让不懂编程的人爱上iPhone开发(2017秋iOS11+Swift4+Xcode9版)-第1篇

说明：本系列教程仅针对入门新手！已有iOS开发经验的请绕行~

本系列教程来源：http://www.raywenderlich.com/store

答疑说明：

因本人时间精力有限，无法跟大家一对一解决相关的技术问题，请大家在QQ交流群中互帮互助，或者去cocoachina,stack overflow,苹果官方开发者论坛，泰然网,51cto等相关技术论坛参与讨论。

联系方式：

本人网站（http://icode.ai)，邮箱(eseedo@gmail.com)，仅进行一般交流（学习方向建议指导等），具体的编程问题答疑请参考答疑说明。

适合看本系列教程的对象：

1.从未学过编程，或者对Swift语言一无所知，但要懂一些常用的英语单词

2.从未学过iPhone/iPad开发

3.喜欢苹果，充满想象力，喜欢创造，同时也愿意承受学习的压力，愿意投入时间和精力

如果你对iOS开发已具备丰富的经验，请不要在此浪费时间！

如果你只是想了解下Swift语言的开发知识，那么建议直接阅读苹果的官方文档，或另外一份教程（Swift开发入门系列教程）。

简介：

人天生就喜欢游戏，所以我们要开发的第一款应用不会是苍白无力的”Hello World”，而是一个小游戏，名为Bull’s Eye(拖拖看)。虽然这个游戏非常简单，但如果你从未接触过编程，可能还是会遇到一些困难。但是不要担心，即便你第一遍接触这些新概念的时候还有些含糊不清，但我们会在整个系列的教程中不断重复，直到它们成功的进入你的潜意识，甚至在梦中都不会忘记~

需要提醒大家的是，学习一门语言或工具的最好方式是练习和实践。因此，对于初学者来说，千万不要只是看过一遍了事，而应该自己手动敲入所有的代码，甚至故意修改其中的代码，刻意制造一些bug，然后想办法解决。而在学完本教程之后，要立即开始实战，同时多看苹果官方的示例代码和Github里的示例。不要害怕麻烦和错误，在解决麻烦和修正错误的过程中，你能更深入的领会为何要这样做，而不仅仅是简单的copy和paste。

在学习的过程中，会要求你自己做一些思考和练习，而不是完全被动的接受。仅仅知道前面有这样一条路和自己亲自走过这条路是完全不同的感觉。学习编程唯一有效的途径就是在思考的同时要自己写代码。

前面说了，本系列教程是针对完全的菜鸟来设计的。也就是说，哪怕你是个完全不懂编程的文科生，我们也有信心让你爱上iPhone开发。当然，如果你懂一点编程知识，学习起来会快很多。

处女座或强迫症患者必读：

开玩笑，处女座的童鞋们表打我~ 如果你在第一遍开教程的时候有些东西不能百分百理解，最好的方式不是立即钻牛角尖死磕到底，毕竟你没有打过很好的编程底子，这样做会让你很快丧失信心，甚至很快放弃。碰到这类情况，哥建议你先把问题放在那儿（或者记下来），然后继续看下去。等到整个教程看完了，回过头再看你之前遇到的问题，如果还是不能理解，那么再看一遍。（其实90%的可能性是，等你回头再看当时觉得比相对论和量子力学还难理解的问题，现在就是小学加减法的水平）。因为在整个教程的学习过程中，我们会对一些重要的概念不断重复，直到你觉得跟吃饭走路一样自然。

如果说Objective-C曾经是学习iOS开发的最大障碍，那么这最后一点点担忧也不需要了，因为在2014年的WWDC上苹果发布了一个新的编程语言-Swift。 它可以让开发者在很短的时间里面轻松上手，如同javascript一样，与此同时它的性能又不会比Objective-C差多少，起码是在Java之上。

对当前的主流开发语言难度排个序，大致如下（从最难到最简单）：

机器语言 >汇编> C++ >Objective-C > C, Lisp,Prolog>C# > Java > Python,PHP, Swift, Javascript, Ruby

