# 实验目的

理解并掌握用户事件的使用。

# 实验内容

根据用户事件的学习内容和课程中提供的resources,完成相关的实验小项目，并结合项目截图完成实验报告。要求实验与课程提供的ppt中的不同。

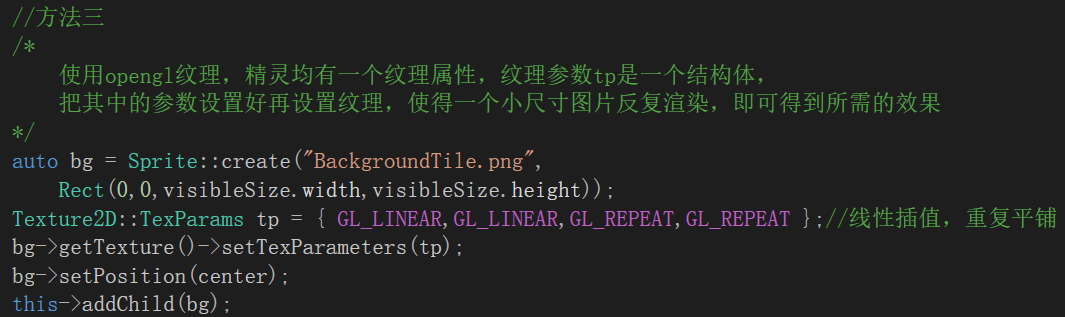
# 实验步骤

本次实验我使用了两个场景，HelloWorld场景用于普通事件，Second场景中使用层事件。

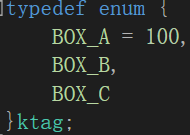
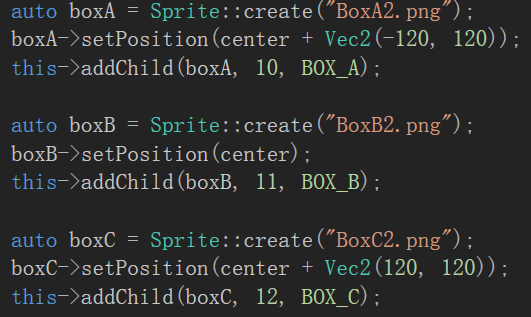
## 首先创建场景，场景中有3个方块。

创建背景

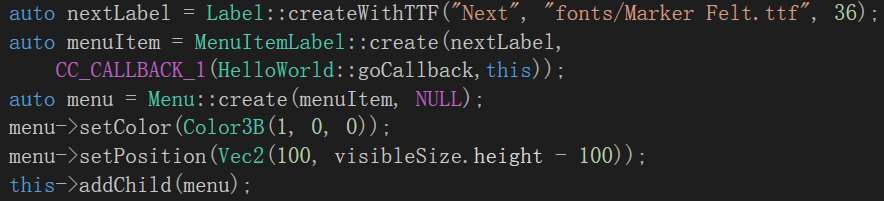
使用opengl纹理，精灵均有一个纹理属性，纹理参数tp是一个结构体，把其中的参数设置好再设置纹理，使得一个小尺寸图片反复渲染，即可得到所需的效果。



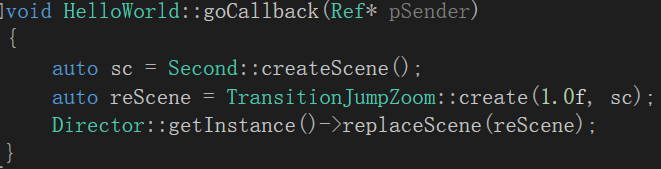
创建三个方块，并设置标签



创建标签菜单，用于跳转到下一个场景



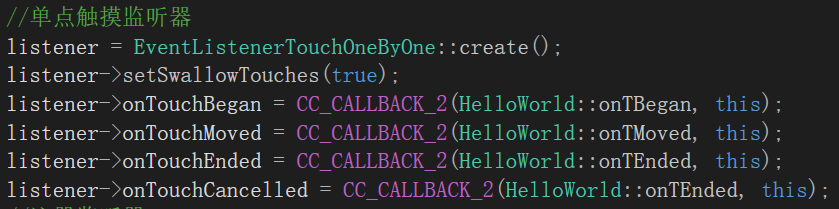
对应的回调函数

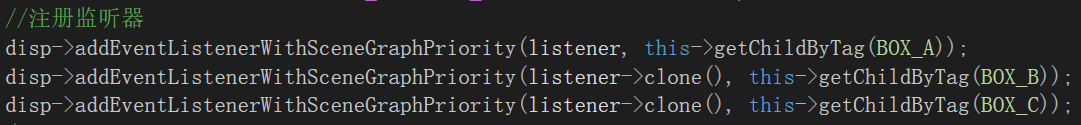


## 单点触摸事件

OnEnter函数中：

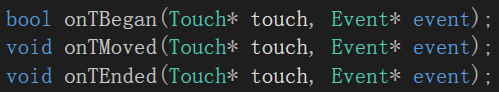




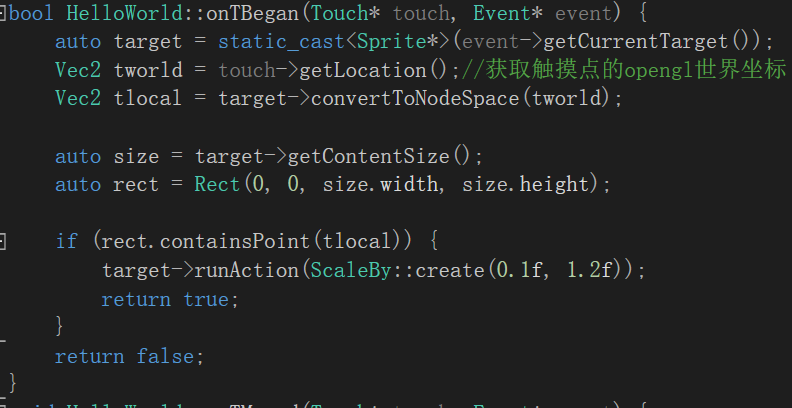


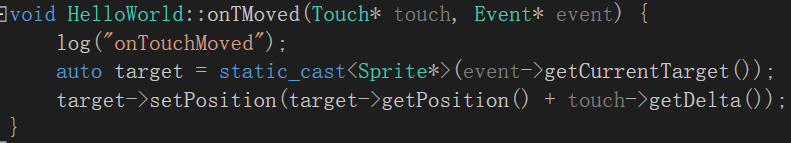
回调函数

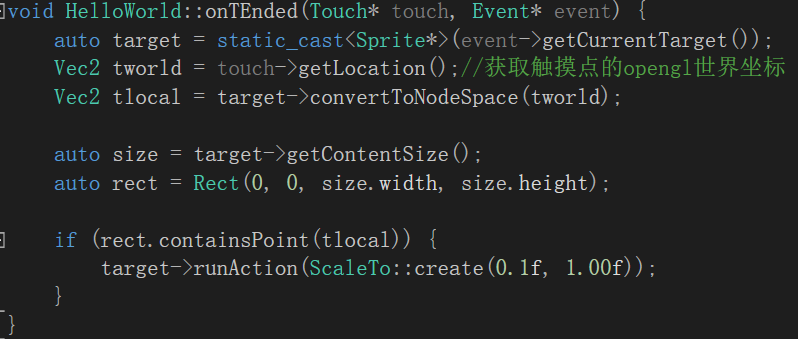
.h中声明



定义







## 使用lambda表达式

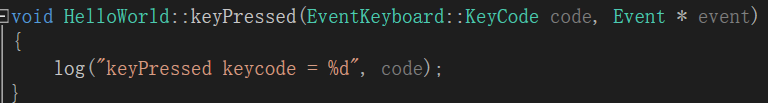
参见键盘事件和鼠标事件。

## 键盘事件

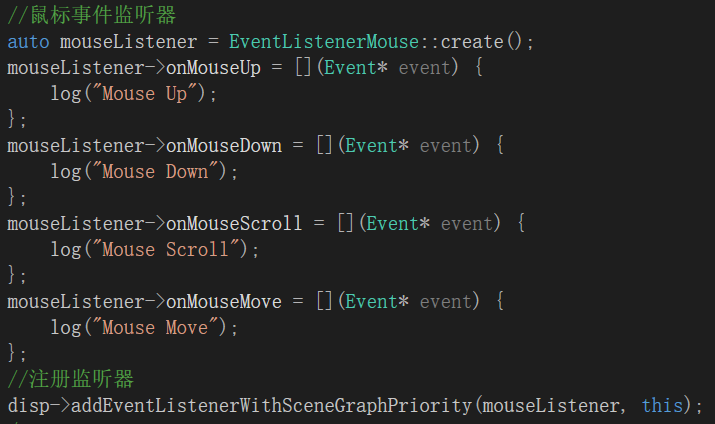
