**SSN College of Engineering**

**Department of Computer Science and Engineering**

## UCS1712 – GRAPHICS AND MULTIMEDIA LAB

**Assignment-1 – Study of Basic Output Primitives**

SATHEESH KUMAR G R

185001136

AIM:

To create a window using OPENGL and to draw the basic output primitives like POINTS,LINES etc.

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 glVertex2d() between glBegin(GL\_POINTS) and glEnd().
4. For lines we have to mention 2 vertices using glVertex2d() between glBegin(GL\_LINES) and glEnd().
5. For line strip we have to mention 3 or more vertices using glVertex2d() between glBegin(GL\_LINE\_STRIP) and glEnd().
6. For line loop we have to mention 4 or more vertices using glVertex2d() between glBegin(GL\_LINE\_LOOP) and glEnd().
7. For triangles we have to mention 3 vertices in anti-clockwise order using glVertex2d() between glBegin(GL\_TRIANGLES) and glEnd().
8. For triangle strips we have to mention 4 or more vertices in anti-clockwise order using glVertex2d() between glBegin(GL\_TRIANGLE\_STRIP) and glEnd().
9. For triangle fan we have to mention 4 or more vertices in anti-clockwise order using glVertex2d() between glBegin(GL\_TRIANGLE\_FAN) and glEnd().
10. For quads we have to mention 4 vertices using glVertex2d() between glBegin(GL\_QUADS) and glEnd().
11. For quad strip we have to mention 5 or more vertices using glVertex2d() between glBegin(GL\_QUAD\_STRIP) and glEnd().
12. For polygons we have to mention 3 or more vertices using glVertex2d() between glBegin(GL\_POLYGON) and glEnd().
13. Color can be set using glColor3d().
14. Finally glFlush() will flush the output into the window.

CODE:

#include<GL/glut.h>

void display()

{

/\* clear window \*/

glClear(GL\_COLOR\_BUFFER\_BIT);

/\*point\*/

glBegin(GL\_POINTS);

glVertex2d(10,100);

glEnd();

/\*line\*/

glBegin(GL\_LINES);

glVertex2d(20,100);

glVertex2d(50,100);

glEnd();

/\*line strip\*/

glBegin(GL\_LINE\_STRIP);

glVertex2d(60,100);

glVertex2d(100,100);

glVertex2d(100,120);

glEnd();

/\*line loop\*/

glBegin(GL\_LINE\_LOOP);

glVertex2d(120,100);

glVertex2d(170,100);

glVertex2d(170,180);

glVertex2d(190,80);

glEnd();

/\*Triangle\*/

glBegin(GL\_TRIANGLES);

glVertex2d(200,100);

glVertex2d(250,100);

glVertex2d(250,150);

glEnd();

/\*Triangle Strip\*/

glBegin(GL\_TRIANGLE\_STRIP);

glVertex2d(300,100);

glVertex2d(350,100);

glVertex2d(350,150);

glVertex2d(400,100);

glVertex2d(450,150);

glEnd();

/\*Triangle Fan\*/

glBegin(GL\_TRIANGLE\_FAN);

glVertex2d(500,100);

glVertex2d(550,220);

glVertex2d(600,340);

glVertex2d(650,220);

glVertex2d(700,320);

glEnd();

/\*Quads\*/

glBegin(GL\_QUADS);

glVertex2d(60, 330);

glVertex2d(75, 320);

glVertex2d(75, 280);

glVertex2d(60, 270);

glEnd();

/\*Quad Strip\*/

glBegin(GL\_QUAD\_STRIP);

glVertex2d(100, 330);

glVertex2d(115, 320);

glVertex2d(115, 280);

glVertex2d(100, 270);

glEnd();

/\*polygon \*/

glBegin(GL\_POLYGON);

glVertex2d(150,450);

glVertex2d(170,450);

glVertex2d(190,350);

glVertex2d(170,250);

glVertex2d(150,250);

glVertex2d(130,350);

glEnd();

/\* flush GL buffers \*/

glFlush();

}

void init() // initialize colors

{

/\* set clear color to black \*/

glClearColor(0.0, 0.0, 0.0, 0.0);

/\* set fill color to white \*/

glColor3f(1.0, 1.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glPointSize(5);

gluOrtho2D(0.0, 1000.0, 0.0, 500.0);

