实验三 中点Bresenham算法生成直线段

时间：2022年3月30日

地点：信息学院2202机房

1、实验内容

熟悉OPENGL，通过中点Bresenham算法生成直线段

2、实验目的

安装OPENGL，能编写代码运行

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）为白色

gluOrtho2D (-150.0, 150.0, -150.0, 150.0); //指定二维坐标系中被显示的区域

}

void setPixel(int x,int y){

glPointSize(1.0f);

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

glFlush();

}

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

int i,x,y,dx,dy;

if(xEnd>=x0){

dx=xEnd-x0;

dy=yEnd-y0;

x=x0;y=y0;

}

else{

dx=x0-xEnd;

dy=y0-yEnd;

x=xEnd;y=yEnd;

}

float k=float(dy)/float(dx);

if(k>=0&&k<1){

float d=0.5-k;

for(i=0;i<fabs(dx);i++){

if(d<0){

x=x+1;

y=y+1;

d=d+1-k;

setPixel(x,y);

}

else{

x=x+1;

d=d-k;

setPixel(x,y);

}

}

}

if(k>=1){

float d=1-0.5\*k;

for(i=0;i<fabs(dy);i++){

if(d<0){

y=y+1;

d=d+1;

setPixel(x,y);

}

else{

x=x+1;

y=y+1;

d=d+1-k;

setPixel(x,y);

}

}

}

if(k>-1&&k<0){

float d=-0.5-k;

for(i=0;i<fabs(dx);i++){

if(d<0){

x=x+1;

d=d-k;

setPixel(x,y);

}

else{

x=x+1;

y=y-1;

d=d-1-k;

setPixel(x,y);

}

}

}

if(k<=-1){

float d=-1-0.5\*k;

for(i=0;i<fabs(dy);i++){

if(d<0){

x=x+1;

y=y-1;

d=d-1-k;

setPixel(x,y);

}

else{

y=y-1;

d=d-1;

setPixel(x,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

myBreseham(30,100,0,0);

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");

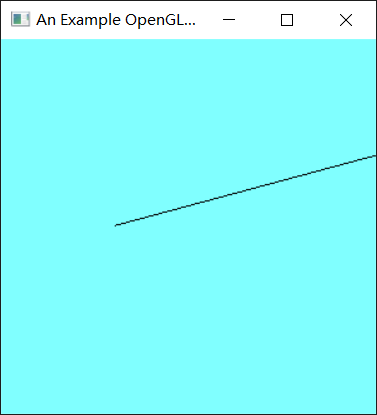
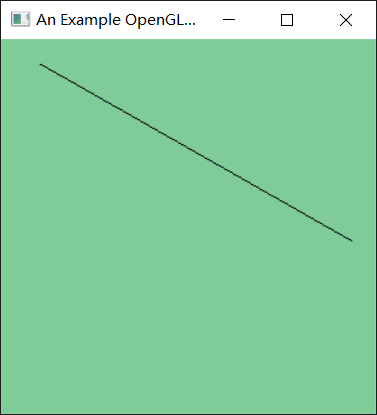
// 用前面指定参数创建窗口，并定义窗口名称

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

1.Bresenham算法主要利用了直线将平面分为三个区域，一个坐标轴移动单位长，另一个移动斜率的长度或者斜率的倒数的大小长度。所以Bresenham算法按照斜率为四个部分，每个部分的递推方程与初始值均不相同。

2.改进的Bresenham算法主要改进的地方是减少了浮点数的运算，通过一步一步地减少浮点运算来达到越来越优。

2.glutDisplayFunc()的使用方式是glutDisplayFunc(&display)， void display(void)函数为绘图函数。