实验二 DDA直线生成算法

时间：2022年3月16日

地点：信息学院2202机房

1、实验内容

熟悉OPENGL，通过DDA算法生成直线段

2、实验目的

安装OPENGL，编写代码运行DDA算法

3、实验代码

#include <windows.h>

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

void init(void)

{

glClearColor (0.5, 1.0, 1.0, 0.0); // 指定清空颜色（背景色）为蓝色

gluOrtho2D (0.0, 300.0, 0.0, 300.0); //指定二维坐标系中被显示的区域

}

/\*

inline int round (const float a) {

return int (a + 0.5);

}

\*/

void setPixel(int x,int y){

glPointSize(1.0f);

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

glFlush();

}

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

int dx=xEnd-x0, dy=yEnd-y0, steps, k;

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

if(fabs(dx)>fabs(dy))

steps=fabs(dx);

else

steps=fabs(dy);

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

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

setPixel(round(x),round(y));

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

x+=xIncrement;

y+=yIncrement;

setPixel(round(x),round(y));

}

}

void myDisplay(void){

glClear(GL\_COLOR\_BUFFER\_BIT);// 清空显示窗口

glColor3f(0.0,0.0,0.0);// 指定前景色（当前绘制颜色）为黑色

lineDDA(0,0,150,100);

glFlush();// 使绘制立即反映到屏幕上

}

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

{

glutInit (&argc, argv); // 初始 GLUT.

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB); //设定显示模式

glutInitWindowPosition (500, 200); // 设定窗口位置

glutInitWindowSize (300, 300); // 设定窗口大小

glutCreateWindow ("An Example OpenGL Program");

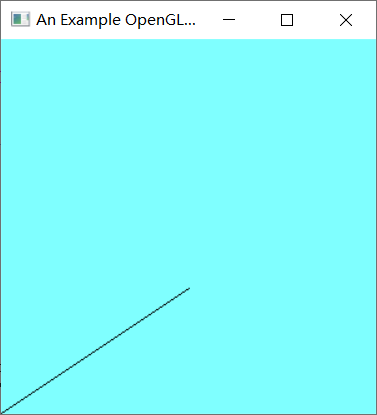
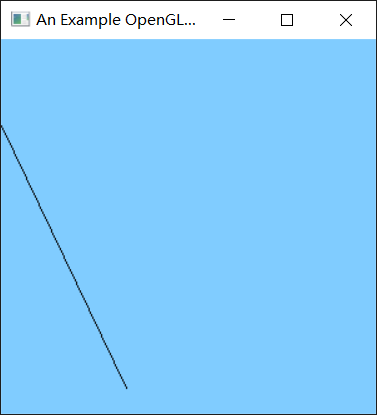
// 用前面指定参数创建窗口，并定义窗口名称

init( ); // 进行一些初始化工作

glutDisplayFunc(&myDisplay); // 指定绘制的回调函数

glutMainLoop ( ); // 进入无穷循环，等待事件处理

}

4、实验结果

5、实验总结

直线DDA算法(数值微分法)是一种增量算法，直观、易实现，但由于里面包含了大量的浮点数运算，不利于用硬件实现,由此引出下面的中点Bresenham算法，并通过一步步的优化减少浮点数的计算量。