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 : Mohamed Hashim G

RegNo : 185001094

---

1. Basic output primitives

Aim

Create a window using OpenGL and draw the following basic output primitives – POINTS, LINES, LINE_STRIP, LINE_LOOP, TRIANGLES, TRIANGLE STRIP, TRIANGLE FAN, QUADS, QUAD_STRIP, POLYGON.

Algorithm

1. Set Window Size using glutInitWindowSize().
2. Set the background color using gluClearColor().
3. To plot points, we have to mention the vertices using glVertex2i() between glBegin(GL_POINTS) and glEnd().
4. For lines we have to mention 2 vertices using glVertex2i() between glBegin(GL_LINES) and glEnd().
5. For line strip we have to mention 3 or more vertices using glVertex2i() between glBegin(GL_LINE_STRIP) and glEnd().
6. For the line loop, we have to mention 4 or more vertices using glVertex2i() between glBegin(GL_LINE_LOOP) and glEnd().
7. For triangles, we have to mention 3 vertices in anti-clockwise order using glVertex2i() between glBegin(GL_TRIANGLES) and glEnd().
8. For triangle strips we have to mention 4 or more vertices in anti-clockwise order using glVertex2i() between glBegin(GL_TRIANGLE_STRIP) and glEnd().

9. For triangle fan, we have to mention 4 or more vertices in anti-clockwise order using glVertex2i() between glBegin(GL_TRIANGLE_FAN) and glEnd().
10. For quads, we have to mention 4 vertices using glVertex2i() between glBegin(GL_QUADS) and glEnd().
11. For the quad strip, we have to mention 5 or more vertices using glVertex2i() between glBegin(GL_QUAD_STRIP) and glEnd().
12. For polygons, we have to mention 3 or more vertices using glVertex2i() between glBegin(GL_POLYGON) and glEnd().
13. Color can be set using glColor3f().
14. Finally, glFlush() will flush the output into the window.

Code

```
#include<GL\glut.h>


void display()

{

   glClear(GL_COLOR_BUFFER_BIT);



   glBegin(GL_POINTS);

   glColor3f(1.0f,1.0f,1.0f);

   glVertex2i(80,950);

   glEnd();

   glPointSize(10);



   glBegin(GL_LINES);

   glColor3f(1.0,0.0,0.0);

   glVertex2i(20,900);
```

