**中点 Bresenham算法绘制圆**

日期 2020.4.6

**一、实验目的**

1.掌握中点Bresenham算法。  
2.掌握圆的绘制过段。  
3 熟练OpenGL编写中点Bresenham算法代码。  
**二、实验要求**1.利用中点Brsenham算法完成圆的绘制。  
2.注释绘制圆的位置及半径。 **三、实验内容(代码及注释)**

#include<windows.h>

#include<GL/glut.h>

void init(int argc,char\*\* argv)

{

glutInit(&argc,argv); //glut初始化API

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); //设置初始显示模式

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

glutInitWindowSize(400,400); //窗口大小

glutCreateWindow("中点Bresenham"); //产生一个顶层的窗口中点Bresenham 作为窗口的名字,也就是窗口标题栏显示的内容.

glClearColor(1.0,1.0,1.0,0); //为颜色缓存区指定确定的值

glMatrixMode(GL\_PROJECTION); //将当前矩阵指定为投影矩阵

gluOrtho2D(0,400,0,400); //指定屏幕区域对应的坐标范围

}

void Bresenham\_Circle(int xc,int yc,int r) //将d的计算放大两倍，同时将初始值改成3 – 2R，这样避免了浮点运算，

{

int x, y, d;

x = 0;

y = r;

d = 3 - 2 \* r;

glVertex2i(x+xc,y+yc); //指定下一个要画的点的标

while(x < y)

{

if(d < 0)

{

d = d + 4 \* x + 6;

}

else

{

d = d + 4 \* ( x - y ) + 10;

y--;

}

x++;

glVertex2i(x+xc,y+yc); //画出其余象限的点

glVertex2i(y+xc,x+yc);

glVertex2i(y+xc,-x+yc);

glVertex2i(x+xc,-y+yc);

glVertex2i(-x+xc,-y+yc);

glVertex2i(-y+xc,-x+yc);

glVertex2i(-x+xc,y+yc);

glVertex2i(-y+xc,x+yc);

}

}

void myDisplay(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT); //清除缓存，将缓存清除为预先的设置值

glColor3f(0.0,0.0,0.0); //绘制的圆为黑色

glPointSize(2); //设置点直径大小

glBegin(GL\_POINTS); //绘制n个点

Bresenham\_Circle(200,200,50); //调用改进的中点画圆算法 Bresenham

glEnd(); //防止模型位置转换begin到end区块结束

glFlush(); //清空缓冲区，将指令送往缓硬件立即执行

}

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

{

init(argc,argv); //初始化窗口

glutDisplayFunc(myDisplay); //绘制窗口

glutMainLoop();

return 0;

}

**四、实验总结**

1、通过此次实验使我更进一步了解中点画圆算法及其改进方法Bresenham。

2、通过opengl程序接口调用gl核心库画点画圆的过程中，熟练掌握调用底层库的流程。例如初始化，绘制，清除等一系列操作。

**五、实验结果**

