.1333))
   glPopMatrix()

   glPushMatrix()
   glTranslatef(0.08, -0.59, 0.0)
   glLineWidth(40.0)
   glRotatef(-45.0, 0.0, 0.0, 1.0)
   glColor3f(0.8863, 0.0, 0.1333)
   smalltriangle(color=(0.8863, 0.0, 0.1333))
   glPopMatrix()

   glPushMatrix()
   glTranslatef(-0.55, -0.26, 0.0)
   glLineWidth(40.0)
   glRotatef(45.0, 0.0, 0.0, 1.0)
   glColor3f(0.8863, 0.0, 0.1333)
   smalltriangle(color=(0.8863, 0.0, 0.1333))
   glPopMatrix()

   glPushMatrix()
   glTranslatef(0.205,  -0.99, 0.0)
   glLineWidth(40.0)
   glRotatef(-45.0, 0.0, 0.0, 1.0)
   glColor3f(0.8863, 0.0, 0.1333)
   smalltriangle(color=(0.8863, 0.0, 0.1333))
   glPopMatrix()

def drawDome() : #circle waala part

   cx=-0.2
   cy=-1.0
   r=0.8
   num_segments=180
   PI=3.14159265358979323846264338327950

 # border
   glColor3f (0.2039, 0.2353, 0.5765)
   glLineWidth(40.0)
   glBegin(GL_LINE_STRIP)
   for i in range(0,90):
       theta = PI * i / num_segments
       x = r * math.cos(theta)
       y = r * math.sin(theta)
       glVertex2f(x + cx, y + cy)

   glEnd()

   glColor3f(1.0, 1.0, 1.0)
   glBegin(GL_POLYGON)
   for i in range(0, 90)   :
       theta =2* PI * i / num_segments
       x = r * math.cos(theta)
       y = r * math.sin(theta)
       glVertex2f(x + cx, y + cy)

   glEnd()

def drawFace():
   glColor3f(0.8863, 0.0, 0.1333)
   glBegin(GL_TRIANGLE_FAN)
   centrex = -0.25
   centrey = 0.075
   for i in range(101):
       angle = 2.0 * math.pi * float(i) / float(100)
       x = centrex + 0.015 * math.cos(angle)
       y = centrey + 0.015 * math.sin(angle)
       glVertex2f(x, y)
```

```
glEnd()


# left eye

glColor3f(0.2039, 0.2353, 0.5765)

glLineWidth(5.0)

glBegin(GL_LINE_LOOP)

glVertex2f(-0.365, 0.0)

glVertex2f(-0.385, 0.0175)

glVertex2f(-0.365, 0.035)

glVertex2f(-0.3, 0.035)

glVertex2f(-0.28, 0.0175)

glVertex2f(-0.3, 0.0)

glEnd()


# left eyeball

glBegin(GL_TRIANGLE_FAN)

eyeCentrex = -0.3325

eyeCentrey = 0.0175

for i in range(101):

    angle = 2.0 * math.pi * float(i) / float(100)

    x = eyeCentrex + 0.015 * math.cos(angle)

    y = eyeCentrey + 0.015 * math.sin(angle)

    glVertex2f(x, y)

glEnd()


# eyebrows

glColor3f(0.2039, 0.2353, 0.5765)

glBegin(GL_LINE_STRIP)

glVertex2f(-0.385, 0.0475)

glVertex2f(-0.365, 0.065)
```

```
glVertex2f(-0.305, 0.065)
glVertex2f(-0.28, 0.0475)
glEnd()

# right eye
glColor3f(0.2039, 0.2353, 0.5765)
glLineWidth(5.0)
glBegin(GL_LINE_LOOP)
glVertex2f(-0.20, 0.0)
glVertex2f(-0.22, 0.017)
glVertex2f(-0.20, 0.035)
glVertex2f(-0.14, 0.035)
glVertex2f(-0.12, 0.017)
glVertex2f(-0.14, 0.0)
glEnd()

# right eyeball
glBegin(GL_TRIANGLE_FAN)
eyeCentrex = -0.17
eyeCentrey = 0.0175
for i in range(101):
    angle = 2.0 * math.pi * float(i) / float(100)
    x = eyeCentrex + 0.015 * math.cos(angle)
    y = eyeCentrey + 0.015 * math.sin(angle)
    glVertex2f(x, y)
glEnd()

