

摒弃传统入门书讲解方式，项目和任务驱动，逐步引导读者在实战中掌握iOS应用开发的工具、方法和技巧

由浅入深、循序渐进，核心知识点蕴藏在10余个设计精巧的案例中，零基础读者也能开发出高质量应用。

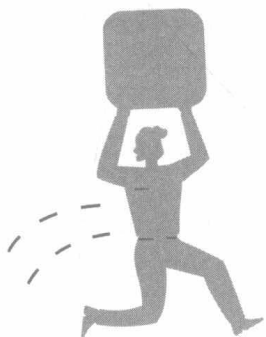
Develop iOS App with Swift

跟着项目学iOS应用开发

基于Swift 4

刘铭 陈雪峰 李钢 秦琼 著





Develop iOS App with Swift

跟着项目学iOS应用开发 基于Swift 4

刘铭 陈雪峰 李钢 秦琼 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

跟着项目学 iOS 应用开发: 基于 Swift 4/ 刘铭等著. —北京: 机械工业出版社, 2018.9
(iOS/ 苹果技术丛书)

ISBN 978-7-111-60907-0

I. 跟… II. 刘… III. 移动终端 - 应用程序 - 程序设计 IV. TN929.53

中国版本图书馆 CIP 数据核字 (2018) 第 214216 号

跟着项目学 iOS 应用开发: 基于 Swift 4

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 李 艺

责任校对: 李秋荣

印 刷: 三河市宏图印务有限公司

版 次: 2018 年 9 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 24.25

书 号: ISBN 978-7-111-60907-0

定 价: 89.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有 · 侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

为什么要写这本书

从目前来看，iOS 是全球最流行的移动端操作系统，这已经成为不争的事实。同时近年来苹果公司不断更新和完善供 iOS OS X 应用编程的开发语言 Swift，使 Swift 在未来赚足眼球成为可能。作为一名 iOS 开发者，相信对于 Swift 语言并不陌生。使用 Swift 语言可以高效开发出高质量的移动应用。如果开发人员现在还不开始学习 Swift，还在留恋着 Objective-C 的话，恐怕用不了多久，就会被 Swift 所带来的技术革新无情淘汰。另一方面，iOS 本身也在快速地进行技术改进。与以往不同，iOS 10、iOS 11 引入和开放了许多别出心裁的新技术，如 Core-ML、ARKit 等。如果开发人员能够充分利用这些新技术，就可以让自己的应用给用户带来前所未有的使用体验，进而在移动应用时代取得一个制高点。

Swift 自身的发展太快了，很多程序员在将自己的项目从 Swift 2 迁移到 Swift 3 的时候就遇到了很多头痛的问题。希望像这样“毁灭性”的升级是最后一次。好在从 Swift 3 到 Swift 4 的变化并不大。通过本书，读者可以将学到的知识点运用到实战中去，真正地将所有知识点融会贯通，从而打通所有“脉络”，在编写程序代码的时候达到“思如泉涌”的效果。

本书结构

本书通过制作真实世界的应用程序来帮助读者学习 Swift 4 编程知识。例如木琴弹奏、问答测试、骰子游戏、聊天应用、待办事项类应用 (TODO) 和天气应用。另外，书中还包括苹果公司的 Core-ML 智能应用程序与机器学习的相关内容，通过学习这些内容你将可以构建图像识别应用程序。

本书是根据由易到难的顺序来安排应用程序项目的，具体如下。

第 1 章：介绍 Xcode 的安装，认识 Xcode 的用户界面及主要面板功能，带领大家创建 Hello World 应用。

第 2 章：使用 Interface Builder 简单搭建用户界面，并制作 I am rich 应用。

第 3 章：介绍如何在 iPhone 物理真机上安装应用程序。

第 4 章：通过制作掷骰子应用，掌握如何通过代码控制界面元素，以及当用户与界面元素发生交互时如何给代码发送消息。

第 5 章：Swift 语言的基础知识讲解。

第 6 章：介绍如何利用 Stack Overflow 网站解决在开发时所遇到的问题，并利用 AVAudioPlayer 类在应用中播放声音。

第 7 章：介绍 MVC 设计模式，并制作问答测试应用程序。

第 8 章：学习自动布局的相关知识，并对掷骰子应用进行迭代更新。

第 9 章：介绍类与对象的相关知识。

第 10 章：通过 CocoaPods 安装第三方链接库，并通过相关 API 从远程 Webservice 获取所需要的数据。

第 11 章：利用目前国内流行的云端数据库建立聊天应用。

第 12 章：介绍版本控制的相关知识。

第 13 章和第 14 章：利用 CoreData、Realm 等数据存储工具实现类 TODO 应用程序。

第 15 章：利用机器学习和 Core-ML 的相关知识，构建图像识别应用程序。

各个部分的功能实现都基于由浅入深、循序渐进的原则，让广大读者在实践操作的过程中不知不觉地学习新方法，掌握新技能。

本书面向的读者

本书适合具备以下几方面知识和硬件条件的群体阅读。

- 有面向对象的开发经验，熟悉类、实例、方法、封装、继承、重写等概念。
- 有 Swift 的开发经验。
- 有 MVC 设计模式的开发经验。
- 有简单图像处理的经验。
- 有一台 Intel 架构的 Mac 电脑（Macbook Pro、Macbook Air、Mac Pro 或 Mac Mini）。

如何阅读本书

每个人的阅读习惯都不相同，而且本书并不是一本从 Swift 语法讲起的基础“开荒”书。所以我还是建议你先从 Swift 3.X 的语法书学起，在有了一定的 Swift 语言基础以后，再开始阅读本书，跟着实践操作一步步完成各章节的项目。

在阅读本书的过程中，你可能会遇到语法错误、编译错误、网络连接错误等情况，不用着急，根据调试控制台中的错误提示，去分析产生 Bug 的原因，或者通过与本书所提供的源码进行对比，找出问题所在。

勘误和支持

由于水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。书中的全部源文件可以从 GitHub (<https://github.com/liumingl/iOS-11-Swift-4-Tutorial>) 下载，也可以从我的网站 (刘铭 .cn) 下载。如果你有任何宝贵意见或建议，欢迎发送邮件至 liuming_cn@qq.com，期待得到你们的真挚反馈。

致谢

首先要感谢伟大到可以改变这个世界的 Steven Jobs，他的精神对我产生了非常大的影响。

其次要感谢机械工业出版社华章公司的编辑杨福川老师和小艺老师，在这段时间中始终支持我的写作，你们的鼓励和帮助使我顺利完成全部书稿。

最后感谢我的爸爸、妈妈、刘颖、刘怀羽、张燕、王海燕，感谢你们对我的支持与帮助，并时时刻刻给我信心和力量！

谨以此书献给我最亲爱的家人，以及众多热爱 iOS 的朋友们！

刘铭

目 录 *Contents*

前 言

第1章 开始iOS 11和Swift 4编程 1

- 1.1 iOS 11 应用程序开发工具 1
- 1.2 下载安装 Xcode 3
- 1.3 浏览 Xcode 开发环境 4
- 1.4 初步剖析 iOS 应用程序 10

第2章 Interface Builder 介绍 12

- 2.1 如何创建 Xcode 项目 13
- 2.2 使用故事板创建用户界面 13
- 2.3 如何定位用户界面元素 16
- 2.4 导入图像素材到 Xcode
项目 18
- 2.5 运行并测试项目 20

第3章 在iPhone真机上安装应用 22

- 3.1 使用 Xcode 将项目下载到
物理真机 22
- 3.2 通过 GitHub 下载项目
样例代码 25

第4章 构建简单的掷骰子游戏 29

- 4.1 如何设计掷骰子游戏 29
- 4.2 建立代码与界面元素的关联 32
- 4.3 IBOutlet/IBActions 调试 35
- 4.4 使用 Swift 创建随机数 37
- 4.5 数据类型、常量、变量 39
- 4.6 解决错误：“The Maximum
Number of Provisioning Profiles
Reached” 42
- 4.7 通过数组改变显示方式 43
- 4.8 为项目添加运动检测功能 46
- 4.9 挑战：Swift 数据类型、
变量和数组 47

