# 实验目的

学习在OpenGL中使用键盘、鼠标、菜单等响应，进行基本的程序交互；学习使用OpenGL的选择拾取，进一步了解OpenGL的交互功能。

# 实验内容

1、教材P74：习题1。

创建一个简单的绘制程序，使得可用 来创建一些简单的形状，如线段、矩形和三角形，并可通过 菜单 或/和 键盘按键 来选择各种形状模式，并允许用户改变绘制颜色。

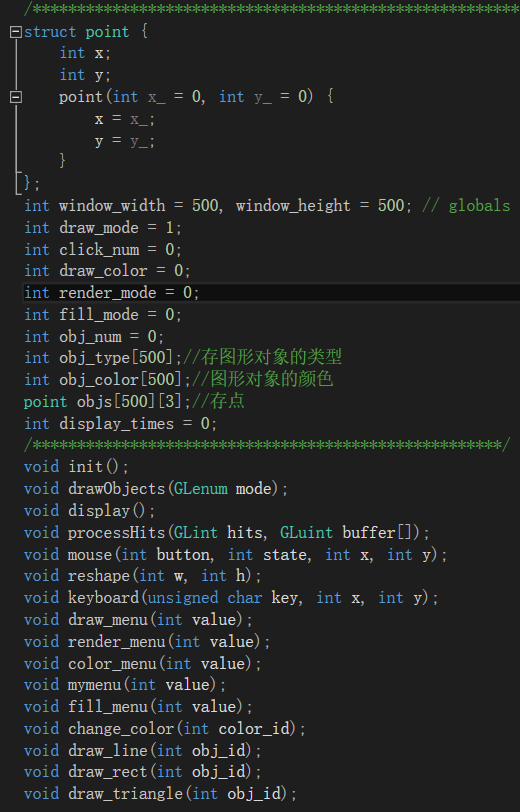
2、在习题1的基础上，添加拾取功能，具体功能要求如下：

① 被拾取到的几何对象要高亮显示或者以其它可以区别的方式进行显示；

② 在控制台窗口中显示当前拾取到的几何对象的相关信息，格式为（类型，编号），比如（Line 1），（Rect 3），（Triangle 3）... 。

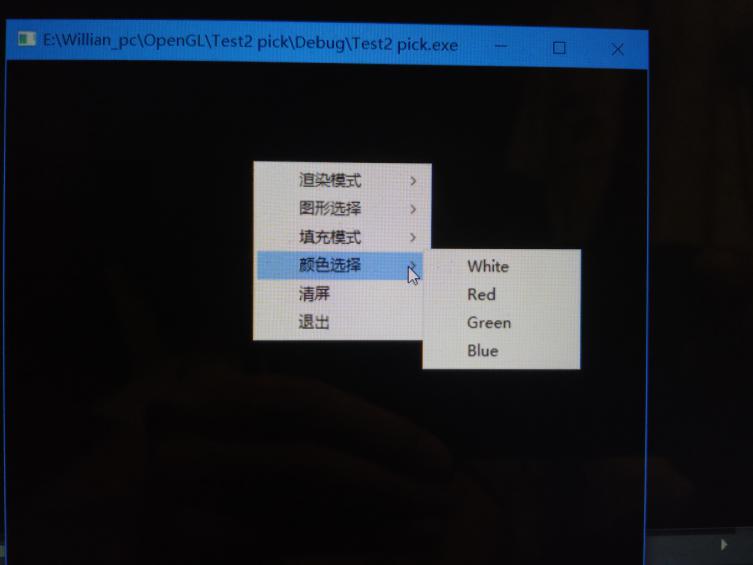
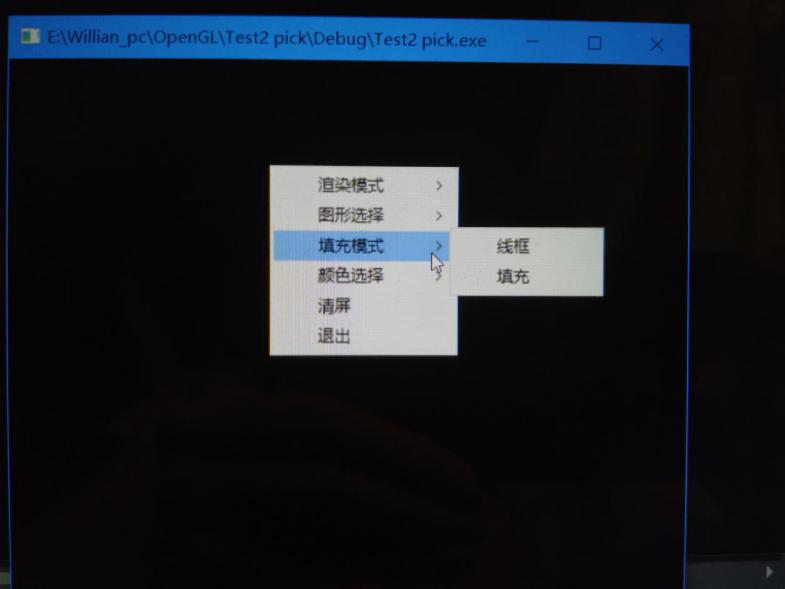
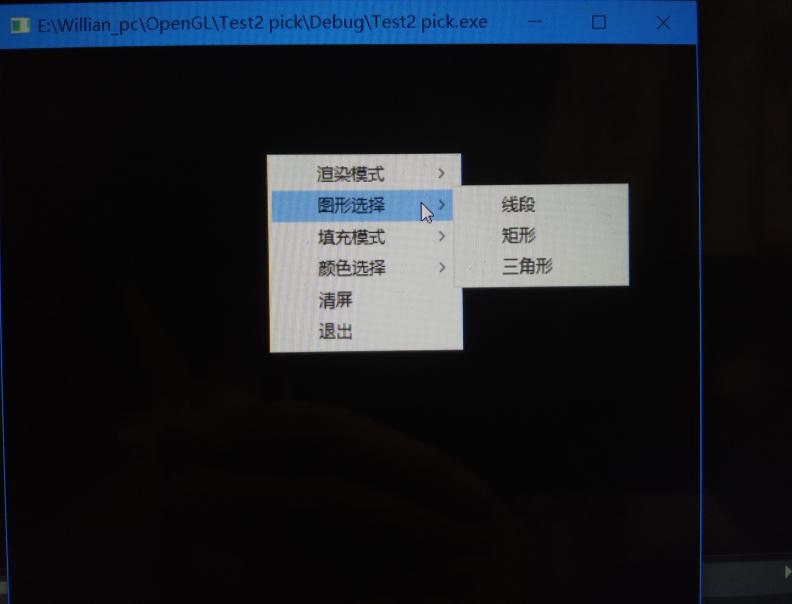
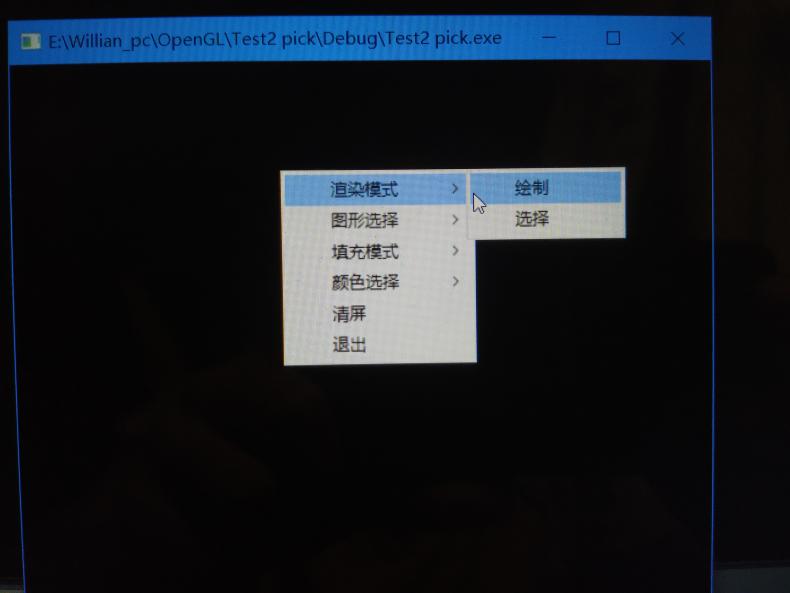
# 实验步骤

**.h文件中的全局变量和函数申明**

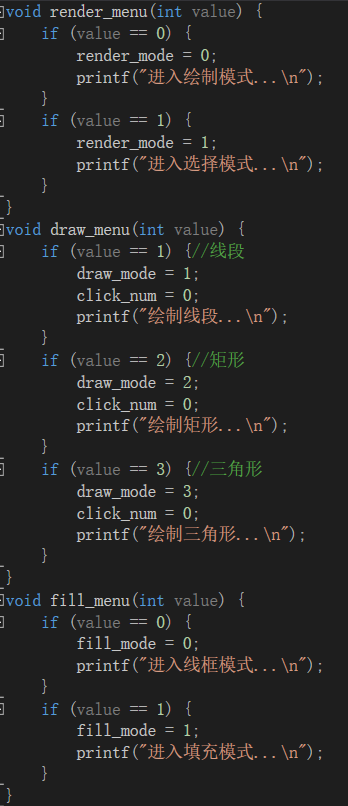
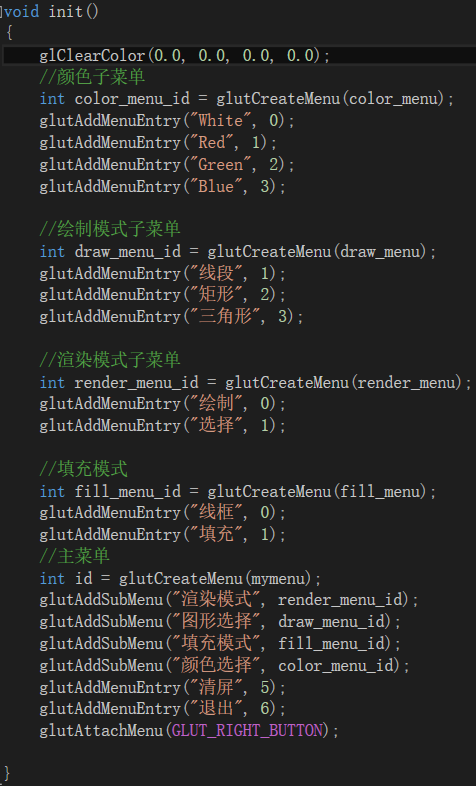


## 菜单的使用

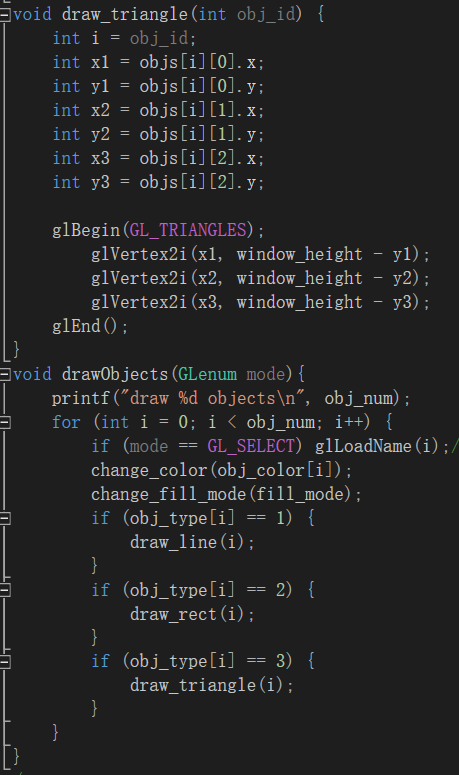
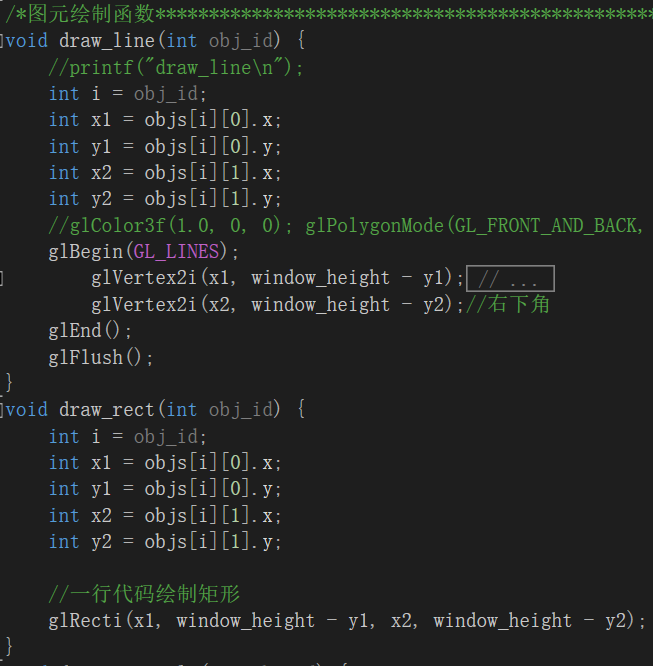
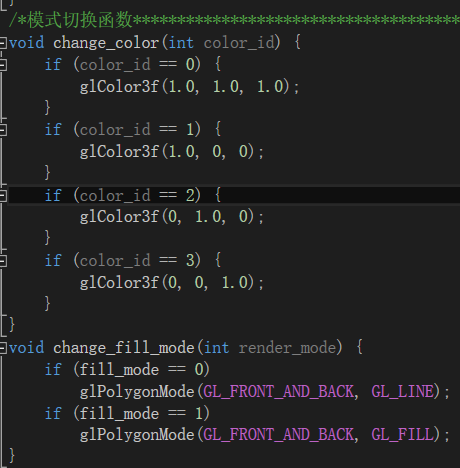
这个比较简单，在init函数中直接添加就行了。除了最近本的清屏和退出，还实现了渲染模式、图形选择、填充模式、颜色选择等子菜单。其中各类图元的绘制都没有问题。在图元绘制



### 在Init函数中添加菜单，对应的菜单函数

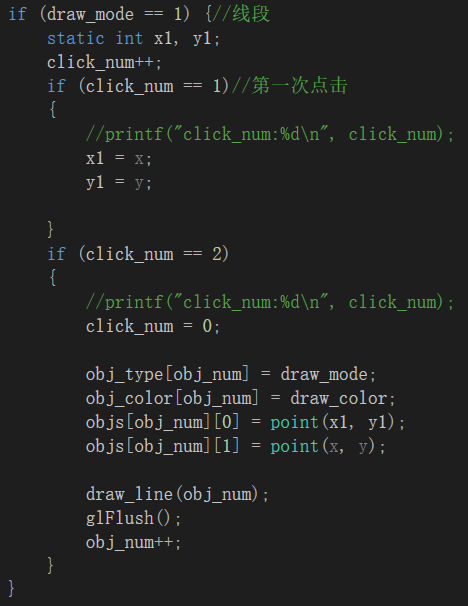
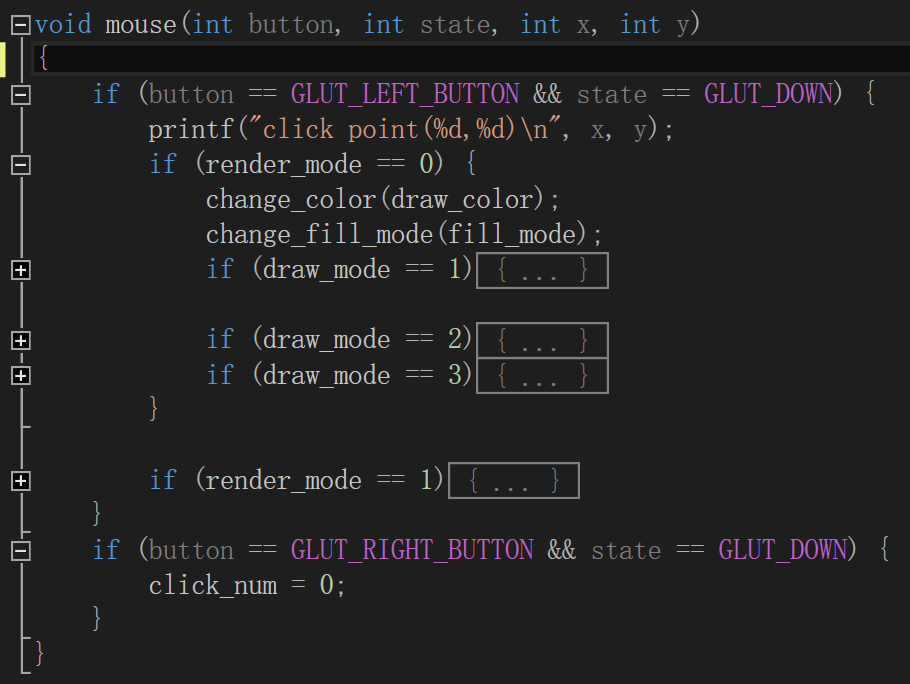


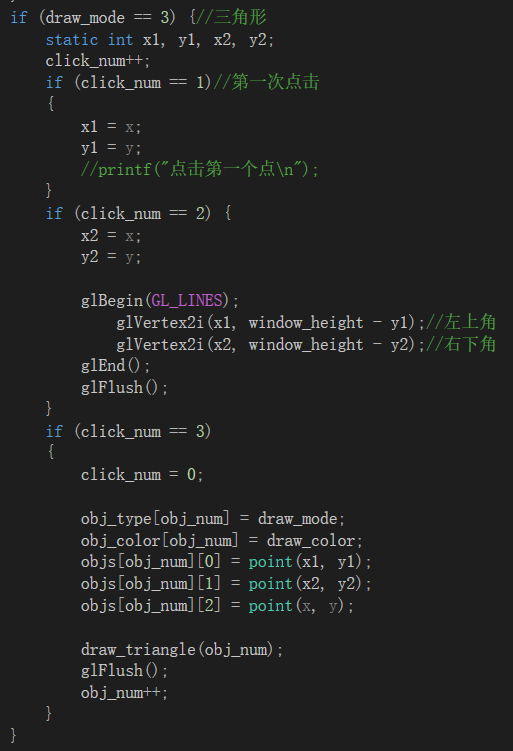
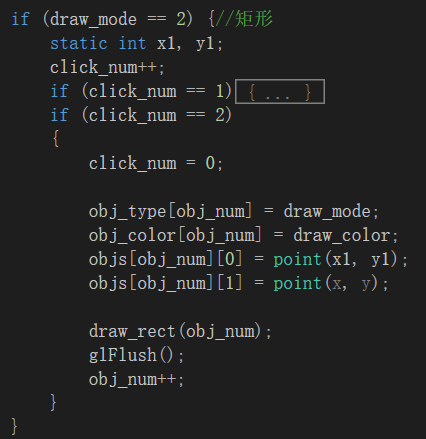
### 模式切换函数，图元绘制函数



### mouse函数框架

首先进来后判断点击的键，如果是右键就将点击次数清零（进入了菜单）；如果是左键，那么继续判断渲染模式，如果是0就是绘制模式，如果是1就是拾取模式。其中绘制模式又分为三图元的绘制。





## 添加拾取功能

这个有点麻烦，机制大概搞懂了。我的代码基本是参照书上标程的，但是要从绘制模式和选择模式之间切换，就总是出现不了预期效果，**遇到许多问题**，最后在老师和蔡婷艳的帮助下发现并解决了一些问题，后面自己通过调试与尝试解决了拾取功能hit总是为0的问题。

**遇到的问题总结如下：**

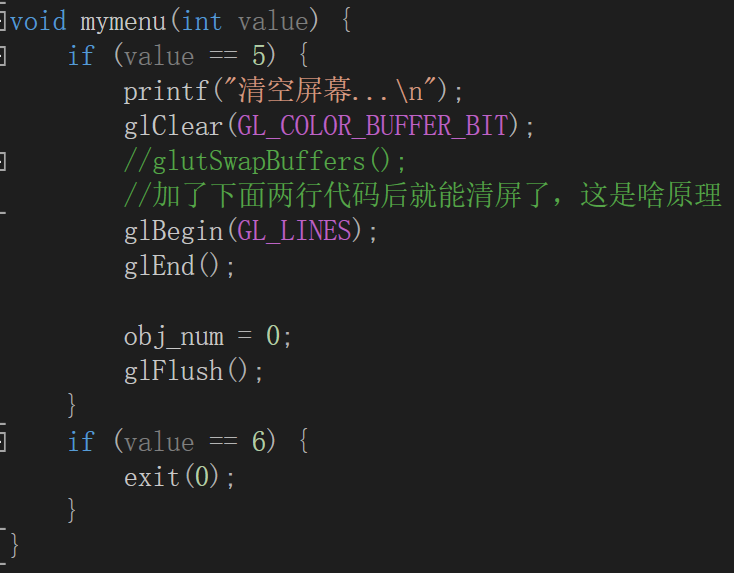
1. OpenGL中的函数是否都是全局的。

比如我绘制一个线段，首先设置颜色，设置填充模式，然后glBegin,glEnd,最后glFlush。这基本是否能够分开（如果这些函数的作用都是全局的那么就可以分开）。但是当我把设置颜色放在一个单独的函数中时，绘制就会出问题（用printf打印显示进入了函数，但是就是不绘制图元），其他的也是。

**答：可以理解为全局的，可以把绘制分在不同的函数中实现。**

1. 点击清屏菜单后实际没有清屏

这我是在其函数中写的很清楚的，但是就是不清屏。后面在老师的帮组下加入了glBegin,glEnd两行代码，成功清屏，这大概就是opengl内部的一个机制了。



1. Hit一直显示为0

我是通过一个全局变量render\_mode来确定进入mouse后是绘制模式还是选择模式。

进入选择模式后，选中图元，但是hit的返回值一直是0，这点很迷。

后面我经过多次尝试，百度查找了一些博客，对各函数进行调用提示输出，交换各函数顺序，终于把这个问题解决了。

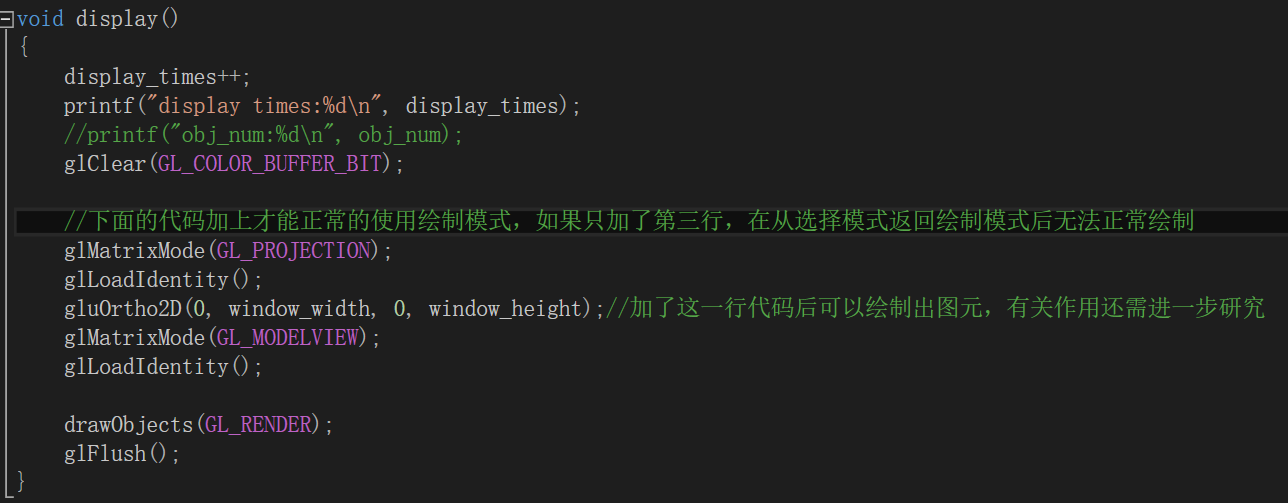
下面是Mouse函数中选择模式对应的代码

其中最主要的一行就是glLoadIdentity(),这是我观察reshape函数找到的规律。解释在注释中。

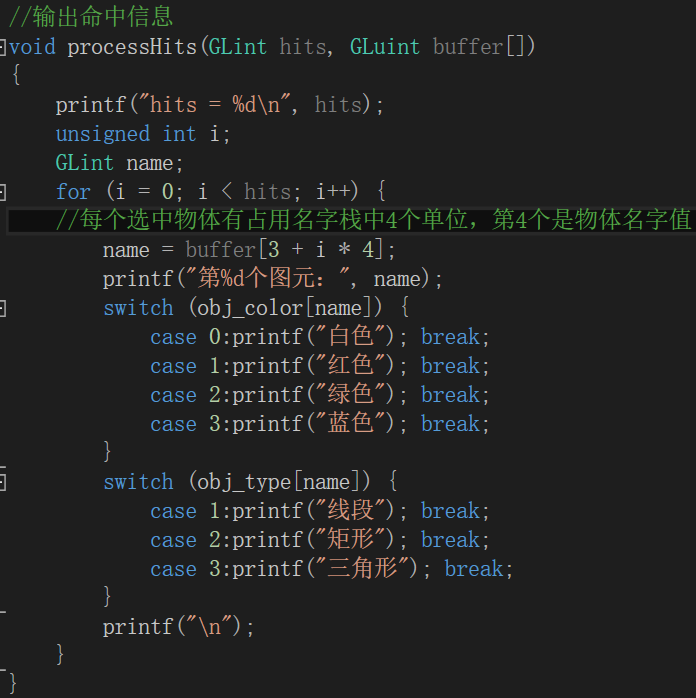


1. 从选择模式返回绘制模式时，图元不显示

最后在display函数中解决了，也是从reshape中找的规律。



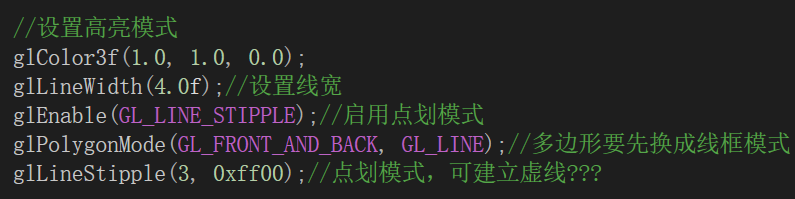
**hits的问题解决后，processHits函数的实现就简单多了，直接上代码吧。**



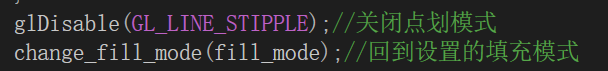
## 新增优化

1. 实现拾取选中图元突出显示的功能（新增）

processHits函数中添加了如下代码

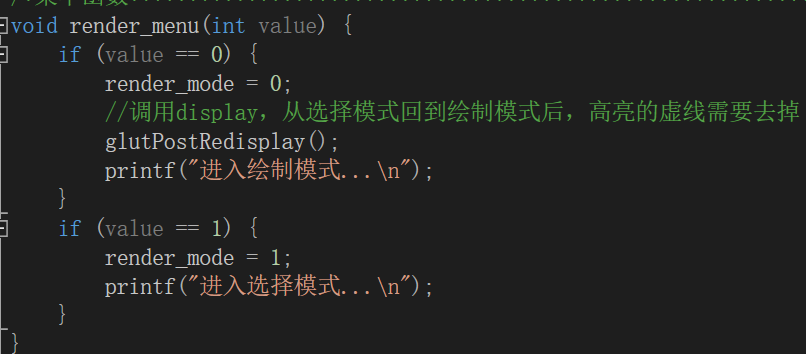


中间是高亮图元的绘制（黄色虚线，如果是矩形或三角形，会多绘制一层黄色线框）



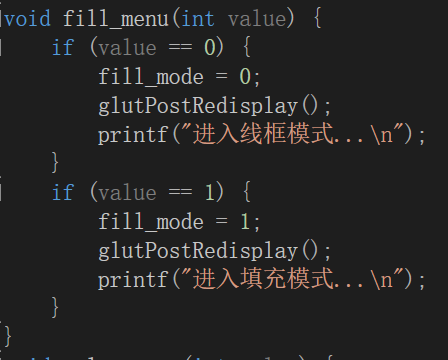
1. 从选择模式回到绘制模式时去掉高亮虚线

在菜单中切换时回调dispaly函数，由于黄色虚线没有保存在数组中，所以不会重新绘制出来。

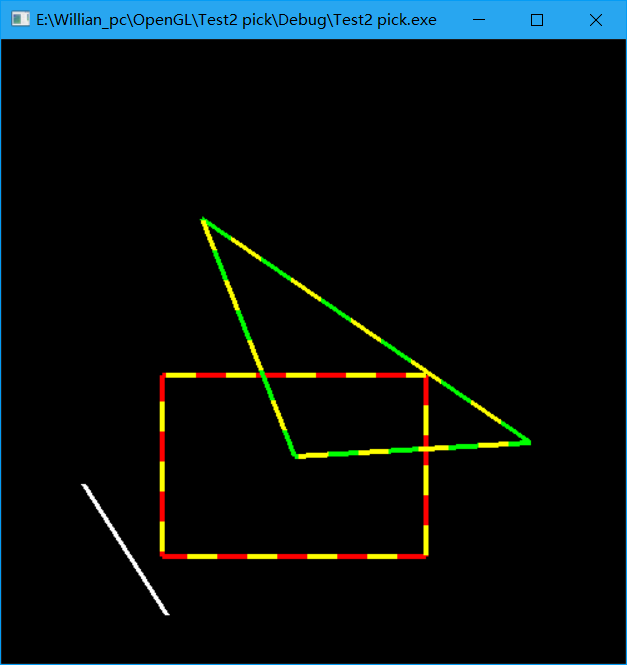
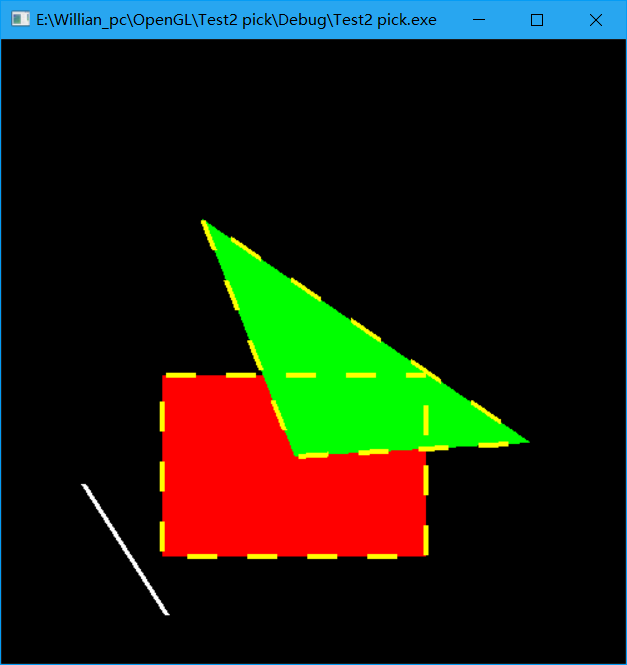


1. 切换填充模式时更新已绘制的图元

同样也是在菜单切换时回调display函数。



1. 效果图



# 实验心得

一开始怎么都解决不了问题的原因：

对opengl内部的机制了解太少

没有和其他人交流，有些调试上的问题完全就是经验，没有必要死磕

然后就是知识上的收获在遇到的问题中已经讲的很清楚了了，这里就不再赘述了。