****

**UCS1712-Graphics and Multimedia Lab**

**Programming Assignment 1**

**Study of Basic Output Primitives in C++ using OpenGL**

Name: Jayannthan P T

Dept: CSE ‘A’

Roll No.: 205001049

a)To create an output window using OPENGL and to draw the following basic output primitives:

**Source code:**

* POINTS

void **points**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_POINTS);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(0.5, 0.5);

    //    glVertex2f(0.0, 0.5);

**glEnd**();

**glFlush**();

}

* LINES

void **lines**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_LINES);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glEnd**();

**glFlush**();

}

* LINE\_STRIP

void **linesstrip**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_LINE\_STRIP);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(1.0, 1.0);

**glVertex2f**(0.7, 0.7);

**glEnd**();

**glFlush**();

}

* LINE\_LOOP

void **lineloop**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_LINE\_LOOP);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(1.0, 1.0);

**glVertex2f**(0.7, 0.7);

**glEnd**();

**glFlush**();

}

* TRIANGLES

void **triangle**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_TRIANGLES);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(0.5, 0.5);

    //    glVertex2f(0.0, 0.5);

**glEnd**();

**glFlush**();

}

* QUADS

void **quadrant**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_QUADS);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(0.5, 0.5);

**glVertex2f**(0.0, 0.5);

**glEnd**();

**glFlush**();

}

* QUAD\_STRIP

void **quadstrip**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_QUAD\_STRIP);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(0.5, 0.5);

**glVertex2f**(0.0, 0.5);

**glVertex2f**(1.0, 1.0);

**glVertex2f**(0.7, 0.7);

**glEnd**();

**glFlush**();

}

* POLYGON

void **polygon**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glBegin**(GL\_POLYGON);

**glVertex2f**(0.0, 0.0);

**glVertex2f**(0.5, 0.0);

**glVertex2f**(0.5, 0.5);

**glVertex2f**(0.0, 0.5);

**glVertex2f**(1.0, 1.0);

**glVertex2f**(0.7, 0.7);

**glEnd**();

**glFlush**();

}

b) To create an output window and draw a checkerboard using OpenGL.

**Source code:**

#include <iostream>

#include <GLUT/glut.h>

const int windowWidth = 400;

const int windowHeight = 400;

const int numCheckers = 8;

const int checkerSize = windowWidth / numCheckers;

void **drawCheckerboard**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(1.0, 1.0, 1.0, 0.0);

    for (int row = 0; row < numCheckers; row++)

    {

        for (int col = 0; col < numCheckers; col++)

        {

            if ((row + col) % 2 == 0)

**glColor3f**(0.0f, 0.0f, 0.0f); // Black

            else

**glColor3f**(1.0f, 1.0f, 1.0f); // White

            int x = col \* checkerSize;

            int y = row \* checkerSize;

**glBegin**(GL\_QUADS);

**glVertex2f**(x, y);

**glVertex2f**(x + checkerSize, y);

**glVertex2f**(x + checkerSize, y + checkerSize);

**glVertex2f**(x, y + checkerSize);

**glEnd**();

        }

    }

**glFlush**();

}

void **myInit**()

{

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glMatrixMode**(GL\_PROJECTION);

**glLoadIdentity**();

**gluOrtho2D**(0, windowWidth, 0, windowHeight);

}

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

{

**glutInit**(&argc, argv);

**glutInitDisplayMode**(GLUT\_SINGLE | GLUT\_RGB);

**glutInitWindowSize**(windowWidth, windowHeight);

**glutCreateWindow**("Checkerboard");

**glutDisplayFunc**(**drawCheckerboard**);

**myInit**();

**glutMainLoop**();

    return 0;

}

****

c) To create an output window and draw a house using POINTS, LINES, TRAINGLES and QUADS/POLYGON.

#include <GLUT/glut.h>

void **myInit**()

{

**glClearColor**(1.0, 1.0, 1.0, 0.0);

**glColor3f**(0.0f, 0.0f, 0.0f);

**glPointSize**(10);

**glMatrixMode**(GL\_PROJECTION);

**glLoadIdentity**();

**gluOrtho2D**(0.0, 640.0, 0.0, 480.0);

}

void **displayPoints**()

{

**glBegin**(GL\_POINTS);

**glVertex2d**(150, 100);

**glVertex2d**(100, 230);

**glVertex2d**(170, 130);

**glVertex2d**(300, 350);

**glEnd**();

}

void **displayRectangle**(int x, int y, int w, int h)

{

**glBegin**(GL\_POLYGON);

**glVertex2d**(x, y);

**glVertex2d**(x + w, y);

**glVertex2d**(x + w, y + h);

**glVertex2d**(x, y + h);

**glEnd**();

}

void **displayTriangle**(int x, int y, int w, int h)

{

**glBegin**(GL\_TRIANGLES);

**glVertex2d**(x, y);

**glVertex2d**(x + w, y);

**glVertex2d**(x + (w / 2), y + h);

**glEnd**();

}

void **displayLine**(int x1, int y1, int x2, int y2)

{

**glBegin**(GL\_LINES);

**glVertex2d**(x1, y1);

**glVertex2d**(x2, y2);

**glEnd**();

}

void **displayHouse**()

{

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glColor4f**(0.268, 0.243, 0.217, 1);

**displayRectangle**(200, 0, 150, 150);

**glColor4f**(0.54, 0.54, 0.51, 1);

**displayTriangle**(200, 150, 150, 100);

**glColor4f**(0.56, 0.55, 0, 1);

**displayRectangle**(250, 0, 50, 80);

**glColor4f**(0, 0, 0, 1);

**displayRectangle**(280, 30, 10, 10);

}

