#include <iostream>

#include <math.h>

#include <GL/glut.h>

using namespace std;

GLdouble X1, Y1, X2, Y2;

void LineBresenham(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

int dx = abs(X1 - X2), dy = abs(Y1 - Y2);

int p = 2 \* dy - dx;

int twoDy = 2 \* dy, twoDyDx = 2 \* (dy - dx);

int x, y, xEnd;

if (X1 > X2)

{

x = X2;

y = Y2;

xEnd = X1;

}

else

{

x = X1;

y = Y1;

xEnd = X2;

}

glBegin(GL\_POINTS);

glVertex2d(x, y);

while (x < xEnd)

{

x++;

if (p < 0)

p += twoDy;

else

{

y++;

p += twoDyDx;

}

glVertex2d(x, y);

}

glEnd();

glFlush();

}

void Init()

{

glClearColor(1.0,1.0,1.0,0);

glColor3f(0.0,0.0,0.0);

glViewport(0, 0, 640, 480);

glMatrixMode(GL\_PROJECTION);

//glLoadIdentity();

gluOrtho2D(0, 640, 0, 480);

}

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

{

cout << "Enter Two Points for Draw LineBresenham:\n";

cout << "\nEnter Point1( X1 , Y1):";

cin >> X2 >> Y2;

cout << "\nEnter Point2( X2 , Y2):";

cin >> X2 >> Y2;

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(0, 0);

glutCreateWindow("LineBresenham");

Init();

glutDisplayFunc(LineBresenham);

glutMainLoop();

return 0;

}