正如刚才所提到的，对于汇编以下难度的语言，只要真正学懂一门， 再学其它的编程语言会轻松很多。毕竟在现实的世界里，真正的程序猿和攻城师很少只会一门开发语言的。

只懂一门开发语言能活到现在的要吗是某个方面的顶级专家，要吗就是走了技术转管理的路线。

在我们的教程中，不会也不可能教你学习所有和iPhone,iPad开发的知识。iOS SDK（开发工具包）非常庞大，除了苹果的官方技术文档，市面上没有任何一个教材可以涵盖iOS开发的全部内容。我们只会教你了解Swift和iOS开发所需具备的核心基础。一旦你掌握了建筑技术，可以自己去探索iOS开发的其它细节。

除了Swift语言和iOS开发工具包的相关知识，我们最重要的目的是让你学会程序猿的思维方式。一旦你具备了这种思维方式，可以完成任何编程任务，不管是游戏，工具，网络应用还是其它你能想到的东西。作为一个程序猿，需要思考解决各种计算问题，并创造性的想出解决方案。一旦掌握了解决问题的方法，不论多复杂的问题都可以解决。这才是本系列教程的终极目的，让不懂编程的人爱上开发！

可以百分百保证的是，你在学习的过程中一定会遇到各种问题。程序代码中会出现无数莫名其妙的bug，让你不知所措。但即便是一个拥有30年以上编程经验的程序猿，也会经常遇到这样的问题。我们只是人类，而人类的大脑在处理复杂计算问题的时候总会出错的。不要害怕出错，但我们会提供一些思维工具，教会你如何填平自己挖的坑。

在我身边有很多人学习iPhone开发的方式是:

从大量的博客和网站中拷贝粘贴代码，而完全不理解这些代码的工作原理，以及该如何将这些代码嵌入到自己的项目之中。从网络中寻找解决方案是一种高效的工作方式，但你必须真正的理解这些代码的作用，才能举一反三。

在本系列教程中，我们从一开始就会学习如何构建真正的应用，而不是所谓的baby应用，或是仅仅为了学习目的而设计的简单示例。我们会详细解释其中的每一步操作，并附上丰富的图片帮助大家来理解。

通过这些步骤，你将在制作这些有趣应用的同时逐渐掌握编程的思维和技能。当你最终学完本系列教程后，应该已经掌握了Swift和iOS开发工具包的精髓。更重要的是，你应该学会了如何用程序猿的思维方式来编程和解决问题，并真正开始制作属于自己的应用。对此，我有百分之一千的信心！

当然，最最重要的是，希望大家在看完教程后，能够爱上iPhone开发，爱上用编程语言来创造世界的乐趣~

那还等什么，让我们就此开始吧！

iOS7, iOS8,iOS9, iOS10,iOS11…

时光飞逝，世事无常。自2007年1月Macworld上乔帮主那一次惊天地泣鬼神的演讲至今，竟然已经10年了！

10年过去了，乔帮主的音容笑貌仿佛还在眼前，只是，人面不知何处去，桃花依旧笑春风。

10年前，Nokia藐视群雄，Motorola和三星争斗不休，众多国产品牌手机和山寨手机还在华强北幸福的收割着打工者腰包里不多的毛爷爷。

10年后，Nokia亏损连连，被Elop的木马计成功收入微软旗下，继而又宣布放弃了这块业务。

Motorola早就被Google收入帐下，当年的手机三雄竟然只剩下三星还在行业里兴风作浪。

早期混得风生水起的HTC也在最近将手机设计业务部门出售给了Google。

反观天朝，小米、魅族、华为一片混战，而采用农村包围城市战略的Vivo和Oppo手机则成功登顶国内市场。

不过回顾历史会发现，当天朝的土豪们占据市场主动的时候，就意味着一个行业发展到了高峰期，即将开始走下坡路了。比如从前的PC，比如再之前的家电，莫不如是。

苹果帝国好不容易占据半壁江山，却不幸遭遇王者的离去，在Tim Cook接手苹果之后，迟迟没有给大家提供足够的惊喜。

熟悉苹果的童鞋都知道，乔帮主总是会将一些重要的产品放在One more thing…上。

然而在Cook时代，除了数年前iPhone6发布会上的Apple Watch, One more thing…竟然从苹果发布会中消失了，直到今年。

除了硬件之外，苹果今年在技术创新了也着实拿出了新的东西。在2017年6月的WWDC上，苹果推出了针对虚拟现实开发的ARKit，以及针对深度学习和人工智能的Core ML。可以说，这两个SDK的发布，再配合iPhone X的诸多创新，确实很有可能引领未来十年的智能手机。

