Kathmandu University

Department of Computer Science and Engineering Dhulikhel, Kavrepalanchowk

A Lab Report
on
"Computer Graphics"

[Code No: COMP342]

Submitted by: Aaditya KC

CS III-I

Roll-No: 05

Submitted to:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submission Date:

2025/12/15

1. **Write your first name using the OpenGL library in a window [Use Polygon Function of OpenGL for writing the name] and fill the polygon with the color of your choice.**

The source code for this problem is:

lab1.py

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


def draw_A(x):
    glBegin(GL_POLYGON)
    glVertex2f(x, 100)
    glVertex2f(x + 20, 200)
    glVertex2f(x + 40, 100)
    glVertex2f(x + 30, 100)
    glVertex2f(x + 25, 130)
    glVertex2f(x + 15, 130)
    glVertex2f(x + 10, 100)
    glEnd()


def draw_D(x):
    glBegin(GL_POLYGON)
    glVertex2f(x, 100)
    glVertex2f(x, 200)
    glVertex2f(x + 30, 180)
    glVertex2f(x + 30, 120)
    glEnd()


def draw_I(x):
    glBegin(GL_POLYGON)
    glVertex2f(x, 100)
    glVertex2f(x + 15, 100)
    glVertex2f(x + 15, 200)
    glVertex2f(x, 200)
    glEnd()


def draw_T(x):
    glBegin(GL_POLYGON)
    glVertex2f(x, 180)
    glVertex2f(x + 40, 180)
    glVertex2f(x + 40, 200)
    glVertex2f(x, 200)
    glEnd()

    glBegin(GL_POLYGON)
    glVertex2f(x + 15, 100)
    glVertex2f(x + 25, 100)
    glVertex2f(x + 25, 180)
    glVertex2f(x + 15, 180)
```

lab1.py

lab1.py > display

```
36  def draw_I(x):
49      glEnd()
50
51
52  def draw_Y(x):
53      glBegin(GL_POLYGON)
54      glVertex2f(x, 200)
55      glVertex2f(x + 20, 160)
56      glVertex2f(x + 40, 200)
57      glVertex2f(x + 30, 200)
58      glVertex2f(x + 20, 180)
59      glVertex2f(x + 10, 200)
60      glEnd()
61
62      glBegin(GL_POLYGON)
63      glVertex2f(x + 18, 100)
64      glVertex2f(x + 22, 100)
65      glVertex2f(x + 22, 160)
66      glVertex2f(x + 18, 160)
67      glEnd()
68
69
70  def display():
71      glClear(GL_COLOR_BUFFER_BIT)
72
73
74      glColor3f(0.1, 0.5, 0.9)
75      draw_A(50)
76      draw_A(110)
77      glColor3f(0.0, 0.0, 0.0)
78      draw_D(170)
79      draw_I(230)
80      glColor3f(1.0, 1.0, 0.0)
81
82      draw_T(260)
83      draw_Y(320)
84      glColor3f(0.6, 0.0, 0.6)
85      draw_A(380)
86
87      glFlush()
88
89
90  def init():
91      glClearColor(1.0, 1.0, 1.0, 1.0)   # White background
92      gluOrtho2D(0, 500, 0, 300)
93
94
95  def main():
```

lab1.py

```
88
89
90  def init():
91      glClearColor(1.0, 1.0, 1.0, 1.0)   # White background
92      gluOrtho2D(0, 500, 0, 300)
93
94
95  def main():
96      glutInit()
97      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
98      glutInitWindowSize(700, 300)
99      glutCreateWindow(b"Name Using OpenGL Polygons")
100
101     init()
102     glutDisplayFunc(display)
103     glutMainLoop()
104
105
106 main()
107
```

The above provided is the source code.

Output:

Theory:

OpenGL (Open Graphics Library) is a cross-platform graphics API used for rendering 2D and 3D graphics. In OpenGL, complex objects are created by combining basic geometric primitives such as points, lines, and polygons.

A polygon is a closed figure formed by connecting multiple vertices in a specified order. The `GL_POLYGON` primitive is used to draw and fill a closed shape automatically.

An orthographic projection using `gluOrtho2D()` is used to map 2D coordinates directly onto the screen.

The program mainly uses `GL_POLYGON` for drawing filled letters, `glVertex2f()` for vertices, `glColor3f()` for coloring, `gluOrtho2D()` for 2D projection, and `glutMainLoop()` for execution.

Algorithm :

1. Import required OpenGL and GLUT modules.
2. Initialize the OpenGL window.
3. Set the background color and projection mode.
4. Define functions to draw individual letters using `GL_POLYGON`.
5. Assign color to the polygons using `glColor3f()`.
6. Display the letters on the screen using the display callback function.
7. Run the GLUT main loop.