

让不懂编程的人爱上iPhone开发(2017秋iOS11+Swift4+Xcode9版)-第14篇

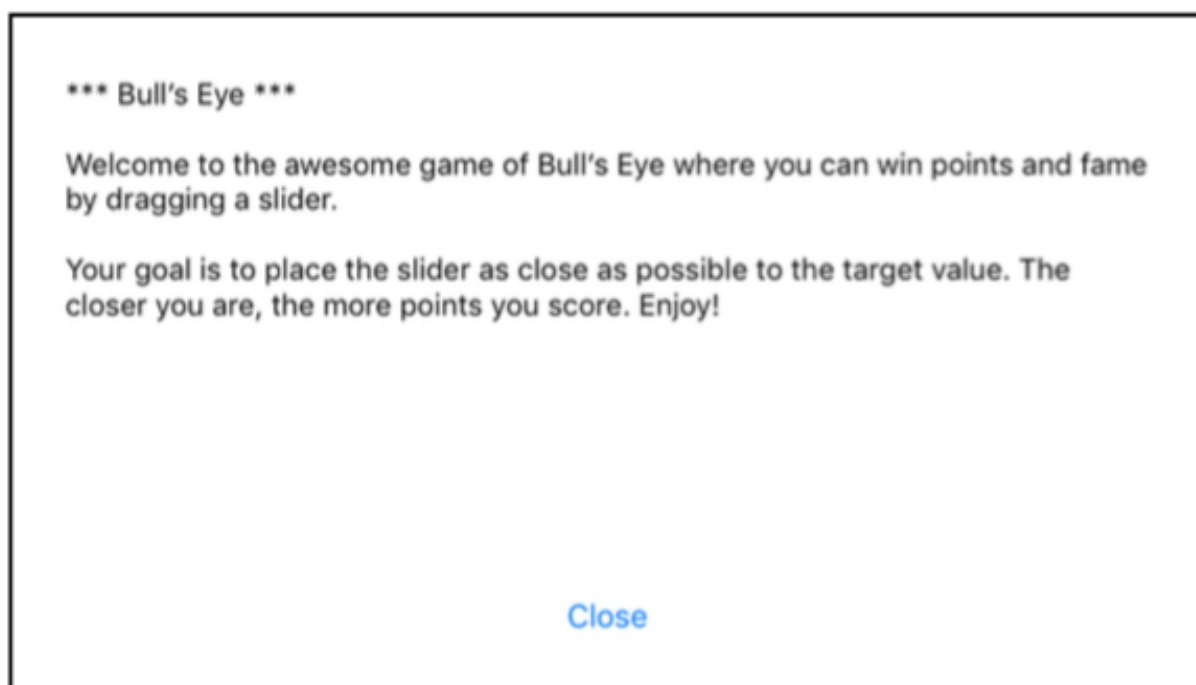
上一篇的内容比较简单，只不过是修修补补而已。而这一篇我们将会把游戏从单一界面拓展到多界面。还记得游戏界面中的i 按钮吗？现在点击这个按钮是没有反应的，接下来我们要解决这一问题。

现在开始吧。

游戏的主体已经基本建立了，作为一个艺术家，最自豪的事情就是在作品上署上自己的大名。作为产品的开发者也不例外，虽然创作的是软件产品，但仍然可以归为创作家的系列。而且，我们还需要给玩家提供一些简单的帮助，让他们知道如何进行游戏。

所以我们将要添加一个about界面，这个新的about 界面中包含了传说中的text view（文本视图），里面放了一堆说明文字当然还有一个用来关闭当前界面的按钮。