还是谈谈iOS11吧。

其实上面要说的是，移动互联网行业的发展速度太快了，iPhone操作系统到现在已经到了iOS11。本系列教程将完全基于iOS11，并采用全新的编程语言Swift4。

既然是学习一门全新的工具，自然要从最新的版本学起，因此，iOS11+Swift4+Xcode9是我们的第一选择。

有舍才有得

学习iPhone开发不但可以作为一门兴趣爱好，同样还可以带来不错的收益（如果你能给用户带来不错的产品，或者找到一份提供给力薪水的工作~）。但天下没有白吃的午餐，进行iOS开发也是要花钱的。以下是你需要投资的：

1.一台iOS设备：

iPhone,iPad,iPad Pro，iPad mini或iPad Touch中的任一种。只用虚拟机永远没法学会真正的开发。当然，为了支持最新的iOS系统，就不要买太老型号的设备了，比如iPhone4之类的。

因为本系列教程将使用iOS11，所以大家也要了解下支持iOS11的设备：

iPhone:

- iPhone 7
- iPhone 7 Plus
- iPhone 6s
- iPhone 6s Plus

iPhone 6

iPhone 6 Plus

iPhone SE

iPhone 5s

iPad:

12.9-inch iPad Pro (2nd generation)

12.9-inch iPad Pro (1st generation)

iPad Pro (10.5-inch)

iPad Pro (9.7-inch)

iPad Air 2

iPad Air

iPad (5th generation)

iPad mini 4

iPad mini 3

iPad mini 2

当然，如果你希望在自己的应用中提供虚拟现实（ARKit）或人工智能（Core ML）相关的特性，那么就必须使用支持A9、A10或A11芯片的设备，具体来说是这些：

- iPhone 6s /6s Plus
- iPhone 7 / 7 Plus
- iPhone SE
- iPhone 8/ 8 Plus
- iPhone X
- iPad Pro
- iPad (2017)

个人建议直接购买可以支持ARKit的设备。

2.一台使用Intel内核处理器的Mac电脑：

需要安装最新的macOS High Sierra。建议电脑的内存在4G以上，否则你会很痛苦的。。。

有人说可以用虚拟机在PC上开发，我的建议是，宁可买一台二手的MAC，也不要用PC开发。否则，你会遇到各种莫名其妙的问题，而且对提高你的编程思维没有任何帮助。

对一般的入门开发者来说，如果不是同时还身兼设计师的重任，那么一台13寸的Macbook Air或Macbook Pro就可以满足要求了。

3.一个付费的iOS开发者账号

有了开发者账号，最简单的装B方式就是当苹果发布了新的beta版系统时（特别是大版本的更新，比如从iOS9到iOS10），你可以第一时间在自己的iOS设备或者电脑上体验。

当然，最重要的是可以将自己开发的产品放到苹果AppStore里面销售，赚取真金白银，当上CEO赢取白富美从此走上人生巅峰。

如何申请付费的iOS开发者账号

狠简单，直接到这里就可以了：

https://developer.apple.com/programs/ios

整个过程其实很简单，不过你需要先注册一个Apple ID，因为你的开发者账号会与之绑定。其实一年99美元的费用并不高，如果你是真心想学iOS开发，建议还是花了这笔钱。而且苹果现在相当厚道的一点是，只要加入一个开发者计划，就可以为iOS设备，Apple Watch, Mac,Apple TV等苹果全系列硬件产品开发应用。而在此前iOS开发和Mac开发是两个完全不同的开发者计划，每个都要单独收费的哦~

当然，如果朋友们实在是吃了上顿没下顿，或者说只是想观望下，还不想花一毛钱。那么你也可以直接用自己的Apple ID来登录，同样可以免费使用Xcode开发工具，也可以在设备上进行测试。但是没法获得最新的Beta版本，更不可能将产品发布到App Store里面。