# eyebrows
glColor3f(0.2039, 0.2353, 0.5765)
glBegin(GL_LINE_STRIP)
```

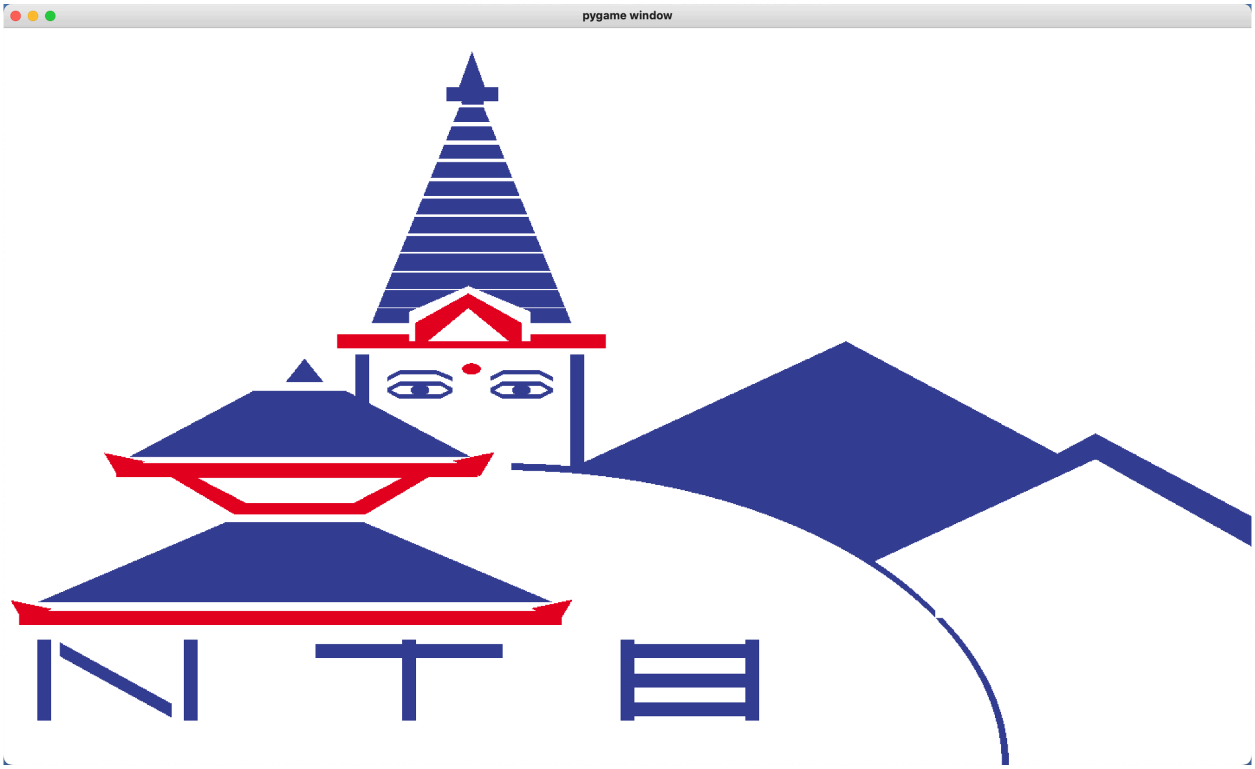
```
   glVertex2f(-0.22, 0.0475)

   glVertex2f(-0.20, 0.065)

   glVertex2f(-0.14, 0.065)

   glVertex2f(-0.12, 0.0475)

   glEnd()


# # Nose

#     glColor3f(0.2039, 0.2353, 0.5765)

#     glBegin(GL_LINE_STRIP)

#     glLineWidth(1.0)

#     for i in range(200, -1, -1):

#         theta = -2.0 * math.pi * i / 100.0

#         x = -0.25 + (0.001 + i * 0.00015) * math.cos(theta)

#         y = -0.035 + (0.001 + i * 0.00015) * math.sin(theta)

#         glVertex2f(x, y)

#     glEnd()



def ntb_text():

   glColor3f (0.2039, 0.2353, 0.5765)


 # for N

   glLineWidth(30.0)


   glBegin(GL_LINES)

   glVertex2f(-0.935, -0.66)

   glVertex2f(-0.935, -0.88)

   glEnd()


   glBegin(GL_LINES)
```

```
  glVertex2f(-0.91, -0.686)     # Top-left

  glVertex2f(-0.73, -0.853)     # Bottom-right

  glEnd()


  glBegin(GL_LINES)

  glVertex2f(-0.7, -0.66)

  glVertex2f(-0.7, -0.88)

  glEnd()



# for T

  glBegin(GL_LINES)


  glVertex2f(-0.2, -0.69)     # Top

  glVertex2f(-0.5, -0.69)     # Top

  glEnd()




  glBegin(GL_LINES)

  glVertex2f(-0.35, -0.66)

  glVertex2f(-0.35, -0.88)




  glEnd()


# for B

  glBegin(GL_LINES)

  glVertex2f(0.0, -0.66)

  glVertex2f(0.0, -0.88)

  glEnd()
```



```
    glBegin(GL_LINES)
    glVertex2f(0.0, -0.69)    # Top
    glVertex2f(0.2, -0.69)    # Top
    glEnd()

    glBegin(GL_LINES)
    glVertex2f(0.0, -0.77)    # Top
    glVertex2f(0.2, -0.77)    # Top
    glEnd()

    glBegin(GL_LINES)
    glVertex2f(0.0, -0.85)  # Top
    glVertex2f(0.2, -0.85)  # Top
    glEnd()

    glBegin(GL_LINES)
    glVertex2f(0.2, -0.66)
    glVertex2f(0.2, -0.88)
    glEnd()

main()
```

Output:


pygame window