onEnter函数中：



回调函数：

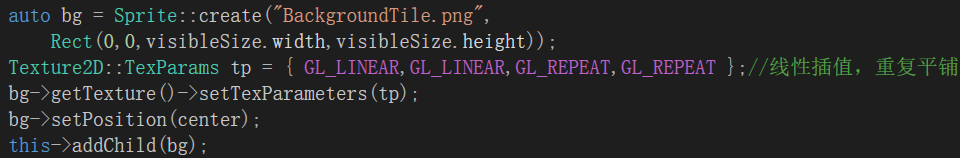


## 鼠标事件

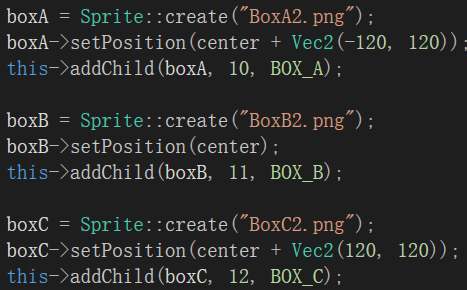
onEnter函数中：  


## 在层中进行事件处理(SecondScene类中)

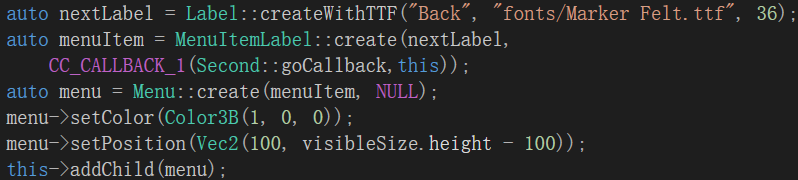
背景的创建：



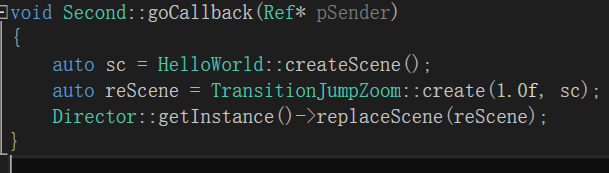
方块的创建，方块被定义为成员变量。



返回菜单

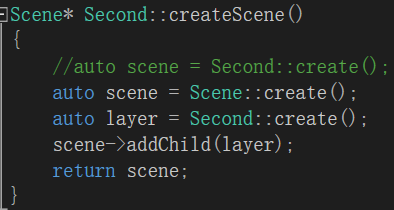


对应回调函数：

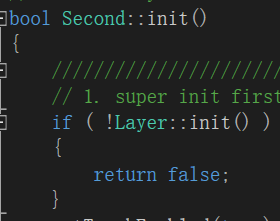


定义上继承Layer。



createScene函数：  


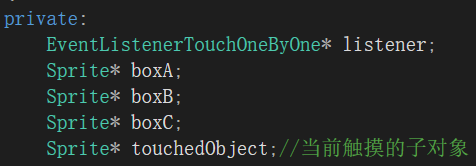
Init函数中调用Layer的init函数。



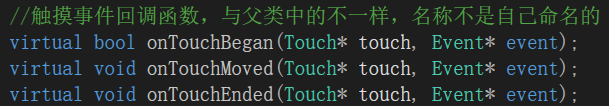
.h中

属性：

方块都定义为私有属性，此外还定义了一个touchedObject，用来存层中当前触摸的子对象，任何时候都只有一个。

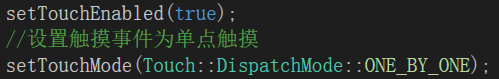


函数的声明：

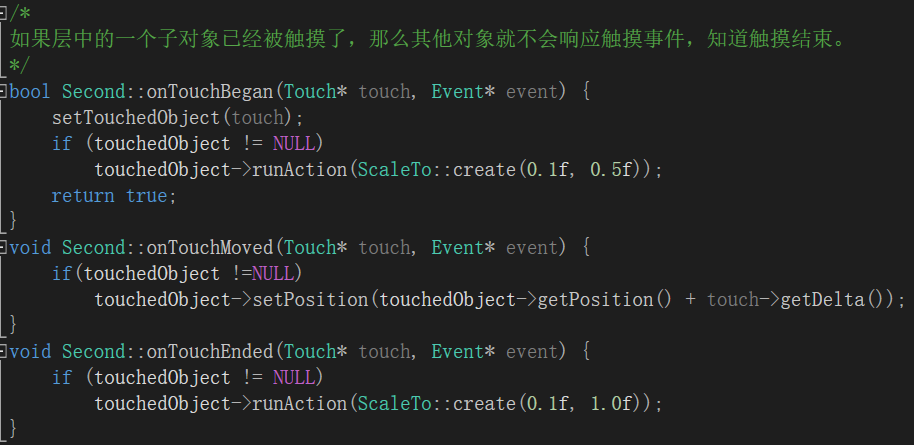




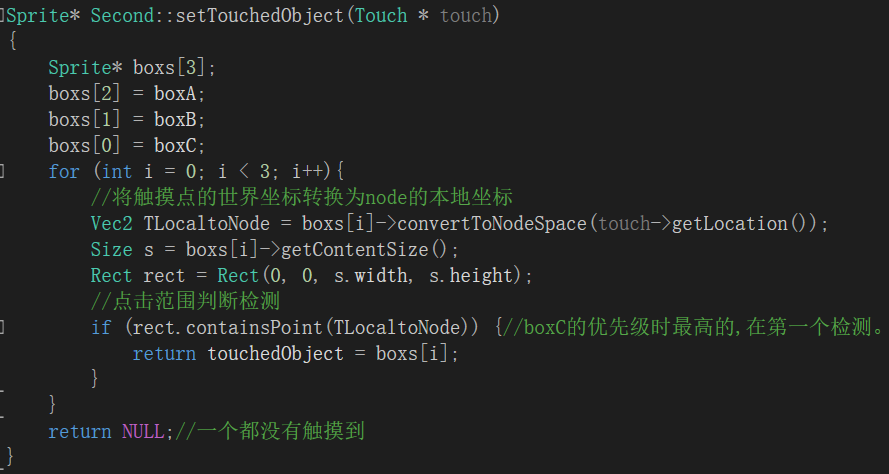
Init函数中启用层触摸事件



触摸事件回调函数：（解释在下面）



setTouchedObject函数：（解释在下面）



**层中触摸事件处理多个子对象的方法**

先看setTouchedObject函数。在该函数中，遍历每一个方块子对象，优先级决定遍历顺序，优先级高的最先被遍历。如果检测出触摸点在该对象的范围内（原ppt中isTap函数中的内容），就将touchedObject设置为该对象，然后返回该对象。如果触摸点没有在任何子对象中，那么就返回空指针。

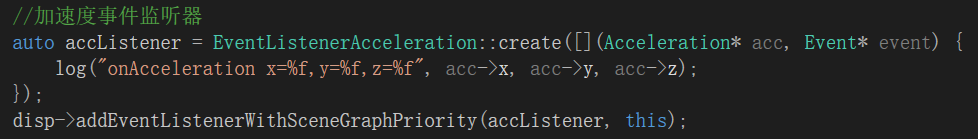
在看几个触摸事件回调函数。onTouchBegan中，先调用setTouchedObject函数，设置被触摸的对象，然后包括后买你的onTouchMoved，touchedObject函数，都只对touchedObject引用的子对象进行操作。这样无论鼠标移动到哪，只要触摸还没有结束，回调函数中的操作都只对touchedObject引用的子对象有效，其他子对象不会受到影响。

由于时通过touchedObject来引用的子对象，所以**不会出现对象缩小后触摸不到的问题**。

touchedObject任何时候都只引用一个对象，**也不会出现同时触摸两个对象的问题**，一开始就点到了两对象按照优先级决定触摸到的时哪个。

## 加速度事件

HelloWorld场景的onEnter函数中



Second场景中，使用层的事件处理

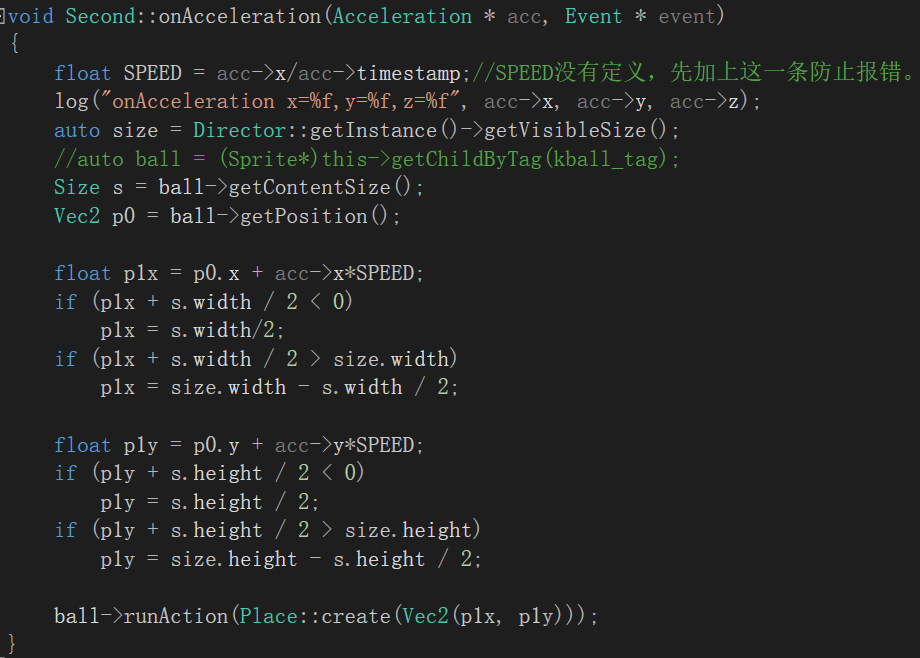
onEnter函数中：



.h中声明回调函数：



函数定义：



# 实验心得：

**遇到的一个问题**：一开始切换到下一个场景，下一个场景中的事件、按钮都不能响应。

**后面发现原因**：在OnExit函数中，我一开始用的是removeAllEventListeners函数。在场景切换中（我这里用的是replaceScene,pushScene前面几步也一样)，生命周期函数调用的顺序是这样的：

Step1 初始化**目标**场景Init

Step2 退出**原**场景的过渡动画开始

Step3 进入**目标**场景onEnter

Step4 退出**原**场景onExit

Step5 进入**目标**场景的过渡动画结束

也就是说在目标场景的onEnter函数(这里创建目标场景的监听器）调用后才调用原场景的onExit, 如果使用removeAllEventListeners函数，所有监听器都会被移除，所以目标场景的事件都不能响应。

**解决方法**：使用removeEventListenersForTarget(this)，只移除自身场景的所有监听器，这样就不会影响到目标场景了。

**日**