既然安装了黑苹果，那么总要做点什么的，给女儿做个app应该是一个不错的选择，看来又要开始新一轮的折腾，哈哈！

那么，既然要开发app，而且要在实体机上运行，似乎要满足2个条件，就是首先你得手机要越狱，还要有一个开发者账户，当然现在可以免账户运行了，但没有开发者账户，自然也就不能发布到app store了，当然，眼下的目标就是能免费运行就很不错了，Xcode 7应该直接就支持了，现在用Xcode6可能你还需要设置一下。

# 一、XCode6.0的iOS免证书真机测试方法

目前在XCode上开发的iOS程序只能在模拟器Simulator中运行，如果要放到真机上测试，需要苹果官方认证的开发者账号，购买开发者证书iDP，99美金一年啊！！！

作为刚开始学习iOS编程的菜鸟，这么大手笔花钱实在是肉疼。。。经过一番折腾，借鉴各种大神的帖子，终于成功在真机上编译运行成功！免费的哦！这里将方法与大家分享。

我的运行环境是：【电脑】XCode6.0.1 + OS X 10.9.5 【真机iPod Touch】iOS6.1

准备工作：

## 1. 越狱

首先，你要有一个已经越狱的iPhone/iPod/iPad，越狱方法此处不提供。

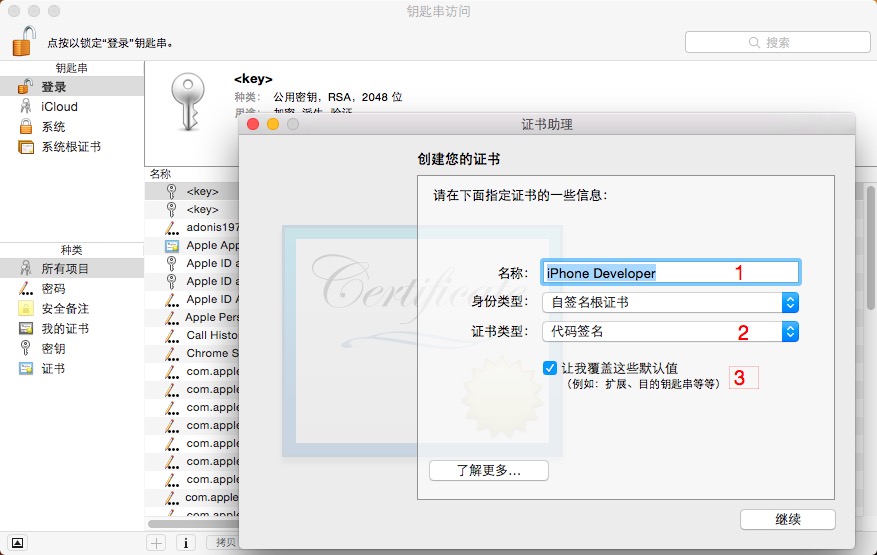
## 2. 安装appSync

在用来测试的真机Cydia中添加源：http://apt.weiphone.com，下载适合真机系统版本的AppSync。

## 3. 创建证书

在电脑上创建一个证书，用于为生成的程序签名。

创建证书的过程比较简单，在电脑上打开“其他-钥匙串访问”。然后在菜单栏里点击“钥匙串访问-证书助理-创建证书...”来打开向导。注意这里的三个地方，必须要把名称命名为iPhone Developer，将类型设定为代码签名，将”让我覆盖这些默认值”选中。之后的步骤什么都不用改，一路点击“确定”和“继续”，最后完成这个向导就可以了。



## 4. 编辑配置文件SDKSettings.plist和Info.plist

如果Xcode在运行状态，请先 Command+Q将其 完全关闭。然后打开“其他-终端”，我们需要用到一些UNIX命令，可能看起来会比较繁琐，但是相信我，这可是在每年节省99美金啊！

① 进入iPhone SDK目录，指令如下：

cd /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS8.0.sdk/

备注：不同版本的Xcode最后的版本号不同。在Xcode 6.0中是 iPhoneOS8.0.sdk ，未来版本可能版本高一些，早期版本的就低一些。请根据实际情况自己改动。（如果实在不知道自己的版本号，可以用“ls”指令查看所有文件。）

② 备份原文件，如果电脑设有密码，系统会提示你输入密码（输入后不会显示出来，敲回车即可）

sudo cp SDKSettings.plist SDKSettings.plist.orig

③ 编辑配置文件SDKSettings.plist

为了方便修改，我们可以直接用Xcode打开这个文件。首先要打开这个文件所在目录：

open .

