DDA Line Drawing Algorithm

Code

#include <GL/glut.h>

#include<stdlib.h>

#include<math.h>

#include<iostream>

using namespace std;

void myinit(void)

{

glClearColor(1.0,1.0,1.0,0.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D (0.0,640.0,0.0,480.0);

}

float xstart,ystart,xend,yend,step,xinc,yinc,x,y;

int Round(float a) //any x i.e 1>=x>=0.5 is rounded to 1

{

if(a-int(a)>=0.5)

return int(a)+1;

else

return int(a);

}

void drawline()

{

float ydiff = yend-ystart;

float xdiff = xend-xstart;

if (abs(xdiff) > abs(ydiff))

step = abs(xdiff); //assign abs(xdiff) to step if xdiff>ydiff

else

step = abs(ydiff); //assign abs(ydiff) to step if xdiff<ydiff

xinc = xdiff/step; //assign xdiff/step to xinc

yinc = ydiff/step; //assign ydiff/step to yinc

x = xstart; //assign xstart to x

y = ystart; //assign ystart to y

for(int k=0; k<step; k++)

{

x = x+xinc; // update x by xinc

y = y+yinc;

glColor3f(0,1,0); // sets the current drawing (foreground) color to blue

glPointSize(3); // sets the size of points to be drawn (in pixels)

glBegin(GL\_POINTS); // writes pixels on the frame buffer with the current drawing color

glVertex2i(Round(x),Round(y));//sets vertex

glEnd();

}

glutPostRedisplay();

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT); // clears the frame buffer and set values defined in glClearColor() function call

drawline();

glFlush(); // flushes the frame buffer to the screen

}

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

{

cout << "Enter x1: ";

cin >> xstart;

cout << "Enter y1: ";

cin >> ystart;

cout << "Enter x2: ";

cin >> xend;

cout << "Enter y2: ";

cin >> yend;

glutInit(&argc,argv);

glutInitWindowSize(640,480);//sets the width and height of the window in pixels

glutInitWindowPosition(50,50);//sets the position of the window in pixels from top left corner

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);//creates a single frame buffer of RGB color capacity.

glutCreateWindow("DDA Line Drawing Algorithm");//creates the window as specified by the user as above.

glutDisplayFunc(display);//links the display event with the display event handler(display)

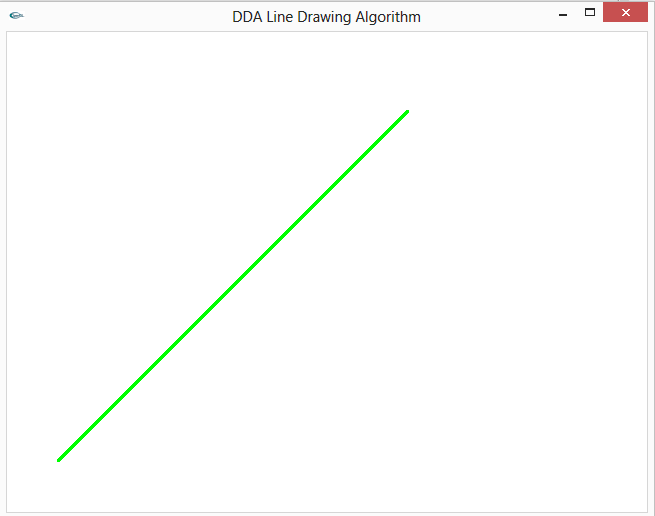
myinit();

glutIdleFunc(drawline);

glutMainLoop();//loops the current event

}

Figure



Taking Input

