Wiley：移动开发工程师实验报告

云课堂昵称：MartinZhou　　实验日期：2015.12.8

一、实验题目

创建简单计算器程序

二、实验要求

1.使用Swift语言创建简单计算器
2.使用Interface Builder来构建UI

三、操作步骤

1.创建项目并选择Swift作为开发语言

(1)启动 Xcode。在 Welcome to Xcode 界面上,选择 Create a new Xcode project 选项,开启新项目创建。

(2)在 Choose a template for your new project 界面,选择 Single View Application 模板,并点 Next

(3)在 Choose options for your new project 界面上,键入项目名为 MZSimpleCal_1,Language 处选择 Swift,然后点 Next。

(4)在下一个界面,选择保存项目的位置,然后点 Create 来创建。

(5)Swift 中有 AppDelegate.swift、ViewController.swift 和 Main.storyboard 文件,作为编码的主文件。

2.通过Interface Builder创建应用程序的UI

(6)选择 Main.storyboard 文件,点击View Controller按钮如图1，使用右边栏的属性检查器调整 Storyboard 尺寸，如图2

图 1

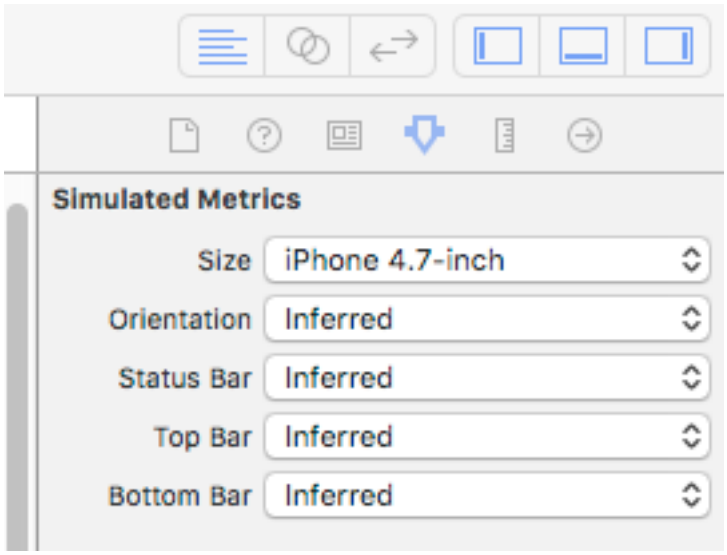

图 2

(7)在实用工具面板中选择库选择器栏,通过选择标签来设计计算器的前端用户界面,如图 3 所示:

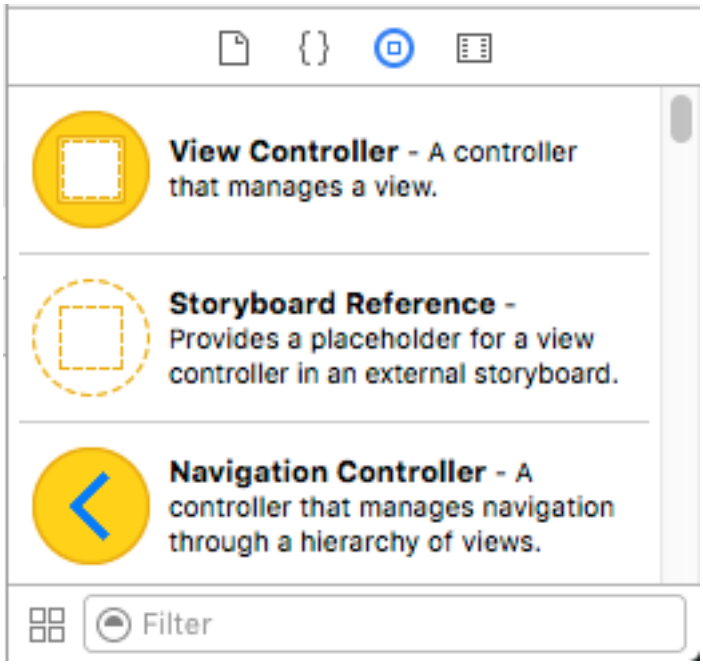

图 3

(8)下面用计算器按钮填充Storyboard。要添加按钮到计算器,你只需要将按钮从实用工具面板拖放到 Storyboard。

(9)按照这种方式设计四个运算按钮,用于计算器的四则运算功能。使用右侧的属性检查器来调整按钮的文本大小(参见图 4)

