Kathmandu University

School of Engineering

Department of Computer Science and Engineering (DoCSE)

Dhulikhel, Kavre

Computer Graphics – Lab1

Submitted To:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submitted By:

Mani Dumaru (15)

CE-2019 3rd year/2nd semester

Submission Date: 2023/03/29

1. Mention the name of Programming Language and Graphics Library you are using this semester for performing your Computer Graphics Lab and Project.

    - I am going to use python as the programming language and PyOpenGL as the graphics library.

2. Write the code snippets for setting graphics environment in your chosen graphics library and display the resolution of your display system through functions, classes provided by your graphics library.

    Source Code: NepalFlag.py

    Setting up Graphical Environment

```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

def clearScreen():
    glClearColor(0.0,0.0,0.0,1.0)
    gluOrtho2D(-0.5,1.0,-0.5,1.0)
```

```
def main():
    glutInit()
    glutInitDisplayMode(GLUT_RGB)
    glutInitWindowSize(650, 650)
    glutInitWindowPosition(100, 100)
    glutCreateWindow("Mani_Dumaru_(15)_Lab1_Nepal_Flag")
    size = numpy.ndarray.tolist(glGetIntegerv(GL_VIEWPORT))
    size = size[2],size[3]
    print(size)
    glutDisplayFunc(flag)
    clearScreen()
    glutMainLoop()
```

Display Resolution:

```
    size = numpy.ndarray.tolist(glGetIntegerv(GL_VIEWPORT))
    size = size[2],size[3]
    print(f"Your Window Resolution is: {size}")
```

Output:

```
Your Window Resolution is: (650, 650)
```

3. Get familiar with the coordinate system and draw a flag of Nepal using the chosen geometrical functions/classes provided by the library and also colour the flag accordingly.

Initial Coordinates:

```
vertices1 = (
    (0.0,0.4),
    (0,0.8),
    (0.6,0.4),
)

upper_vertices = (
    (-0.01,0.39),
    (-0.01,0.82),
    (0.63,0.39),
)

vertices2 = (
    (0.0,0.0),
    (0.0,0.55),
    (0.6,0.0),
)

lower_vertices = (
    (-0.01, -0.01),
    (-0.01, 0.57),
    (0.63,-0.01),
)

circle_moon = (
    (0.17, 0.535)
)

circle_sun = (
    (0.17, 0.18)
)
```

Border:

```
def flag():
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(0.0,0.0,1.0)
    ################ Border Triangles ###################
    glBegin(GL_TRIANGLES)
    for vertex in upper_vertices:
        x = vertex[0]
        y = vertex[1]
        glVertex2f(x,y)
    glEnd()

    glBegin(GL_TRIANGLES)
    for vertex in lower_vertices:
        x = vertex[0]
        y = vertex[1]
        glVertex2f(x,y)
    glEnd()
```

Triangles:

```
    glColor3f(1.0,0.0,0.0)
    ################ Upper Triangle ##################
    glBegin(GL_TRIANGLES)
    for vertex in vertices1:
        x = vertex[0]
        y = vertex[1]
        glVertex2f(x,y)
    glEnd()
    ################ Lower Triangle ##################
    glBegin(GL_TRIANGLES)
    for vertex in vertices2:
        x = vertex[0]
        y = vertex[1]
        glVertex2f(x,y)
    glEnd()
```

