

## UCS1712 – GRAPHICS AND MULTIMEDIA LAB

### Ex. No. 1 Study of Basic Output Primitives in C++ using OpenGL

Date: 19/7/21

Name: Srinath S

Class: CSE-C

Roll: 185001205

---

#### **Question:**

- a). To create an output window using OPENGL and to draw the following basic output primitives – POINTS, LINES, LINE\_STRIP, LINE\_LOOP, TRIANGLES, QUADS, QUAD\_STRIP, POLYGON.
- b) To create an output window and draw a checkerboard using OpenGL.
- c) To create an output window and draw a house using POINTS, LINES, TRIANGLES and QUADS/POLYGON.

#### **Code:**

##### **A) Shapes:**

```
#include<GL/glut.h>
#include<stdio.h>
void myInit()
{
    glClearColor(0.0,0.0,0.0,1.0);
    gluOrtho2D(-1.0,1.0,-1.0,1.0);
}
void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);

    // 1) POINTS

    glColor3f(1.0,1.0,1.0);
    glBegin(GL_POINTS);
        glVertex2f(-0.1,-0.7);
    glEnd();

    // 2) LINES

    glColor3f(1.0,1.0,1.0);
    glBegin(GL_LINES);
        glVertex2f(-0.3,-0.3);
        glVertex2f(0.3,-0.3);
    glEnd();

    // 3) LINE_STRIP

    glColor3f(0.0,0.0,1.0);
```

```

glBegin(GL_LINE_STRIP);
    glVertex2f(-0.1,-0.3);
    glVertex2f(0.0,0.2);
    glVertex2f(0.2,0.5);
glEnd();

// 4) LINE_LOOP

glColor3f(1.0,0.0,0.0);
glBegin(GL_LINE_LOOP);
    glVertex2f(-0.1 - 0.15,-0.3);
    glVertex2f(0.0 - 0.15,0.2);
    glVertex2f(0.2 - 0.15,0.5);
glEnd();

// 5) TRIANGLES

glColor3f(0.3,0.7,0.4);
glBegin(GL_TRIANGLES);
    glVertex2f(-0.1 - 0.4,-0.3 + 0.5);
    glVertex2f(0.0 - 0.4,0.2 + 0.5);
    glVertex2f(0.2 - 0.4,0.5 + 0.5);
glEnd();

// 6) TRIANGLE STRIP

glColor3f(0.0,0.7,0.8);
glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(-0.1 +0.4,-0.3 - 0.2);
    glVertex2f(0.0 +0.4,0.2 - 0.2);
    glVertex2f(0.2 +0.4,0.5 - 0.2);
    glVertex2f(0.4 +0.4, 0.6 - 0.2);
glEnd();

// 7) TRIANGLE FAN

glColor3f(0.7,0.1,0.8);
glBegin(GL_TRIANGLE_FAN);
    glVertex2f(-0.1 -0.75,-0.3 - 0.55);
    glVertex2f(0.0 -0.75,0.2 - 0.55);
    glVertex2f(0.2 -0.75,0.5 - 0.55);
    glVertex2f(0.4 -0.75, 0.6 - 0.55);
glEnd();

// 8) QUADS

glColor3f(0.1,0.1,0.8);
glBegin(GL_QUADS);

```

```

        glVertex2f(-0.1, -0.1 - 0.6);
        glVertex2f(-0.1, 0.1 - 0.6);
        glVertex2f(0.1, 0.1 - 0.6);
        glVertex2f(0.1, -0.1 - 0.6);
    glEnd();

    // 9) QUAD_STRIP

    glColor3f(0.6,0.1,0.5);
    glBegin(GL_QUAD_STRIP);
        glVertex2f(-0.1 + 0.5, -0.1 - 0.6);
        glVertex2f(-0.1 + 0.5, 0.1 - 0.6);
        glVertex2f(0.1 + 0.5, -0.1 - 0.6);
        glVertex2f(0.1 + 0.5, 0.1 - 0.6);

        glVertex2f(0.3 + 0.5, -0.1 - 0.6);
        glVertex2f(0.3 + 0.5, 0.2 - 0.6);

    glEnd();

    // 10) POLYGON

    glColor3f(0.6,0.1,0.5);
    glBegin(GL_POLYGON);

        glVertex2f(-0.1 + 0.5, -0.1 + 0.6);
        glVertex2f(-0.1 + 0.5, 0.1 + 0.6);
        glVertex2f(0.0 + 0.5, 0.2 + 0.6);
        glVertex2f(0.1 + 0.5, 0.1 + 0.6);
        glVertex2f(0.1 + 0.5, -0.1 + 0.6);

    glEnd();

    glFlush();
}
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutCreateWindow("check");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}

```

## B) Checkerboard:

```

#include <GL/glut.h>
void initGL() {

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    GLint x;
    GLint y;
    GLint colorCode = 1;

    for(y = 50; y <= 350; y += 50) {

        for(x = 50; x <= 350; x += 50) {
            if (colorCode == 1) {
                glColor3f(0.0, 0.0, 0.0);
                colorCode = 0;
            }
            else {
                glColor3f(1.0, 1.0, 1.0);
                colorCode = 1;
            }
            glBegin(GL_QUADS);
            glVertex2i(x, y);
            glVertex2i(x, y + 50);
            glVertex2i(x + 50, y + 50);
            glVertex2i(x + 50, y);
            glEnd();

        }

    }

    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutCreateWindow("Understandable Checkers");
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(50, 50);

```

```

        glutDisplayFunc(display);
        initGL();
        glutMainLoop();
        return 0;
}