第5章 Swift程序设计基础 51

- 5.1 备注、打印语句和调试
控制台 51
- 5.2 Swift 函数：Part 1 - 简单
函数 54
- 5.3 Swift 函数：Part 2 - 函数
的输入 56

5.4	Swift 函数 : Part 3 - 函数的输出	58	7.9	如何实现 UIAlertController 以及弹出窗口给用户	110
5.5	Swift 中的条件语句 (IF/ELSE)	59	7.10	高级别的重写	113
5.6	挑战: 在 Playgrounds 中制作人体体重指数计算器	62	7.11	统计分数	113
5.7	Swift 中的循环语句	63	7.12	合并 Objective-C 代码到 Swift	116
5.8	在程序中使用循环	64	7.13	挑战: 制作情商测试应用	119
5.9	挑战: 脑筋急转弯	68			
第6章	利用iOS API制作音乐应用	71	第8章	iOS的自动布局和设置约束	124
6.1	使用故事板中的 Tags	71	8.1	通过代码定位 UI 元素	126
6.2	学会使用 Stack Overflow 和 Apple Documentation	74	8.2	自动布局	128
6.3	利用 AVFoundation 播放声音	82	8.2.1	在界面生成器中实时预览布局效果	129
6.4	Swift 4 中的错误捕获——Do、Catch 和 Try	84	8.2.2	使用自动布局将 square 居中	130
6.5	创建一个播放声音的方法	86	8.2.3	解决布局约束的问题	132
6.6	让 App 每次播放不同的声音	88	8.2.4	另一种预览故事板的方式	133
6.7	程序中的“作用域”	89	8.2.5	添加一个标签	135
			8.2.6	安全区域	137
第7章	使用Model-View-Controller 设计模式制作小测验App	92	8.2.7	编辑约束	139
7.1	初始化 Quizzler 项目	92	8.3	自动布局实战——设置约束	139
7.2	创建数据模型	96	8.4	挑战自动布局	145
7.3	面向对象	98	8.5	在自动布局中使用堆叠视图	146
7.4	创建答题库类	101	第9章	Swift 4中阶知识	152
7.5	Model View Controller (MVC) 设计模式	102	9.1	类和对象	152
7.6	初始化第一个题目	104	9.2	创建全新的类	153
7.7	处理后续题目	106	9.3	创建枚举	155
7.8	使用 Xcode 调试控制台	108	9.4	根据类创建一个对象	156
			9.5	类的初始化	157

- 9.6 Designated 和 Convenience
初始化方法 158
- 9.7 创建一个方法 160
- 9.8 类的继承 161
- 9.9 重写一个继承的方法 162
- 9.10 Swift 语言中的可选 164
- 第10章 利用Cocoapods、GPS、APIS、REST制作天气应用 169**
 - 10.1 设置项目 170
 - 10.2 注册免费的 API Key 172
 - 10.3 为什么需要 Cocoapods ? 173
 - 10.3.1 在你的 Mac 上安装和设置
Cocoapods 174
 - 10.3.2 在你的 Xcode 项目中
安装 Pods 176
 - 10.4 设置 Location Manager 并从
iPhone 获取 GPS 数据 177
 - 10.5 定位权限 181
 - 10.6 在 WeatherViewController
中获取 GPS 数据 184
 - 10.7 委托、字典和 API 190
 - 10.7.1 委托 190
 - 10.7.2 字典 190
 - 10.7.3 API 191
 - 10.8 使用 Alamofire 192
 - 10.9 JSON 以及如何解析 JSON 193
 - 10.10 创建气象数据模型 196
 - 10.11 Segues 的相关介绍 201
 - 10.12 在项目中委托和协议 204
 - 10.13 如何在视图控制器间
传递数据 207
 - 10.14 基于城市名称的天气
数据请求 212
 - 10.15 挑战：利用 Cocoapods、REST
和 APIs 构建比特币价格跟踪
应用 213
- 第11章 利用云端数据库、iOS 动画
和高级Swift特性构建聊天
应用 219**
 - 11.1 关于 Bmob 219
 - 11.1.1 在 LeanCloud 上注册账户 220
 - 11.1.2 设置 Bmob 220
 - 11.2 保存数据到 Bmob 222
 - 11.2.1 创建桥接头文件 222
 - 11.2.2 测试云端数据库的读写 223
 - 11.2.3 在应用上注册一些用户 224
 - 11.3 Swift 闭包 227
 - 11.4 事件驱动、应用程序生存期 231
 - 11.4.1 事件驱动——应用运行
的本质 231
 - 11.4.2 应用程序的生存期 232
 - 11.4.3 什么是完成处理? 232
 - 11.5 导航控制器是如何工作的? 235
 - 11.6 编写登录屏幕代码 236
 - 11.7 表格视图 237
 - 11.8 了解 UI 动画 243
 - 11.9 发送消息 247
 - 11.10 通过 Bmob 监听数据
表的变化 248

