CG programs

Program 1:

#include <iostream>

#include <GL/glut.h>

#include <time.h>

using namespace std;

int x1, x2, yc1, y2;

int flag = 0;

void draw\_pixel(int x, int y)

{

glColor3f(1, 0, 0);

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

glFlush();

}

void draw\_line()

{

int dx, dy, i, e;

int incx, incy, inc1, inc2;

int x, y;

dx = x2 - x1;

dy = y2 - yc1;

if (dx < 0)dx = -dx;

if (dy < 0)dy = -dy;

incx = 1;

if (x2 < x1)

incx = -1;

incy = 1;

if (y2 < yc1)

incy = -1;

x = x1;

y = yc1;

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);

}

}

glFlush();

}

void myinit()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

gluOrtho2D(-250, 250, -250, 250);

}

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

{

switch (button)

{

case GLUT\_LEFT\_BUTTON:

if (state == GLUT\_DOWN)

{

if (flag == 0)

{

printf("Defining x1,y1");

x1 = x - 250;

yc1 = 250 - y;

flag++;

cout << x1 << " " << yc1 << " \n";

}

else

{

printf("Defining x2,y2");

x2 = x - 250;

y2 = 250 - y;

flag = 0;

cout << x2 << " " << y2 << " \n";

draw\_line();

}

}

break;

}

}

void display()

{}

int main(int ac, char\* av[])

{

/\*

//FOR KEYBOARD

cout<<"X1\n";

cin>>x1;

cout<<"Y1\n";

cin>>yc1;

cout<<"X2\n";

cin>>x2;

cout<<"Y2\n";

cin>>y2;

//END KEYBOARD

\*/

glutInit(&ac, av);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 200);

glutCreateWindow("LINE");

myinit();

glutMouseFunc(MyMouse); //INCLUDE TO USE MOUSE, REMOVE WHILE USING KEYBOARD

//draw\_line(); //INCLUDE TO USE KEYBOARD, REMOVE WHILE USING MOUSE

glutDisplayFunc(display);

glutMainLoop();

}

Output:









