**/\**Program to draw simple line, dotted line, dashed line and solid line using DDA line drawing algorithm*\*/**

#include <GL/glut.h>

#include <stdlib.h>

#include<iostream>

using namespace std;

void displayPoint(int x, int y)

{

glColor3f(0, 1, 0);

glPointSize(2);

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

}

float x01, x2, y01, y2;

int ch;

void SimpleLine(float x1, float y1, float x2, float y2)

{

float step;

float dx = x2 - x1;

float dy = y2 - y1;

if (abs(dx) >= abs(dy))

{

step = abs(dx);

}

else

step = abs(dy);

float Xinc = dx / (float)step;

float Yinc = dy / (float)step;

float x = x1;

float y = y1;

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

{

displayPoint(x, y);

x = x + Xinc;

y = y + Yinc;

}

glFlush();

}

void DottedLine(float x1, float y1, float x2, float y2)

{

float step;

float dx = x2 - x1;

float dy = y2 - y1;

if (abs(dx) > abs(dy))

{

step = abs(dx);

}

else

step = abs(dy);

float Xinc = dx / (float)step;

float Yinc = dy / (float)step;

float x = x1;

float y = y1;

displayPoint(x, y);

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

{

if(i%4==0)

{

displayPoint(x, y);

}

x = x + Xinc;

y = y + Yinc;

}

glFlush();

}

void DashedLine(float x1, float y1, float x2, float y2)

{

float step;

int count=0;

float dx = x2 - x1;

float dy = y2 - y1;

if (abs(dx) > abs(dy))

{

step = abs(dx);

}

else

step = abs(dy);

float Xinc = dx / (float)step;

float Yinc = dy / (float)step;

float x = x1;

float y = y1;

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

{

count++;

if (count<7)

{

displayPoint(x, y);

x=x+Xinc;

y=y+Yinc;

}

else if(count<=10 && count>=7)

{

x=x+Xinc;

y=y+Yinc;

}

else{

x=x+Xinc;

y=y+Yinc;

count=0;

}

}

glFlush();

}

void SolidLine(float x1, float y1, float x2, float y2)

{

float step;

float dx = x2 - x1;

float dy = y2 - y1;

if (abs(dx) >= abs(dy))

{

step = abs(dx);

}

else

step = abs(dy);

float Xinc = dx / (float)step;

float Yinc = dy / (float)step;

float x = x1;

float y = y1;

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

{

glColor3f(0, 1, 0);

glPointSize(5);

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

x = x + Xinc;

y = y + Yinc;

}

glFlush();

}

void myMouse(int button, int state, int x, int y)

{

static int xst, yst, pt = 0;

if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

if (pt == 0)

{

xst = x;

yst = y;

x01 = xst;

y01 = yst;

pt = pt + 1;

}

else

{

x2 = x;

y2 = y;

if (ch == 1)

{

SimpleLine(xst, yst, x, y);

}

else if (ch == 2)

{

DottedLine(xst, yst, x, y);

}

else if (ch == 3)

{

DashedLine(xst, yst, x, y);

}

else if(ch==4)

{

SolidLine(xst,yst,x,y);

}

xst = x;

yst = y;

}

}

else if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN)

pt = 0;

//Clear Screen

glFlush();

}

void keyboard(unsigned char key, int x, int y)

{

switch (key)

{

case 's':

ch = 1;

glutMouseFunc(myMouse);

break;

case 'd':

ch = 2;

glutMouseFunc(myMouse);

break;

case 'D':

ch = 3;

glutMouseFunc(myMouse);

break;

case 'S':

ch=4;

glutMouseFunc(myMouse);

break;

}

glutPostRedisplay();

}

void initialize(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

// gluOrtho2D(l,r,b,t)

gluOrtho2D(0, 600, 600, 0);

}

void primitives(void)

{

//glClearColor(1.0, 1.0, 1.0, 1.0);

//glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1, 0, 0);

SimpleLine(0, 300, 600, 300);

SimpleLine(300, 0, 300, 600);

glutKeyboardFunc(keyboard);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowPosition(0, 0);

glutInitWindowSize(600, 600);

glutCreateWindow("OpenGL - DDA Algo");

initialize();

cout<<"--------------------";

cout<<"\ns. Simple Line";

cout<<"\nd. Dotted Line";

cout<<"\nD. Dashed Line";

cout<<"\nS. Solid Line";

cout<<"\n--------------------\n";

glutDisplayFunc(primitives);

glutMainLoop();

return 0;

}