- 11.11 进一步完善用户体验和
用户界面 253
 - 11.11.1 利用 Progress Spinner
改善用户体验 253
 - 11.11.2 区别不同的用户 254
- 第12章 Git、GitHub和版本控制** 256
 - 12.1 版本控制和 Git 256
 - 12.2 使用 Git 和命令行进行
版本控制 257
 - 12.3 GitHub 和远程仓库 260
 - 12.4 Gitignore 263
 - 12.5 克隆 266
 - 12.6 分支和迁移 267
 - 12.7 在 Xcode 9 中使用 Git 和
GitHub 273
- 第13章 使用Core Data、User Defaults
学习本地数据存储** 279
 - 13.1 创建 UITableViewController 的
子类 280
 - 13.2 在 UIAlert 中使用文本框
创建新的条目 285
 - 13.3 持续本地数据存储 290
 - 13.3.1 为什么需要持续的本地
数据存储 290
 - 13.3.2 使用 UserDefaults 实现持续
本地数据存储的功能 291
 - 13.3.3 UserDefaults 说明 295
 - 13.3.4 Swift 中的单例模式 296
 - 13.3.5 创建自定义数据模型 297
 - 13.3.6 UserDefaults 的弊端 301
 - 13.4 认识 NSCoder 301
 - 13.4.1 使用 NSCoder 编码
对象数组 301
 - 13.4.2 使用 NSCoder 解码 304
 - 13.5 在应用中使用数据库 305
 - 13.5.1 设置和配置 Core Data 306
 - 13.5.2 如何使用 Core Data
存储数据 311
 - 13.5.3 查看 SQLite 后端数据库 313
 - 13.5.4 Core Data 基础 314
 - 13.5.5 从 Core Data 读取、
修改和删除数据 315
 - 13.6 借助 Core Data 的查询功能
实现搜索 317
 - 13.7 借助 Core Data 创建关系图 323
- 第14章 使用Realm进行本地数据
存储** 333
 - 14.1 在项目中集成 Realm 334
 - 14.2 使用 Realm 保存数据 336
 - 14.3 使用 Realm 读取数据 340
 - 14.4 使用 Realm 修改和
移除数据 343
 - 14.5 使用 Realm 检索数据 344
 - 14.6 回顾 Realm 的操作流程 345
 - 14.7 让单元格可以滑动 347
 - 14.8 让 App 的界面更加
丰富多彩 352
 - 14.9 调整导航栏的 UI 357

第15章 机器学习和Core-ML	362
15.1 介绍机器学习	362
15.1.1 机器学习	362
15.1.2 监督式学习	363
15.1.3 非监督式学习	366
15.1.4 强化学习	366
15.2 Core-ML——整合机器学习到 iOS 应用中	367
15.2.1 什么是 Core-ML?	367
15.2.2 Core-ML 能做什么	369
15.2.3 如何识别图像并反馈结果	374
15.2.4 判断图片中的食物	376

开始 iOS 11 和 Swift 4 编程

大家好，本书的目的是教会大家如何使用 iOS 11 SDK、Xcode 9 和 Swift 4 编程语言创建 iOS 应用程序。

不管你是 iOS 开发的初学者，想通过本书学习如何使用 Swift 语言编写应用程序；还是之前已经有在 iOS 10 中开发应用程序的经验，想进一步快速掌握 iOS 11 的功能和最新版本的 Swift 4 语言。请放心，本书都可以满足你的需求。

