Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

Lab I: Drawing Nepal Tourism Board Logo using OpenGL

COMP 342

Submitted by:

Manish Shivabhakti
Roll no: 63
CE, III year/ II Semester

Drawing the Nepal Tourism Board (NTB) Logo using OpenGL

Programming language

1. Python

Graphics Library

1. Pygame

Pygame is a popular Python library used for creating video games and multimedia applications. It provides modules for handling graphics, sound, and user input, making it easy to develop interactive applications. Pygame is built on top of the Simple DirectMedia Layer (SDL) library, which offers cross-platform support for various operating systems.

2. PyOpenGL

PyOpenGL is a Python binding to the OpenGL library, enabling the use of OpenGL's powerful graphics rendering capabilities within Python applications. It supports a wide range of OpenGL features, including advanced 3D graphics and shaders, allowing for the creation of complex visualizations and simulations. PyOpenGL is often used in conjunction with other libraries like GLUT or Pygame to handle windowing and user input.

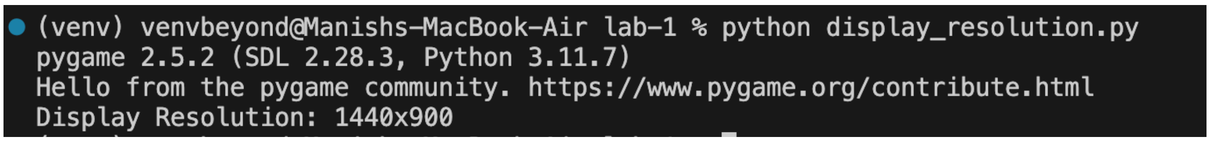
Displaying Resolution

```python
from OpenGL.GL import *
from OpenGL.GLUT import *
import pygame

def display_resolution_pyopengl():
    # Initialize Pygame to get display info
    pygame.init()
    # Get the resolution of the display
    display_info = pygame.display.Info()
    screen_width = display_info.current_w
    screen_height = display_info.current_h
    # Print the resolution
    print(f"Display Resolution: {screen_width}x{screen_height}")
    # Clean up and quit Pygame
    pygame.quit()

if __name__ == "__main__":
    display_resolution_pyopengl()
```

Output



```
(venv) venvbeyond@Manishs-MacBook-Air lab-1 % python display_resolution.py
pygame 2.5.2 (SDL 2.28.3, Python 3.11.7)
Hello from the pygame community. https://www.pygame.org/contribute.html
Display Resolution: 1440x900
```

Drawing NTB logo

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


def draw_polygon(color, vertices):
    glColor3f(*[c/255.0 for c in color])
    glBegin(GL_POLYGON)
    for vertex in vertices:
        glVertex2f(*vertex)
    glEnd()


def draw_circle(color, center, radius, segments=100):
    glColor3f(*[c/255.0 for c in color])
    glBegin(GL_POLYGON)
    for i in range(segments):
        theta = 2.0 * 3.1415926 * i / segments
        x = radius * cos(theta)
        y = radius * sin(theta)
        glVertex2f(x + center[0], y + center[1])
    glEnd()


def draw_rect(color, rect):
    x, y, width, height = rect
    draw_polygon(color, [(x, y), (x + width, y),
                 (x + width, y + height), (x, y + height)])


def draw():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    # Blue color
```

```python
blue = (54, 60, 146)

# White color
white = (255, 255, 255)

# Red color
red = (221, 0, 39)

# Draw first mountain
draw_polygon(blue, [(680, 414), (295, 776), (1067, 776)])

# Draw second mountain
draw_polygon(blue, [(928, 564), (563, 834), (1292, 834)])

# Draw snow on first mountain
draw_polygon(white, [(680, 432), (590, 522), (770, 522)])

# Draw front of second mountain
draw_polygon(white, [(928, 599), (630, 843), (1226, 843)])

# Draw stupa border
draw_circle(blue, (418, 1024), 437)

# Draw face border
draw_rect(blue, (291, 439, 203, 172))

# Draw shrine
draw_polygon(blue, [(393, -17), (296, 441), (489, 441)])

# Draw stupa front
draw_circle(white, (419, 1024), 402)

# Draw face front
draw_rect(white, (307, 461, 167, 138))

# Draw head top
draw_polygon(red, [(393, 370), (337, 441), (449, 441)])

# Draw head top light
draw_polygon(white, [(393, 389), (353, 441), (433, 441)])

# Draw head hat
draw_rect(red, (281, 439, 223, 27))
```

```python
    # Draw mask
    draw_rect(white, (0, 539, 433, 485))

    # Draw bottom beam
    draw_rect(red, (22, 783, 472, 31))

    # Draw bottom layer
    draw_polygon(blue, [(252, 609), (27, 776), (477, 776)])

    # Draw gap
    draw_rect(white, (137, 633, 229, 32))

    # Draw top beam
    draw_rect(red, (92, 607, 314, 26))

    # Draw top layer
    draw_polygon(blue, [(247, 461), (98, 598), (395, 598)])

    # Draw gap
    draw_polygon(white, [(247, 461), (190, 511), (303, 511)])

    # Draw gap in shrine
    for i in range(72, 356, 31):
        draw_rect(white, (306, i, 178, 14))

    # Draw shrine
    draw_polygon(blue, [(246, 466), (228, 499), (263, 499)])

    # Draw text rectangles
    draw_rect(white, (19, 874, 45, 150))
    draw_rect(white, (167, 874, 45, 150))
    draw_rect(white, (370, 874, 49, 150))
    draw_rect(white, (300, 874, 189, 37))
    draw_rect(white, (579, 874, 48, 150))
    draw_rect(white, (709, 874, 46, 150))
    draw_rect(white, (579, 930, 176, 37))
    draw_rect(white, (579, 977, 176, 37))
    draw_rect(white, (579, 874, 176, 37))