}

int main(int argc, char\*\* argv)

{

/\* Initialize mode and open a window in upper left corner of/\* screen \*/

/\* Window title is name of program (arg[0]) \*/

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(1000, 500); // Set window Size

glutInitWindowPosition(0, 0); //Set Window Position

glutCreateWindow("simple"); //Create Window and Set title

glutDisplayFunc(display); //Call the Displaying function

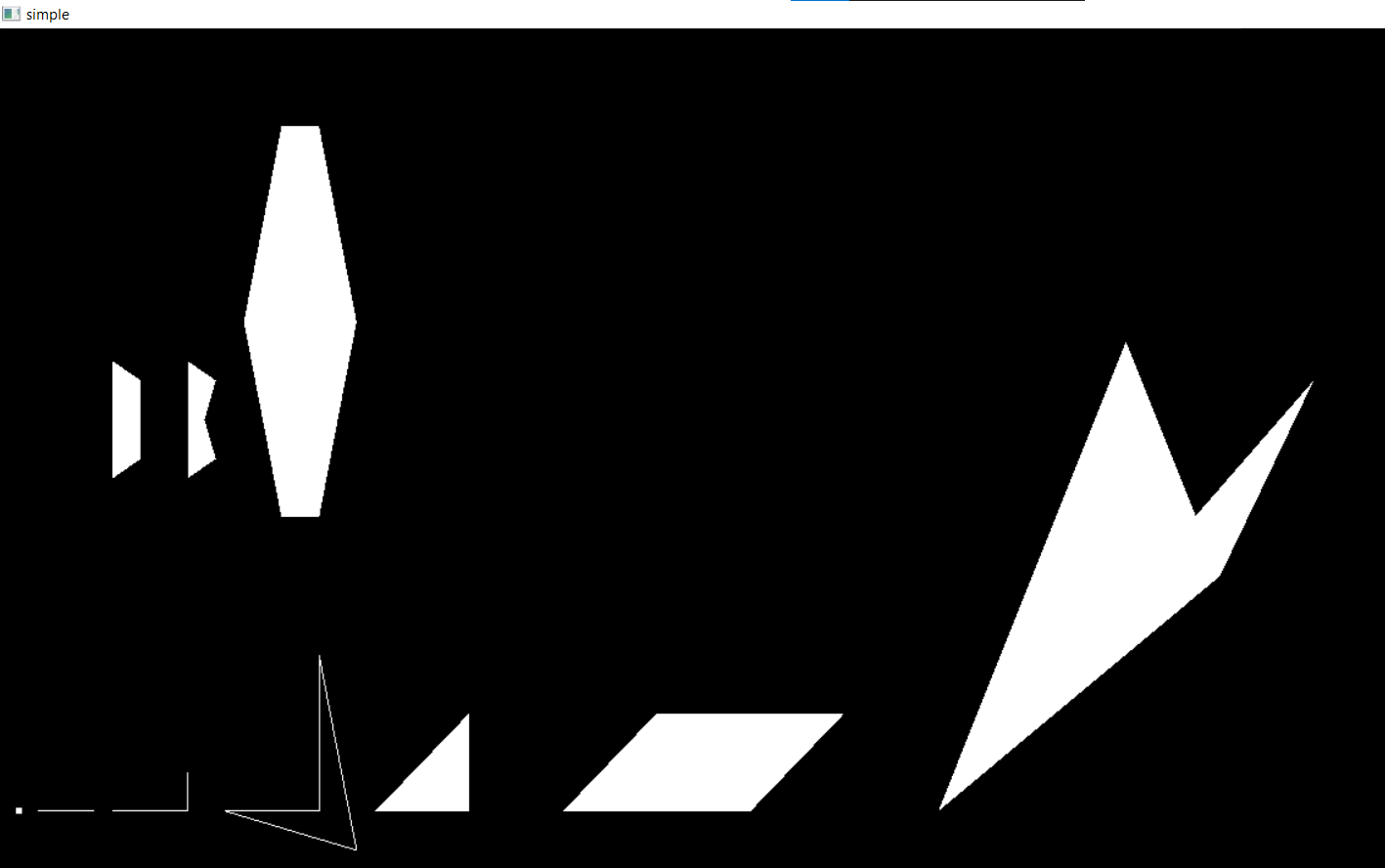
init(); //Initialize Drawing Colors

glutMainLoop(); //Keep displaying until program is closed.

return 1;

}

OUTPUT:



2)

AIM:

To draw a simple house in OpenGL window using output primitives.