```

### C) House:

```

#include<GL/glut.h>
#include<stdio.h>
void myInit()
{
    glClearColor(0.0,0.0,0.0,1.0);
    gluOrtho2D(-1.0,1.0,-1.0,1.0);
}
void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    // glColor3f(1.0,0.0,0.0);
    // glBegin(GL_LINES);
    //     glVertex2f(-0.7,-0.7);
    //     glVertex2f(-0.1,-0.7);

    //     glVertex2f(-0.7,-0.7);
    //     glVertex2f(-0.7,-0.1);

    //     glVertex2f(-0.1,-0.7);
    //     glVertex2f(-0.1,-0.1);

    //     glVertex2f(-0.7,-0.1);
    //     glVertex2f(-0.1,-0.1);

    // glEnd();

    // glColor3f(0.0,1.0,0.0);
    // glBegin(GL_LINES);

    //     glVertex2f(-0.7,-0.1);
    //     glVertex2f(-0.1,-0.1);

    // glEnd();

    glColor3f(0.0,1.0,0.0);
    glBegin(GL_QUADS);

        glVertex2f(-0.7, -0.7);
        glVertex2f(-0.7, 0.3);
        glVertex2f(0.7, 0.3);

```

```
        glVertex2f(0.7, -0.7);

    glEnd();

    glColor3f(1.0,0.0,0.0);
    glBegin(GL_TRIANGLES);

        glVertex2f(-0.7, 0.3);
        glVertex2f(0.7, 0.3);
        glVertex2f(0.0, 0.75);

    glEnd();

    // door
    glColor3f(0.0,0.0,1.0);
    glBegin(GL_QUADS);

        glVertex2f(-0.15, -0.7);
        glVertex2f(-0.15, -0.25);
        glVertex2f(0.15, -0.25);
        glVertex2f(0.15, -0.7);

    glEnd();

    // left window
    glColor3f(0.0,0.5,0.5);
    glBegin(GL_QUADS);

        glVertex2f(-0.3, -0.6);
        glVertex2f(-0.3, -0.4);
        glVertex2f(-0.5, -0.4);
        glVertex2f(-0.5, -0.6);

    glEnd();

    // right window
    glColor3f(0.0,0.5,0.5);
    glBegin(GL_QUADS);

        glVertex2f(0.3, -0.6);
        glVertex2f(0.3, -0.4);
        glVertex2f(0.5, -0.4);
        glVertex2f(0.5, -0.6);

    glEnd();

    glColor3f(0.0, 0.0, 0.0);
    glBegin(GL_LINES);
```

```

        glVertex2f(-0.7, -0.15);
        glVertex2f(0.7, -0.15);

    glEnd();

    // upper window

    glColor3f(0.5,0.5,0.0);
    glBegin(GL_QUADS);

        glVertex2f(-0.6, -0.05);
        glVertex2f(-0.6, 0.15);
        glVertex2f(0.6, 0.15);
        glVertex2f(0.6, -0.05);

    glEnd();

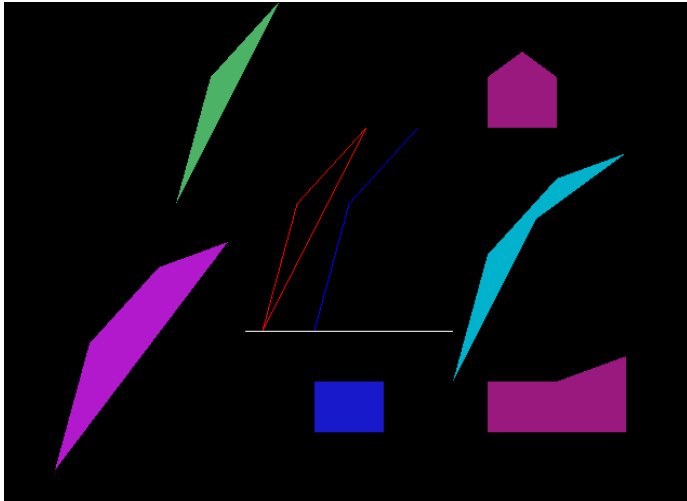
    // glBegin(GL_POINTS);
    //     glVertex2f(-0.1,-0.7);
    // glEnd();

    glFlush();
}
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutCreateWindow("check");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}

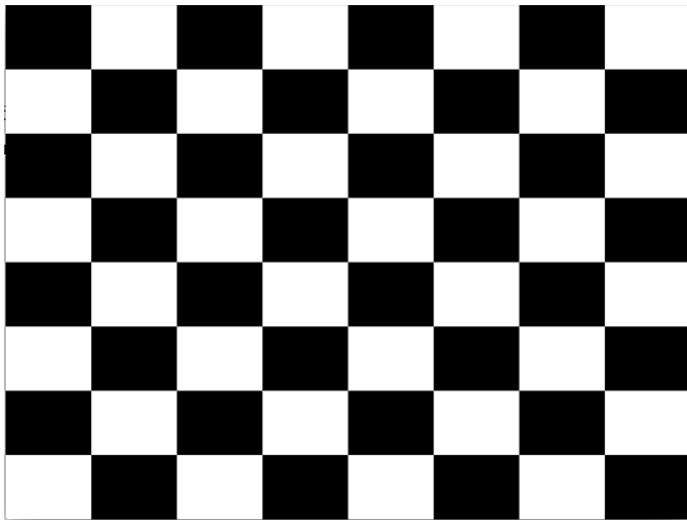
```

## OUTPUTS:

A)



**B)**



**c)**