注意这是一个osx的命令，open相当于go to的意思，“.”这个就不用多说了把，就是当前目录的意思了。

在弹出的Finder窗口中双击 SDKSettings.plist ，会启动Xcode的图形界面，展开 DefaultProperties 分支，将下面的ENTITLEMENTS\_REQUIRED和CODE\_SIGNING\_REQUIRED两个属性改为 NO。

备注：这里可能会遇到系统提示【“SDKSettings.plist” is locked for editing......】，这是因为文件读写的权限问题，解决办法如下：

首先，修改文件夹权限，此时我们所在目录是iPhoneOS8.0.sdk，如果输入指令“cd ..”返回上级目录，“ls -al”查看所有文件，会发现系统显示iPhoneOS8.0.sdk -> iPhoneOS.sdk，这表示iPhoneOS8.0.sdk是指向iPhoneOS.sdk的快捷方式，所以我们要修改的真正文件夹是iPhoneOS.sdk，指令如下：

sudo chmod -R 777 iPhoneOS.sdk

然后，修改文件夹内所有文件的读写权限：

sudo chmod 777 \*

现在双击打开 SDKSettings.plist， 你会发现，刚才不能修改的属性可以修改了！

④ 编辑另外一个配置文件Info.plist

cd /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform

sudo cp Info.plist Info.plist.orig

open .

在弹出的Finder窗口中双击打开 Info.plist 。分别在 DefaultProperties 分支下、 RuntimeRequirements 分支下和 OverrideProperties 分支下，共有3处，将全部的 XCiPhoneOSCodeSignContext 修改成 XCCodeSignContext 。

备注：同第③步，可能会遇到权限问题，修改Platforms 文件夹权限和该文件夹内所有文件的权限 即可，步骤同上。

## 5. 准备自定义的生成后脚本（此步需在联网状态下进行）

sudo mkdir /Applications/Xcode.app/Contents/Developer/iphoneentitlements

cd /Applications/Xcode.app/Contents/Developer/iphoneentitlements

sudo curl -O http://www.alexwhittemore.com/iphone/gen\_entitlements.txt

sudo mv gen\_entitlements.txt gen\_entitlements.py

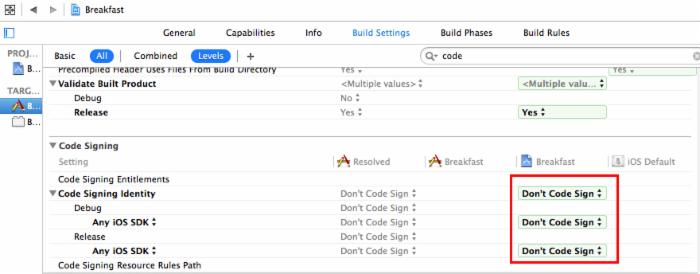
sudo chmod 777 gen\_entitlements.py

## 6. 修改工程设置

备注：本阶段之前的修改配置文件、准备脚本等，只需要做一次。但本阶段的操作，对每个需要真机调试的工程都要做一遍。

① 禁用Xcode自动的签名操作

将工程配置“Build Settings”中所有的Code Signing选项全部设为Don’t Code Sign，如图。



② 添加自定义的生成后脚本 在Build Phases中，点击左上角的”+“号，添加一个”New Run Script Phase“，输入以下脚本:

export CODESIGN\_ALLOCATE=/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/codesign\_allocate

if [ "${PLATFORM\_NAME}" == "iphoneos" ] || [ "${PLATFORM\_NAME}" == "ipados" ]; then

/Applications/Xcode.app/Contents/Developer/iphoneentitlements/gen\_entitlements.py "my.company.${PROJECT\_NAME}" "${BUILT\_PRODUCTS\_DIR}/${WRAPPER\_NAME}/${PROJECT\_NAME}.xcent";

codesign -f -s "iPhone Developer" --entitlements "${BUILT\_PRODUCTS\_DIR}/${WRAPPER\_NAME}/${PROJECT\_NAME}.xcent" "${BUILT\_PRODUCTS\_DIR}/${WRAPPER\_NAME}/"

fi

至此配置全部完成！！！连上你的iPhone/iPod/iPad，编译运行，看见真机屏幕上程序跑起来的时候，99美金就省下了！！！

# 二、Xcode教程

apple还是比较良心的企业，Xcode完全免费，并且iosSDK现在也都自动集成到Xcode中了，就是说你也不用再单独下载了，ios模拟器似乎你还需要单独下载，由于目前Xcode还没有正式发布，并且这里只是给大家介绍Xcode的基本使用方法，还是以正式发布版为准吧，并且，学会了这些东西，就是升级也无所谓了，很多东西也就是换了地方而已，至于语言的选择，你可以直接选择swift，当然你也可以选择object-c，这个还要看你的实际需要。

