#include <GL/freeglut.h>

#include <GL/gl.h>

#include <math.h>

#include<iostream>

void SetPixel(int x, int y){

glColor3f(0,0,0);

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

}

int sign(int k) {

if(k==0)

return 1;

return k / abs(k);

}

void bresenham(int xs, int ys, int xe, int ye) {

int s1, s2, sw;

int x, y, dx, dy;

dx = abs(xe - xs);

dy = abs(ye - ys);

s1 = sign(xe - xs);

s2 = sign(ye - ys);

if (dy > dx) {

sw = dy;

} else {

sw = dx;

}

int n = 1;

int p = 2 \* dy - dx;

x = xs;

y = ys;

SetPixel(x, y);

while(n<sw){

if(dx>dy){

if(p<0){

x = x + s1;

p = p + 2 \* (dy);

}

else{

x = x + s1;

y = y + s2;

p = p + 2 \* (dy - dx);

}

}

else{

if(p<0){

y = y + s2;

p = p + 2 \* (dx);

}

else{

x = x + s1;

y = y + s2;

p = p + 2 \* (dx - dy);

}

}

SetPixel(x,y);

n++;

}

}

void renderFunction(){

glClearColor(1.0, 1.0, 1.0, 0.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0,0,0);

gluOrtho2D(0, 700, 0, 500);

glBegin(GL\_LINES);

std::cout<<"abc";

//front

bresenham(175, 350, 175, 280);

bresenham(215, 280, 215, 350);

bresenham(175, 350, 215, 350);

bresenham(175, 280, 215, 280);

//window

bresenham(185, 295, 205, 295);

bresenham(205, 295, 205, 325);

bresenham(205, 325, 185, 325);

bresenham(185, 325, 185, 295);

//rest of the house

bresenham(215, 280, 300, 305);

bresenham(165, 330, 195, 390);

bresenham(225, 330, 195, 390);

bresenham(225, 330, 310, 355);

bresenham(195, 390, 280, 415);

bresenham(280, 415, 310, 355);

bresenham(300, 305, 300, 353);

bresenham(165, 330, 175, 335);

bresenham(249, 290, 249, 325);

bresenham(266, 295, 266, 330);

bresenham(266, 330, 249, 325);

glEnd();

glFlush();

std::cout<<"abc";

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(700,500);

glutInitWindowPosition(100,100);

glutCreateWindow("OpenGL - house using bresenham");

glutDisplayFunc(renderFunction);

glutMainLoop();

return 0;

}