1.1 iOS 11 应用程序开发工具

在本节将会向大家介绍开发 iOS 应用程序需要用到的软件以及相关的硬件。首先，我们必须拥有一台 Mac 电脑，不需要是当下最新最快的，但它一定要能运行 macOS 10.12.6 及以上版本的操作系统。因为苹果的特殊政策，我们只能在 macOS 上安装 iOS 应用程序开发工具 Xcode。这也就意味着仅仅使用 iPad 或 iPad Pro 是不可能完成 iOS 应用程序开发的任务。如果你拥有一台 iMac、MacBook 甚至是 Mac Mini 的话，就足以满足开发的需求。

如果你现在手头确实有些“银子”不足的话，可以考虑购买一台二手的 Mac Mini，性价比还是很高的。如果你手头只有 PC 的话，可以考虑借助 Mac in Cloud 平台（网址：www.macincloud.com）。在网站上它提供了如同 Mac 一样的在线云端服务，这样就可以通过现有的 PC 和互联网实现 Mac 功能。你只需要在远程系统中下载并安装 Xcode 就好。如图 1-1 所示。

另外，还有一种叫作 Hackintosh 的方式，也就是将 macOS 操作系统通过非正常的手段安装到自己的 PC 上，比如通过 VMWare、Delphi XE4 等方式。但是不管是 Mac in Cloud

还是用 PC 安装的 Hackintosh，都不能通过这种方式将写好的应用传到 iPhone 真机上进行测试，唯一的方法就是使用真正的 Mac 电脑。

虽然不能在真机上运行，但是我们还是可以在 Xcode 模拟器中运行所编写的 iOS 项目。而且，即便是在 macOS 系统上，我们也会在大部分时间利用 Xcode 模拟器测试项目代码。在模拟器中包含了各种版本的 iOS 系统，所以可以很好地测试和运行项目。如图 1-2 所示。



图 1-1 MacinCloud 网站主页



图 1-2 在模拟器中运行并测试 iOS 项目

开发所用到的软件叫作 Xcode，是由苹果公司研发的 IDE 开发环境。我们可以在 Xcode 中编写代码、设计界面和调试应用程序，Xcode 是完全免费的。

只有在 macOS 10.12.6 及以上，或者是 macOS 10.13 及以上环境下才可以下载并安装 Xcode 9。强烈建议大家将 Mac 的操作系统升级到 macOS 10.13 的最新版本。如何检测你的 macOS 是否为最新的版本呢？单击屏幕左上角的苹果图标后会弹出一个对话框，在概览标签中就可以看到运行操作系统的版本，如图 1-3 所示。或者单击对话框右下角的**软件更新**升级你的操作系统版本。另外，我们还可以在 Mac Store 中搜索最新的 macOS high Sierra（也就是 macOS 10.13 版本），然后下载安装。

除了在开发的时候需要安装 Xcode 以外，最好再安装一款图像编辑软件。比如 Adobe 的 Lightroom、Photoshop，或者是 Sketch，如图 1-4 所示。

在测试应用程序的时候，或是将其上架到 App Store 之前，你最好有一台 iOS



图 1-3 在关机本机菜单中查看 macOS 系统的版本

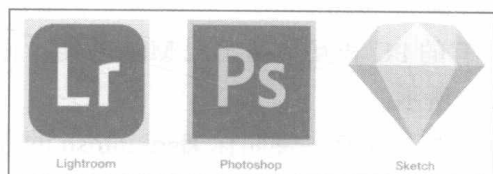


图 1-4 Lightroom、Photoshop 和 Sketch 软件

物理真机，并进行必要的测试。到底是 iPhone 还是 iPad，这需要根据你的开发目标需求而定。

Xcode 模拟器就像一个运行在 macOS 系统上的虚拟 iPhone，我们可以旋转它，并进行简单的手势操作和实现摇晃的功能，可以对其放大或缩小。但是模拟器也会有一定的限制，比如在模拟器中我们无法实现通知、健康或 HomeKit 功能。

