

SSN COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UCS1712 – GRAPHICS AND MULTIMEDIA LAB

EX NO: 1 – Study of Basic output primitives in OpenGL

Name : Likhitha Verma A

Reg : 185001084

Date : 28/07/2021

1. **AIM :** To create a window using OPENGL and to draw the following basic output primitives – POINTS, LINES, LINE_STRIP, LINE_LOOP, TRIANGLES, TRIANGLE STRIP, TRIANGLE FAN, QUADS, QUAD_STRIP, POLYGON.

ALGORITHM :

1. Open Visual Studio
2. Create a new Empty C++ project
3. Add a new main.cpp file to the project
4. Add the GLUT library file
5. Add the below code in the main.cpp file
6. Run the code

Code :

```
#include<glut.h>
void myInit()
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glColor3f(1.0, 0.2, 0.0);
    glPointSize(7);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    //points
    glBegin(GL_POINTS);
```

```
glVertex2d(40,470);
glVertex2d(65, 470);
glVertex2d(40, 450);
glEnd();
//line
glBegin(GL_LINES);
glVertex2d(100, 400);
glVertex2d(130, 475);
glEnd();
//line strip
glBegin(GL_LINE_STRIP);
glVertex2d(150, 400);
glVertex2d(200, 475);
glVertex2d(250, 400);
glVertex2d(300, 475);
glVertex2d(350, 400);
glVertex2d(400, 475);
glEnd();
//line loop
glBegin(GL_LINE_LOOP);
glVertex2d(450, 400);
glVertex2d(430, 475);
glVertex2d(520, 440);
glVertex2d(550, 400);
glEnd();
//triangle
glBegin(GL_TRIANGLES);
glVertex2d(40,250);
glVertex2d(80, 350);
glVertex2d(105, 200);
glEnd();
//triangle strip
glBegin(GL_TRIANGLE_STRIP);
glVertex2d(140, 250);
glVertex2d(140, 350);
glVertex2d(200, 350);
glColor3f(0.0f, 0.2f, 1.0f);
glVertex2d(200, 250);
glEnd();
//triangle fan
glBegin(GL_TRIANGLE_FAN);
```

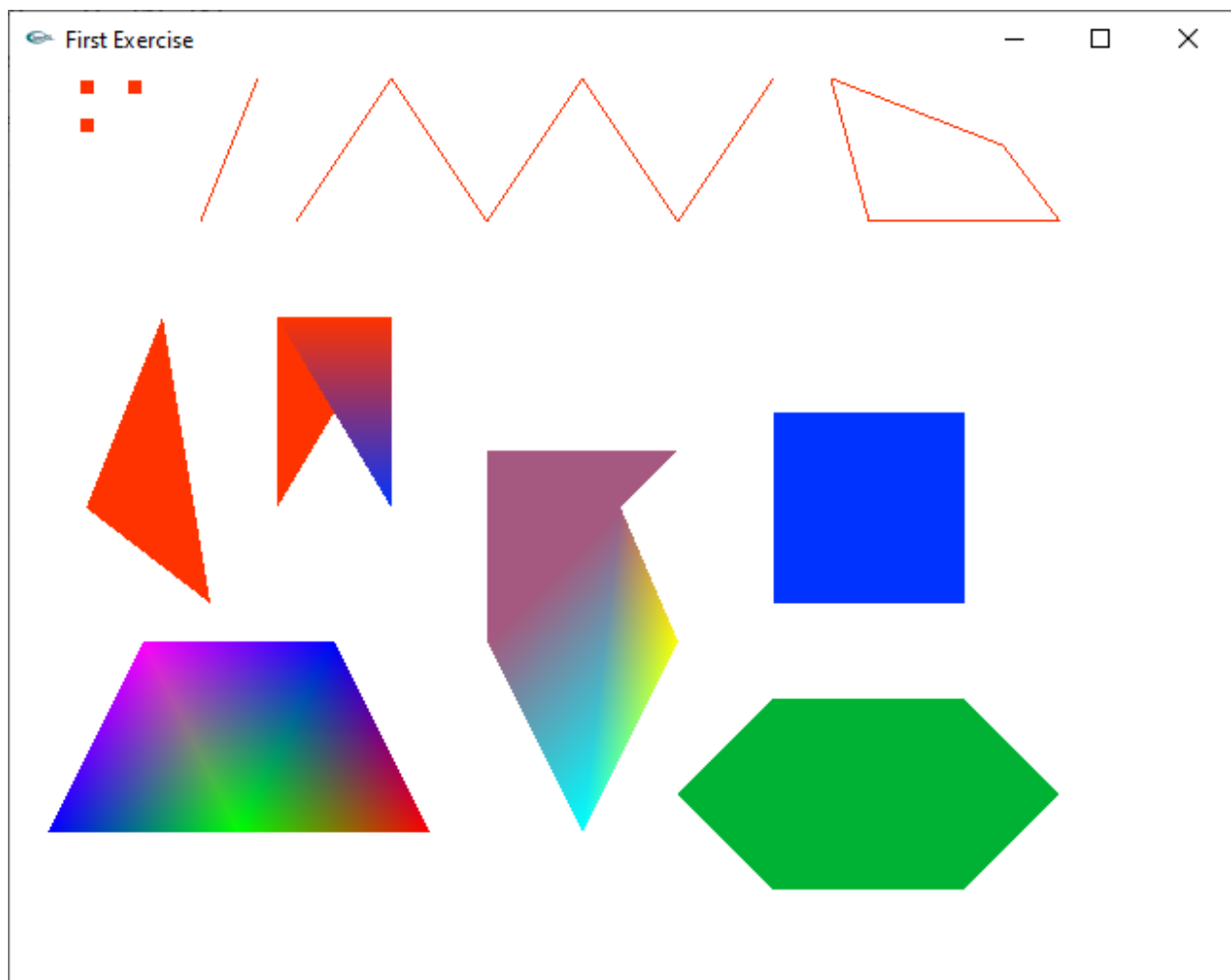
```
glColor3f(0.0f, 1.0f, 0.0f);
glVertex2f(120, 80);
glColor3f(1.0f, 0.0f, 0.0f);
glVertex2f(220, 80);
glColor3f(0.0f, 0.0f, 1.0f);
glVertex2f(170, 180);
glColor3f(1.0f, 0.0f, 1.0f);
glVertex2f(70, 180);
glColor3f(0.0f, 0.0f, 1.0f);
glVertex2f(20, 80);
glEnd();
//quads
glBegin(GL_QUADS);
glColor3f(0.0f, 0.2f, 1.0f);
glVertex2d(400, 200);
glVertex2d(500, 200);
glVertex2d(500, 300);
glVertex2d(400, 300);
glEnd();
//quad strip
glBegin(GL_QUAD_STRIP);
glColor3f(0.65f, 0.35f, 0.5f);
glVertex2d(350, 280);
glVertex2d(250, 280);
glVertex2d(320, 250);
glVertex2d(250, 180);
glColor3f(1.0f, 1.0f, 0.0f);
glVertex2d(350, 180);
glColor3f(0.0f, 1.0f, 1.0f);
glVertex2d(300, 80);
glEnd();
//PoLygon
glBegin(GL_POLYGON);
glColor3f(0.0f, 0.7f, 0.2f);
glVertex2d(400, 50);
glVertex2d(500, 50);
glVertex2d(550, 100);
glVertex2d(500, 150);
glVertex2d(400, 150);
glVertex2d(350, 100);
glEnd();
```

```

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

```

OUTPUT :



RESULT :

Thus, drew basic output primitives POINTS, LINES, LINE_STRIP, LINE_LOOP, TRIANGLES, TRIANGLE STRIP, TRIANGLE FAN, QUADS, QUAD_STRIP, POLYGON successfully.

2. **AIM :** To create a window using OPENGGL and to draw the following basic output primitives – POINTS, LINES, LINE_STRIP, LINE_LOOP, TRIANGLES, TRIANGLE STRIP, TRIANGLE FAN, QUADS, QUAD_STRIP, POLYGON.

ALGORITHM :

1. Open Visual Studio
2. Create a new Empty C++ project
3. Add a new main.cpp file to the project
4. Add the GLUT library file
5. Add the below code in the main.cpp file
6. Run the code

Code :

```
#include <glut.h>
void initGL() {
    glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
    glPointSize(10);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 600.0, 0.0, 600.0);
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    //rectangle
    glBegin(GL_QUADS);
    glColor3f(0.74f, 0.45f, 0.95f);
    glVertex2f(120, 10);
    glVertex2f(380, 10);
    glVertex2f(380, 200);
    glVertex2f(120, 200);
    glEnd();

    //triangle
    glBegin(GL_TRIANGLES);
```

```
glColor3f(0.74f, 0.45f, 0.95f);
glVertex2f(80, 200);
glVertex2f(420, 200);
glVertex2f(250, 350);
glEnd();
```

```
//line under triangle
```

```
glBegin(GL_QUADS);
glColor3f(0.0f, 0.0f, 0.0f);
glVertex2f(80, 200);
glVertex2f(80, 210);
glVertex2f(420, 210);
glVertex2f(420, 200);
glEnd();
```

```
//chimney
```

```
glBegin(GL_QUADS);
glColor3f(0.74f, 0.45f, 0.95f);
glVertex2f(320, 240);
glVertex2f(370, 240);
glVertex2f(370, 300);
glVertex2f(320, 300);
glEnd();
```

```
//door
```

```
glBegin(GL_QUADS);
glColor3f(0.68, 1, 0);
glVertex2f(200, 10);
glVertex2f(300, 10);
glVertex2f(300, 150);
glVertex2f(200, 150);
glEnd();
```

```
//doorknob
```

```
glBegin(GL_POINTS);
glColor3f(0.0f, 0.0f, 0.0f);
glVertex2f(220.0, 70.0);
glEnd();
```

```
//steps
```

```
glBegin(GL_QUADS);
```

```
glColor3f(1, 0.47, 0);
glVertex2f(150, 10);
glVertex2f(350, 10);
glVertex2f(350, 30);
glVertex2f(150, 30);
glEnd();

//left-window
glBegin(GL_QUADS);
glColor3f(0.68, 1, 0);
glVertex2f(140, 130);
glVertex2f(190, 130);
glVertex2f(190, 180);
glVertex2f(140, 180);
glEnd();

//left window grills
glBegin(GL_LINES);
glColor3f(0.0f, 0.0f, 0.0f);
glVertex2f(140, 156);
glVertex2f(190, 155);
glVertex2f(165, 130);
glVertex2f(165, 180);
glEnd();

//right-window
glBegin(GL_QUADS);
glColor3f(0.68, 1, 0);
glVertex2f(310, 130);
glVertex2f(360, 130);
glVertex2f(360, 180);
glVertex2f(310, 180);
glEnd();

//right window grills
glBegin(GL_LINES);
glColor3f(0.0f, 0.0f, 0.0f);
glVertex2f(310, 156);
glVertex2f(360, 155);
glVertex2f(335, 130);
glVertex2f(335, 180);
```

```
    glEnd();  
    glFlush();  
}  
  
int main(int argc, char** argv) {  
    glutInit(&argc, argv);  
    glutCreateWindow("House");  
    glutInitWindowSize(500, 500);  
    glutInitWindowPosition(0, 0);  
    glutDisplayFunc(display);  
    initGL();  
    glutMainLoop();  
    return 0;  
}
```

OUTPUT :



RESULT :

Thus created a simple house with windows using OpenGL successfully.