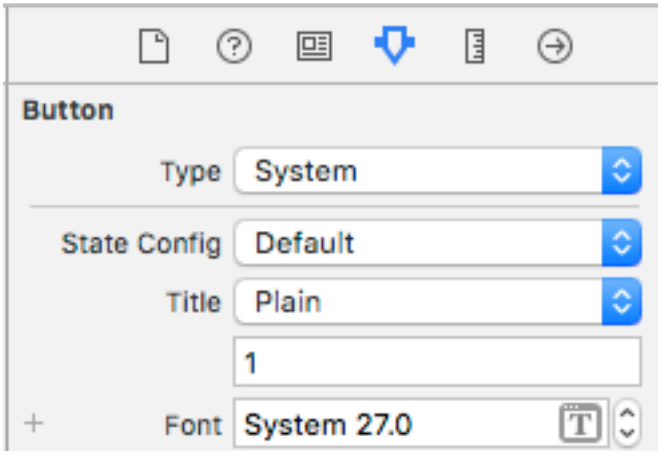

图 4

(10)下面需要添加每个运算符按钮背后的逻辑,确保用户界面能够作为计算器发挥作用。点辅助编辑器来分屏显示,如图5所示:

图 5

(11)右键按住（或者按住control键，左键按住）,拖拽计算器界面的label到ViewController.swift区域。在出现的弹窗中(参见图 6),进行下面这 些操作:

Connection 字段选择 Outlet
Name 字段输入 OutScreen
Type 字段选择 UILabel
Event 中选择 Touch Up Inside
Argument 选择为 Weak
点击 Connect

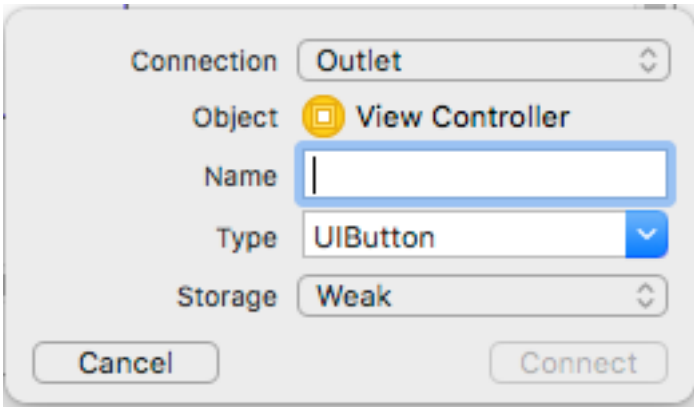

图6

(12)下一步,右键按住，拖拽每一个数字按钮到ViewController.swift 区域。在出现的弹窗中,进行 下面这些操作:

Connection 字段选择 Action
Name 字段输入 calButton
从 Type 下拉列表选择 AnyObject
Event 字段中选择 Touch Up Inside
Argument 字段中选择 Sender
点 Connect

（13）下一步,右键按住，拖拽每一个运算按钮到 `ViewController.swift` 区域。在出现的弹窗中,进行下 面这些操作:

从 Connection 下拉列表选择 Action
Name 字段输入 operator
从 Type 下拉列表选择 AnyObject
Event 字段中选择 Touch Up Inside
Argument 字段中选择 Sender
点 Connect