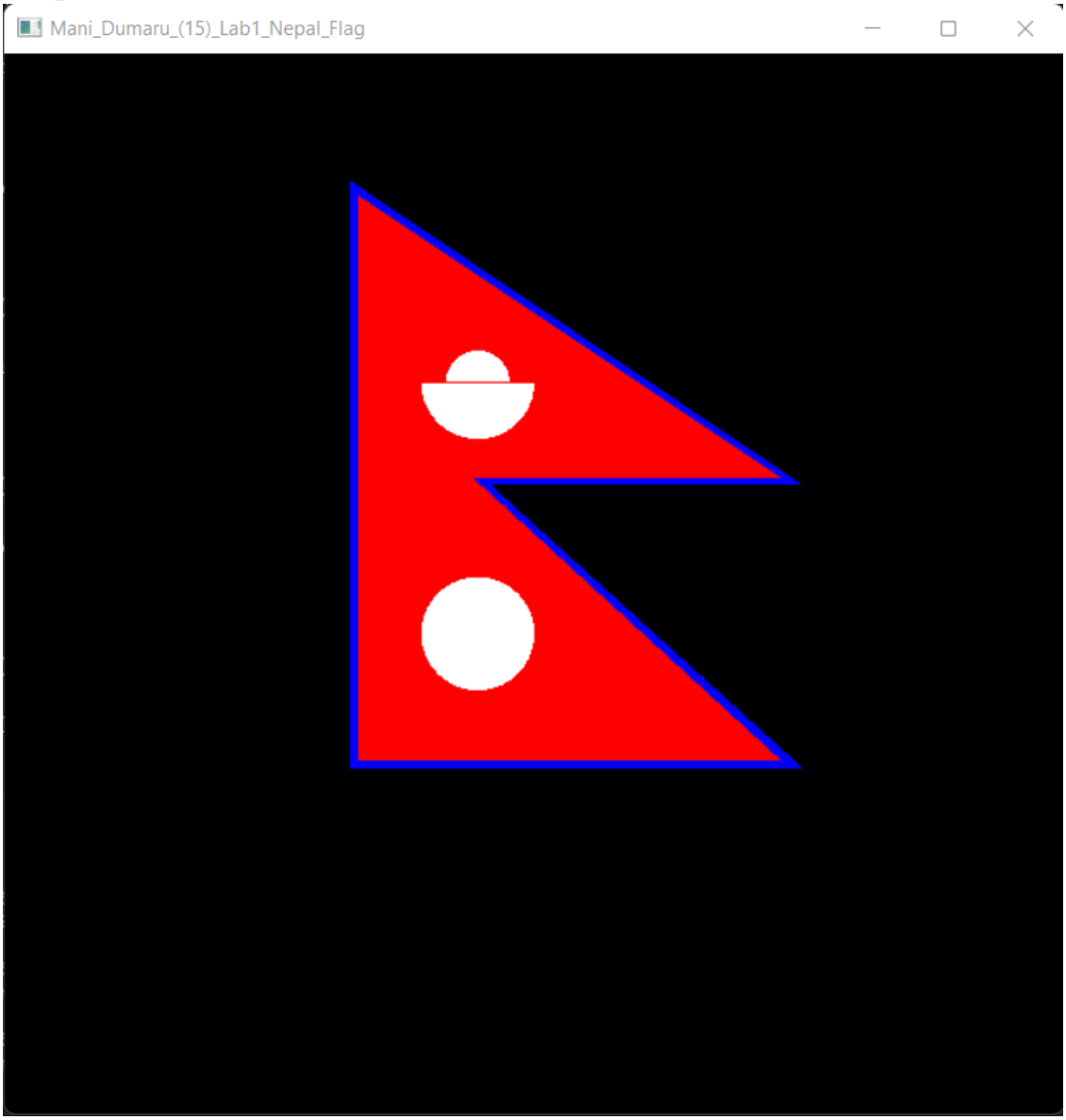
Moon:

```
################ Moon #################
glColor3f(1.0,1.0,1.0)
glLineWidth(2)
glBegin(GL_TRIANGLE_FAN)
cx = circle_moon[0]
cy = circle_moon[1]
r = 0.08
for i in range(180,360):
    theta = 3.1415926 * float(i) / float(180)
    x = r * math.cos(theta)
    y = r * math.sin(theta)
    glVertex2f((x + cx), (y + cy))
glEnd()
glBegin(GL_TRIANGLE_FAN)
cx = circle_moon[0]
cy = circle_moon[1]
r = 0.045
for i in range(0,360):
    theta = 3.1415926 * float(i) / float(360)
    x = r * math.cos(theta)
    y = r * math.sin(theta)
    glVertex2f((x + cx), (y + cy))
glEnd()
```

Sun:

```
################ Sun ####################
glBegin(GL_TRIANGLE_FAN)
cx = circle_sun[0]
cy = circle_sun[1]
r = 0.08
for i in range(0,360):
    theta = 2 * 3.1415926 * float(i) / float(360)
    x = r * math.cos(theta)
    y = r * math.sin(theta)
    glVertex2f((x + cx), (y + cy))
glEnd()
glFlush()
```

Output:



Conclusion:

Using the functions and classes provided by PyOpenGl graphics library, flag of Nepal was designed. Through this project, I was able to understand the coordinate system and learn how to use the methods and classes in graphics.