强大的Xcode

Xcode是开发iPhone应用的主要工具。Xcode带有一个文本编辑器，可以让你敲入自己的代码，同时还有一个可视化的工具由来设计应用的用户界面。Xcode可以将你编写的源代码编译成可执行的应用，并在模拟器(Simulator)或设备上进行测试。同时，Xcode还带有一个debugger(调试器)，用于帮助你发现代码中的错误（很遗憾，目前它还没法自动帮你修复bug，这一天的到来还需要更给力的人工智能)

下载Xcode的方法很简单，直接从这个链接下载：

http://itunes.apple.com/app/xcode/id497799835?mt=12

当然，最通常的做法是在Mac App Store里面搜索Xcode，然后下载安装就好了。

再次提醒，下载Xcode，Unity和Cocos2d-x等开发工具时一定要从官网下载，且只使用最慢的http下载链接下载，或使用官方的下载工具（比如Unity)，不要使用任何第三方下载工具（比如迅雷之类的）。

如果你的操作系统不是macOS High Sierra，那么请先升级操作系统。

本系列教程用的Xcode版本是最新的Xcode 9.0(9A235)，而iOS版本是10.3。

很多老的iOS教程还在用Xcode和iOS之前的版本，所以当你阅读那些教程的时候，请注意开发工具的细节差异。

关于计算机语言

语言是一种沟通工具。很多时候我们以为iPhone只是一部手机，其实它的内核是一个非常先进的微型计算机，只是同时具备打电话的功能而已。和其它计算机一样，iPhone是通过数字电路的0，1指令来工作的。如果我们编写软件在iPhone上运行，就必须把源代码翻译成计算机可以理解的0，1指令。

几十年前，人们不得不使用0，1指令和计算机直接交流。而随着汇编和高级语言的出现，大多数的编程语言变得更接近于日常生活所使用的英语。这样一来，人们更容易理解编程语言的使用。但同时也需要将人类可以理解的语言翻译成计算机可以理解的0，1指令。

举例而言，计算机内部会使用以下的语言：（不要关注其中的细节，你现在还看不懂）：

```
Ltmp96:
    .cfi_def_cfa_register %ebp
    pushl   %esi
    subl    $36, %esp
Ltmp97:
    .cfi_offset %esi, -12
    calll   L7$pb
L7$pb:
    popl    %eax
    movl    16(%ebp), %ecx
    movl    12(%ebp), %edx
    movl    8(%ebp), %esi
    movl    %esi, -8(%ebp)
    movl    %edx, -12(%ebp)
    movl    %ecx, (%esp)
    movl    %eax, -24(%ebp)
    calll   _objc_retain
    movl    %eax, -16(%ebp)
    .loc    1 161 2 prologue_end
Ltmp98:
    movl    -16(%ebp), %eax
    movl    -24(%ebp), %ecx
    movl    L_OBJC_SELECTOR_REFERENCES_51-L7$pb(%ecx), %edx
    movl    %eax, (%esp)
    movl    %edx, 4(%esp)
    calll   _objc_msgSend_fpret
    fstps   -20(%ebp)
    movss   -20(%ebp), %xmm0
    movss   %xmm0, (%esp)
    calll   _lroundf
```

事实上，计算机真正看到的指令如下：

```
00011001010011110100100011001111100101O
00101000100111101011011100111010110100l
01010001110011111010111011000011100011O
10010000011100010100110100111110011001l1
```

上面的movl和calll指令只是为了方便人类理解。但即便如此，对我个人来说这种语言还是令人望而生畏。

今天的编程语言是下面这样的（先不要深入细节，看看而已）：

```
void HandleMidiEvent(char byte1, char byte2, char byte3, int deltaFrames)
{
char command =(byte1 & 0xf0);

if(command == MIDI_NOTE_ON && byte3 !=0)
{
      PlayNote(byte2 + transpose, velocityCurve[byte3]/ 127.0f, deltaFrames);
}
elseif((command == MIDI_NOTE_OFF)
       ll  (command == MIDI_NOTE_ON && byte3 ==0))
{
      StopNote(byte2 + transpose, velocityCurve[byte3]/ 127.0f, deltaFrames);
}
elseif(command == MIDI_CONTROL_CHANGE)
{
if(data2 ==64)
          DamperPedal(data3, deltaFrames);
elseif(data2 == 0x7e ll data2 == 0x7b)
          AllNotesOff(deltaFrames);
}
}
```

