

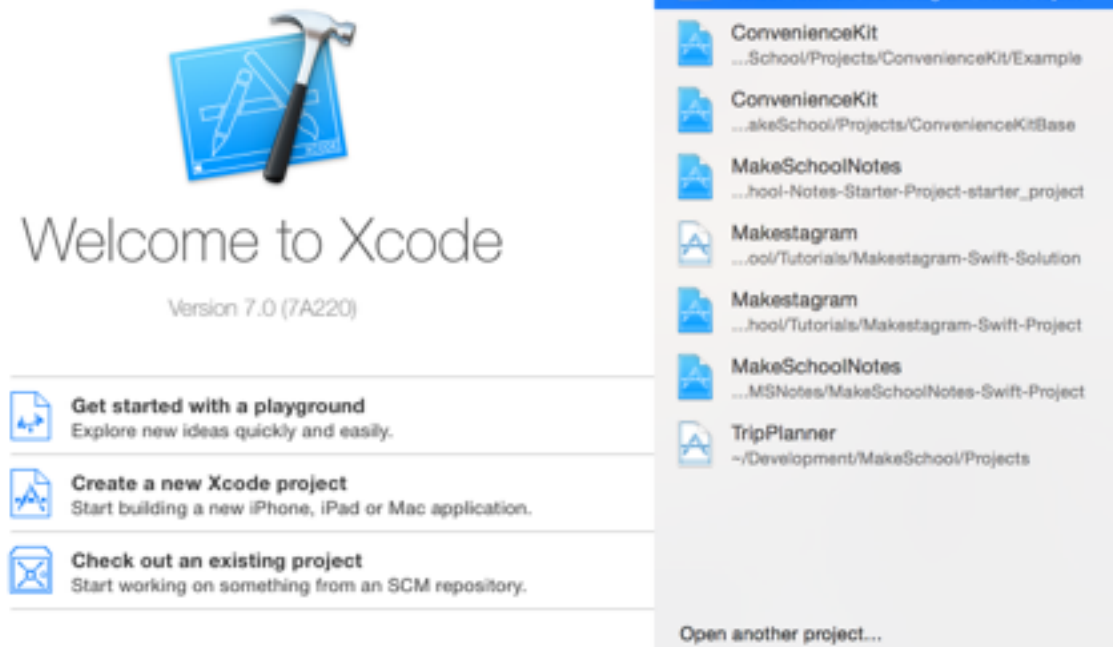
如何与Swift愉快的玩耍

基础入门（可以跳过）

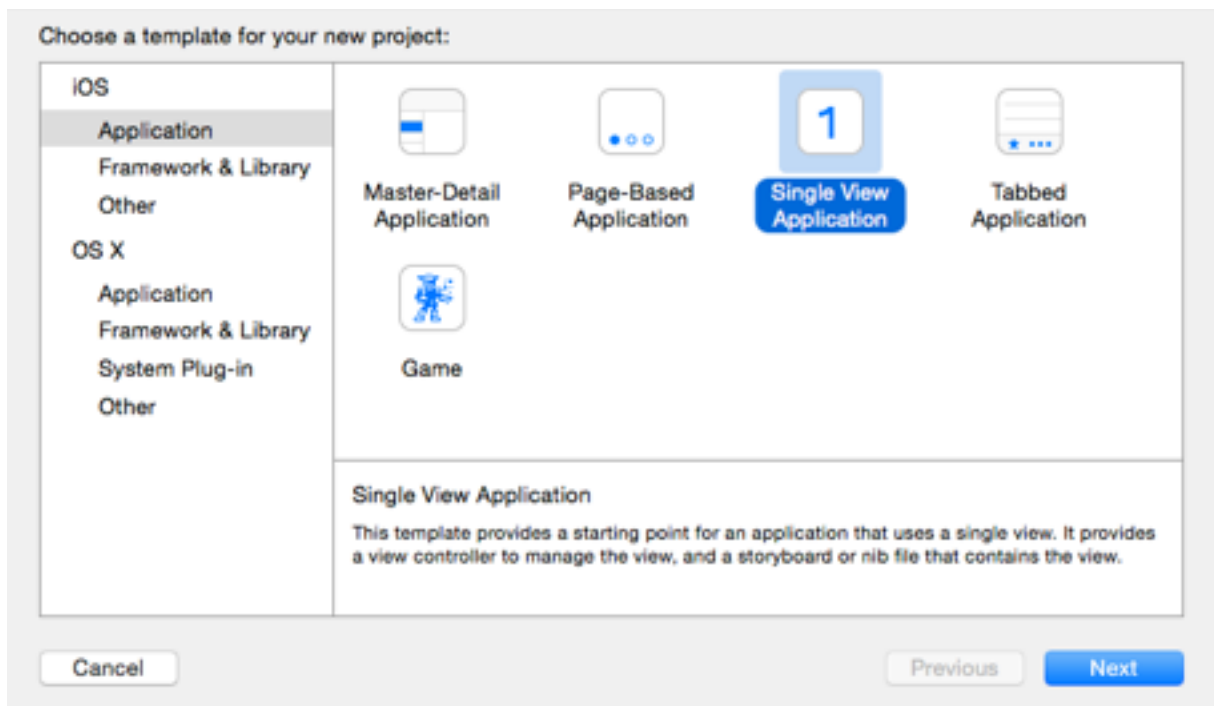
- var 和 let的区别
 - let为常数专门设计
 - var适用于普通的变量
- 在swift中，如果为声明变量的种类，则它会自行推导其种类
- 种类未确定的被称为optional而其值为nil
- 在一个种类不确定的变量后加上“!”就可以使其种类等同于swift猜测的种类
- 另一种方式是将种类不确定的变量赋值，来使等式两边的种类相等
- 在array中只能含有同一种类的元素
- 在if语句中不需要括号
- 标记名字与种类时，名字在前，种类在后，以“名字：种类”的形式

如何使用Xcode（可以跳过）

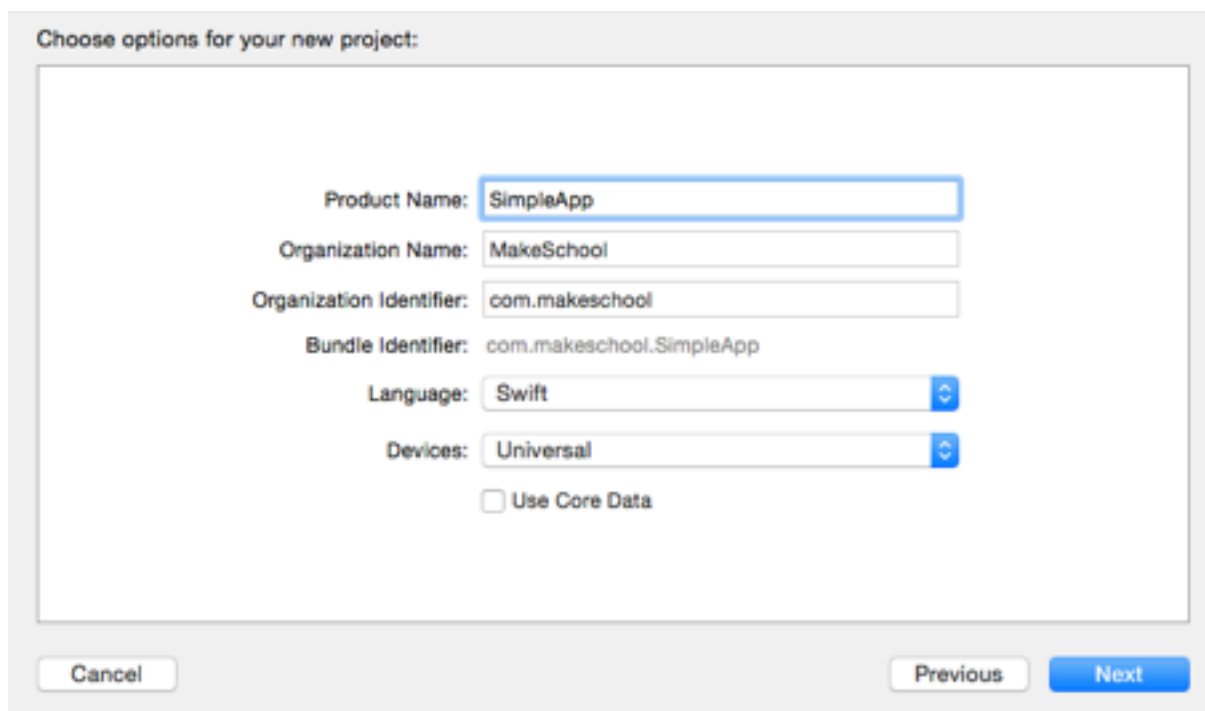
1. 首先，你需要去App Store下载一个Xcode



2. 选择“create a new Xcode project”并且在接下来的页面中选择Application中的“Single View Application”然后点击“next”



3. 接下来，随意给你的项目起个名字，并且在语言中选择Swift（如果在下列空格中还有那一项空着，请自行补足）



Choose options for your new project:

Product Name: SimpleApp

Organization Name: MakeSchool

Organization Identifier: com.makeschool

Bundle Identifier: com.makeschool.SimpleApp

Language: Swift

Devices: Universal

☐ Use Core Data

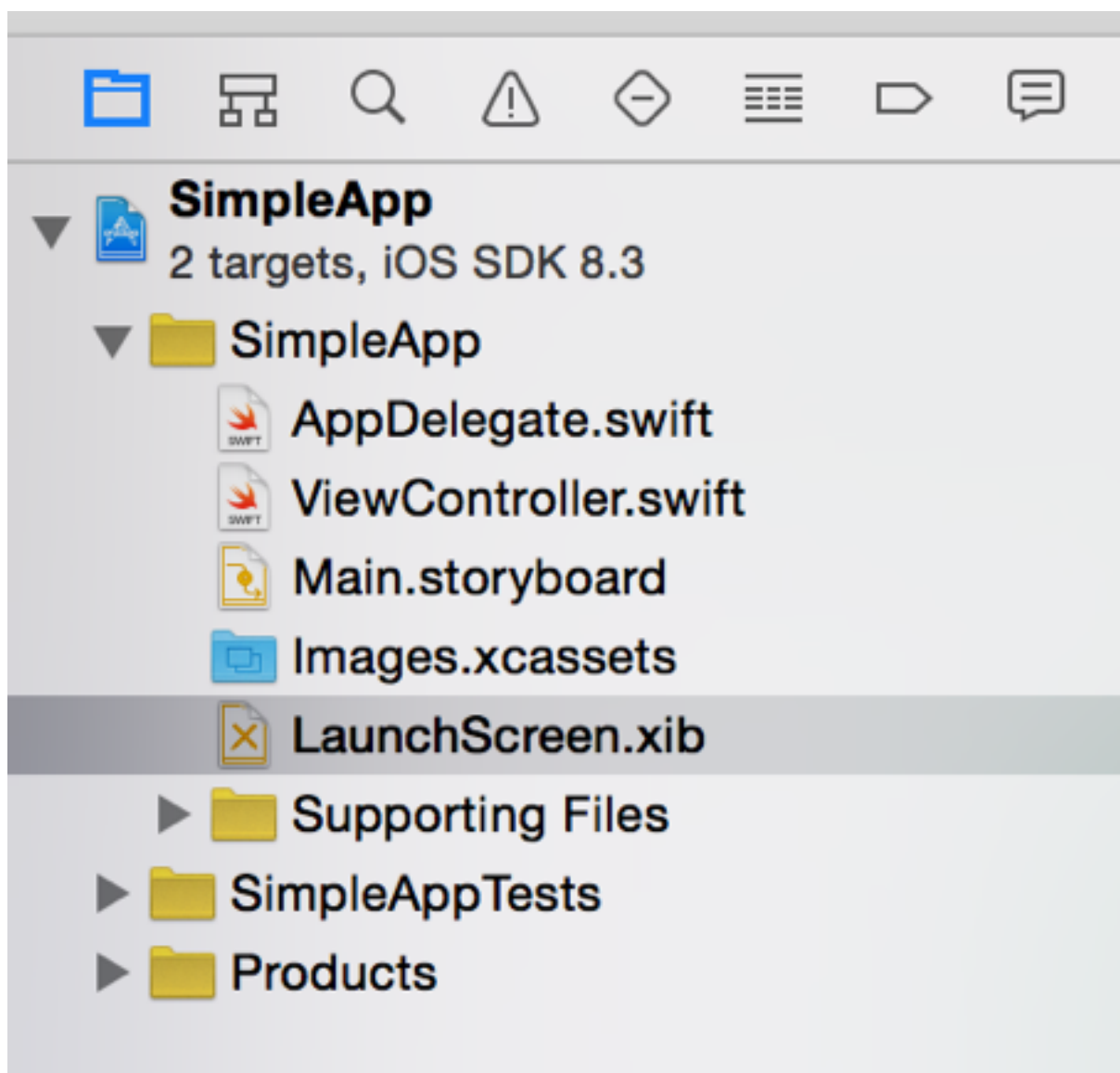
Cancel Previous Next

4. 选择运行项目（看到下面这个页面就非常对了）

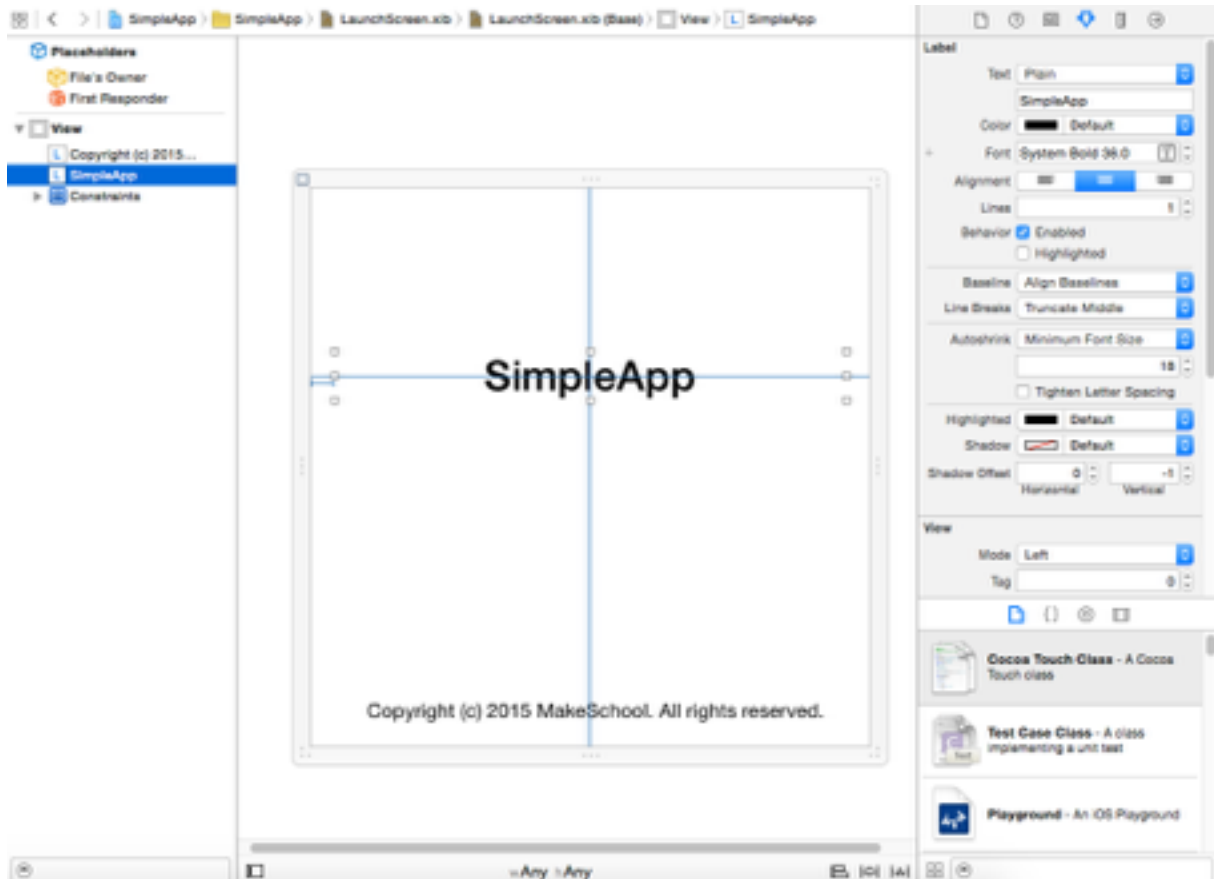
Friday, June 17, 2016



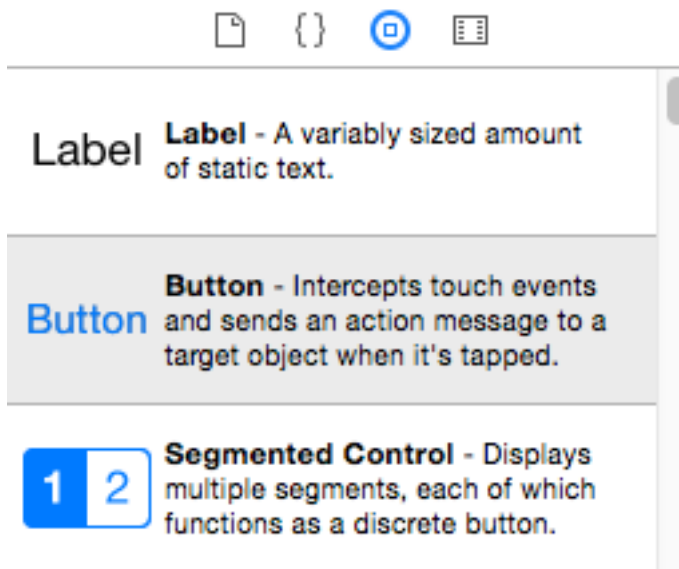
5. 在最左边那一栏选择LaunchScreen.xib



6. 然后在View那一分支下选择你的项目名称（你会注意到右侧有个蓝色的小图标亮了，那个图标指示的是属性）



7. 点击右下角的铜钱形状的标志，比较常用的是label（文字），button（按钮）和 segmented control（苹果手机中解锁时的那种滑动效果）

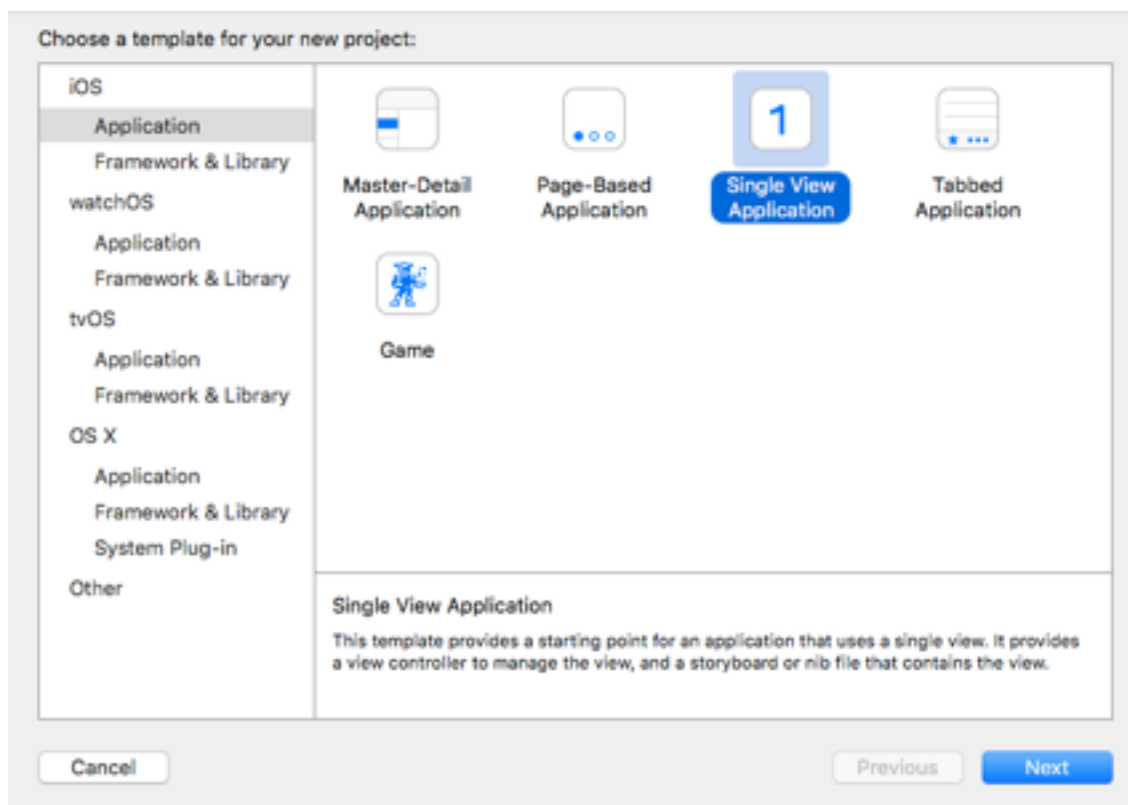


Learn how to make a calculator

1. 首先，打开Xcode（最新版，目前是7.3.1，不同版本的内部功能相同，但不同功能的位置可能会不一样）并选择Create a new Xcode project