安装Xcode，就不多说了，你既可以下载dmg安装包，也可以直接到app store中下载，如果网络速度允许的话，还是建议你去app store下载。

打开xcode会出现欢迎界面，界面如下。接下来一个一个为大家介绍。

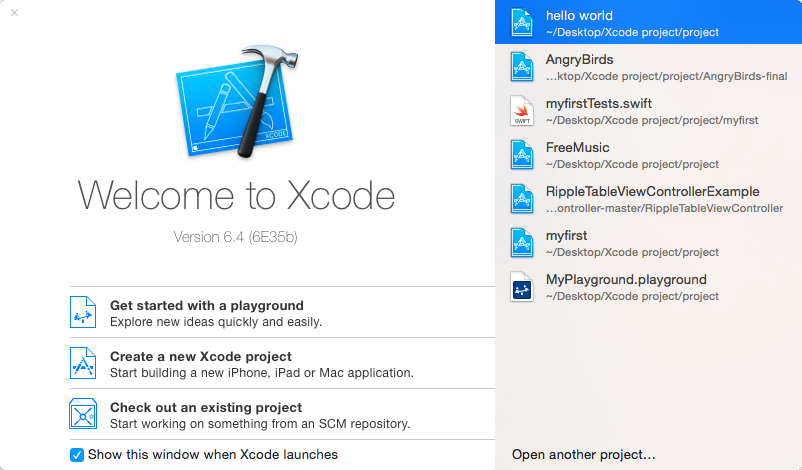
首先是左边的三个图标。

1. Get start with a playground ——这是xcode为了方便我们了解和快速上手swift特意为我们建造的”游乐场”.我们先略过这项

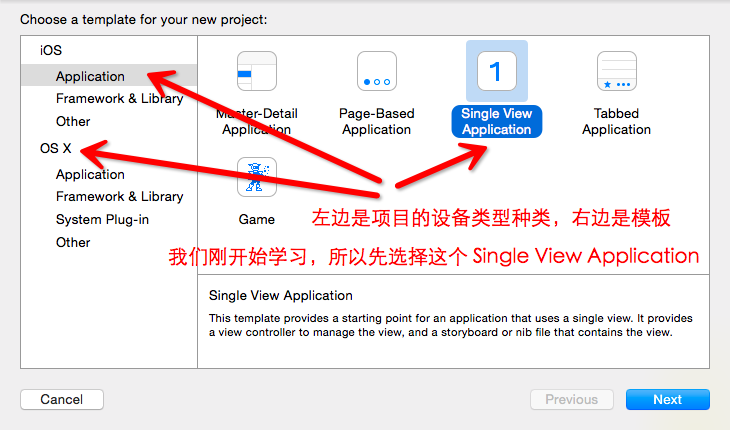
2. Creat a new Xcode Project —— 创建一个Xcode项目。 当我们打算开发一款软件的时候，一般会使用第二个选项创建一个“项目”

3. Check out an existing project —— 取回一个已有的项目。一般开发会使用svn或者git等版本控制软件来进行多人协作开发，如果你已经有svn或者git的基础，有开发人员提供git或者 svn服务权限，那么可以直接打开一个现成的工程进行研究。这一项我们在后边几章的进阶中详细介绍。