    # Draw text triangles
    draw_polygon(
        white, [(62.5, 875.5), (170.5, 963.5), (173.5, 1022.5)])
    draw_polygon(
        white, [(62.5, 875.5), (62.5, 940.5), (173.5, 1022.5)])
    pygame.display.flip()


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

    # Set up the display
    global screen
    screen = pygame.display.set_mode(
        (1024, 1024), DOUBLEBUF | OPENGL)  # Set screen size
    pygame.display.set_caption("Mountain Scene")    # Set window title

    # Set up OpenGL
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    # Invert the y-axis to match Pygame's coordinate system
    glOrtho(0, 1024, 1024, 0, -1, 1)
    glMatrixMode(GL_MODELVIEW)

    glClearColor(1.0, 1.0, 1.0, 1.0)  # Set background color to white

    running = True

    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False  # Exit the loop when the user closes
the window

        draw()  # Call the draw function to draw the logo with "NTB"
integrated

    pygame.quit()  # Clean up and quit Pygame when the loop exits


if __name__ == "__main__":
    main()
```

Output

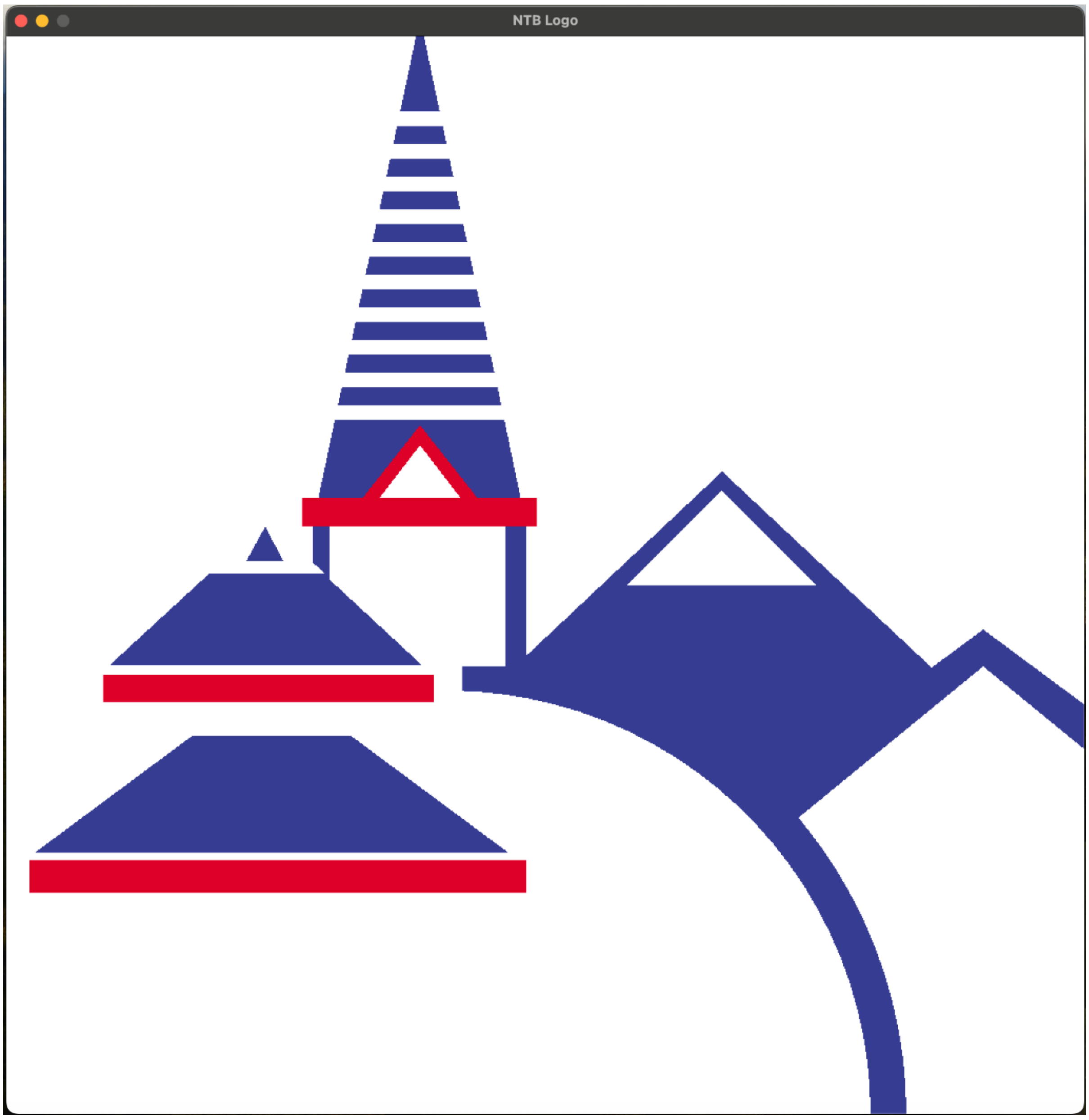


Figure 1. NTB Logo