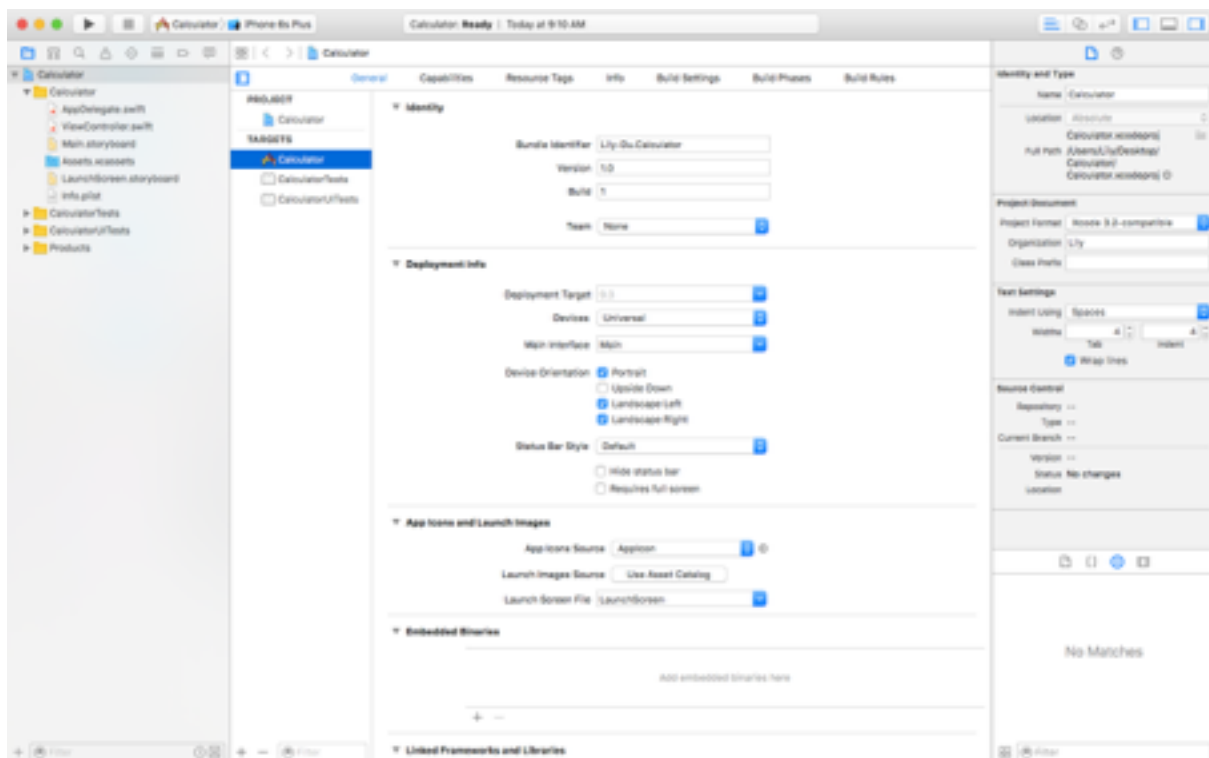


2. 然后选择iOS→Application→Single View Application

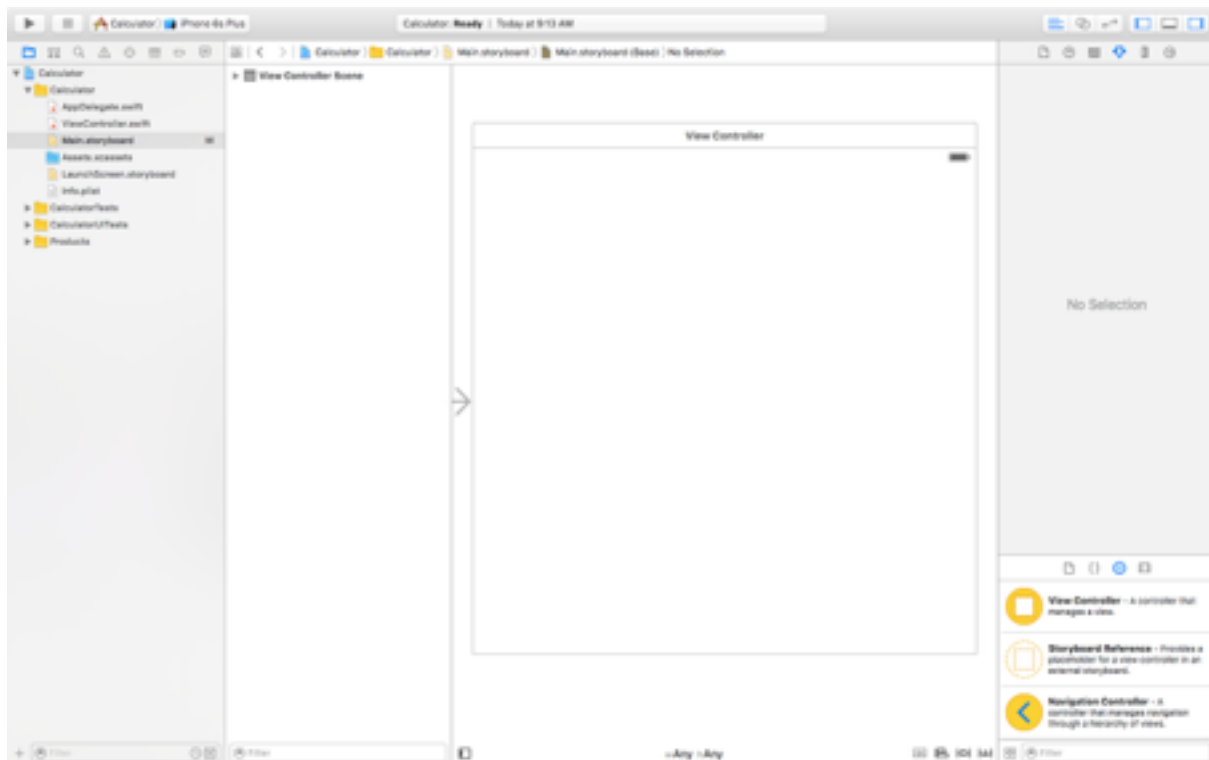


Friday, June 17, 2016

3. 将product的名字改为Calculator然后保存此文档，你就能看到这个页面（如别的栏有空着的也请随意填上）

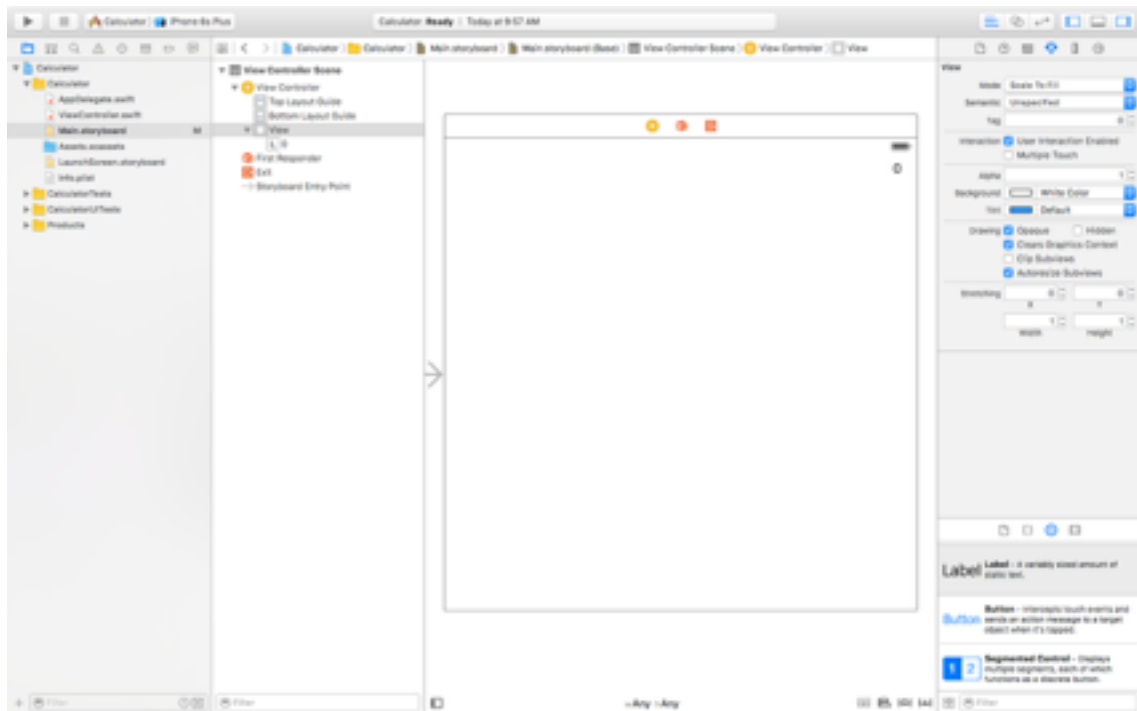


4. 选中Main.storyboard（在左侧栏）

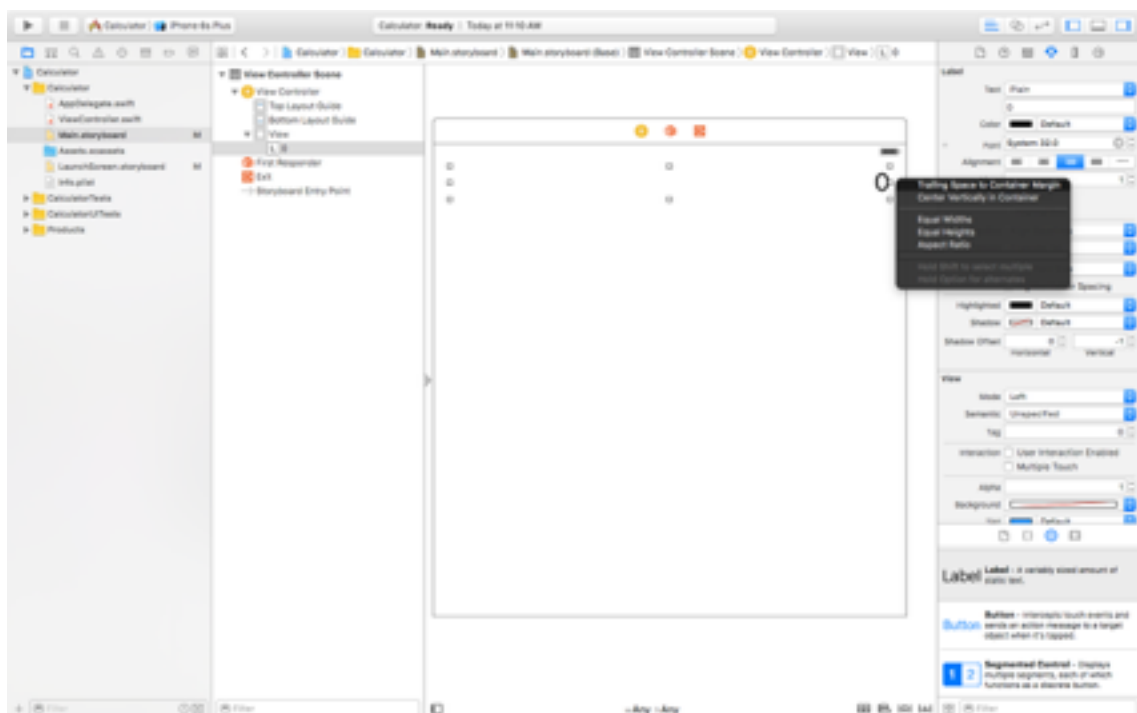


Friday, June 17, 2016

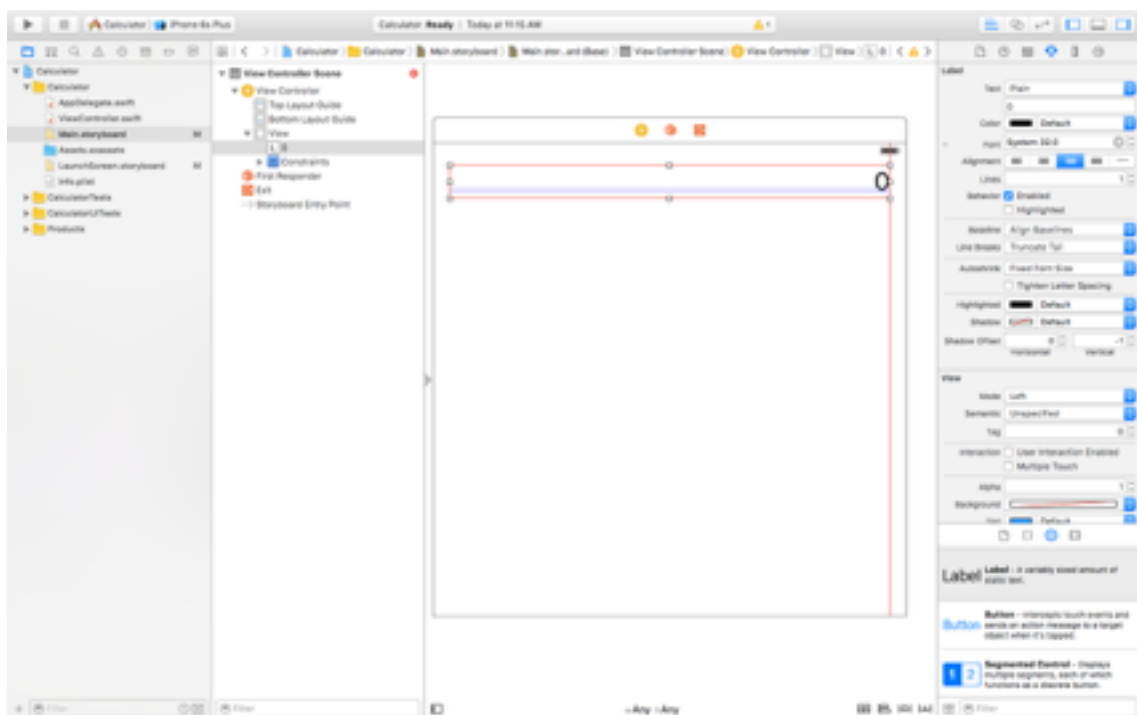
