

让不懂编程的人爱上iPhone开发(2017春iOS10+Swift3+Xcode8版)-第5篇

好了，继续我们上次的课程。

提醒下大家，本系列教程的每一个章节都有对应的项目源代码，如果大家在看教程的过程中遇到任何问题，可以参考项目源代码。

源代码地址：

<https://github.com/eseedo/iOSCourse.git>

我个人使用的开发环境：

电脑：Macbook Pro Retina 15-inch Mid 2014

操作系统：macOS High Sierra(10.13)

iOS系统版本：iOS 11

编程语言：Swift 4

开发工具包：Xcode 9.0

在上一篇的内容中我们花了很多功夫进行一些科普和理论知识的介绍，现在又到了动手时间了。

首先回到ViewController.swift，更改sliderMoved()动作的内容如下：

```
@IBAction func sliderMoved(slider: UISlider){  
  
    //    print("滑动条的当前数值是： \(slider.value)")  
    currentValue = lroundf(slider.value)  
  
}
```

这里我们注释了print()这行代码，然后增加了下面的这行代码代替

```
currentValue = lroundf(slider.value)
```

在Swift中，只要在代码行的最前面添加双斜杠（半角），就可以屏蔽一行代码。

关于代码注释后面还会详细解释，这里就不多说了。

那么这行代码是什么意思呢？之前我们曾看到过slider.value，它代表当前时刻滑动条的数值。该数值的范围在1到100之间，当然后面还有小数点几位。currentValue是我们之前所创建的变量名称。

当我们要把一个新的数值放到一个变量里，只需要这样来写：

variable = the new value

这就是术语党们号称的“赋值(assignment)”。我们把一个新的值“赋予”变量。这里我们把滑动条的数值赋予currentValue这个变量。

这个应该比较好理解，但lroundf又是怎么回事？刚才我们说过，滑动条的数值不是一个整数，而是小数点后面还有几位(在我的Xcode里面小数点后面有6位)。

刚才使用print()的时候，只要拖动滑动条，就可以在调试区看到这个数值。

当然，这样一来游戏的难度也大大增加了。你不可能猜得那么准，甚至连小数点后面6位都猜到，除非你是神。为了让游戏难度不至于这么变态，我们需要把数值转换成一个最接近的整数。这就是为神马我们把currentValue定义为一个int整数值，因为int类型的变量里面会保存整数（integers）。

这里的lroundf()是一个函数，它可以把一个带小数点的数值四舍五入到最接近的整数，然后我们会把转换后的整数保存到currentValue这个变量里。

科普时间：

什么是函数？函数和方法有什么区别？

我们已经知道了方法会提供功能，而函数则是为应用添加功能的另一种方式。

Swift语言中使用函数把多行代码整合到一个连贯的整体中。

方法和函数唯一也是本质的区别是，函数是代码世界里独立的生命体，它不依赖于某个对象而生存。

但方法不同，任何一个方法都是某个对象的附属生命（寄生体？），同时也只有它的宿主对象才能调用这个方法。

因此我们在Swift中使用func关键词来定义方法。不过通用的函数，或是所谓的free functions（自由函数）可以在任意地方使用。

Swift提供了一个巨大的可用函数库。比如lroundf()就是其中的一个，我们在教程中还将用到其它的函数，比如print()。

函数的名称后面带着括号，括号里面有一个或多个参数。

现在让我们更改showAlert()方法如下：

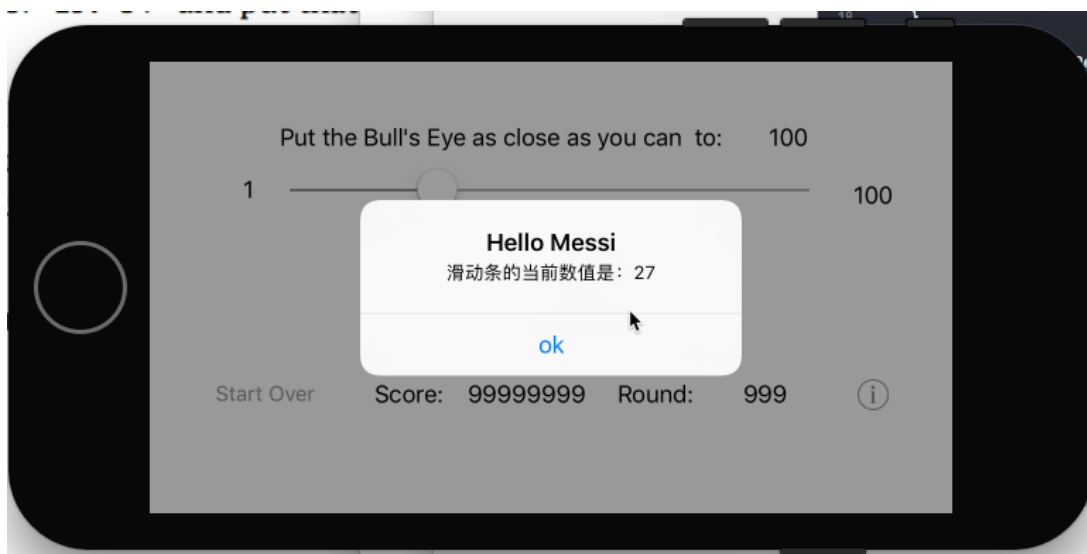
```
@IBAction func showAlert(){  
  
    let message = "滑动条的当前数值是： \(currentValue)"  
  
    let alert = UIAlertController(title:"Hello Messi",  
                                message:message,  
                                preferredStyle: .alert)  
    let action = UIAlertAction(title:"ok",style: .default,handler: nil)  
    alert.addAction(action)  
  
    present(alert, animated: true, completion: nil)  
}
```

在以上代码中，黄色高亮部分是新添加或修改的部分，其它保持不变。
和之前一样，我们创建并显示一个提示对话框UIAlertController，不过这一次的内容变了，我们使用message代替之前的文字内容，而message的内容则使用currentValue的数值来定义。

假定currentValue是34，这就意味着滑动条大概在靠左边1/3的位置。那么我们上面的代码就会把字符串转化为：“滑动条的当前数值是： 34”，然后把这个字符串放到名为message的对象中。

print()之前做了类似的事情，只不过它是直接把结果打印在调试面板中。这里我们则使用提示对话框，因为玩家看不到调试面板的内容，但可以看到提示对话框。

点击Run运行应用，拖动滑动条，然后触碰按钮。现在就会弹出一个提示对话框告诉你滑动条的当前数值。



不错，又前进了一步。虽然我们前进每一步都很慢，但每一步都很坚实。

这里我们使用一个变量来保存滑动条当前位置四舍五入后的数值，这样应用中的其它地方就可以用到变量里面的值。比如这里在提示对话框中就用到了。

如果我们不移动滑动条就触碰按钮，那么对话框中的数值也会保持不变。变量会始终记住保存在它里面的数值，直到我们放入新的数值。

狩猎时间

在代码的世界里面，bug（虫子）很可恶，它的作用是让玩家所看到和感受到的东西和你所设计的东西南辕北辙。大多数程序猿终其一生（好吧，是在他们的职业生命周期里）都在和虫子打交道。无虫子，不编程。

虽然这个游戏还没多少东西，里面已经出现了bug。现在让我们看看这个bug的表现：点击Xcode上的Stop完全关闭应用。然后点击Run重新运行，先不要拖动滑动条，直接按按钮。

这时对话框提示，“滑动条的当前数值是：0”。但显然滑动条的拖动手柄在中间，这个数值应该是50才对啊魂淡！可恶的虫子！

小练习：想想看为神马这个数值会是0.

答案：线索在这里，只有当我们不拖动滑动条就按按钮的时候才会出现这个bug.

显然，当我们还没有拖动滑动条的时候，sliderMoved()这个消息根本没机会发送，我们也不可能读取滑动条的数值，把它保存到currentValue变量中。

而在Swift里面，实例变量的默认值是0，所以我们就看到了这个bug。

为了消灭这个虫子，我们需要更改一个方法。在Xcode中点击ViewController.swift，然后找到currentValue定义的代码，将其修改为：

```
var currentValue: Int = 50
```

点击Run运行应用，然后看看虫子被消灭没有。

科普-什么是注释

你可能已经多次看到了用//开始的绿线了吧？这些被称为注释。你可以在这个符号后面写任何文字（包括吐槽，情诗，政治宣言，入党申请书，笑话，YY小说，财务计划。。。）反正编译器看到//就会自动无视后面的东西。

// 我是注释行。窗外闪电雷鸣，我虎躯一震，发现哥竟然穿越了，有木有！

除了//之外，我们还可以使用/*和*/。区别在于//只能注释一行文字，而/*和*/则可以注释它们中间的多行文字。

通常使用/*和*/来注释掉一整块代码。

不过在实际的使用中，哥推荐//。在Xcode中，把光标放在要注释的那行代码上，然后按快捷键Command 和/，就可以了。或者选中一整块代码，用这个快捷键，也是一样。

注释行的最佳用处是解释你的代码的作用。虽然说好的源代码自己可以说明自己，但打点广告，添加点说明和解释是很有必要的。向谁解释呢？很多时候向你自己，不过是一个月后的自己，半年后的自己，甚至是几年后的自己。除非你的记忆如同计算机一样的精确，而且还不死机，那么注释就很有必要。

你可能看到了，Xcode在你的项目文件中自动为你加了这样的一段注释：

```
//  
// ViewController.swift  
// BullsEye  
//  
// Created by eseedo on 10/11/17.  
// Copyright © 2017 eseedo. All rights reserved.  
//
```

这个其实没什么用，看看就好，不爽了可以删掉。

好了，今天的学习到此结束，送福利1P，我最爱的美剧之一，《太空堡垒》系列中的女主。



对于吃货妹子们可以看下面~