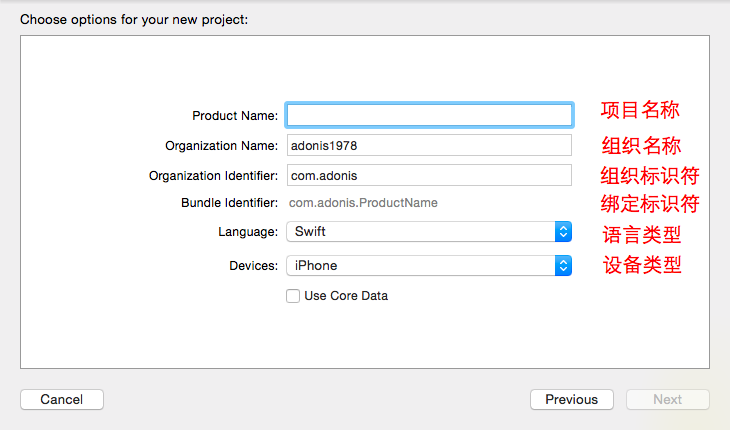
右侧的一堆文件图标，那是你打开过的历史文件记录。



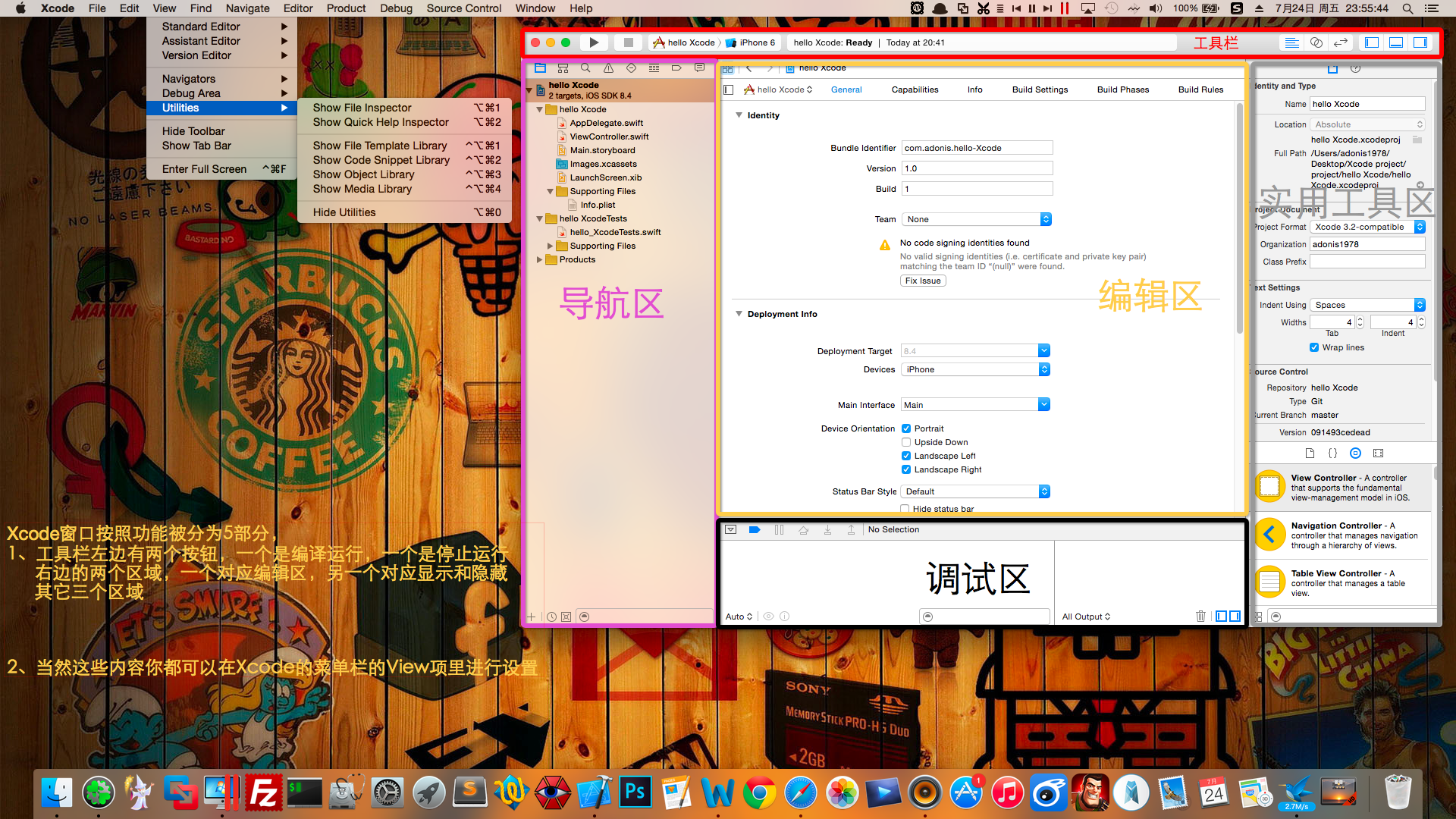
为了快速了解xcode，与这个神秘的软件交朋友，我们选择Creat a new Xcode project，新建一个软件工程，在弹出的界面中选择Single View Application，点击next继续。（其它选项我们现在用不到，这里先不做讲解。后期我们会详细介绍。）