5. 选择右下方的铜钱形状的按钮，向下滚动，直到看到label，将其拖拽入View中，将label的内容改为数字0并拉至电量下方（利用蓝色虚线），点击右上角的箭头，改alignment为右侧对齐，并将字体（font中可以进行调整）调为32（拉大这个label的框就可以看到整个数字）



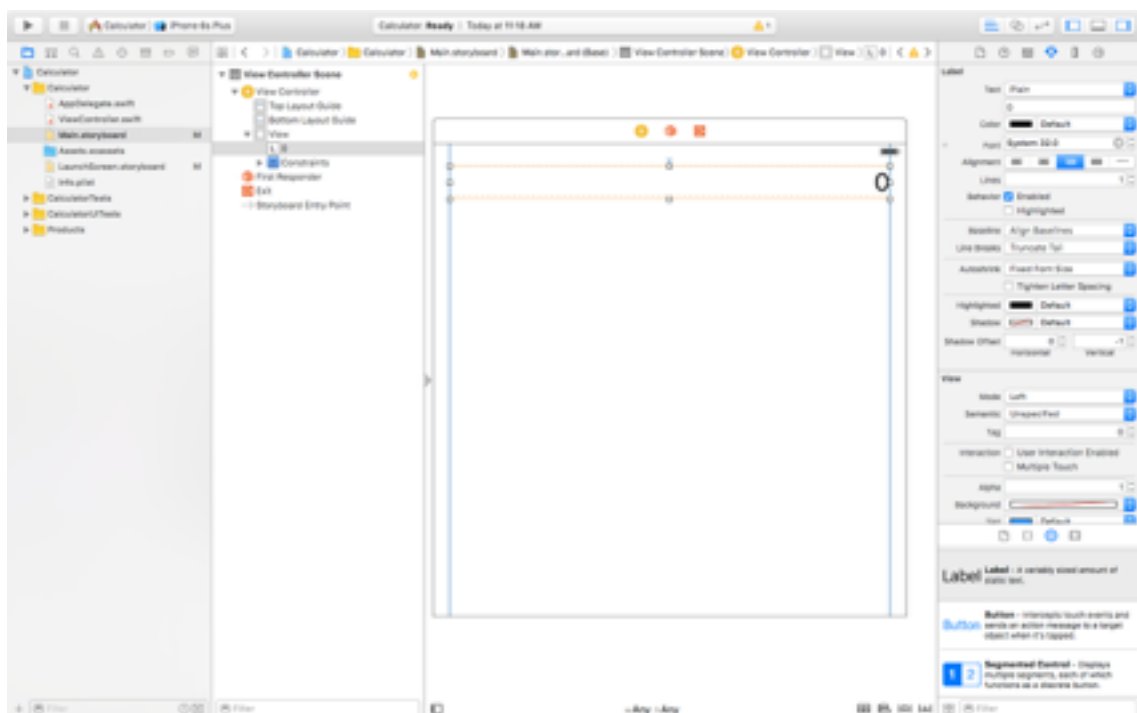
6. 按住ctrl键，然后从这个label的右边框开始拖拽到View的右边框，来给它定一条rule（选择第一条“Trailing...”）



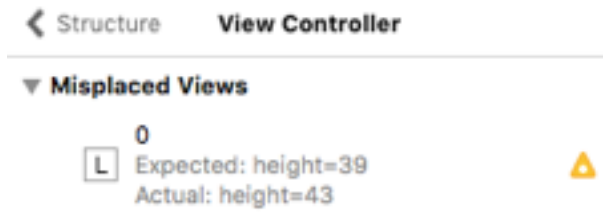
7. 完成之后，你看到的图应该是这样的



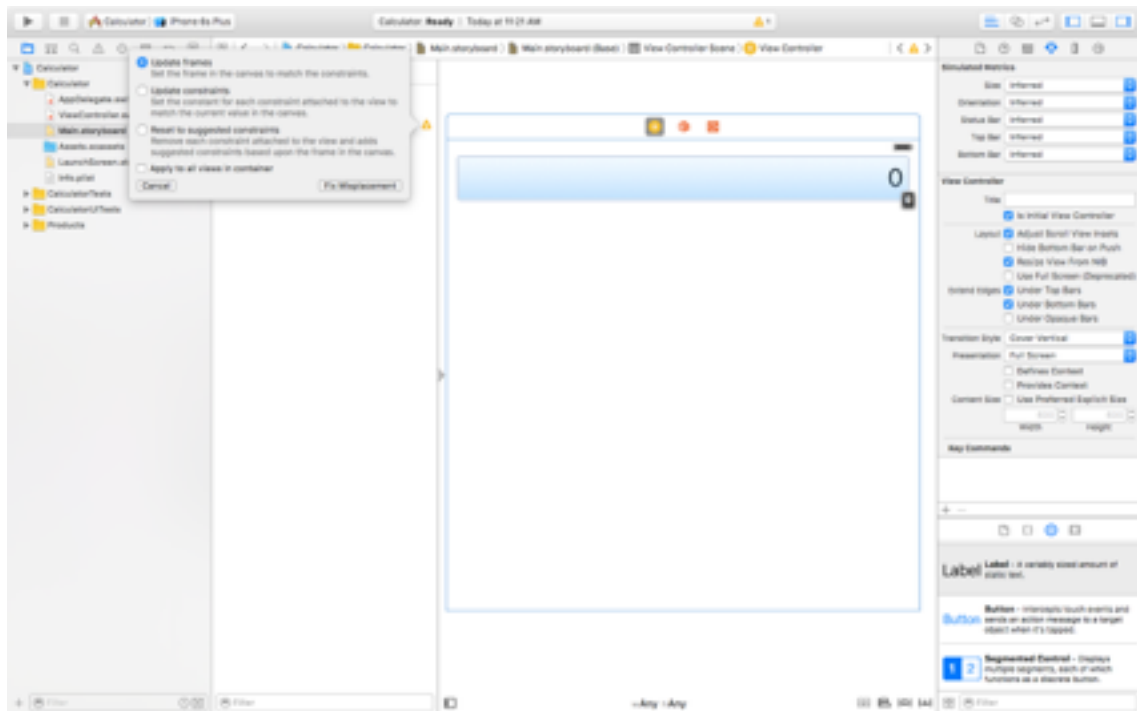
8. 那么为了使数字零永远悬于上方，用刚才提到的同样的方法，将这个label的右边，左边以及上边分别于View的这三边形成关联，你看到的页面是这样的



