

OpenGL 画线

原理

在显示屏上所有的图形都是一个个的像素点组成。我们看到说有直线，曲线都是像素点拟合出来的产物。

这里使用 Bresenham 算法，来实现一个基本的画线函数

Bresenham 的基本原理是: 求出 x 坐标增加 1 时，y 坐标的增加值，每当 x 加 1 时，y 坐标增加一个斜率 k，再进行四舍五入。

OpenGL 函数解释

```
int main(int argc, char** argv) {
    glutInit(&argc, argv);

    glutInitDisplayMode( GLUT_SINGLE | GLUT_RGB );
    glutInitWindowPosition( 50, 100 );
    glutInitWindowSize( WIDTH, HEIGHT );
    glutCreateWindow( "Bresenhan Line" );

    glutDisplayFunc( drawFunc );

    glutMainLoop( );

    return 0;
}
```

main 函数

main 函数模板就是这样。

可以自定义 glutInitWindowPosition，glutInitWindowSize，glutCreateWindow 函数

glutInitWindowPosition 函数

窗口所处位置。 原点是屏幕右上角。参数是整型。

glutInitWindowSize 函数

窗口的初始化大小。

`glutCreateWindow` 函数

该函数负责创建窗口。参数为字符串。

`glutDisplayFunc` 函数

所有画图相关的函数在，`glutDisplayFunc` 里的 `drawFunc` 里面实现。

作画函数

```
void drawFunc() {  
    glClear( GL_COLOR_BUFFER_BIT );  
    glColor3f( 1.0, 1.0, 1.0 );  
    glBegin( GL_LINES );  
        glVertex2f( 0, -1.0 );  
        glVertex2f( 0, 1.0 );  
    glEnd();  
    glFlush();  
}
```

`glClear()`

清空缓冲区

`glColor3f()`

设置绘画出的线条的颜色，RGB 形式

`glBegin()` & `glEnd()`

两个函数必须成对出现。

`glBegin` 的参数代表其两个函数之间所画图形的种类。

其可能是参数有 `GL_LINES`，`GL_POINTS`，`GL_TRIANGLES` 。。。

`glVertex2f()`

是 `glVertex*()` 系列函数中的一个。其参数的值为 $-1 \sim 1$

缺点

原算法假定斜率在 $0 \sim 1$ 之间。