看到这里或许你才有点感觉了。即便你没有任何编程经验，但只要懂英语，就大概能判断出上面代码的意思。以上代码是从一个音效同步工具的程序中截取的。它使用C语言编写，这门语言是上世纪60年代开发的，人们用它开发了著名的Unix操作系统（今天所有操作系统的鼻祖，包括Windows,Mac,Linux）。当然，iOS的内核也是基于Unix系统的。

而这里我们要着重提一下苹果的新编程语言Swift。Swift集成了传统面向对象编程语言的特性，同时又具备函数式编程的一些特征。如果你曾经学过C#,Python，Ruby或者JavaScript，那么会发现Swift有很多相似之处，很容易上手。

在2014年WWDC之前，用来开发iOS应用的语言被称为Objective-C，它是标准C语言的扩展。使用Objective-C可以完成C语言所能完成的任何工作。同时它还添加了很多有用的特性，比如最重要的面向对象编程（Objective-Oriented）。Objective-C在前些年可谓门

庭冷落，无人问津，除了铁杆的Mac粉丝，几乎濒临灭绝。但随着2007年那一次伟大的iPhone产品发布后之后，几乎要被历史遗忘的Objective-C语言再次进入人们的视线，甚至成为今的主流开发语言。Objective-C是2012和2013年的年度编程语言No.1。

目前仍然有大量的iOS项目使用Objective-C开发，毕竟一个新的编程语言普及需要几年甚至更长的时间，不过iOS开发的未来显然是属于Swift的。

这里不得不提到C++语言，事实上C++和Objective-C语言几乎是同时出现的。和Objective-C语言的简洁不同，C++语言几乎包含了所有可能的特性。作为一门编程语言，它非常强大，且执行效率超高。事实上，所有的操作系统，以及大量的网络游戏，主机游戏和PC游戏，游戏引擎都会使用C++来开发。C++的问题在于，对于一个新手来说，它异常复杂，包括了基本语言结构，面向对象开发和模板、标准库等诸多内容。学习C++还是颇有难度的，仅次于汇编语言。不过C++11（2011年的新标准）这一C++的最新版本在很多方面做了大的改进，相信会让这门“古老”而又强大的编程语言更加褶褶生辉。

在进行iOS应用或游戏开发的时候，我们可以混合使用C,C++和Objective-C（简称为Objective-C++）。

此外，Facebook在2015年开源的React Native可以使用Javascript开发原生的iOS和Android应用。但是目前因为授权的问题，React受到众多开发者的抵制。所以虽然很多公司的项目用的是React，作为新手的你也可能经常听人提到，但是，谨慎考虑是否用它。

对于iPhone手机游戏开发来说，由于Cocos2d-x引擎的迅速普及，C++的使用频率也大大增加。此外，主流的3D商业引擎Unreal Engine4使用的就是C++语言。而另一个主流的手游开发引擎Unity则主要使用C#和Javascript脚本语言。

总之，对于iPhone应用开发来说，最主要接触的语言是Swift和Objective-C，部分情况下也会用到C++和C，以及javascript等脚本语言。

考虑到本教程的很多读者从未接触过任何编程语言，这里对其它几个主流语言的特点和作用稍微说明一下：

1.Java语言是当今最普遍使用的开发语言，它简单易学（相对C++,C和Objective-C），且跨平台性非常强，对网络开发的支持令人称赞。很多企业使用Java语言来开发商业相关的网络应用。此外，Java语言也是开发Android应用的必备工具。

2.C语言是几个主流开发语言(Java,C++,C#,Objective-C)的根基所在。常有人说，学好C语言，其它的语言就会一通百通。因为对硬件底层性能的支持超强，它的主要应用领域是

