Bangladesh University of Business and Technology

Lab no: 2

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 : 17/01/2023

Lab No: 02

Lab Task Name: Mid-Point Line Drawing Algorithm

Objective:

In computer graphics, a line drawing algorithm is an algorithm for approximating a line segment on discrete graphical media, such as pixel-based displays and printers. On such media, line drawing requires an approximation (in nontrivial cases). Basic algorithms rasterize lines in one color.

Algorithm:

```
int dx = X2 - X1;
   int dy = Y2 - Y1;

   if(dy<=dx) //m<=0
   {
      int d = dy - (dx/2);
      int x = X1, y = Y1;
      cout << x << "," << y
<< "\n";

      while (x < X2)
      {
         x++;

         if (d < 0)
            d = d + dy;

         else
         {
            d = d+(dy - dx);
            y++;
         }
         cout << x << ","
<< y << "\n";
      }
   }

   else if(dx<dy) //m>1
   {
      int d = dx - (dy/2);
```

```
        int x = X1, y = Y1;
        cout << x << "," << y
<< "\n";

        while (y < Y2)
        {
           y++;

           if (d < 0)
              d = d + dx;

           else
           {
              d = d+(dx - dy);
              x++;
           }
           cout << x << ","
<< y << "\n";
        }
     }
```

Program:

```
#include<bits/stdc++.h>
using namespace std;
#include<stdio.h>
#include <GL/gl.h>
#include <GL/glut.h>


int X1 = 2, Y1 = 2, X2 = 8,
Y2 = 5;
void midPoint(void)
{
   glClear
(GL_COLOR_BUFFER_B
IT);
   glEnd();
   glColor3f (0.0, 1.0, 0.0);
   glBegin(GL_POINTS);
```

```
    int dx = X2 - X1;
    int dy = Y2 - Y1;

    if(dy<=dx) //m<=0
    {
        int d = dy - (dx/2);
        int x = X1, y = Y1;
        cout << x << "," << y
<< "\n";

        while (x < X2)
        {
            x++;

            if (d < 0)
                d = d + dy;

            else
            {
                d = d+(dy - dx);
                y++;
            }
            cout << x << "," <<
y << "\n";
        }
    }

    else if(dx<dy) //m>1
    {
        int d = dx - (dy/2);
        int x = X1, y = Y1;
        cout << x << "," << y
<< "\n";

        while (y < Y2)
        {
            y++;

            if (d < 0)
```

```
            d = d + dx;

         else
         {
            d = d+(dx - dy);
            x++;
         }
         cout << x << "," <<
y << "\n";
      }
   }
   glEnd();
   glFlush ();
}

void init (void)
{
   glClearColor (0.0, 0.0,
0.0, 0.0);

glMatrixMode(GL_PROJE
CTION);
   glLoadIdentity();
   glOrtho(0.0, 1.0, 0.0,
1.0, -1.0, 1.0);
}

int main(int argc, char**
argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode
(GLUT_SINGLE |
GLUT_RGB);
   glutInitWindowSize
(500, 500);
   glutInitWindowPosition
(100, 100);
   glutCreateWindow
("hello");
```

```
    init ();

 glutDisplayFunc(midPoint)
 ;
    glutMainLoop();
    return 0;
 }
```

Input & Output: