**DDA算法绘制直线**

日期 2020/5/11

**一、实验目的**

1、掌握DDA算法；

2、掌握直线的绘制过程；

3、熟练OpenGL编写直线代码。

**二、实验要求**

1、利用DDA算法完成直线的绘制；

2、绘制两条垂直直线。

**二、实验内容（代码及注释）**

#include "stdafx.h"

#include <gl/glut.h>

#include <math.h>

#define WIDTH 500 //窗口宽度

#define HEIGHT 500 //窗口高度

#define DRAWLINE1 DDALine(100,300,300,300); //画直线

#define DRAWLINE2 DDALine(100,200,400,100); //画直线

void Init() //初始化

{

glClearColor(1.0f,1.0f,1.0f,1.0f); //设置背景颜色，完全不透明

glColor3f(0.0f,0.0f,0.0f); //设置画笔颜色

glMatrixMode(GL\_PROJECTION); //设置投影

gluOrtho2D(0.0, WIDTH, 0.0, HEIGHT); //设置投影区域

}

void DDALine(int x0,int y0,int x1,int y1) //DDA算法画线

{

int dx,dy,eps1,i;

float x,y,xIncre,yIncre;

dx = x1-x0;

dy = y1-y0;

x = x0,y = y0;

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

eps1 = abs(dx);

else

eps1 = abs(dy);

xIncre = (float)dx / float(eps1);

yIncre = (float)dy / float(eps1);

for(i=0;i<=eps1;i++){

glBegin(GL\_POINTS);

glVertex2i(int(x+0.5),int(y+0.5));

glEnd();

x+=xIncre;

y+=yIncre;

}

}

void Display() //显示函数

{

glClear(GL\_COLOR\_BUFFER\_BIT); //清空颜色堆栈

DRAWLINE1

DRAWLINE2

glFlush(); //清空缓冲区指令

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); //初始化显示模式

glutInitWindowSize(WIDTH,HEIGHT); //设置窗口尺寸

glutInitWindowPosition(100,100); //设置窗口位置

glutCreateWindow("用DDA算法画直线"); //创建窗口

glutDisplayFunc(Display); //注册显示函数

Init(); //初始化

glutMainLoop(); //进入程序循环

return 0;

}

1. **实验总结**

DDA算法的主要核心思想是在直线段的扫描转换算法的基础上，去掉效率比较低下的乘法操作，引入相关的增量思想。

而增量思想利用斜截式方程进行一个增量的推导：yi=kxi+b,yi+1=kxi+1+b=k(xi+1)+b=kxi+k+b=yi+k,yi+1=yi+k;

这个式子的含义是：当前步的y值等于前一步的y值加上斜率k，这样就把乘法变成了一个加法，提高了在计算机中的运行效率。

**四、实验结果**

