**SSN COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**EX NO: 3 – Drawing 2D Primitives –Line – Bresenham’s Algorithm**

Name : **S.Nachammai Devi Pooja**

Register No : **185001096**

Date : 11/08/2021

Class & Sec : CSE B

**AIM :**

-> To plot points that make up the line with endpoints (x0,y0) and (xn,yn) using Bresenham’s line drawing algorithm for the following case (i) |m|< 1 (ii) |m|>=1

-> To Write a program using OPENGL to write any Alphabet with the help of Bresenham’s line drawing algorithm.

**ALGORITHM :**

Step 1: Read two points (X1, Y1) and (X2, Y2) and assign (X1, Y1) to (X, Y)

Step 2: Compute the difference between X and Y coordinates as dx and dy.

Step 3: Calculate and store the absolute values of dx and dy

Step 4: If x2<x1, set incx as -1, else 1

Step 5: If y2<y1, set incy as -1, else 1

Step 6: If slope < 1 (dx>dy)

* Plot (x,y)
* compute e = 2 \* dy - dx; inc1 = 2 \* (dy - dx); inc2 = 2 \* dy
* loop through dx
  + if e>0, increment y by incy and e by inc1
  + else increment e by inc2
  + increment x by incx
  + plot (x,y)

Step 7: Else

* plot (x,y)
* compute e = 2 \* dy - dx; inc1 = 2 \* (dx - dy); inc2 = 2 \* dx
* loop through dy
  + if e>0, increment x by incyx and e by inc1
  + else increment e by inc2
  + increment y by incy
  + plot (x,y)

**CODE:**

#include<gL/glut.h>

#include<math.h>

#include<stdio.h>

#include<iostream>

float x\_1, y\_1, x\_2, y\_2;

void init() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0.9, 0.8, 0.8, 0.9);

glColor3f(0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0, 500, 0, 500);

}

void draw\_pixel(int x, int y) {

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

}

void bresen\_line(int x1, int x2, int y1, int y2) {

int dx, dy, i, e;

int incx, incy, inc1, inc2;

int x, y;

dx = x2 - x1;

dy = y2 - y1;

if (dx < 0)

dx = -dx;

if (dy < 0)

dy = -dy;

incx = 1;

if (x2 < x1) incx = -1;

incy = 1;

if (y2 < y1) incy = -1;

x = x1; y = y1;

if (dx > dy) {

draw\_pixel(x, y);

e = 2 \* dy - dx;

inc1 = 2 \* (dy - dx);

inc2 = 2 \* dy;

for (i = 0; i < dx; i++) {

if (e >= 0) {

y += incy;

e += inc1;

}

else

e += inc2;

x += incx;

draw\_pixel(x, y);

}

}

else {

draw\_pixel(x, y);

e = 2 \* dx - dy;

inc1 = 2 \* (dx - dy);

inc2 = 2 \* dx;

for (i = 0; i < dy; i++) {

if (e >= 0) {

x += incx;

e += inc1;

}

else

e += inc2;

y += incy;

draw\_pixel(x, y);

}

}

}

void myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

//left to right m<1

bresen\_line(10, 200, 200, 300);

//right to left m<1

bresen\_line(100, 200, 300, 100);

//left to right m>1

bresen\_line(100, 50, 400, 300);

//right to left m>1

bresen\_line(80, 90, 120, 150);

//letter A

bresen\_line(300, 350, 130, 320);

bresen\_line(350, 400, 320, 130);

bresen\_line(325, 375, 225, 225);

glFlush();

}

int main(int argc, char\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(1024, 1024);

glutInitWindowPosition(100, 100);

glutCreateWindow("Bresenham's Line Algorithm");

init();

glutDisplayFunc(myDisplay);

glutMainLoop();

return 0;

}

**Output Screenshot:**

****

**RESULT:**

The plot points that make up the line with endpoints and an alphabet have been drawn using Bresenham’s line drawing algorithm.