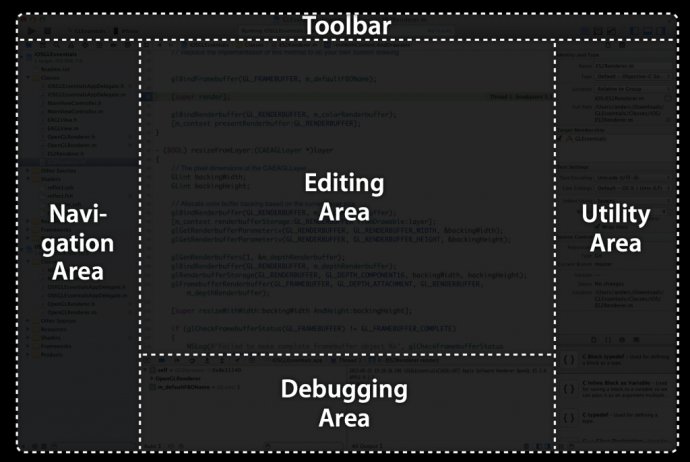


接着为我们的项目起一个名字。下面的选项不用管。直接Next，然后选择一个保存位置，这个随便选就好。

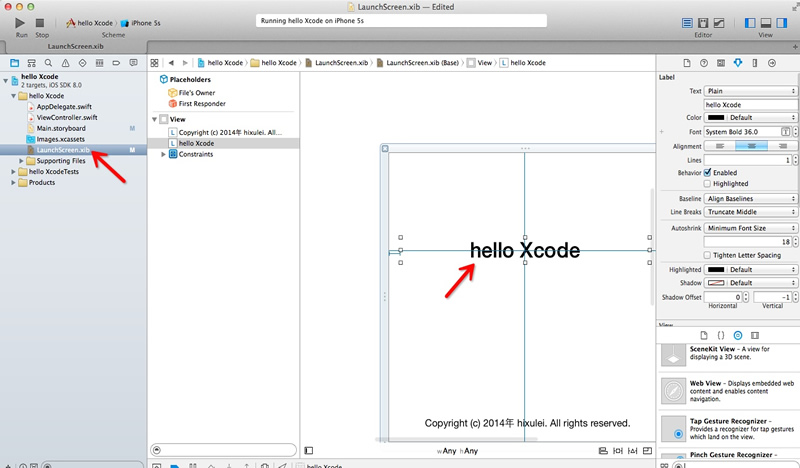


确认后我们看到如下界面。 有些朋友第一次安装可能没有上面的菜单，这个时候，在如下箭头所示的位置鼠标右键选择 icon & text。然后选择 最右边箭头，把三个全点成蓝色。 所有的界面就出现了。





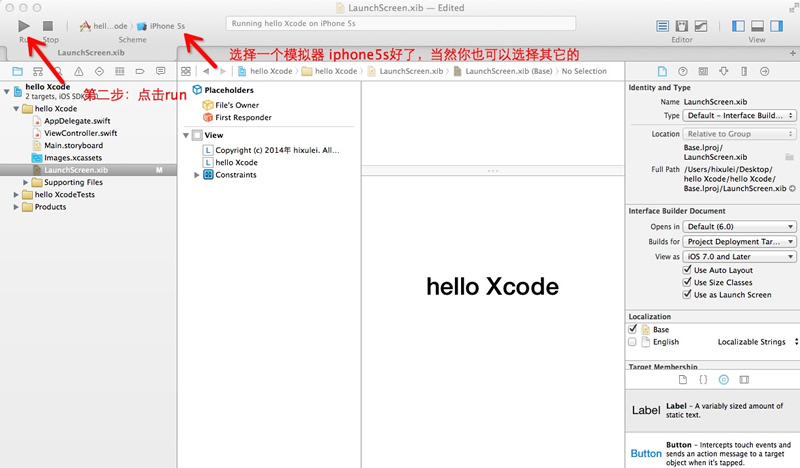
哇，好复杂有木有，大家会不会一头雾水？ 别急，我们一步一步来，选择下图所示的文件，接下来把屏幕右侧的文字随便改成自己想要的吧。



接下来按图点击

1. 选择一个模拟器

2. 点击run



如果没有什么问题的话，你会在模拟器的启动画面中看到hello xcode的字样，这样 一个空白的，有着hello xcode字样的简单app就完成了（启动画面小消失很快，记得盯着模拟器看哦。否则你会看到大白边，啥都没有）怎么样，简单吧？ 如果大家没看到，请点击一次stop，然后再运行一遍。

好的，这一次的文字，我们首先跟大家一起来熟悉一下xcode，让大家不会有太多陌生感。再次声明一下，本教程主要面向的对象是广大的app设计师 以及对软件有兴趣的朋友，专业人士请不要嘲笑静电。毕竟，这个在你们眼里，真的是再简单不过了。好的，接下来的n篇文章中，让我们来一场设计师的逆袭之旅 吧。 别被复杂的功能吓到，其实，Xcode的很多操作，都可以借助可视化界面及右侧的属性面板完成，这样很多界面的工作我们可以直接进行调整了。

下一章我们继续来讲解这个实例，让我们看看怎么在这个简单的app中加入更多好玩的功能。