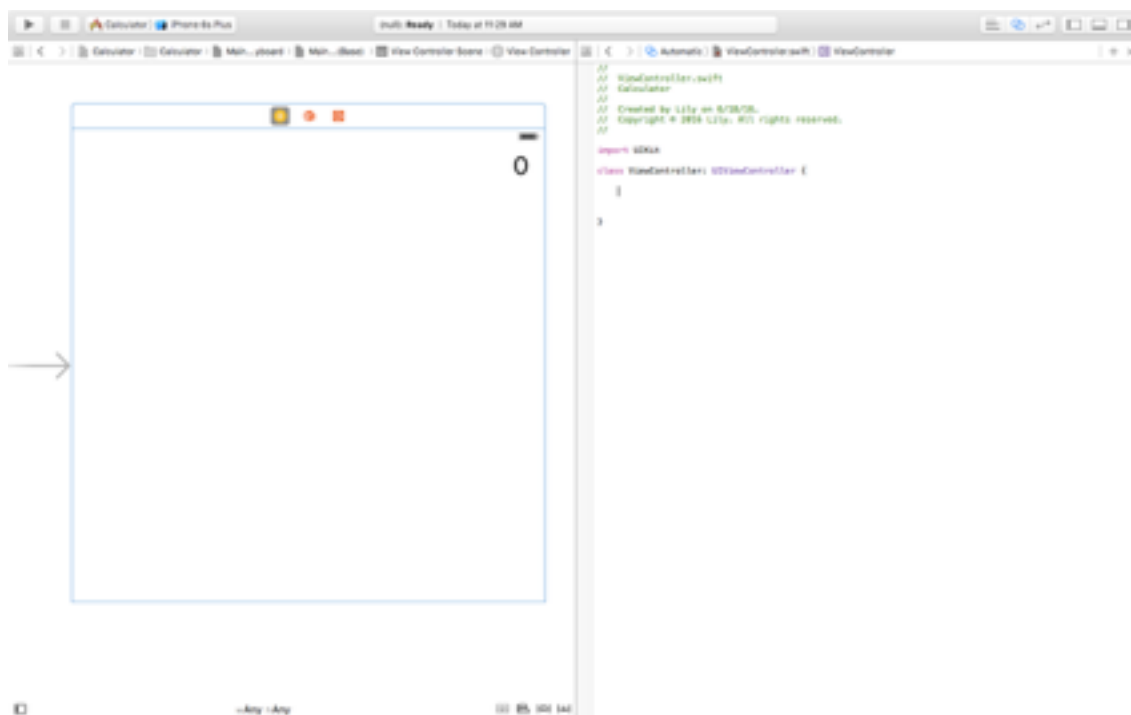
9. 注意View Controller Scene的右边出现了一个黄色的小箭头， 点击它



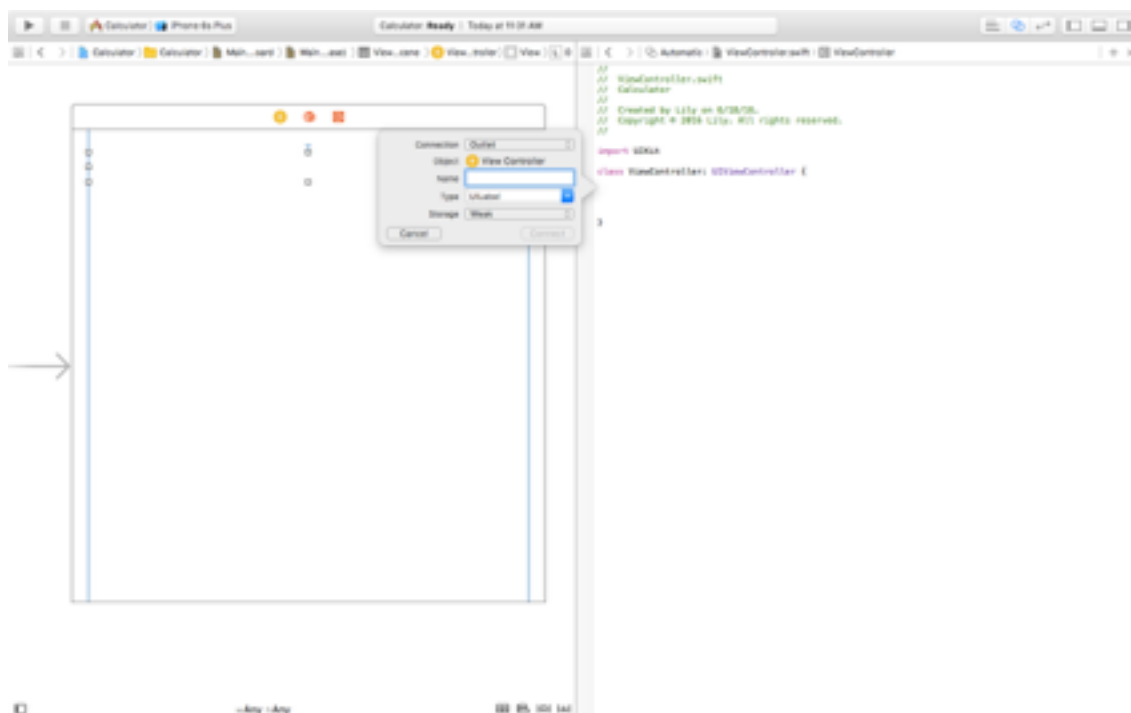
10. 然后再点击那个小三角



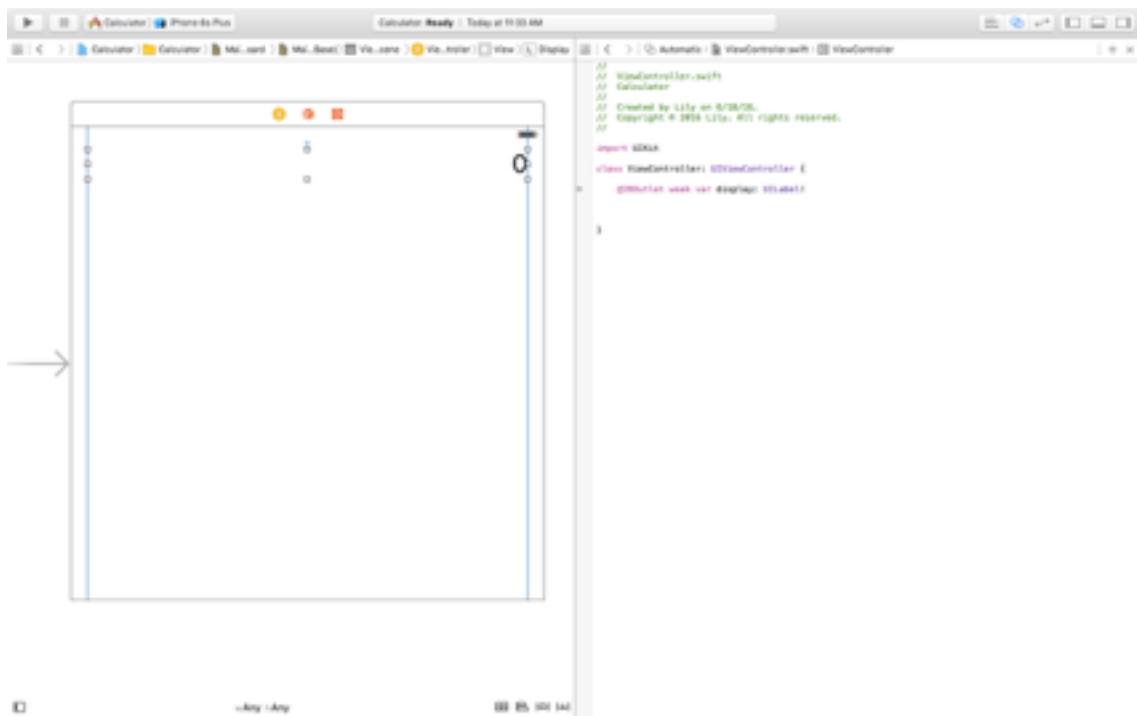
13. 首先删除class中的代码，我们的程序中用不上



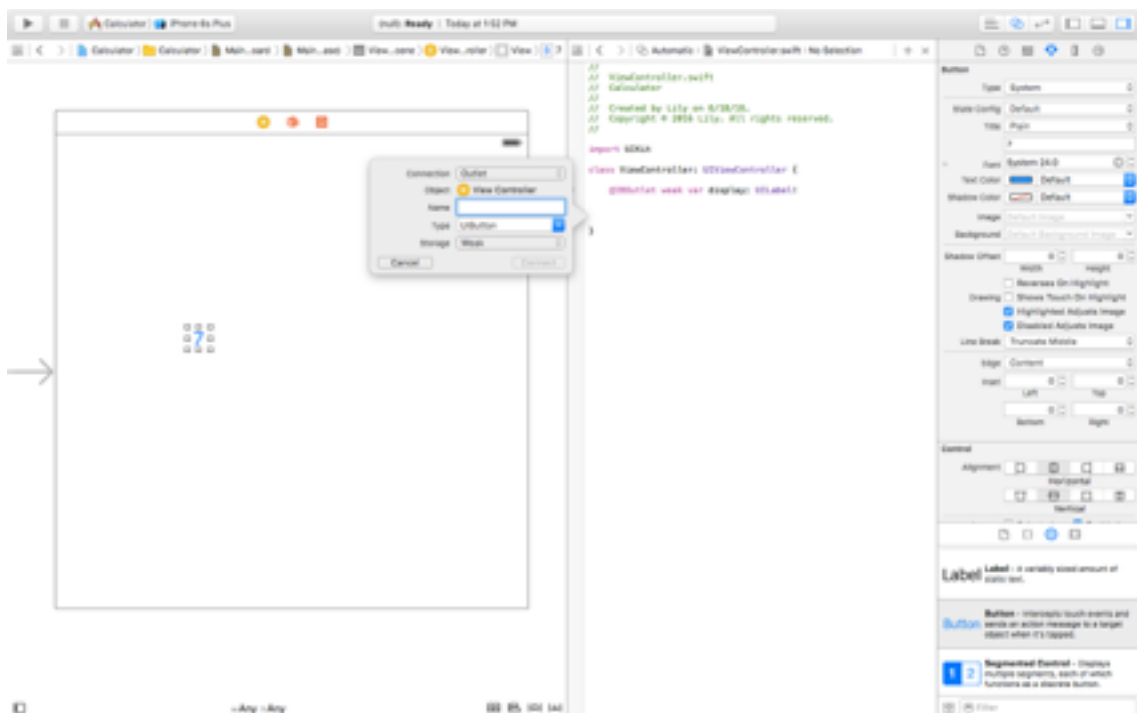
14. 接下来，再一次按住Ctrl键，然后拖住刚才那个label的右端并且连接到代码中去，如果你操作正确的话，应该会看到代码的位置出现这样的框



15. Name那一栏我们填display然后选择connect，代码就自动生成了

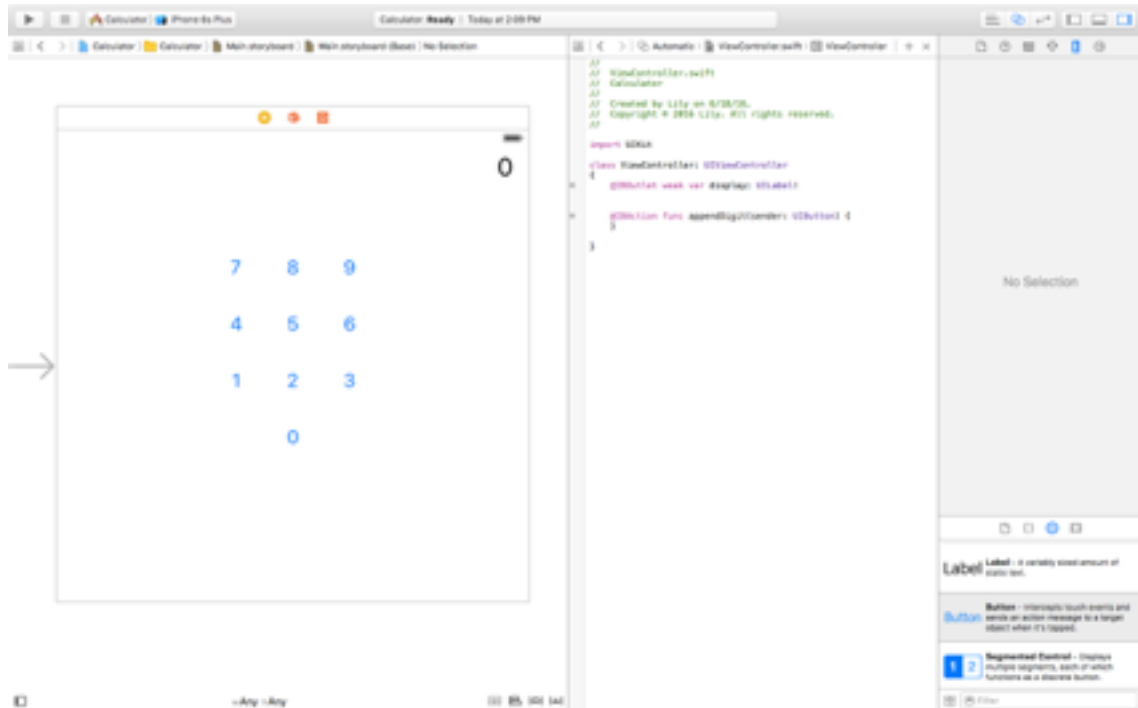


16. 接着，重新调出utility，然后在刚才获得label的地方，选择button并拖入屏幕，将此button改为数字7，然后以相同的方式将这个button与代码关联，并在attribute那一栏将字号改为24，点击右上角的尺，将长宽改为64x64

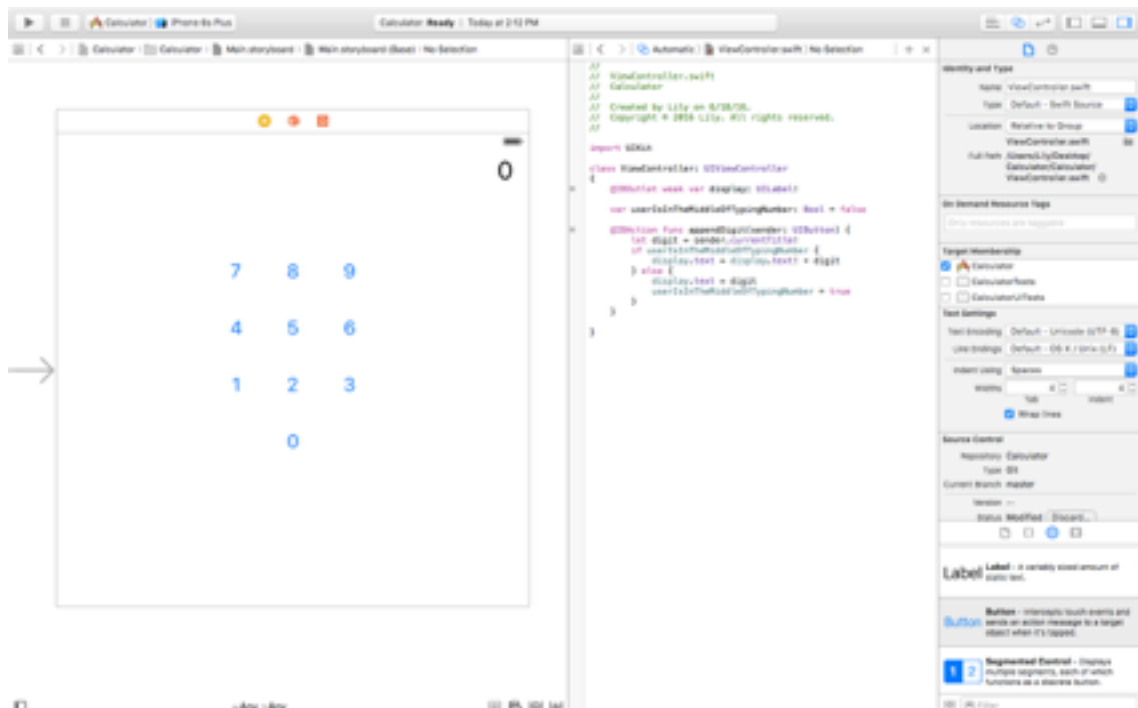


17. 选择Action，Name改为appendDigit，type改为UIButton然后确定

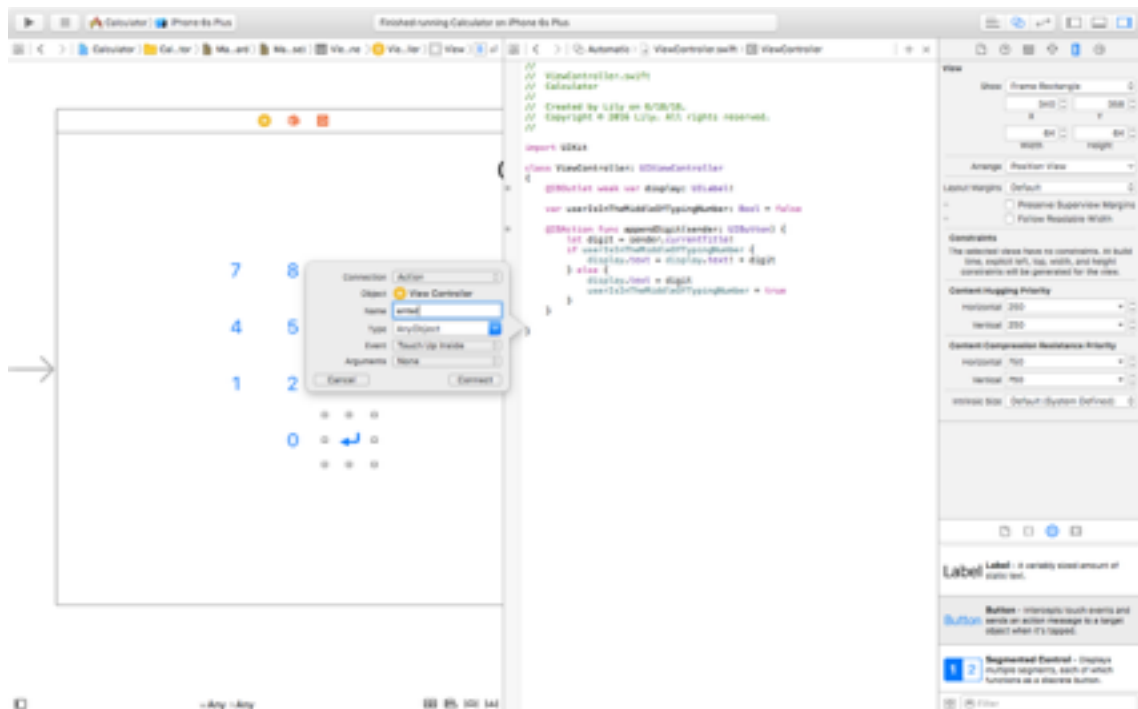
18. 复制黏贴刚才的button，完成计算器的数字部分（注意，在将数字对齐过程中应使用蓝色虚线的提示，以便之后Auto Layout的操作）



