# Bangladesh University of Business and Technology

**Lab no: 3**

Course Name: Computer Graphics Lab

Course Code:  CSE 342

**Submitted By:**

Name: Iftekhar Ahamed Siddiquee
ID : 19202103239
Intake : 44
Section: 6

**Submitted  To:**

Ms. Sweety  Lima
Lecturer
Department of CSE
BUBT

Submission Date :  26/01/2023

## Lab No: 03

## Lab Task Name: Mid-Point Circle Drawing Algorithm

## Objective:

The mid-point circle drawing algorithm is an algorithm used to determine the points needed for rasterizing a circle. We use the mid-point algorithm to calculate all the perimeter points of the circle in the first octant and then print them along with their mirror points in the other octants.

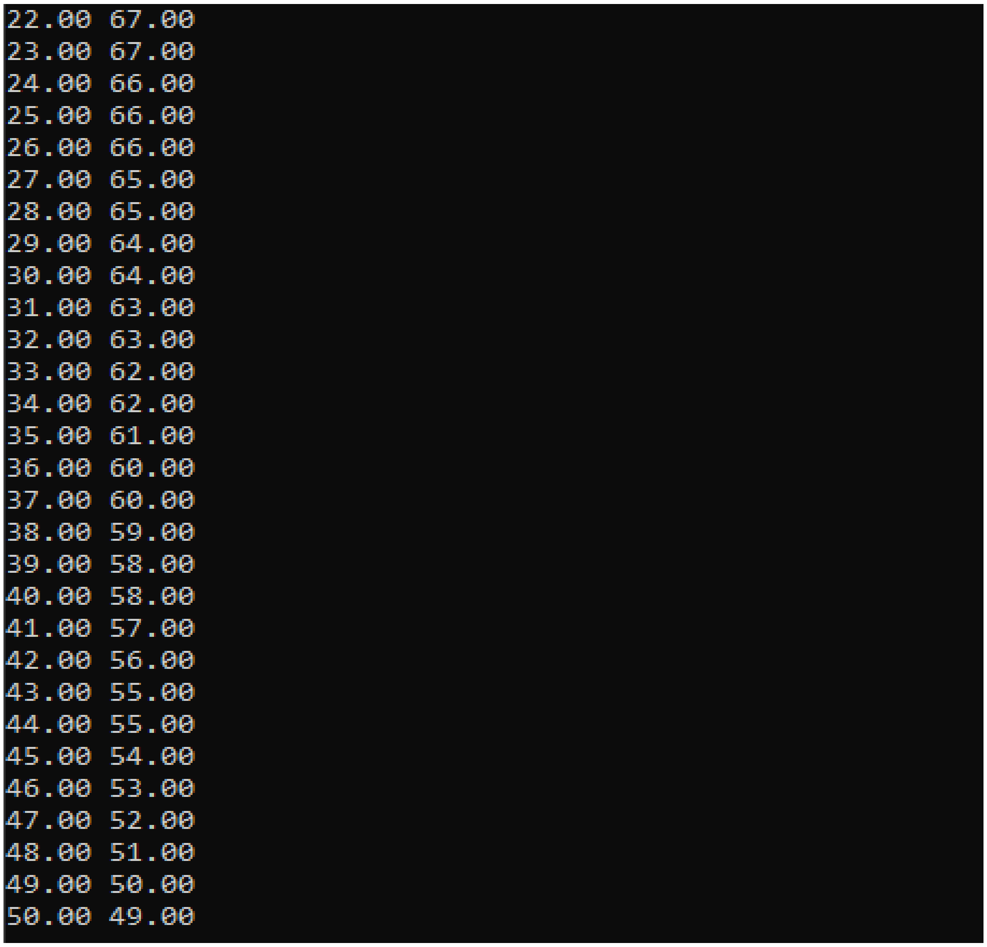
## Algorithm:

```
p=1-r;
   while((x<=y))
   {
      if(p<0)
      {
         x++;
         printf("%0.2f
%0.2f\n",x,y);
         p=p+(2*x)+3;
      }


      else
      {
         x++;
         y--;
         printf("%0.2f
%0.2f\n",x,y);
         p=p+(2*x)+5-
(2*y);
      }
   }
```

## **Program:**

```
#include <stdio.h>
#include <GL/gl.h>
#include <GL/glut.h>
float x=0,y,x2,y2,m,i,j,p;
int dx=0,dy=0,r;
void display(void)
{
    /* clear all pixels */
    glClear
(GL_COLOR_BUFFER_BIT);
    /* draw white polygon
(rectangle) with corners at
    * (0.25, 0.25, 0.0) and (0.75,
0.75, 0.0)
    */
    glEnd();

    glColor3f (0.0, 2.0, 1.0);
    glBegin(GL_POINTS);
    p=1-r;
    while((x<=y))
    {
        if(p<0)
        {
            x++;
            printf("%0.2f %0.2f\
n",x,y);
            p=p+(2*x)+3;
        }



        else
        {
            x++;
            y--;
            printf("%0.2f %0.2f\
n",x,y);
            p=p+(2*x)+5-(2*y);
        }
        glVertex3f (((x/100)),
((y/100)), 0.0);
        glVertex3f (((y/100)),
((x/100)), 0.0);
        glVertex3f ((-(x/100)), (-
(y/100)), 0.0);
        glVertex3f ((-(x/100)),
((y/100)), 0.0);
        glVertex3f (((x/100)), (-
(y/100)), 0.0);
        glVertex3f (((y/100)), (-
(x/100)), 0.0);
        glVertex3f ((-(y/100)), (-
(x/100)), 0.0);
        glVertex3f ((-(y/100)),
((x/100)), 0.0);
    }

    glEnd();
    glFlush ();
}
void init (void)
{
    /* select clearing
(background) color */
    glClearColor (0.0, 0.0, 0.0,
0.0);
    /* initialize viewing values */

glMatrixMode(GL_PROJECTI
ON);
    glLoadIdentity();
    glOrtho(-1.0, 1.0, -1.0, 1.0, -
1.0, 1.0);
    /**
    gluOrtho2D(-300, 300, 0,
600);

    */
}
/*
* Declare initial window size,
position, and display mode
* (single buffer and RGBA).
Open window with "hello"
* in its title bar. Call
initialization routines.
* Register callback function to
display graphics.
* Enter main loop and process
events.
*/
int main(int argc, char** argv)
{
    printf("Enter radius: ");
    scanf("%d",&r);
    y=r;




    glutInit(&argc, argv);
    glutInitDisplayMode
(GLUT_SINGLE |
GLUT_RGB);
    glutInitWindowSize (500,
500);
    glutInitWindowPosition
(100, 100);
    glutCreateWindow ("hello");
    init ();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0; /* ISO C requires
main to return int. */
}
```

## Input & Output:



```
22.00 67.00
23.00 67.00
24.00 66.00
25.00 66.00
26.00 66.00
27.00 65.00
28.00 65.00
29.00 64.00
30.00 64.00
31.00 63.00
32.00 63.00
33.00 62.00
34.00 62.00
35.00 61.00
36.00 60.00
37.00 60.00
38.00 59.00
39.00 58.00
40.00 58.00
41.00 57.00
42.00 56.00
43.00 55.00
44.00 55.00
45.00 54.00
46.00 53.00
47.00 52.00
48.00 51.00
49.00 50.00
50.00 49.00
```