**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**Ex. No. 3 Bresenham’s Line Drawing Algorithm in C++ using OpenGL**

**Date:** 2/8/21 **Name:** Srinath S

**Class:** CSE-C  **Roll:** 185001205

**Question:**

To plot points that make up the line with endpoints (x0,y0) and (xn,yn) using Bresenham’s line drawing algorithm.

Case 1: +ve slope Left to Right line

Case 2: +ve slope Right to Left line

Case 3: -ve slope Left to Right line

Case 4: -ve slope Right to Left line

Each case has two subdivisions (i) |m|<= 1 (ii) |m|>1

Note that all four cases of line drawing must be given as test cases.

**Code:**

#include<GL/glut.h>

#include<stdio.h>

#include<iostream>

using namespace std;

float x1\_arr[8], y1\_arr[8], x2\_arr[8], y2\_arr[8];

void myInit()

{

glClearColor(0.0,0.0,0.0,1.0);

gluOrtho2D(-20,100,-20,100);

}

void myDisplay()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

float dy,dx,step,x,y,k,Xin,Yin,p;

float x1, y1, x2, y2;

for (int i = 0; i < 3; i++) {

x1 = x1\_arr[i];

y1 = y1\_arr[i];

x2 = x2\_arr[i];

y2 = y2\_arr[i];

dx=x2-x1;

dy=y2-y1;

        p=2dy-dx;

x= x1;

y=y1;

        while(x<=x2 && y <= y2)

        {

            if(p>=0)

            {

                glColor3f(1.0,1.0,1.0);

                glBegin(GL\_POINTS);

                    glVertex2i(x,y);

                glEnd();

                y=y+1;

                p=p+2dy-2dx;

            }

            else

            {

                glColor3f(1.0,1.0,1.0);

                glBegin(GL\_POINTS);

                    glVertex2i(x,y);

                glEnd();

                p=p+2dy;

                x=x+1;

            }

        }

        glColor3f(1.0,1.0,1.0);

        glBegin(GL\_POINTS);

            glVertex2i(x,y);

        glEnd();

}

glFlush();

}

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

{

x1\_arr[0] = 10;

y1\_arr[0] = 10;

x2\_arr[0] = 80;

y2\_arr[0] = 10;

x1\_arr[1] = 10;

y1\_arr[1] = 70;

x2\_arr[1] = 80;

y2\_arr[1] = 70;

    x1\_arr[2] = 45;

y1\_arr[2] = 10;

x2\_arr[2] = 45;

y2\_arr[2] = 70;

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (100,100);

glutCreateWindow("check");

glutDisplayFunc(myDisplay);

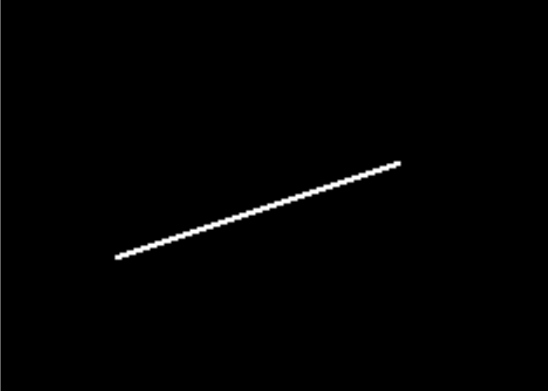
myInit();

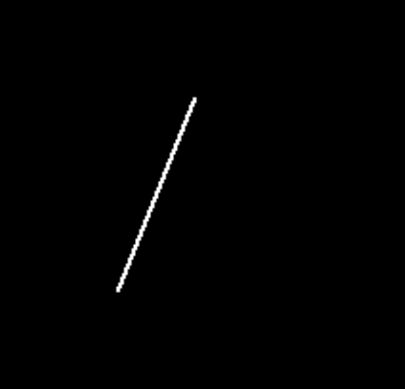
glutMainLoop();

return 1;

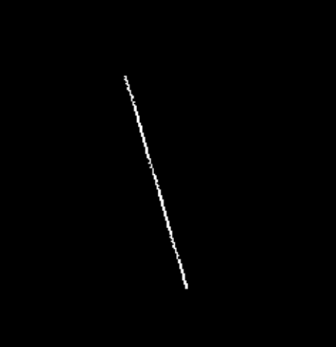
}

**Output:**

****

****

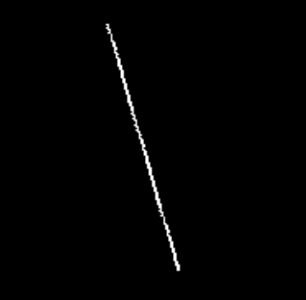
****

****

****

****

****

****