（14）类似地,你还可以为等号按钮指定动作，同时要注意“=”并不是运算符，是一个单独的存在，不能和其他运算符一起使用一种方法

3.在ViewController.swift中编写与UI关联的代码
（15）然后通过导航面板,选择 ViewController.swift 并添加如下代码:

```
import UIKit

class ViewController: UIViewController {
```

```
var isTyping = false
var firstnum = 0
var secondnum = 0
var result = 0
var operation = ""

@IBOutlet weak var OutScreen: UILabel!

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
}

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


@IBAction func calButton(sender: AnyObject) {
    let number = sender.currentTitle
    if isTyping {
        OutScreen.text = OutScreen.text! + number!!
    } else {
        OutScreen.text = number
        isTyping = true
    }
}

@IBAction func Clear(sender: AnyObject) {
    firstnum = 0
    secondnum = 0
    operation = ""
    isTyping = false
    result = 0
    OutScreen.text = "\(result)"
}

@IBAction func operation(sender: AnyObject) {
    isTyping = false

    //之前代码为firstnum = OutScreen.text!.toInt()!，现在Swift为2.0版本，语法变化
    firstnum = Int(OutScreen.text!)!

    operation = sender.currentTitle!!
    print(operation)
}

@IBAction func equal(sender: AnyObject) {

     //之前代码为secondnum = OutScreen.text!.toInt()!，现在Swift为2.0版本，语法变
化
    secondnum = Int(OutScreen.text!)!

    isTyping = false

    if (operation == "+")
```

```
        {  result = firstnum + secondnum
           OutScreen.text = "\(result)"
           print(result)}

           else if (operation == "-" )
        {   result = firstnum - secondnum
           OutScreen.text = "\(result)"
           print(result) }

           else if (operation == "x")
        {  result = firstnum * secondnum
           OutScreen.text = "\(result)"
           print(result)}

           else if (operation == "/")
        {   result = firstnum / secondnum
           OutScreen.text = "\(result)"
           print(result)
        }
    }

}
```

4.测试程序

（16）最后,通过点击 Run 按钮来运行模拟器

四、实验结果

运算5 x 5 = 25，25 - 7 = 18并查看结果。（详细代码见程序文件）

五、总结反思

在完成实验的过程中，遇到并解决了已下问题：

1.在练习用StoryBoard写UI时，Outlet和Action使用不熟练，可能多次拖动同一个按钮到代码上，导致在模拟器上运行时出错，且错误信息不能准确描述错误位置，解决的方式是重新建立工程文件，并一次性写好了UI，之后无报错。虽然StoryBoard很人性化，但这也带来了一些问题，特别是在StoryBoard出错后主线程奔溃，但却找不到问题原因，故按照官方文档的使用方式来写UI是避免出错的好办法。

2.本课程的Swift的代码是1.0版本的，苹果最近刚升级Swift语言到2.0版本，这就导致示例代码中的部分代码报错，在百度上搜索到解决方式后，修改代码，编译成功。变更代码如下：

```
@IBAction func operation(sender: AnyObject) {
        isTyping = false

       //之前代码为firstnum = OutScreen.text!.toInt()!，现在Swift为2.0版本，语法变化。

        firstnum = Int(OutScreen.text!)!

@IBAction func equal(sender: AnyObject) {

         //之前代码为secondnum = OutScreen.text!.toInt()!，现在Swift为2.0版本，语法变化。
        secondnum = Int(OutScreen.text!)!
```

println改为print

3.Swift工程与OC有所不同，OC是把接口（.h）和实现（.m）文件分开，然而Swift中变量，函数的声明与其实现代码都放在一起，同时Swift的语法中？和！较多，对于刚开始使用时还是有一些不习惯，和示例代码结合来看，这些问题也都克服了。

六、作业提交

1. 请将本文档按照《Wiley移动开发_X章X节_云课堂昵称》的名称命名；
2. 请将本文档、源代码文件打包以附件形式上传到课程作业部分