我们可以通过触碰游戏主界面右下角的(i)按钮进入这个界面。



当然，上面的界面不是最终的，很快我们就会对其美化~

绝大多数的iOS应用或游戏中都不止有一个界面，即便是最简单的游戏也是如此，因此我

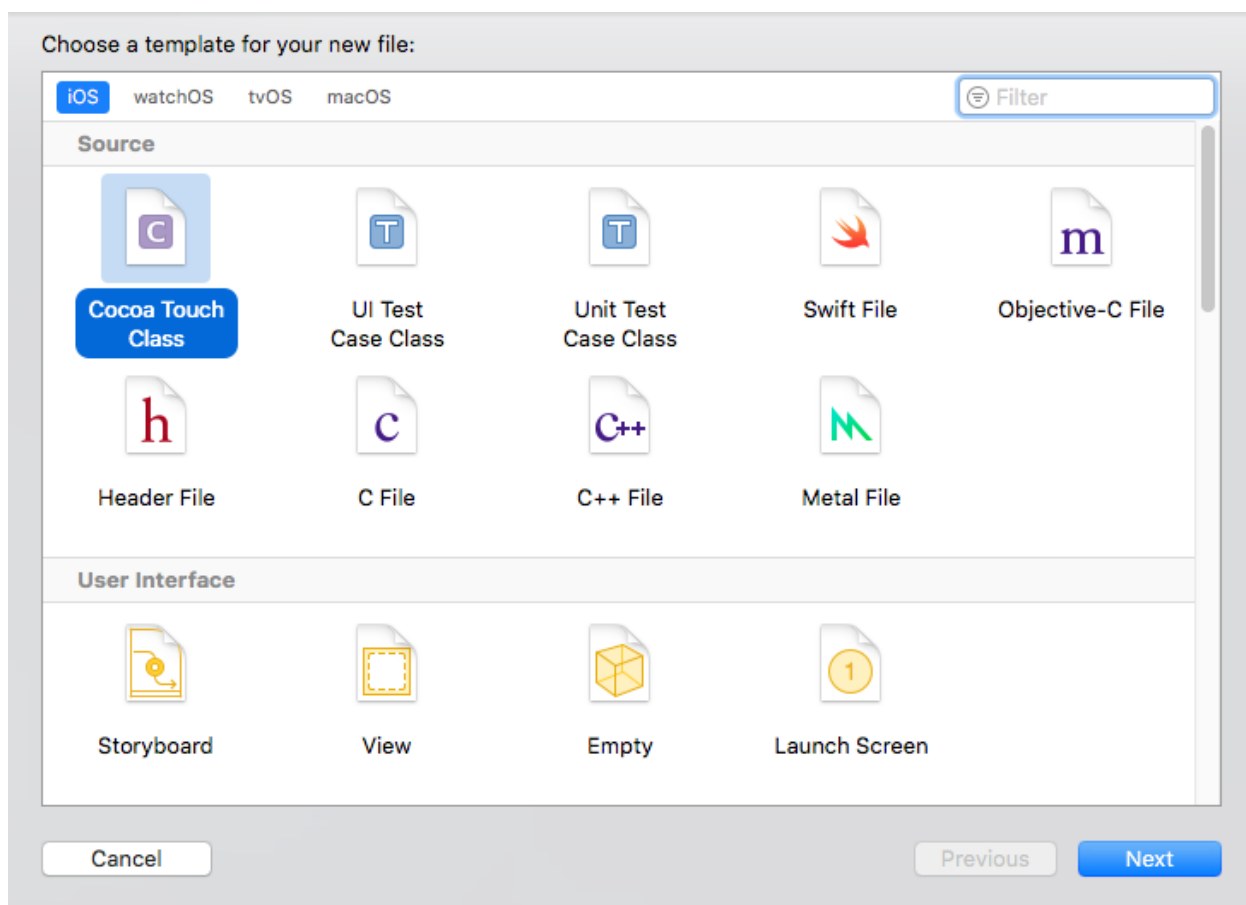
们需要学习如何在应用中添加多个界面。

还记得我之前说过的吗？iOS应用中的每个界面都对应一个独立的视图控制器。当你看到“界面”的时候，就应该想到“视图控制器”。

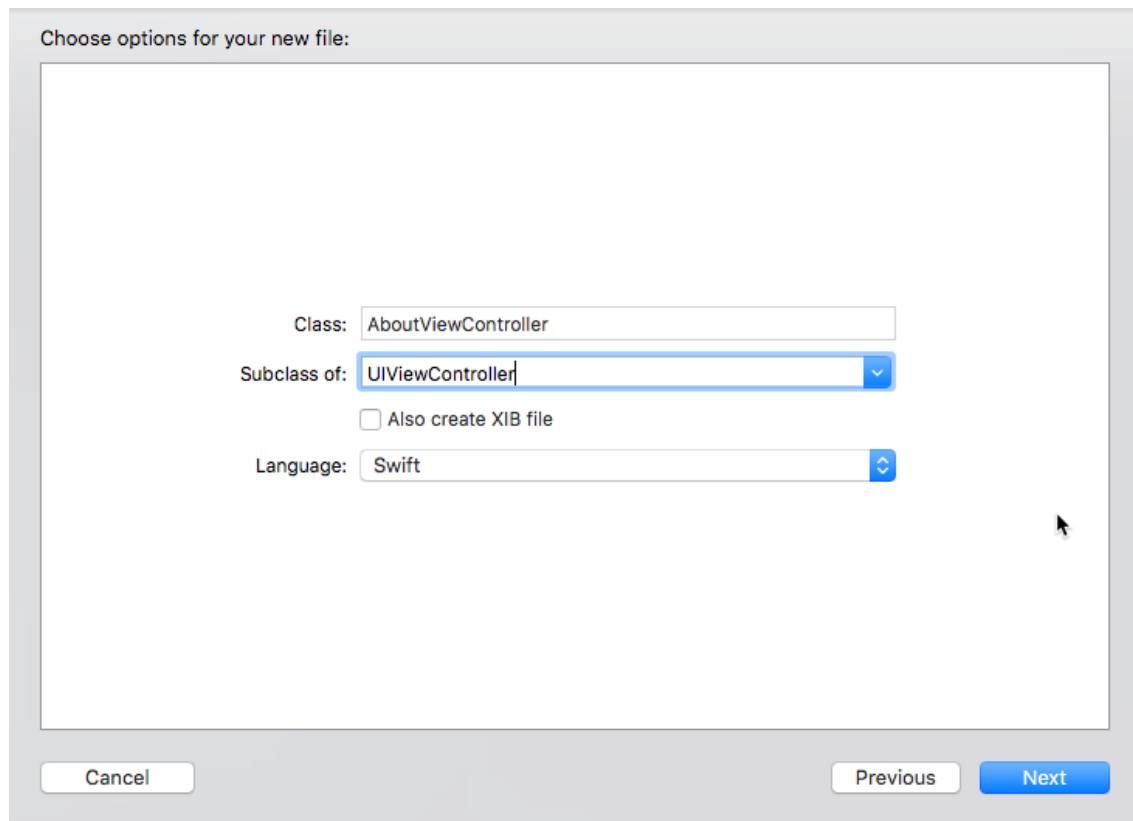
Xcode已经为我们自动创建了ViewController这个视图控制器，但这个新的关于界面的视图控制器则需要我们自己手动来创建。

下面的东西可是全新的，不要眨眼啊。

在Xcode中，点击顶部菜单栏中的File，然后选中New File...在弹出的窗口中选中Cocoa Touch Class。



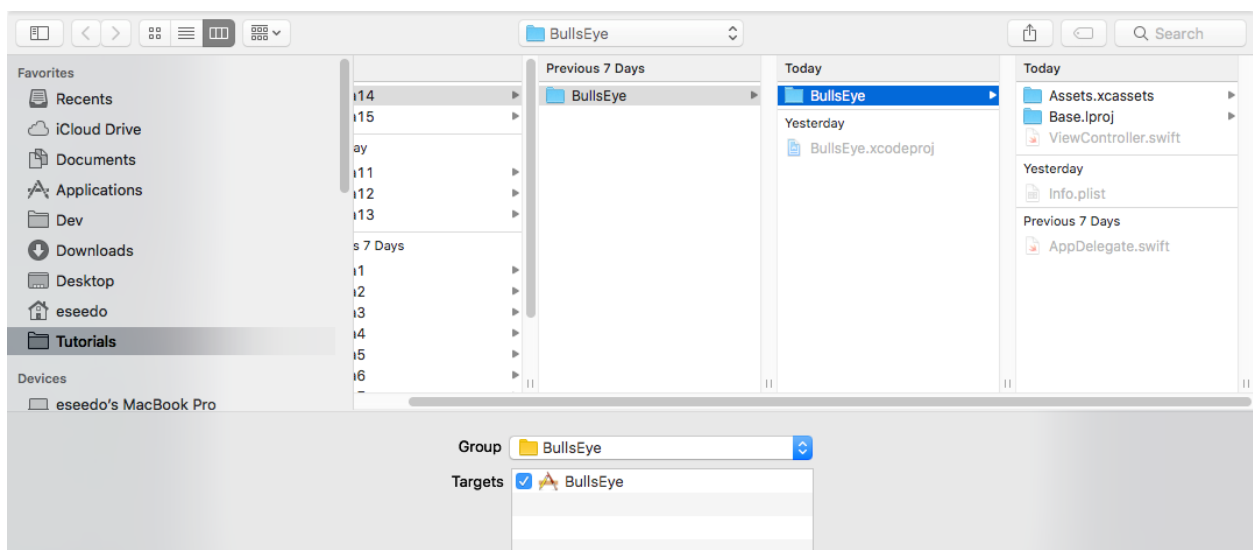
点击Next，看到类似下面的界面：



照着上面输入相关信息：

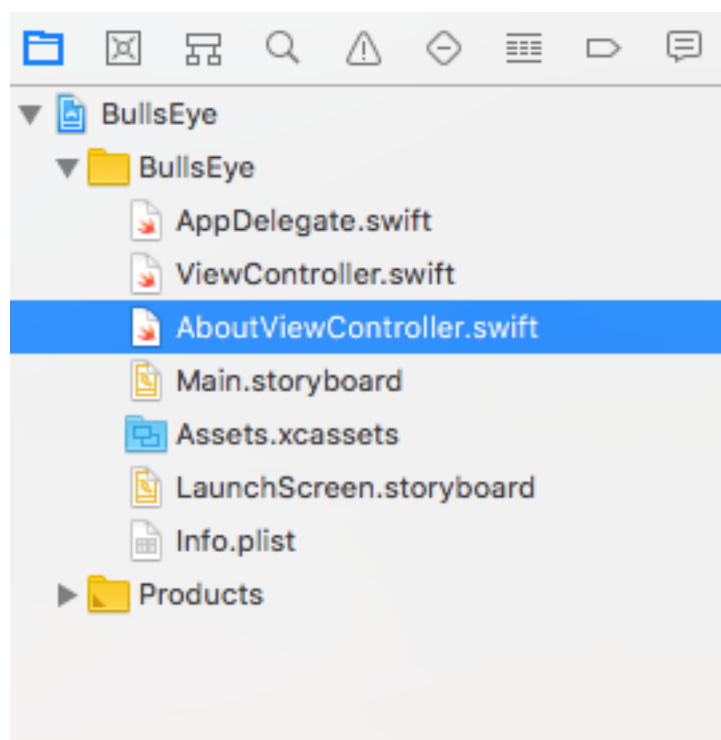
- Class: AboutViewController
- Subclass of: UIViewController
- Also create XIB file: 不选择
- Language: Swift

点击Next，Xcode会让你选择保存这个新视图控制器的地方，不要想太多，直接默认Create就好了。不过要记得勾选Targets后面的那个选框。



此时会看到在Xcode的项目导航部分出现了一个新的文件，名为AboutViewController.swift。

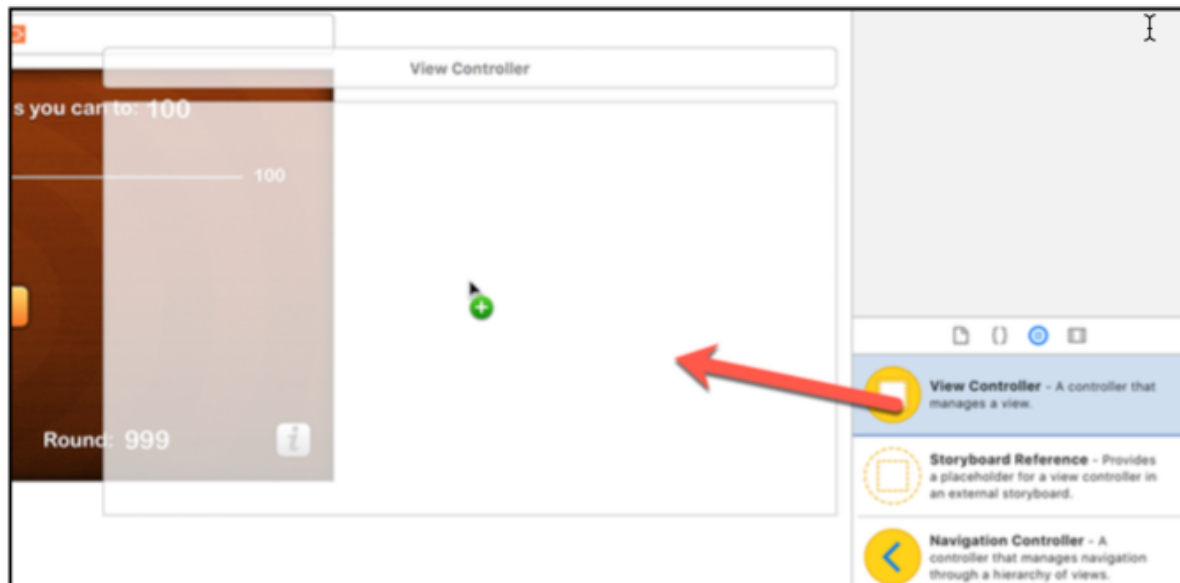
如下图所示：



为了设计这个全新的视图控制器，我们需要再次切换到内置的Interface Builder。

在Xcode的左侧导航部分点击打开Main.storyboard。目前里面并没有关于界面的视图控制器，因此我们需要手动创建一个新的。

从Xcode右下方的Object Library中拖出一个View Controller到画布上，放在之前的视图控制器的右侧。



此时两个视图控制器可能重叠到一起了，为了看起来顺眼，不妨手动拖动调整它们的位置。Interface Builder貌似不是处女座的。

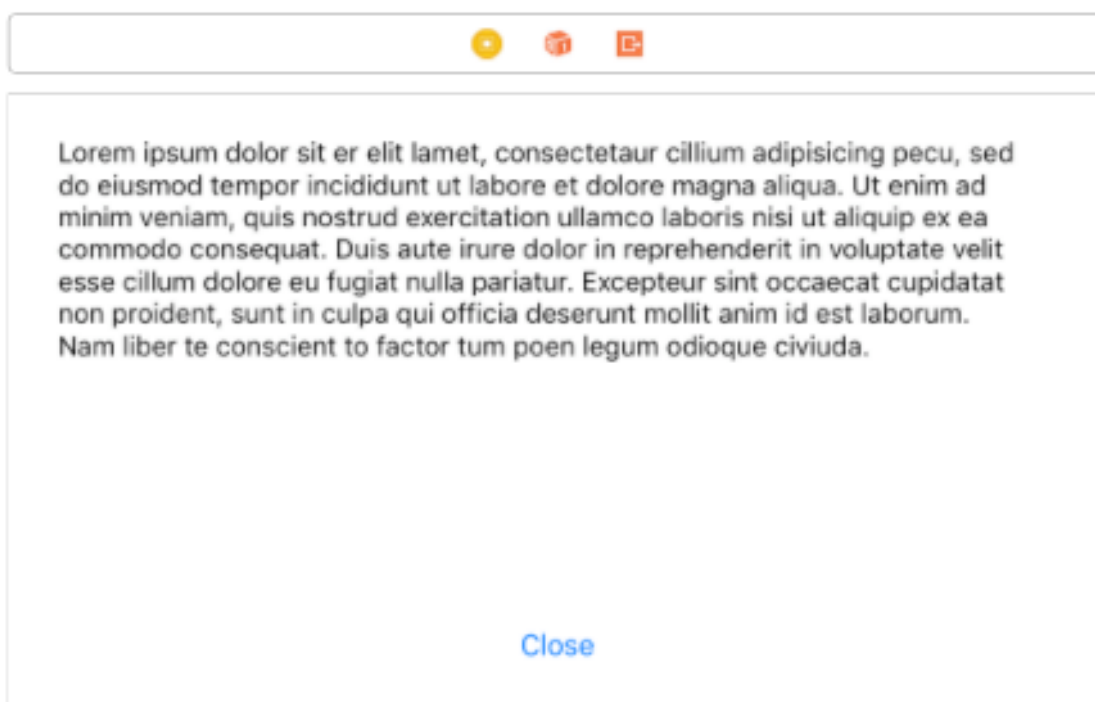
接下来在Xcode右下角的面板中切换到Object Library,找到 Button,把它拖动到视图上，把它改名为“Close”，然后放在视图底部的中央位置。

然后找到Text View，用鼠标拖动到白色的视图上，然后让它覆盖按钮之上的大部分空间。

注意这里还有一个Text Field，不过它是用来输入单行文本的（比如用户名和密码）。

Text View（文本视图）里面则可以放置多行文本。

最后我们得到的是类似下面的东西：



默认情况下，文本视图里面包含了一大堆拉丁语占位文字（也就是传说中的“Lorem Ipsum”，如果你恰好懂得拉丁语，或许会感到很幸福，无人爱苦，亦无人寻之欲之，乃因其苦... 参考<http://baike.baidu.com/view/2339924.htm>）。

双击文本视图，然后修改其中的文字，随便你用怎么样的文字都可以。

如果你想偷懒，可以直接用下面的这些文字（或者随便什么文字都行）。

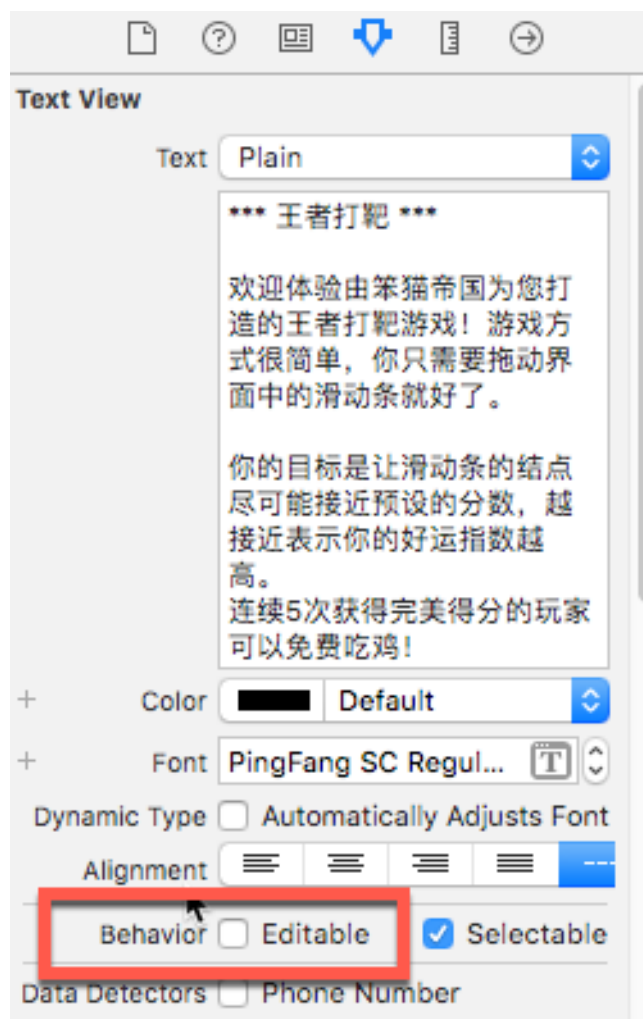
*** 王者打靶 ***

欢迎体验由笨猫帝国为您打造的王者打靶游戏！游戏方式很简单，你只需要拖动界面中的滑动条就好了。

你的目标是让滑动条的结点尽可能接近预设的分数，越接近表示你的好运指数越高。
连续5次获得完美得分的玩家可以免费吃鸡！

你也可以直接把文字拷贝粘贴到text view的Attributes inspector中。

别忘了在右侧面板的Attributes Inspector选项中取消选中Editable，否则玩家就可以自己输入文本了。在我们这款游戏里面，暂时不想搞这些东西。



好了，关于界面的界面设置就是这样了。

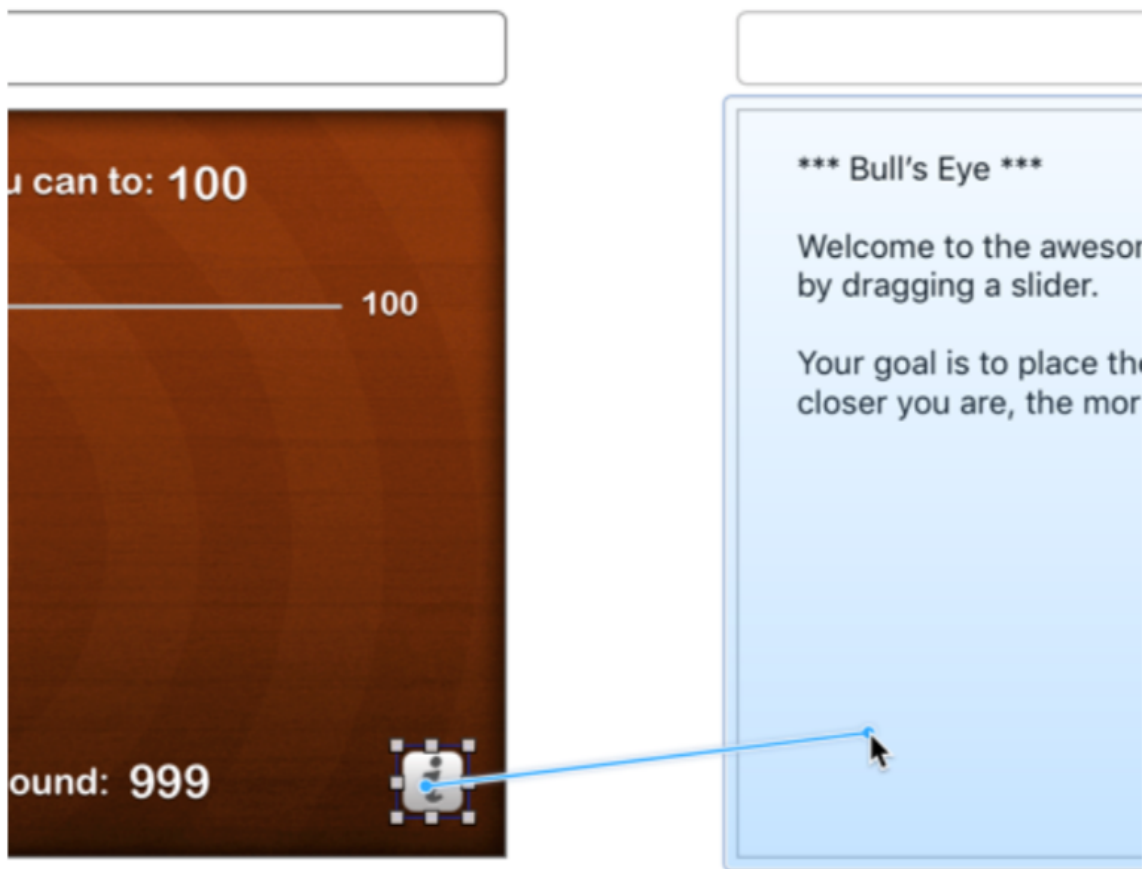
那么我们如何让用户触碰(i)按钮时打开新的关于界面呢？Storyboards里面提供了一个很简单的方法：segue（你可以发音seg-way，是的，就是那个土掉渣但又屌炸天的电动平



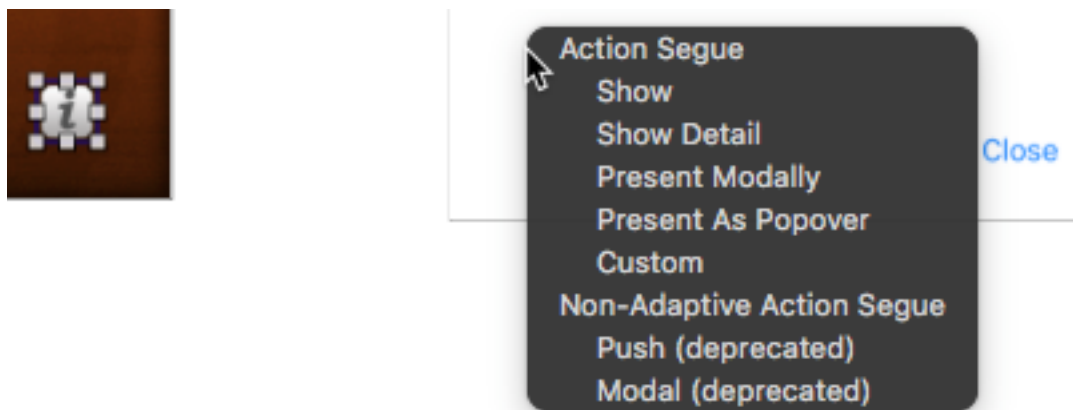
衡车。

一个segue其实就是从一個界面切换到另一个界面的过渡中枢。

在Xcode中切换到Main.storyboard，然后选中(i)这个按钮，按住Ctrl键，按住鼠标左键拖出一条线到关于界面上。



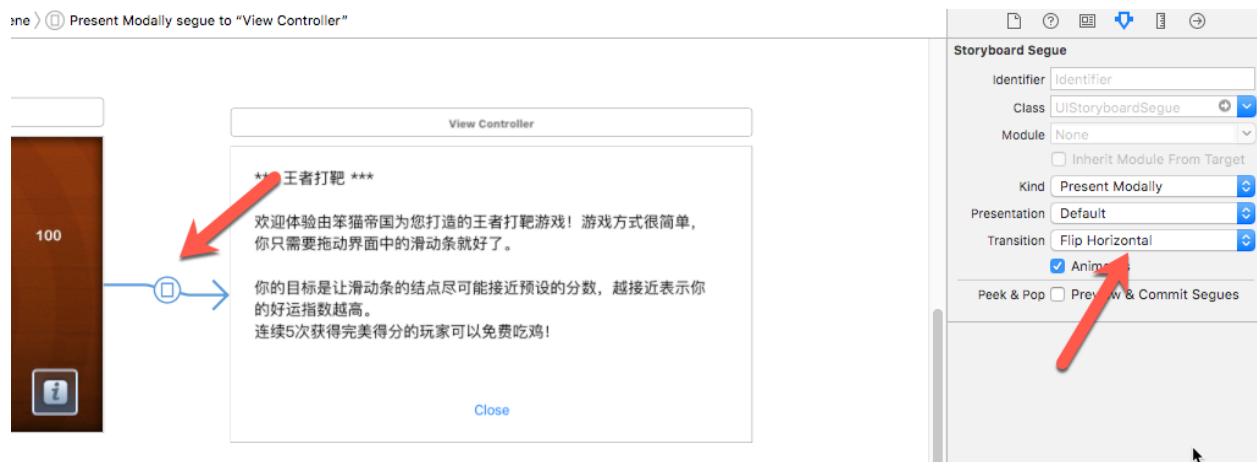
松开鼠标，可以看到一个弹出菜单，从其中选择modal。



此时在两个界面之间会出现一个箭头，这个箭头就代表一个segue。



点击这个箭头选中它，segue也有自己的属性。在Xcode右侧面板中切换到Attributes inspector，选择Transition, Flip Horizontal。这样UIKit将在两个界面之间切换时使用这样的动画特效。



现在你可以尝试编译运行应用，触碰(i)按钮看看效果。

你会看到一个简单的翻转动画效果。

不过似乎还有点小问题，如果我们触碰关于界面中的关闭按钮，什么也不会发生。

一旦玩家进入关于界面，他就被锁定在这个空间了。想想卡布里卡里面的虚拟人，被长期关闭在一个单独的空间里面暗无天日，是多么的残忍。

为什么会发生这种事情呢？因为segue虽然是界面切换的利器，遗憾的是它们都是单向的。为了让虚拟世界和现实世界中的人都能开心起来，我们必须让关闭按钮和某个动作方

法关联在一起。

不过和之前不同的是，这次我们需要让动作方法和AboutViewController中的界面元素关联在一起，而非ViewController，因为关闭按钮是关于界面的一部分，而不是主界面的组成部分。

打开AboutViewController.swift，然后使用以下代码替代其中的内容。

```
import UIKit

class AboutViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBAction func close(){
        dismiss(animated: true, completion: nil)
    }

}
```

close()这个动作方法的作用就是通知UIKit用动画效果关闭关于界面。

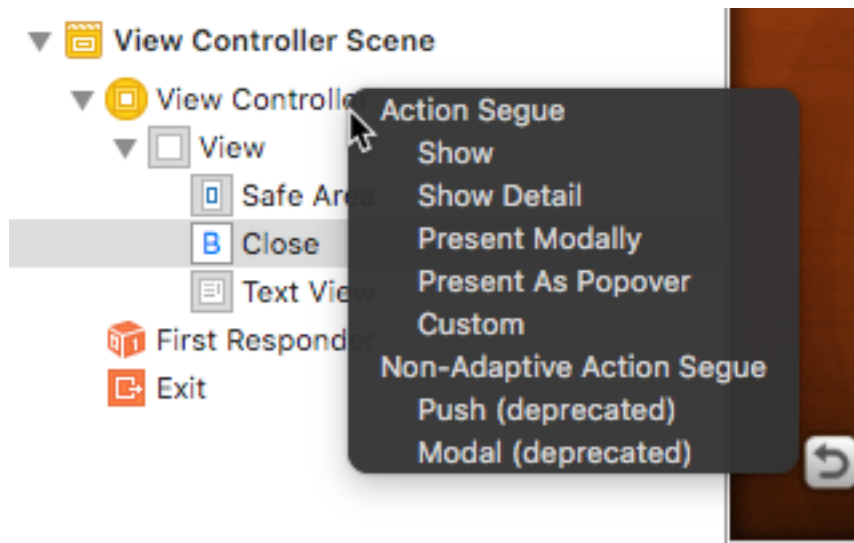
如果上面的那行长代码更改为dismiss(false,...)，那么就不会有任何翻转效果，主界面会立即呈现出来。

为了让用户体验更好，显然使用某种精巧的动画特效要比较贴近人心。

最后就是在界面元素和动作方法之间创建关联了。

在Xcode中打开Main.storyboard，然后选中关闭按钮，按住Ctrl键，用鼠标拖出一条线到关于界面的View Controller。

不对，在弹出式菜单中没有出现我们期待的close动作方法，实际上我们会看到类似下面的弹出菜单。



思考：

想想看究竟哪里出错了。非常常见而又让人头大的错误。

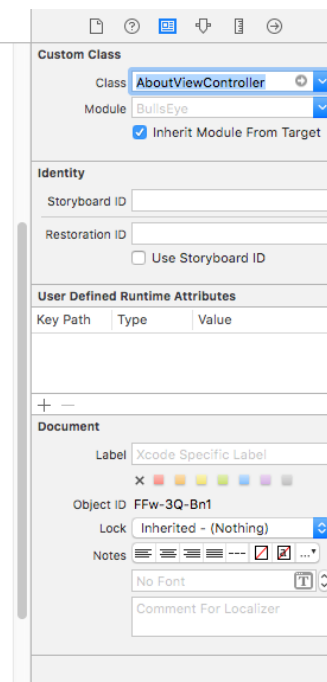
问题其实很简单，storyboard中的界面不知道自己其实代表着AboutViewController（因为没人告诉它）。

我们在前面的操作中只是添加了AboutViewController.Swift的源文件，同时往storyboard中拖入一个新的视图控制器，但是我们没有告诉storyboard这个新的视图控制器实际上是AboutViewController（这就是为什么你在outline 面板中看到的名称是View Controller，而不是About View Controller。）

好在这个问题不难解决。在Interface Builder中选中这个关于界面的视图控制器，然后在Xcode右侧面板切换到Identity inspector。

在Custom Class下面，将Class后面的类型选择为AboutViewController

现在我们终于可以把关闭按钮和动作方法关联在一起了。



按住Ctrl键，有鼠标从关闭按钮拖一条线到About View Controller，这件事情你应该已经轻车熟路了。此时你会看到弹出菜单中有一个选项是close动作方法，选中它，关联就创立了。

再次编译运行，现在你可以很轻松的从关于界面返回游戏主界面了。

好吧，我们的游戏似乎已经完工了。所有的功能都具备了，而且也没有什么bug。现在的问题是-界面太难看！如果你现在把这款应用提交给App Store，百分百会被拒掉！

苹果官方提供了一个iOS 产品设计的人机交互指南，无论是程序猿，产品经理，还是设计人员，都应该反复仔细阅读。

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html#//apple_ref/doc/uid/TP40006556

即便你以后不打算做iOS开发，想做android,网页或者桌面开发，了解iOS的人机交互指南也是非常重要的。毕竟，苹果可是设计领域当之无愧的神级公司！

对于关于界面，我们做一个小小的修改吧，更改其背景图片为之前主界面的图片即可，具体怎么操作就不再赘述了。

好了，今天的学习先到此结束吧，这一篇的内容可能对新手有点吃力，不过我们也学到了更多有用的东西。



老习惯，福利一张。
乔帮主座驾奔驰SL55AMG