最后需要提示大家的是：在 Xcode 7 之前，如果要将编写好的程序传到物理真机中，需要向苹果支付 99 美金的年费。从 Xcode 7 开始，我们在不需要缴纳年费的情况下也可以进行物理真机测试，你只需要注册一个开发者账号即可。但是，如果想要将应用程序上架到 App Store 进行销售或推广，则需要缴纳年费。

1.2 下载安装 Xcode

接下来，我们需要下载和安装 Xcode。Xcode 是运行在 macOS 系统上的一个应用程序，我们会使用它来编写程序代码并创建 iOS 应用。Xcode 是完全免费的，所以我们不用担心会有任何的花销。但是，在我们正式安装 Xcode 以前，还需要再确认一些事情。

首先要确保我们的 Mac 有足够的硬盘空间。Xcode 安装文件大概是 4.5G，所以需要 10G 的剩余空间来下载和安装它。要确保这一步非常简单，只需单击桌面左上角的苹果图标，然后找到储存空间，查看硬盘的剩余空间是否够 10G，如图 1-5 所示。

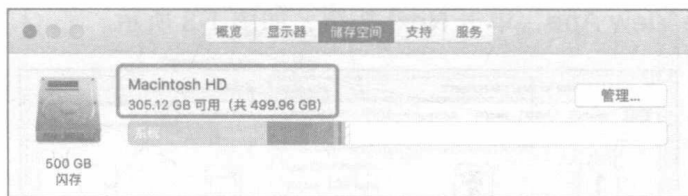


图 1-5 查看 Mac 中的剩余空间

其次，就是需要确定我们的 macOS 版本是否为最新。检查的方法也非常简单。还是单击屏幕左上角的苹果，然后在概览标签中查看系统的版本是否为 10.13 或更高。

在确定好前两件事以后，最后一件事，就是确保我们所下载的 Xcode 版本不是 Beta 版本。如果下载的 Xcode 是正式发行版的话，就不用担心它会产生任何问题，而 Beta 版会包含很多 Bug，进而产生很多让你头疼的问题。

在 Mac App Store 中搜索 Xcode，然后单击获取按钮进入 Xcode 详细页面，这里可以看到当前的 Xcode 版本是 9.2。单击安装按钮，经过一段时间的等待后，Xcode 就安装好了，如图 1-6 所示。



图 1-6 在 Mac App Store 中安装 Xcode

1.3 浏览 Xcode 开发环境

我们在启动 Xcode 后会看到欢迎界面，这里可以选择以 playground 开始（Get started with a playground）或者是创建一个新的 Xcode 项目（Create a new Xcode project），如图 1-7 所示。

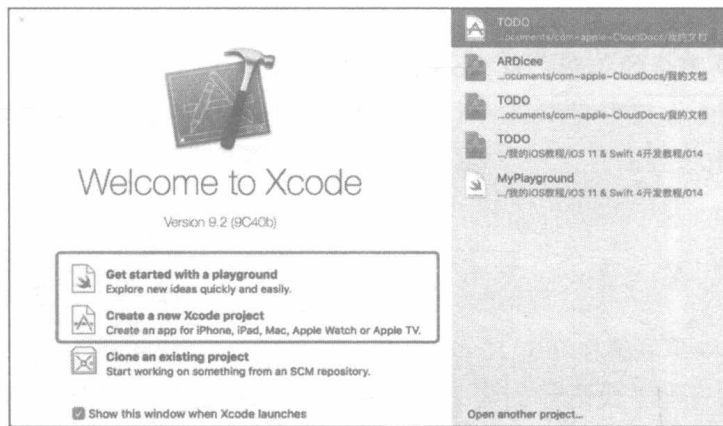


图 1-7 Xcode 的欢迎界面

实战：快速创建一个全新的 Xcode 项目。

步骤 1：单击 Create a new Xcode project 向导，在选择项目模板中选择 iOS / Application / Single View App，单击 Next 按钮，如图 1-8 所示。

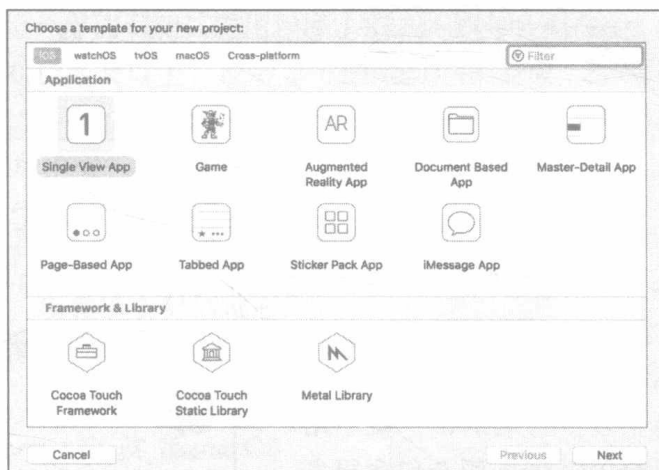


图 1-8 在选择项目模板中选择 Single View App



提示 在模板中还有 Game、Master-Detail App、Page-Based App 和 Tabbed App 模板，我们可以根据不同的需求选择不同的模板，除非是创建游戏项目，大部分的开发者都会选择单视图应用程序（Single View App）模板。因为不管是 Master-Detail、Page-Based 还是 Tabbed App 模板，都会自动在项目中添加很多代码，而这些代码并不实

用。反观 **Single View App** 模板，它具有很大的灵活性，可以最大限度地以自定义的方式添加所需要的内容，具体操作方法会在后面详细介绍。

步骤 2：在 **Product Name** 中需要输入应用程序的名称，这个名字要简单并且最重要的是 **Cool**。这里输入 **Hello World**。**Team** 设置为 **None**。在后面的章节中会讲述如何将 App 上传到 iPhone 真机，到时会具体介绍如何设置。

步骤 3：**Organization Name** 设置为你公司的名字，如果是个人开发则输入本人的名字即可，比如 **Liu Ming**。

步骤 4：**Organization Identifier** 是你的域名的反向，比如你的域名是 **liuming.cn**，这里就需要填写 **cn.liuming**。

步骤 5：**Language** 设置为 **Swift**，代表我们使用 Swift 语言进行项目的开发。

步骤 6：在对话框下面还有三个可选框：**Core Data** 是与数据库存储相关；**Unit Tests** 是单元测试相关；**UI Tests** 是用户界面测试相关。在本实例中请不要勾选任何一个选项，如图 1-9 所示，单击 **Next** 按钮。

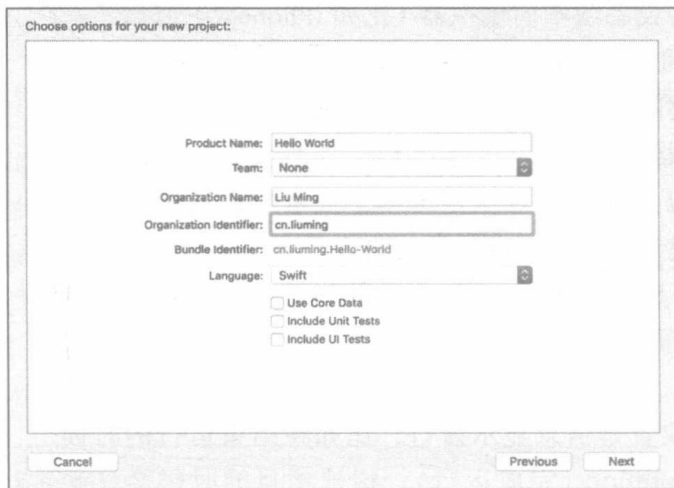


图 1-9 项目设置选项

步骤 7：在接下来的对话框中请确定项目保存的位置。这里选择 **Desktop**（桌面），（可以方便我们快速找到它），单击 **Create** 按钮。

在项目打开以后，我们就可以看到 Xcode 所显示的所有不同组件，如图 1-10 所示。

在界面的顶部是 Xcode 状态栏，从左侧开始是一个**播放**（play）按钮，单击它会构建并在模拟器或物理真机上运行项目代码，单击**停止**（stop）按钮则会终止项目在模拟器或物理真机上的运行。在它们之后的 **Hello World** 选项中，我们可以设置在哪个环境里运行应用程序项目。当 Mac 与 iPhone 物理真机连接以后，我们就可以在真机上运行，或者手动选择在 Xcode 模拟器中运行。