嵌入式开发、游戏引擎开发等偏底层的部分。C语言基本上已经取代了汇编语言和机器语言在底层开发的作用。

3.PHP语言主要用于开发网络应用（特别是web服务器端，也就是用户不可见的部分，如结合MySQL进行后台数据传输处理等），相对其它几门语言，它非常容易上手。但它的局限性在于除了web应用，对其它应用的开发力不从心。

4.Javascript语言主要用于开发Web前端（也就是用户可见的部分），随着HTML5技术的兴起，Javascript语言必将是未来三到五年的主流Web开发工具。在主流的3D游戏开发引擎Unity中，同样支持使用Javascript作为脚本编程语言。此外，Facebook的开源项目React Native可以让开发者使用Javascript语言轻松开发原生的iOS和Android应用。

5.C#语言是微软为了对抗Java语言的强势而自行开发的一种编程语言。它和Java一样简单易学（同样是相对的），但只能支持微软的平台。闻名业界的.NET就是C#语言的最佳搭配。但随着微软在移动互联网领域的式微，C#的地位和前几年比起来大有下降。

不过虽然.NET和C#在Web开发领域的空间越来越小，但C#也有新的应用领域，目前最火爆的移动平台3D游戏开发引擎Unity3D主要支持C#和javascript开发，而windows手机平台的卷土重来也让C#有了新的机会。

6.Python,Ruby,Perl同PHP语言的作用类似，属于脚本语言，对于开发网络应用非常高效。其中Python和另一种脚本语言Lua还常在游戏中作为脚本语言使用。

对于Python要特别补充一点，在2016年下半年开始异常火爆的AI（人工智能）技术中，特别是深度学习等领域，科学家和开发者大量使用Python语言和相关框架。曾经的超轻量级脚本语言在一个超重量级的应用领域重获新生，而且越来越受人欢迎。

7.Go语言，一门全新的系统级语言，由Google开发，于2009年发布。虽然它的历史非常短暂，但根据目前的发展来看,Go语言有望在未来十年成为一款成功的系统级语言。Go语言功能强大，可以替代C++。

在TIBOE2017年最新的编程语言排行榜上，Go语言已经从去年的38位急速攀升到14位。

8.Basic（Visual Basic)语言，曾经风骚一时，若干年前很多编程入门课程必教的开发语言。其学习曲线非常平缓，易于上手，但实际项目中用到的不是很多。

9.SQL语言，这是目前最重要的关系数据库操作语言，其影响已经超出数据库领域，在很多其它领域得到采用，比如人工智能领域的数据检索，软件开发工具中嵌入SQL的语言等。SQL语言是一种交互式查询语言，允许用户直接查询存储数据，但它并不是完整的程序语言，没有DO或FOR类似的循环语句，但可以嵌入到另一种语言中，通过接口发送到数据库管理系统。

10.汇编语言，虽然现在是高级编程语言的天下，但性能超强的直接面向硬件的汇编语言仍然在嵌入式开发领域占据着一席之地。只是汇编语言和硬件本身的关联很大，所以普及性一般。目前汇编语言的江湖地位已被C语言替代，但在某些领域还有自己的一席之地。

其它语言相对来说比较冷僻，或者曾经热门但如今使用的人很少，用不到的时候可以不管。

为了让大家对各种编程语言的江湖地位有所了解，不妨看看TIOBE 最新的2017年9月编程语言排行榜~

可以很明显的看到，2014年9月的时候，苹果的御用开发语言Objective-C是Top3的编程语言。但是自从2014年苹果WWDC上发布了新的Swift语言后，Objective-C的排名迅速下降到如今的Top18，而Swift则上升到了Top13。

科普知识到此结束，我们不打算对Swift语言的特性做详细的介绍，不然很可能5分钟不到你就睡着了。我们将在创建项目的过程中一步步解释你所遇到的语言。包括什么是变量，什么是对象，如何调用方法（发送信息）等等。

当然，如果你需要一本随时可以查询的工具书，我们强烈推荐苹果的官方Swift指南。

好了，有了这么多的基础做铺垫，我们可以进入正式的开发了！