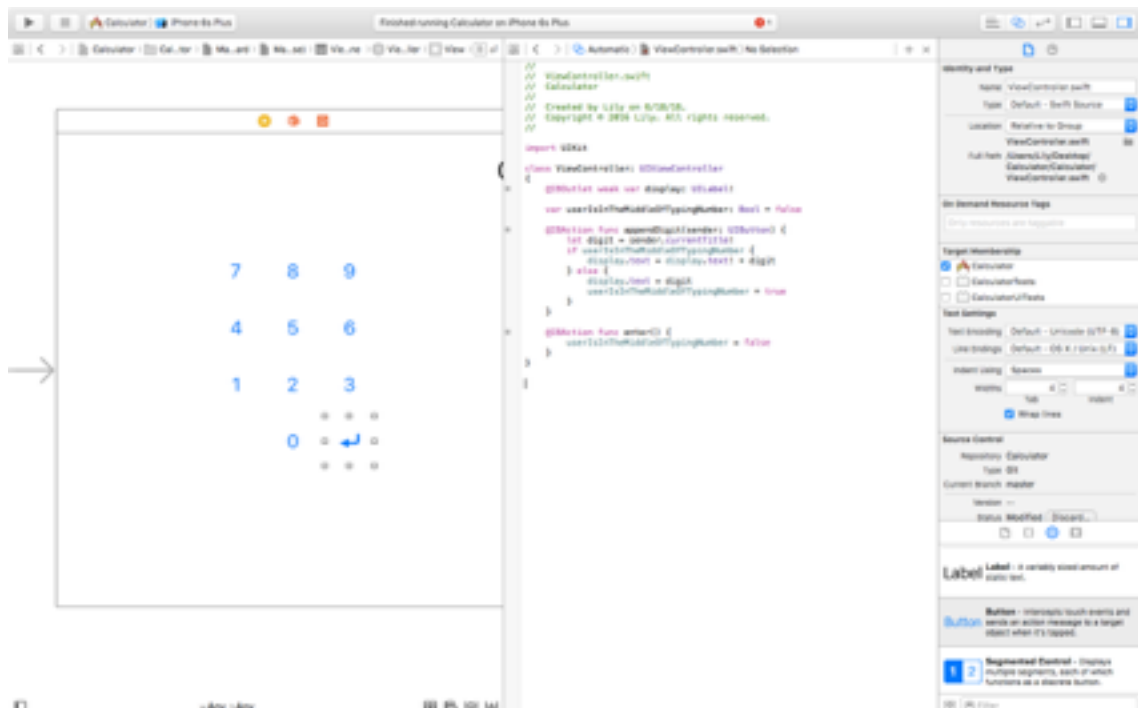
19. 在ViewController部分键入如下代码



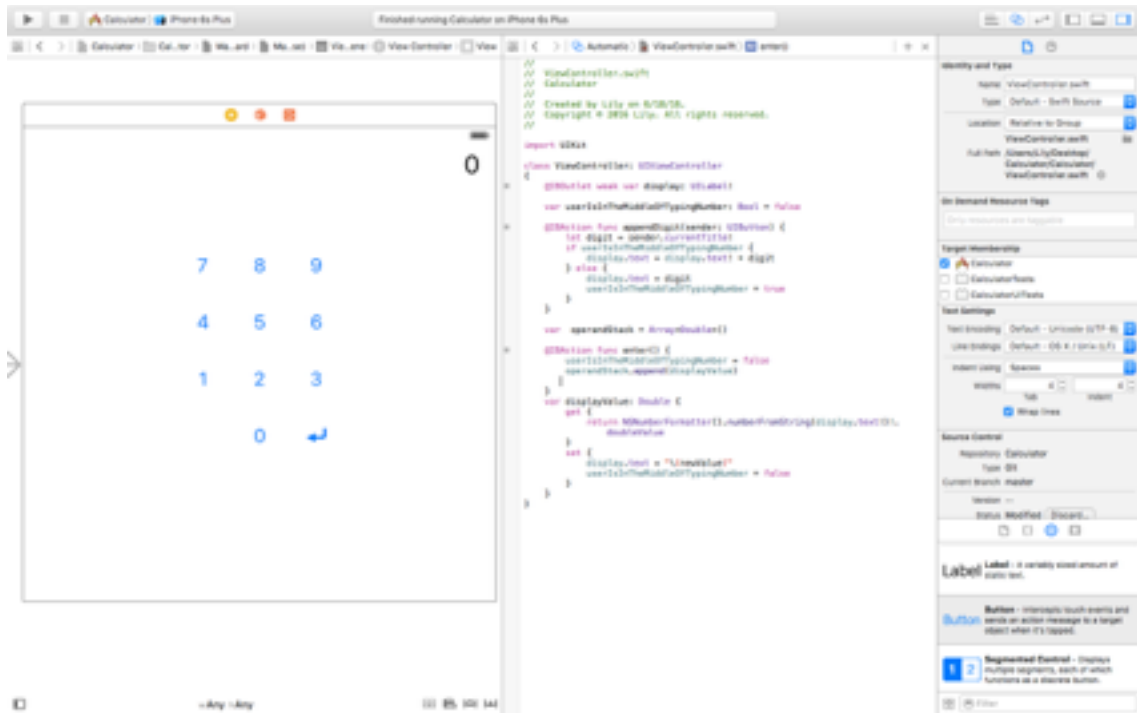
20. 再制作一个enter键（不要复制黏贴之前的按键，与代码的关联也会被复制过来）并用同样的方式生成代码