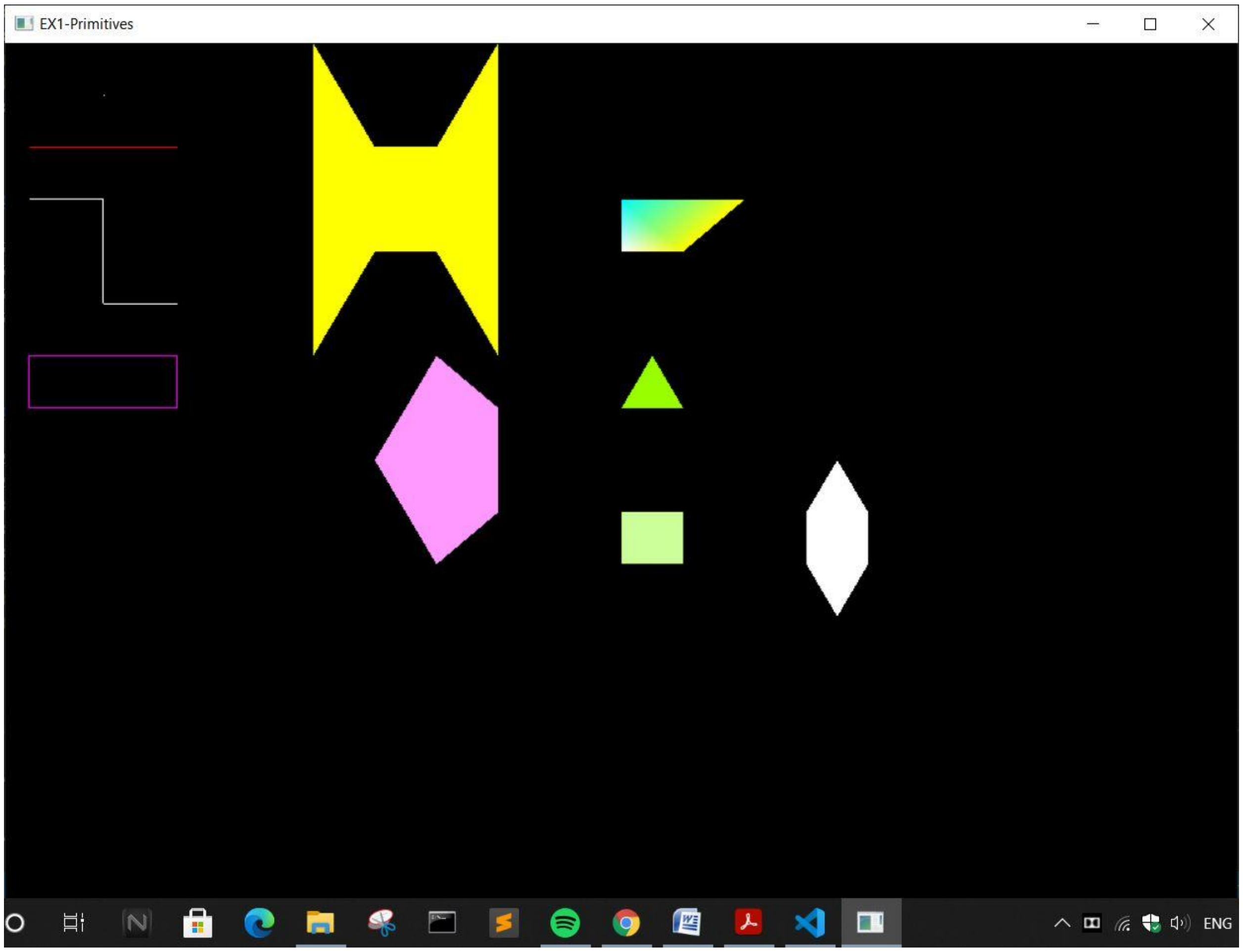
```
glVertex2i(140,900);

glEnd();


glBegin(GL_LINE_STRIP);

glColor3f(1.0,1.0,1.0);

glVertex2i(20,850);

glVertex2i(80,850);

glVertex2i(80,750);

glVertex2i(140,750);

glEnd();


glBegin(GL_LINE_LOOP);

glColor3f(2.0,0.0,1.0);

glVertex2i(20,700);

glVertex2i(140,700);

glVertex2i(140,650);

glVertex2i(20,650);

glEnd();



glBegin(GL_TRIANGLES);

glColor3f(0.6,1.0,0.0);

glVertex2i(500,650);
```

```
glVertex2i(550,650);

glVertex2i(525,700);

glEnd();


glBegin(GL_TRIANGLE_STRIP);

glColor3f(1.0f, 0.99f, 1.0f);

glVertex2i(500,800);

glColor3f(1.0f, 0.99f, 0.0f);

glVertex2i(550,800);

glColor3f(0.0f, 0.99f, 1.0f);

glVertex2i(500,850);

glColor3f(1.0f, 1.1f, 0.0f);

glVertex2i(600,850);

glEnd();


glBegin(GL_TRIANGLE_FAN);

glColor3f(1.0f, 0.6f, 1.0f);

glVertex2i(300,600);

glVertex2i(350,700);

glVertex2i(400,650);

glVertex2i(400,550);

glVertex2i(350,500);

glEnd();
```

```
glBegin(GL_QUADS);

glColor3f(0.8f, 3.0f, 0.6f);

glVertex2i(500,500);

glVertex2i(550,500);

glVertex2i(550,550);

glVertex2i(500,550);

glEnd();



glBegin(GL_QUAD_STRIP);

glColor3f(2.0f, 2.0f, 0.0f);

glVertex2i(250,700);

glVertex2i(250,1000);

glVertex2i(300,800);

glVertex2i(300,900);

glVertex2i(350,800);

glVertex2i(350,900);

glVertex2i(400,700);

glVertex2i(400,1000);

glEnd();
```

```
    glBegin(GL_POLYGON);

    glColor3f(1.5f, 3.0f, 1.0f);

    glVertex2i(650,500);

    glVertex2i(675,450);

    glVertex2i(700,500);

    glVertex2i(700,550);

    glVertex2i(675,600);

    glVertex2i(650,550);

    glEnd();


    glFlush();

}


int main(int argc, char ** argv)

{

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutInitWindowPosition(400, 100);

    glutInitWindowSize(1000,1000);

    glutCreateWindow("EX1-Primitives");

    glClearColor(0,0,0,1);

    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0.0, 1000, 0.0, 1000);

    glutDisplayFunc(display);

    glutMainLoop();

}
```

Output



2. Simple House

Aim

Create a window and draw a simple House using OpenGL.

Algorithm

1. Set Window Size using glutInitWindowSize().
2. Set the background color using gluClearColor().
3. The mainframe of the house is a quad which can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_QUADS) and glEnd().
4. The door of the house is a quad which can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_QUADS) and glEnd().
5. Each window of the house is a quad which can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_QUADS) and glEnd().
6. The doorknob is a point with a size set as 10 and the vertex is specified using glVertex2i() between glBegin(GL_POINTS) and glEnd().
7. The rectangular structure above the mainframe is a quad which can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_QUADS) and glEnd().
8. The roof of the house is a polygon that can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_POLYGON) and glEnd().
9. The chimney of the house is a quad that can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_QUADS) and glEnd().
10. The rectangular structure at the top of the chimney is a quad which can be specified using 4 vertices with the function glVertex2i() between glBegin(GL_QUADS) and glEnd().
11. Color can be set using glColor3f().
12. Finally, glFlush() will flush the output into the window.

Code

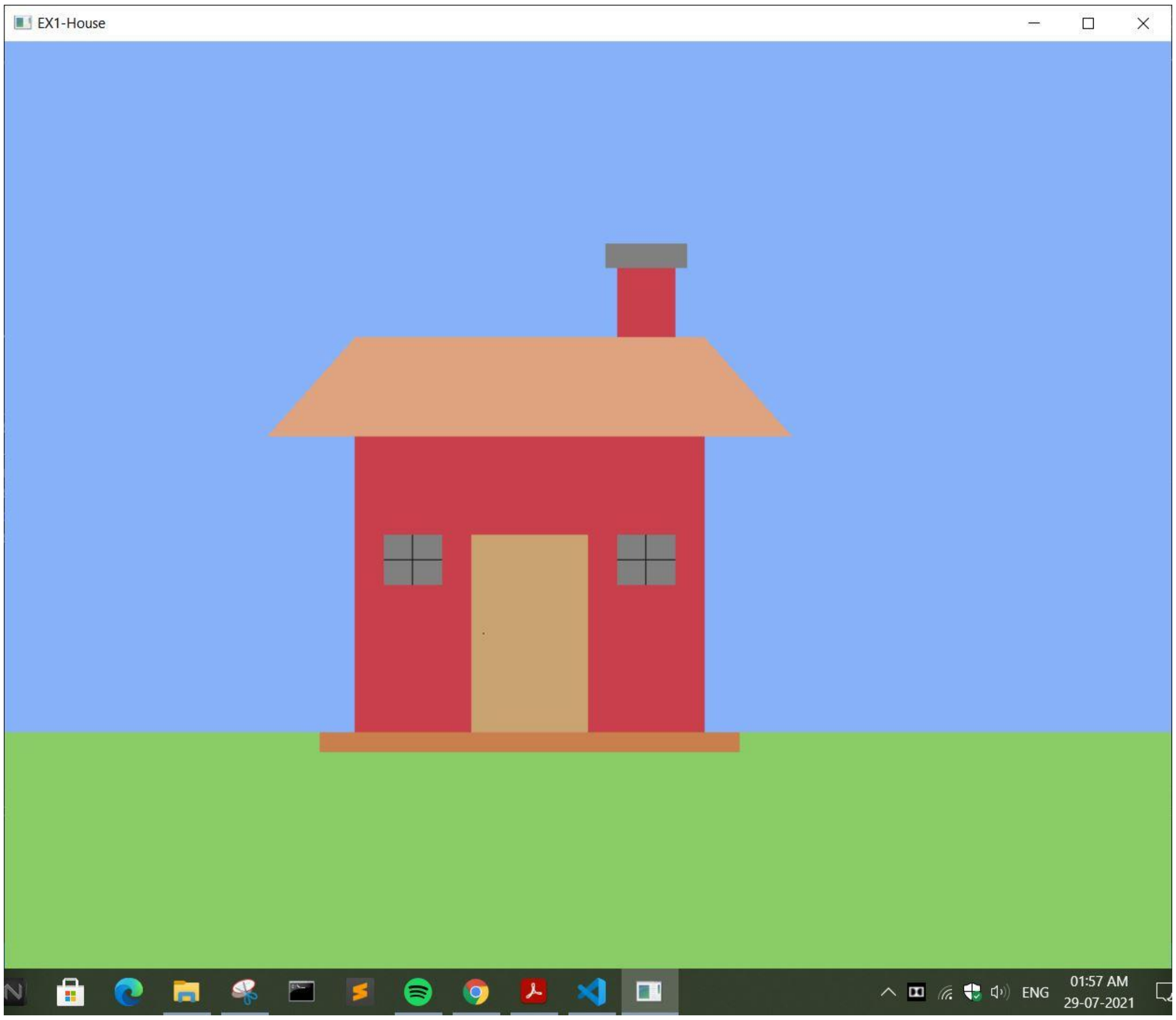
```
#include<GL\glut.h>

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    // Grass
    glBegin(GL_QUADS);
    glColor3f(0.54f, 0.8f, 0.4f);
    glVertex2i(0,0);
    glVertex2i(0,300);
    glVertex2i(1000,300);
    glVertex2i(1000,0);
    glEnd();

    // Walls
    glBegin(GL_QUADS);
    glColor3f(0.79f, 0.25f, 0.3f);
    glVertex2i(300,300);
    glVertex2i(600,300);
    glVertex2i(600,600);
    glVertex2i(300,600);
    glEnd();

    // Step
    glBegin(GL_QUADS);
    glColor3f(0.79f, 0.5f, 0.3f);
    glVertex2i(270,280);
    glVertex2i(630,280);
    glVertex2i(630,300);
    glVertex2i(270,300);
    glEnd();

    // Door
    glBegin(GL_QUADS);
    glColor3f(0.79f, 0.64f, 0.44f);
    glVertex2i(400,300);
    glVertex2i(500,300);
```

```
glVertex2i(500,500);
glVertex2i(400,500);
glEnd();

// Doorknob
glBegin(GL_POINTS);
glColor3f(0.0f,0.0f,0.0f);
glVertex2i(410,400);
glEnd();
glPointSize(100);

// Window L
glBegin(GL_QUADS);
glColor3f(0.5f,0.5f,0.5f);
glVertex2i(325,450);
glVertex2i(375,450);
glVertex2i(375,500);
glVertex2i(325,500);
glEnd();

// Grills
glBegin(GL_LINES);
glColor3f(0.1,0.1,0.1);
glVertex2i(350,450);
glVertex2i(350,500);
glEnd();

glBegin(GL_LINES);
glColor3f(0.1,0.1,0.1);
glVertex2i(325,475);
glVertex2i(375,475);
glEnd();

// Window R
glBegin(GL_QUADS);
glColor3f(0.5f,0.5f,0.5f);
glVertex2i(525,450);
glVertex2i(575,450);
glVertex2i(575,500);
```

```
glVertex2i(525,500);
glEnd();

glBegin(GL_LINES);
glColor3f(0.1,0.1,0.1);
glVertex2i(550,450);
glVertex2i(550,500);
glEnd();

glBegin(GL_LINES);
glColor3f(0.1,0.1,0.1);
glVertex2i(525,475);
glVertex2i(575,475);
glEnd();

// Roof
glBegin(GL_POLYGON);
glColor3f(0.88f, 0.64f, 0.5f);
glVertex2i(225,600);
glVertex2i(675,600);
glVertex2i(600,700);
glVertex2i(300,700);
glEnd();

// Chimney
glBegin(GL_QUADS);
glColor3f(0.79f, 0.25f, 0.3f);
glVertex2i(525,700);
glVertex2i(575,700);
glVertex2i(575,770);
glVertex2i(525,770);
glEnd();

glBegin(GL_QUADS);
glColor3f(0.5f, 0.5f, 0.5f);
glVertex2i(515,770);
glVertex2i(585,770);
glVertex2i(585,795);
glVertex2i(515,795);
```

```
    glEnd();

    glFlush();
}



int main(int argc, char ** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(500, 0);
    glutInitWindowSize(1000,1000);
    glutCreateWindow("EX1-House");
    glClearColor(0.53, 0.7, 0.98, 1);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0.0, 1000, 0.0, 1000);
    glutDisplayFunc(display);
    glutMainLoop();
}
```

Output



Result

Thus basic primitives have been displayed and a simple house has been designed.