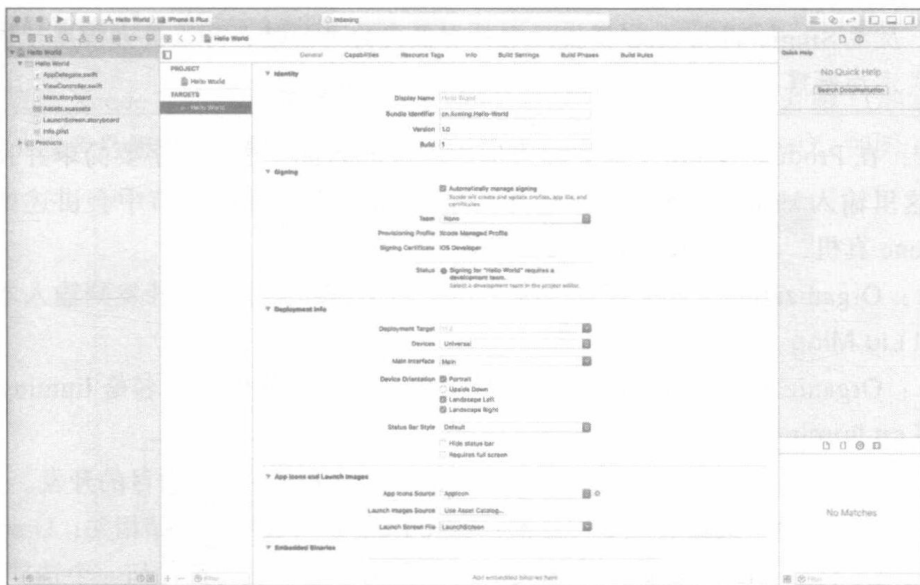


图 1-10 Xcode 的工作界面

当我们选择一种设备版本的模拟器（比如 iPhone 7）以后，一旦我们单击 Play 按钮，就会在 macOS 上面启动 iPhone 模拟器，我们可以用鼠标修改它的尺寸。模拟器默认是带曲边的，这意味着可以单击 iPhone 模拟器左右边缘的仿真按键，实现相应的功能。比如单击 Home 键可以让 iPhone 回到主屏幕，或者单击音量键调整播放声音的大小。建议大家去掉模拟器的曲边显示，在菜单中选择 Window，然后取消 Show Device Bezels 的勾选状态。这样，可以将 iPhone 屏幕设置得更大一些，方便我们进行调试，如图 1-11 所示。

状态栏的中间位置是信息显示窗口，比如在结束运行的时候会显示：Finished running（完成运行）；构建项目的时候会显示 Building 进度条。当项目出现错误或警告的时候，还会在窗口的右下角出现相应的图标和错误或警告的数量。

位于信息窗口右侧的一组按钮负责切换编辑器的状态，其中前两个按钮的使用频率非常高。第一个是默认的标准编辑器（Standard Editor），它会将 Xcode 中间部分的区域设置为一个。

当我们单击第二个有两个圆圈图标的按钮时，Xcode 会进入辅助编辑器（Assistant Editor）模式，Xcode 中间的部分将被分割为两个区域。我们可以将设计的用户界面放在左侧，代码放在右侧，这样方便进行代码与用户界面元素的关联，在后面的章节会对关联有详细介绍。

单击第三个有两个箭头的按钮，会进入版本编辑器（Version Editor）模式，它允许我

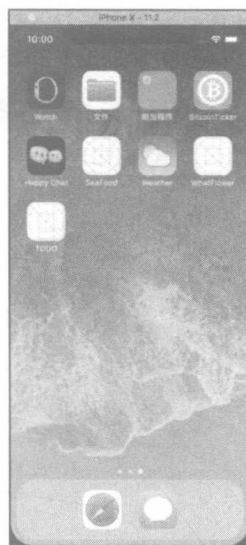


图 1-11 取消 iPhone 模拟器的曲边效果