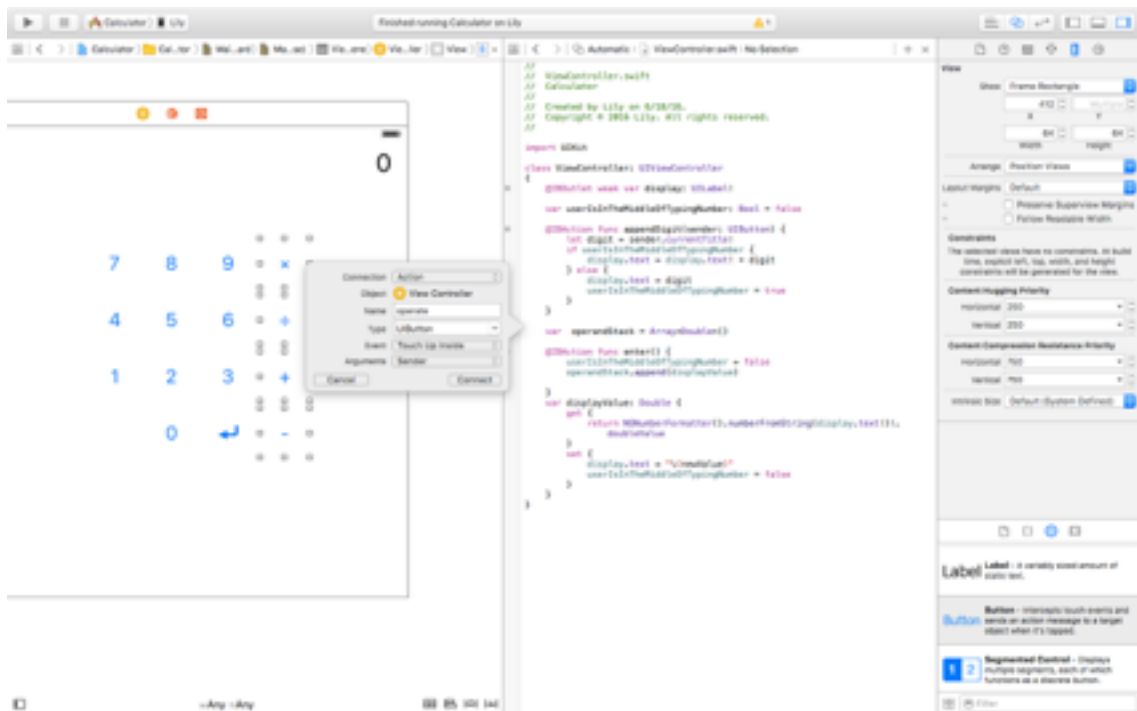
21. 再加入一行代码以后，代码就可以运行了，以上是数字的显示部分



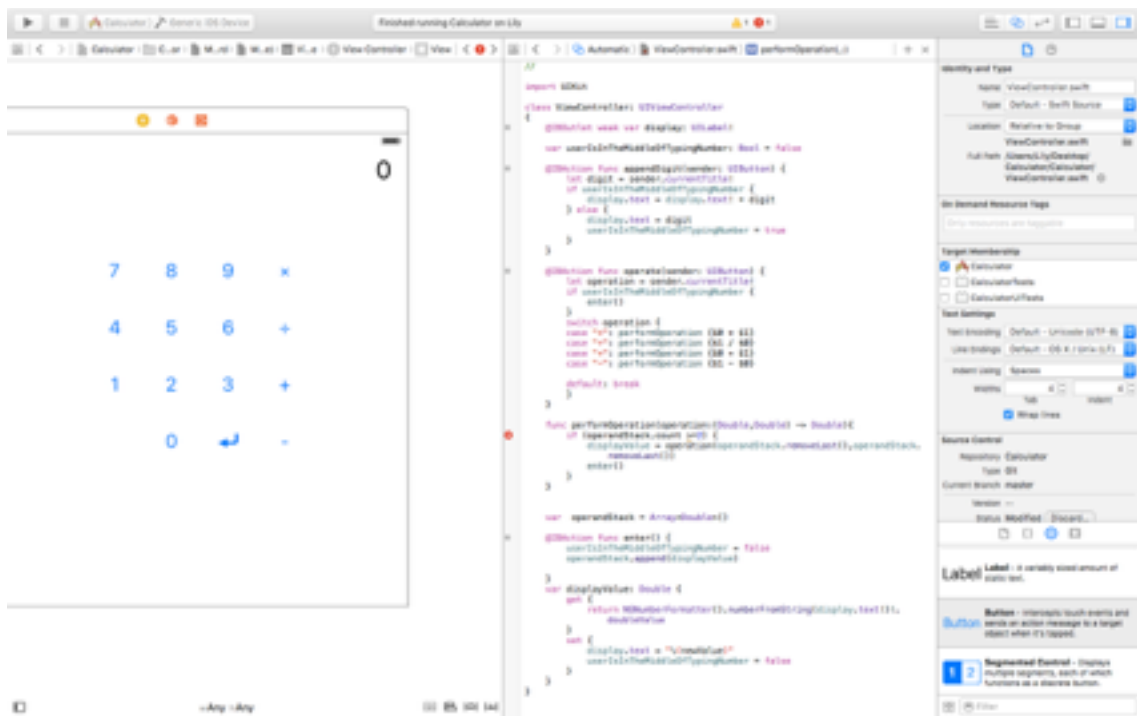
22. 接下来，我们设置一个栈来储存数据以便于我们进行运算



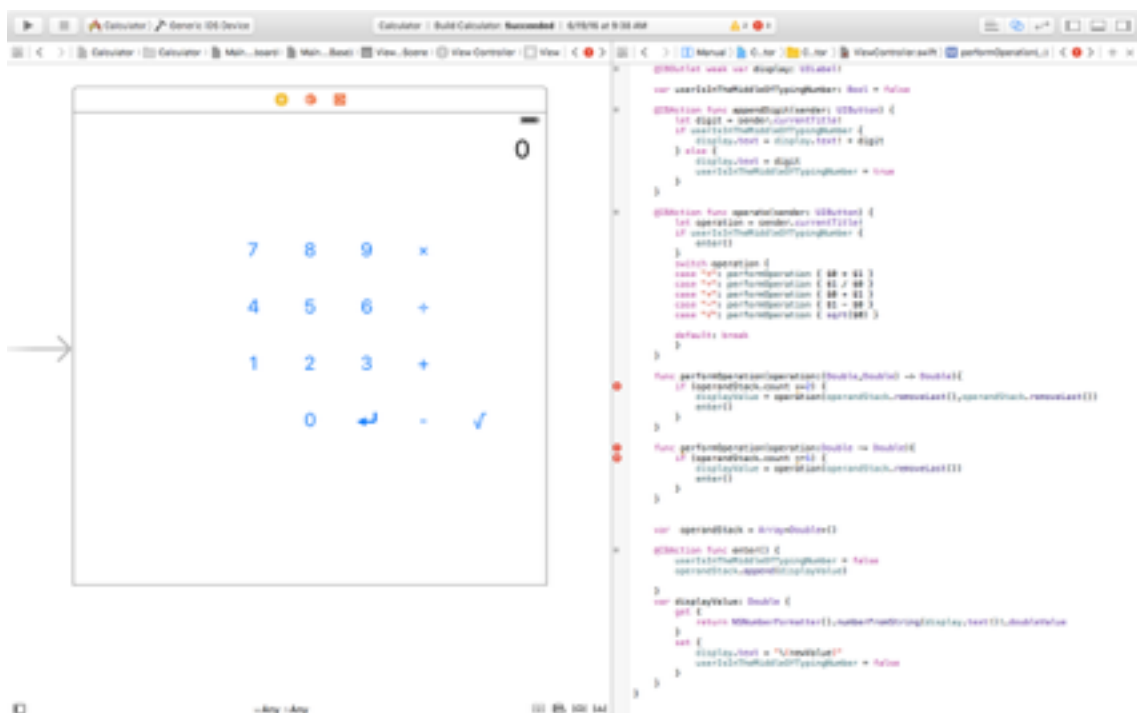
23. 制作加减乘除的按键，并且生成代码



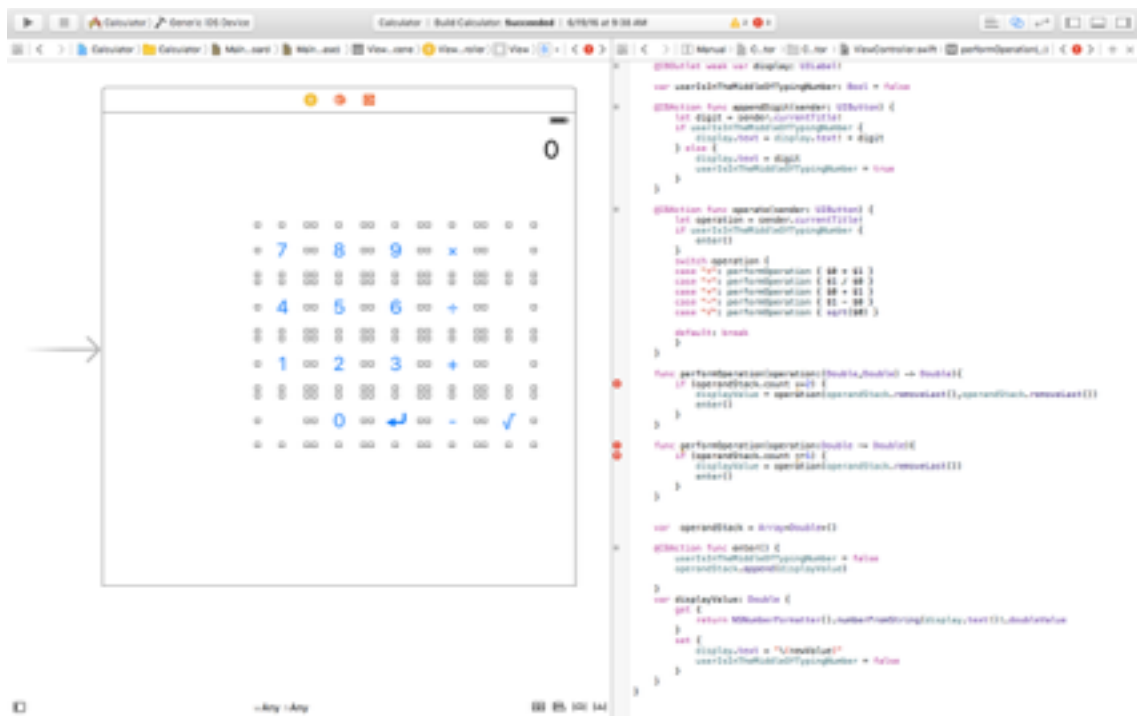
24. 接下来我们完成加减乘除的运算，代码如下：



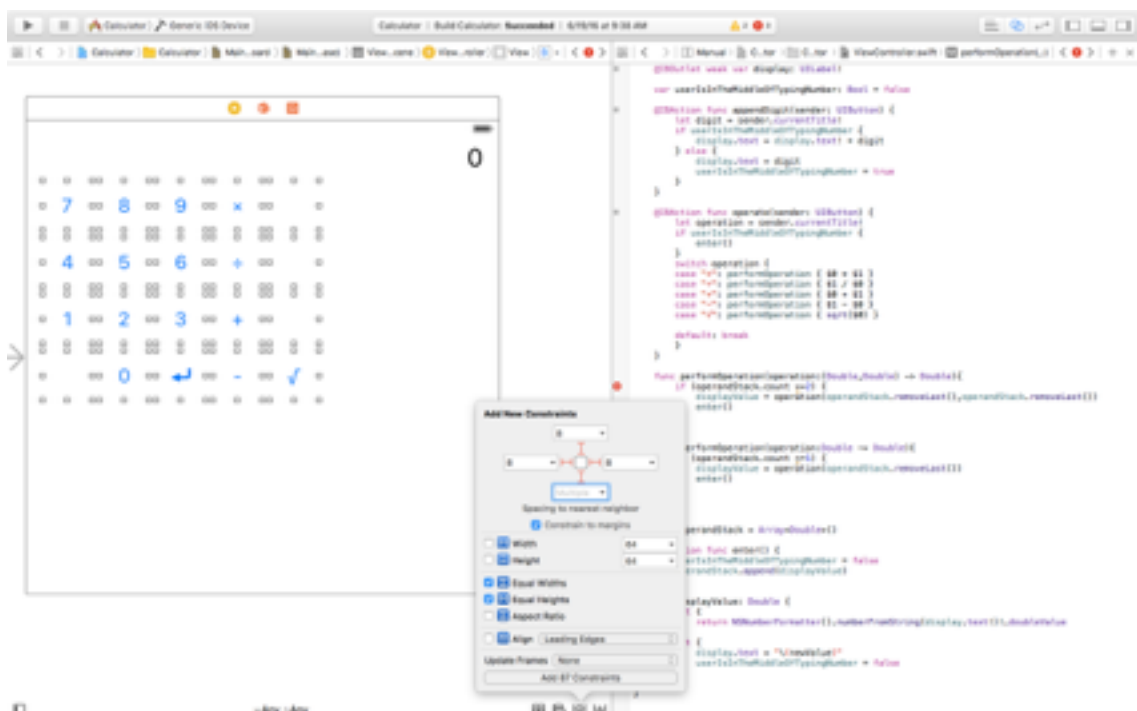
25. 接下来添加根号的运算，复制之前的按钮，并在右边的case中增加一条，在原performOperation的function之下再定义一个同样功能的function



26. 复制黏贴数字零并且将所有格子对齐成如图所示，便于AutoLayout的操作

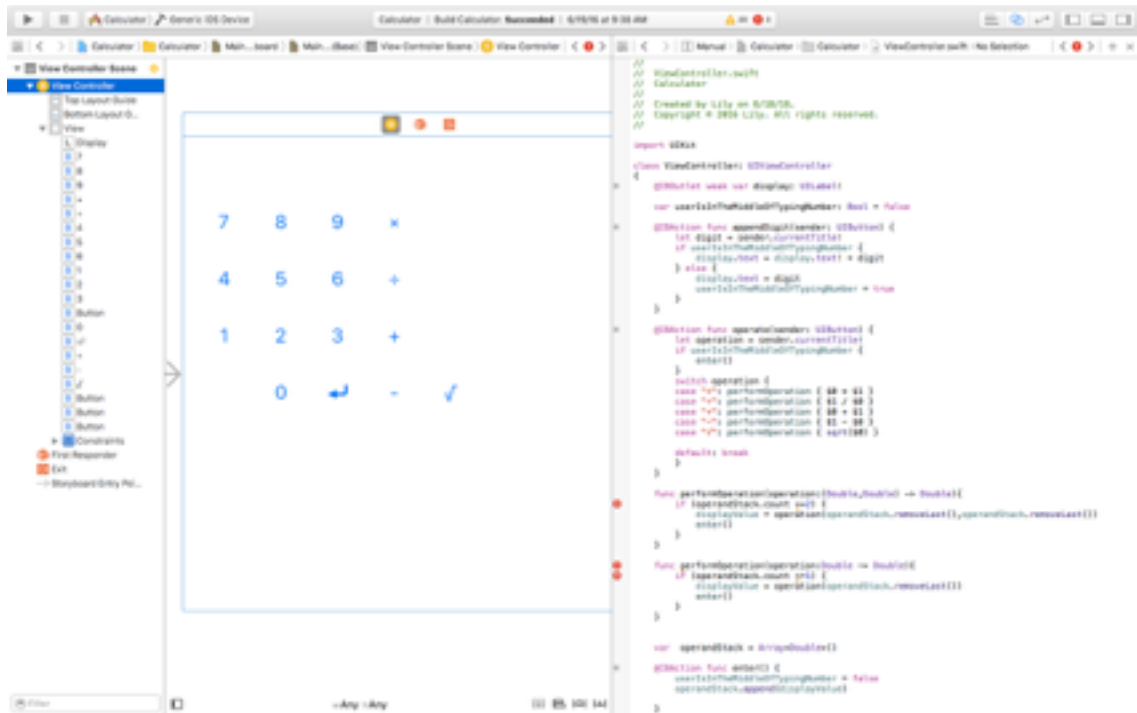


27. 将这个方阵拖到如图所示的位置（用蓝线），并且选中下图中的按钮，并填写需要用到的constraints，点击Add 87



constraints，如果数字不对，建议重做一次

28. 用相同的方法，对黄色三角标示的rules进行调整



29. 然后在其中选择Apply to all views in the container, 然后fix misplacement

