直线裁剪算法

时间：

地点：

实验内容：使用opengl，结合DDA算法使用surtherland算法来对直线进行裁剪

实验目的：实现DDA算法生成直线，更进一步的了解DDA算法，将DDA算法生成的直线与裁剪窗口进行运算，保留和取舍，求交。。。最后得到裁剪后的直线

实验代码：

#include <stdlib.h>

#include <math.h>

#include <iostream>

#include<stdio.h>

#include <Windows.h>

#include <GL/GL.h>

#include <GL/glut.h>

//全局变量

int a,b,a1,b1,pp0,pq0,pp1,pq1;

//画线

void setPixel(GLint x,GLint y){

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

}

void init(void){

glClearColor(1.0,1.0,1.0,0.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0.0,200.0,0.0,150.0);

}

void LineDDA(int x0,int y0,int xEnd,int yEnd){

int dx = xEnd - x0;

int dy = yEnd - y0;

int steps,k;

float xIncrement,yIncrement,x = x0,y = y0;

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

steps = abs(dx);

else

steps = abs(dy);

xIncrement = float(dx)/float(steps);

yIncrement = float(dy)/float(steps);

for (k = 0;k<steps;k++)

{

x+=xIncrement;

y+=yIncrement;

setPixel(x,y);

}

}

//裁剪

class wcPt2D{

public:

GLfloat x,y;

};

inline GLint round(constGLfloat a){

return GLint(a+0.5);

}

constGLintwinLeftBitCode = 0x1;

constGLintwinRightBitCode = 0x2;

constGLintwinBottomBitCode = 0x4;

constGLintwinTopBitCode = 0x8;

inlineGLint inside(GLint code){

returnGLint(!code);

}

inlineGLint reject(GLint code1,GLint code2){

returnGLint(code1&code2);

}

inlineGLint accept(GLint code1,GLint code2){

returnGLint(!(code1|code2));

}

GLubyteencode(wcPt2D pt,wcPt2D winMin,wcPt2D winMax){

GLubyte code = 0x00;

if(pt.x<winMin.x)

code = code|winLeftBitCode;

if(pt.x>winMax.x)

code = code|winRightBitCode;

if(pt.y<winMin.y)

code = code|winBottomBitCode;

if(pt.y>winMax.y)

code = code|winTopBitCode;

return(code);

}

void swapPts(wcPt2D \*p1,wcPt2D \*p2){

wcPt2Dtmp;

tmp = \*p1;

\*p1 = \*p2;

\*p2 = tmp;

}

void swapCodes(GLubyte \*c1,GLubyte \*c2){

GLubytetmp;

tmp = \*c1;

\*c1 = \*c2;

\*c2 = tmp;

}

void lineClipCohSuth(wcPt2D winMin,wcPt2D winMax,wcPt2D p1,wcPt2D p2){

GLubyte code1,code2;

GLint done = false,plotLine = false;

GLfloat m;

int x0=0;int y0=0;int x1=0;int y1=0;

while(!done){

code1 = encode(p1,winMin,winMax);

code2 = encode(p2,winMin,winMax);

if(accept(code1,code2)){

done = true;

plotLine = true;

}//简取

else if(reject(code1,code2))//简弃

done = true;

else{

if(inside(code1)){

swapPts(&p1,&p2);

swapCodes(&code1,&code2);

}

if(p2.x!=p1.x)

m=(p2.y-p1.y)/(p2.x-p1.x);//计算k

if(code1 &winLeftBitCode){

p1.y+=(winMin.x-p1.x)\*m;

p1.x = winMax.x;

}else if(code1 &winBottomBitCode){

if(p2.x != p1.x)

p1.x+=(winMin.y - p1.y)/m;

p1.y=winMin.y;

}else if(code1 &winTopBitCode){

if(p2.x != p1.x)

p1.x +=(winMax.y - p1.y)/m;

p1.y = winMax.y;

}

}//else

}//while

if(plotLine){

LineDDA(round(p1.x),round(p1.y),round(p2.x),round(p2.y));

}

}

void cutwindow(intxmin,intymin,intxmax,intymax){

LineDDA(xmin,ymin,xmin,ymax);

LineDDA(xmin,ymin,xmax,ymin);

LineDDA(xmin,ymax,xmax,ymax);

LineDDA(xmax,ymin,xmax,ymax);

}

void display(){//DDA演示

printf("DDA演示\n");

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0,0.0,0.0);

glBegin(GL\_LINES);

wcPt2D min;

min.x = a;

min.y = b;

wcPt2D max;

max.x=a1;

max.y=b1;

wcPt2D p1;

p1.x=pp0;

p1.y=pq0;

wcPt2D p2;

p2.x=pp1;

p2.y=pq1;

cutwindow(min.x,min.y,max.x,max.y);//绘制窗口

lineClipCohSuth(min,max,p1,p2);

glEnd();

glFlush();

}

void main(intargc,char \*\* argv){

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition(500,300);

glutInitWindowSize(400,300);

glutCreateWindow("直线裁剪");

init();

printf("请输入裁剪窗口左下角坐标\n");

scanf("%d %d",&a,&b);

printf("请输入裁剪窗口右上角坐标\n");

scanf("%d %d",&a1,&b1);

printf("请输入要裁剪的直线第一点坐标\n");

scanf("%d %d",&pp0,&pq0);

printf("请输入要裁剪的直线第二点坐标\n");

scanf("%d %d",&pp1,&pq1);

glutDisplayFunc(display);

glutMainLoop();

}

实验结果



