****

**UCS1712-Graphics and Multimedia Lab**

**Programming Assignment 4**

**Midpoint Circle Drawing Algorithm in C++ using OpenGL**

Name: Jayannthan P T

Dept: CSE ‘A’

Roll No.: 205001049

a) To plot points that make up the circle with center (xc,yc) and radius r using Midpoint circle drawing

algorithm. Give atleast 2 test cases.

Case 1: With center (0,0)

Case 2: With center (xc,yc)

b) To draw any object using line and circle drawing algorithms.

**Source code:**

#include <stdlib.h>

#include <GLUT/glut.h>

#include <iostream>

using namespace std;

int xc, yc, r;

void **myInit**()

{

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

**glColor3f**(0.4, 0.4, 0.9);

**glMatrixMode**(GL\_PROJECTION);

**glLoadIdentity**();

**glPointSize**(2);

**gluOrtho2D**(-250.0, 250.0, -250.0, 250.0);

}

void **plotAll**(int x, int y, int xc, int yc)

{

**glVertex2d**(x + xc, y + yc);

**glVertex2d**(x + xc, -y + yc);

**glVertex2d**(-x + xc, y + yc);

**glVertex2d**(-x + xc, -y + yc);

**glVertex2d**(y + xc, x + yc);

**glVertex2d**(y + xc, -x + yc);

**glVertex2d**(-y + xc, x + yc);

**glVertex2d**(-y + xc, -x + yc);

}

void **circle**()

{

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

    int x = r, y = 0, pk = 1 - r;

**glBegin**(GL\_POINTS);       // Draw the x-axis and y-axis

**glColor3f**(0.0, 0.0, 0.0); // Set color to black for axes

    // X-axis

**glVertex2d**(-250, 0);

**glVertex2d**(250, 0);

    // Y-axis

**glVertex2d**(0, -250);

**glVertex2d**(0, 250);

    // Draw the circle using Mid-Point Circle Algorithm

**lColor3f**(0.4, 0.4, 0.9); // Set color back to blue for the circle

**plotAll**(x, y, xc, yc);

    while (x > y)

    {

        y++;

        if (pk < 0)

        {

            pk += (2 \* y) + 1;

        }

        else

        {

            x--;

            pk += (2 \* y) - (2 \* x) + 1;

        }

**plotAll**(x, y, xc, yc);

    }

**glEnd**();

**glFlush**();

}

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

{

    cout **<<** "Enter circle center coordinates:";

    cin **>>** xc **>>** yc;

    cout **<<** "Enter radius:";

    cin **>>** r;

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

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

**glutInitWindowSize**(500, 500);

**glutCreateWindow**("Mid Point Circle Algorithm");

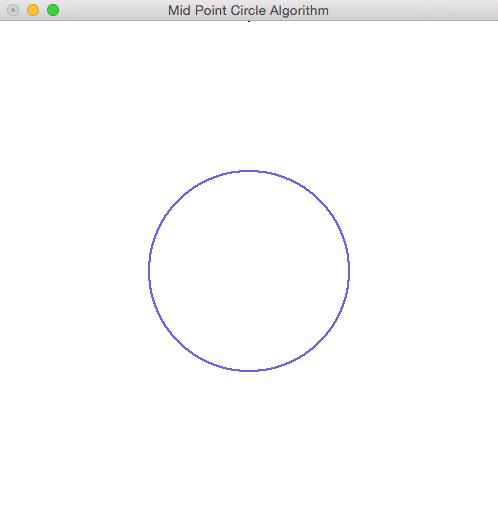
**glutDisplayFunc**(**circle**);

**myInit**();

**glutMainLoop**();

    return 1;

}

****