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 glVertex2d() 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 glVertex2d() 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 glVertex2d() between glBegin(GL\_QUADS) and glEnd().
6. The door knob is a point with size set as 10 and the vertex is specified using glVertex2d() 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 glVertex2d() between glBegin(GL\_QUADS) and glEnd().
8. The roof of the house is a polygon which can be specified using 4 vertices with the function glVertex2d() between glBegin(GL\_POLYGON) and glEnd().
9. The chimney of the house is a quad which can be specified using 4 vertices with the function glVertex2d() between glBegin(GL\_QUADS) and glEnd().
10. The rectangular structure in the top of the chimney is a quad which can be specified using 4 vertices with the function glVertex2d() between glBegin(GL\_QUADS) and glEnd().
11. Color can be set using glColor3d().
12. Finally glFlush() will flush the output into the window

CODE:

#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

//bottom

glColor3d(0.0,0.7,0.0);

glBegin(GL\_QUADS);

glVertex2d(0,0);

glVertex2d(640,0);

glVertex2d(640,100);

glVertex2d(0,100);

//left mid

glColor3d(0.7,0.7,0.2);

glVertex2d(50,100);

glVertex2d(250,100);

glVertex2d(250,300);

glVertex2d(50,300);

//right mid

glColor3d(0.7,0.2,0.2);

glVertex2d(250,100);

glVertex2d(590,100);

glVertex2d(590,300);

glVertex2d(250,300);

//right top

glColor3d(0.6,0.3,0.1);

glVertex2d(250,300);

glVertex2d(590,300);

glVertex2d(490,420);

glVertex2d(150,420);

//door

glColor3d(0.6,0.6,0.6);

glVertex2d(110,100);

glVertex2d(190,100);

glVertex2d(190,200);

glVertex2d(110,200);

//window

glColor3d(0.6,0.6,0.6);

glVertex2d(390,175);

glVertex2d(490,175);

glVertex2d(490,230);

glVertex2d(390,230);

glEnd();

//left top

glColor3d(0.7,0.1,0.8);

glBegin(GL\_TRIANGLES);

glVertex2d(50,300);

glVertex2d(250,300);

glVertex2d(150,420);

glEnd();

//window cross and inner border

glColor3d(0.0,0.0,0.0);

glLineWidth(3.0);

glBegin(GL\_LINES);

glVertex2d(390,202.5);

glVertex2d(490,202.5);

glVertex2d(440,175);

glVertex2d(440,230);

glVertex2d(50,300);

glVertex2d(590,300);

glVertex2d(250,300);

glVertex2d(150,420);

glVertex2d(250,100);

glVertex2d(250,300);

glEnd();

//window border

glBegin(GL\_LINE\_LOOP);

glVertex2d(390,175);

glVertex2d(490,175);

glVertex2d(490,230);

glVertex2d(390,230);

glEnd();

//door border

glBegin(GL\_LINE\_STRIP);

glVertex2d(190,100);

glVertex2d(190,200);

glVertex2d(110,200);

glVertex2d(110,100);

glEnd();

//border

glBegin(GL\_LINE\_STRIP);

glVertex2d(50,100);

glVertex2d(50,300);

glVertex2d(150,420);

glVertex2d(490,420);

glVertex2d(590,300);

glVertex2d(590,100);

glEnd();

//bottom border

glBegin(GL\_LINE\_STRIP);

glVertex2d(640,100);

glVertex2d(0,100);

glEnd();

//door lock

glPointSize(12.0);

glBegin(GL\_POINTS);

glVertex2d(125,155);

glEnd();

glFlush();

}

void init()

{

glClearColor(0.0,0.5,1.0,0.0);

}

void reshape(int w,int h)

{

//viewport

glViewport(0,0,w,h);

//projection

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0,640,0,480);

glMatrixMode(GL\_MODELVIEW);

}

int main(int argc,char \*argv[])

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowPosition(300,100);

glutInitWindowSize(640,480);

glutCreateWindow("House");

glutDisplayFunc(display);

glutReshapeFunc(reshape);

init();

glutMainLoop();

return 1;

}

OUTPUT:

