

- 通过三个示例程序，全方位揭示命令行开发、iPhone App开发和Mac App开发的实用技术、技巧和最佳实践
- 详细介绍Xcode各项功能和使用方法，涵盖从项目初始到项目发布的完整开发流程，所有示例程序都使用Swift语言编写

Xcode 实战



Apple平台开发实用技术、技巧及最佳流程

[美] 弗里茨·安德森 (Fritz Anderson) 著 王东明 译

Xcode 6 Start to Finish

iOS and OS X Development



机械工业出版社
China Machine Press

Xcode 实战

Apple平台开发实用技术、技巧及最佳流程

Xcode 6 Start to Finish

iOS and OS X Development

[美] 弗里茨·安德森 (Fritz Anderson) 著 王东明 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Xcode 实战: Apple 平台开发实用技术、技巧及最佳流程 / (美) 安德森 (Anderson, F.) 著; 王东明译. —北京: 机械工业出版社, 2016.1

(iOS/ 苹果技术丛书)

书名原文: Xcode 6 Start to Finish: iOS and OS X Development

ISBN 978-7-111-52550-9

I.X… II. ①安… ②王 III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2015) 第 309947 号

本书版权登记号: 图字: 01-2015-5404

Authorized translation from the English language edition, entitled *Xcode 6 Start to Finish: iOS and OS X Development*, 9780134052779 by Fritz Anderson, published by Pearson Education, Inc., Copyright © 2015.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2016.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和中国香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

Xcode 实战

Apple 平台开发实用技术、技巧及最佳流程

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: 关 敏

责任校对: 殷 虹

印 刷: 北京文昌阁彩色印刷有限责任公司

版 次: 2016 年 1 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 31.5

书 号: ISBN 978-7-111-52550-9

定 价: 99.00 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

Xcode 运行于苹果公司的 Mac 操作系统下，是苹果公司向开发人员提供的集成开发环境，主要用于开发 Mac OS X 和 iOS 应用程序。之前基本上只有 Mac OS X 应用的开发才会用到 Xcode，所以它一直很小众，处于默默无闻的状态。随着移动设备的普及，移动应用的开发也越来越火热，iPhone 占据了智能机市场的重要地位，越来越多的开发人员开始加入 iPhone 应用程序开发的阵营。从 Xcode 3.1 开始，它也可作为 iOS 的开发环境，所以 Xcode 也随之进入了移动开发人员的日常工作中。

本书深入浅出地介绍了 Xcode 6 的使用，作者根据自己丰富的 Apple 平台开发经验，通过 3 个示例程序，分别介绍了命令行开发、iPhone app 开发和 Mac app 开发的最佳流程，在这些开发流程中，详细介绍了 Xcode 6 的各项功能。示例程序全部用最新的 Swift 语言编写。通过阅读本书，你可以掌握 Xcode 6 的使用方法和常见功能。熟练使用 Xcode 6 的各项功能可以极大地提高工作效率。

自从支持 iOS 开发后，Xcode 的更新速度也加快了许多。每次都和 iOS 的新版本一起发布，翻译完本书的时候，苹果公司在 WWDC 2015 上刚刚发布了 Xcode 7。Xcode 7 最大的特性就是支持 Swift 2.0 和可以不用加入付费开发者计划就能在真机上调试，极大地方便了开发者。而且，苹果公司在 WWDC 上声称之后会将 Swift 开源，这将会极大地促进 Swift 社区的发展，本书中的示例程序都用 Swift 语言重新编写称得上是明智之举。书中介绍的各个特性基本都适用于 Xcode 7，所以读者不必担心内容过时的问题。

本书内容翔实，循序渐进，覆盖了从项目初始到项目发布流程中涉及的 Xcode 功能，既可以作为经验丰富的开发人员的参考手册，也可以作为新手开发人员的入门指引书籍。

鉴于译者水平有限，书中难免有差强人意的地方，欢迎广大读者批评指正。

前言 Preface

欢迎阅读本书。本书将展示如何使用 Apple 的集成开发环境在进行高质量产品开发的过程中达到事半功倍的效果。

Xcode 6 是一个开发工具家族的后裔，这些开发工具最早可以追溯到 20 年前 NeXT 的 ProjectBuilder。它最初是一个集文本编辑器、用户界面设计器和 UNIX 前端开发为一体的工具集。现在 Xcode 6 已经发展成为一个复杂的系统，用于构建应用程序和系统软件。它充分利用了全面的索引系统和精确的增量解析器的功能，可以用来帮助你在项目中集成正确的代码，并一次性让这些代码执行成功。

这种强大的功能会让人望而生畏。本书旨在揭开 Xcode 的神秘面纱，通过一些案例用循序渐进的方式展示如何在日常工作中使用它。不要被一般的方法蒙蔽了双眼：本书将会揭示 Xcode 完整开发流程的最佳实践。本书不涵盖“高级话题”，我会在开发过程的恰当时机介绍版本控制和单元测试相关的内容。

本书的组织方式

首先，对我的计划进行一个整体概括：这是一本关于开发者工具的书。如果它教会了你如何使用 Cocoa 框架或者一些关于编程的知识，那也不错，不过这些内容都是用来介绍 Xcode 工作流程所附带的内容。学习各种框架有众多优秀的书籍和资源，在附录 B 中你可以找到大量这类资源。

每一段旅程都需要有一条路线，同样每一堂课都需要一个故事作为引线。本书前 3 部分通过 3 个应用程序来介绍 Xcode，这 3 个程序是：一个命令程序、一个 iOS 应用程序以及一个 OS X 应用程序，它们的功能是计算和展示美国橄榄球比赛中的部分统计信息。这些应用程序都不实用，图形应用几乎全部基于示例数据。但是在这 3 个应用程序的制作过程中，需要用到很多开发工具，能让你充分了解如何使用这些工具。

Xcode 支持一些技术，如：Core Data 和 OS X 绑定（binding），这些技术都不是针对入门

读者的。本书直接深入这些技术内部，忽略了从概念上讲更简单的方法，这样我可以解释清楚 Xcode 的工作原理。其他一些“高级”技术，比如单元测试和版本控制，都会在实践中需要它们的时候才会介绍。这就是 Xcode 支持的工作流程。

第二部分和第三部分分别使用 iOS 和 OS X 应用程序作为示例程序，即使你仅对其中一个平台感兴趣，也应该阅读另外一个平台对应的内容。这些应用程序的制作不过是一个引子，相应的技术两个平台都适用。

第一步

第一部分会介绍 Xcode 的安装，然后通过一些基本的调试技巧运行第一个项目。你将会完成一个小型的命令行应用程序。这个应用程序可能很简单，但是你会学到在制作复杂的图形应用程序之前必备的一些基本技术。

第 1 章，获取 Xcode——下载 Xcode 6 之前需要考虑的一些问题，两种下载和安装方法。

第 2 章，初识 Xcode——初步了解 Xcode，建立一个小项目并运行。

第 3 章，简单的工作流程和被动调试——编写、构建并运行一个简单的应用程序，再对崩溃做出处理。

第 4 章，主动调试——在程序中设置断点（breakpoint）和追踪（tracing）来负责调试。该章会展示如何组织工作空间（workspace）。

第 5 章，编译——停下来介绍应用程序的构建流程。

第 6 章，添加库 target——在项目中添加一个库 target，然后学习如何从多个目标（target）构建产品。

第 7 章，版本控制——揭示为什么源代码管理这么重要，如何通过 Git 和 Subversion 充分利用 Xcode 内置的版本管理。

iOS 应用程序的生命周期

第二部分介绍一个小型 iPhone 应用程序，以及如何使用 Apple 开发者工具构建这个应用程序。它会引领你使用图形编辑器制作用户界面，展示如何配置应用程序以优化应用程序的速度和内存负荷。

第 8 章，开始制作 iOS 应用程序——从创建一个 iOS 项目开始，学习 iOS Cocoa 框架中的模型 - 视图 - 控制器（Model-View-Controller）设计思想，这个思想也适用于 OS X。

第 9 章，iOS 应用程序：模型——设计一个 Core Data 计划（schema），并在自己的代码中支持它。

第 10 章，iOS 应用程序：控制器——创建一个将模型与界面视图链接在一起的控制器。同时，该章会讲解有关重构以及 Xcode 持续错误检查的知识。

第 11 章，构建新视图——使用集成在 Xcode 中的 Interface Builder 为应用程序设计用户界面视图，充分利用源代码的自动补全功能。

第 12 章，新视图的自动布局——在 Xcode 6 中，自动布局（Autolayout）更多的是帮助完成工作而不是与工具做斗争。学习如何运用 Cocoa 布局得到你想要的效果。

第 13 章，添加表格单元格——当向应用程序添加一个界面内组件（in-screen component）的时候，你将会调试内存管理，并通过 Scheme 编辑器控制 Xcode 对应用程序的编译、运行以及测试过程。

第 14 章，添加编辑器——添加一个编辑器视图，深入学习 Storyboard。

第 15 章，单元测试——单元测试会加快开发进程并让应用程序更稳定。该章将会介绍 Xcode 是如何将单元测试作为开发过程中头等重要的部分的。

第 16 章，度量和分析——使用 Instruments 追踪性能和内存问题。

第 17 章，iOS 扩展——创建一个系统级别的扩展和共享库，提升应用程序的价值。

第 18 章，配置——完成开发后，获取 Apple 的许可让应用程序运行在 Apple 设备上的流程复杂且易变。该章将会介绍如何使用 Xcode 减少绝大部分麻烦，并给出一些避免陷阱的小技巧。

Mac OS X 开发中的 Xcode

第三部分将焦点转移到 OS X 开发上。有一些概念在 OS X 中比在 iOS 中更重要，不过你也会学到一些与开发平台无关的技术。

第 19 章，开始制作 OS X 应用程序——将 iOS 组件移植到 OS X 上；介绍什么是响应链（responder chain），以及 Interface Builder 如何让它更易于使用。

第 20 章，绑定：连接 OS X 应用程序——当构建一个弹出窗口（popover window）的时候，你将会使用 OS X 绑定来简化数据与界面之间的链接。你还会使用自动调整大小的功能，这是一种布局视图结构的传统技术。

第 21 章，本地化——如何将 Mac 和 iOS 应用程序翻译成其他语言。

第 22 章，程序包——你将会掌握大部分 Mac 和 iOS 产品的基本结构，以及在这两个平台上，如何使用 Info.plist 文件将应用程序与操作系统相匹配。

第 23 章，属性列表——学习基本的类 JSON 的文件类型，用于在 OS X 和 iOS 平台上存储数据。

Xcode 任务集

当看到这里的时候，你已经学习了深入了解 Xcode 工具集内部细节的基础知识。第四部分将会学习一些比第二部分和第三部分更需要认真对待的内容。

第 24 章，Xcode 中的文档——该章将会介绍 Xcode 如何为你提供 API 的即时帮助，并浏览 Cocoa 开发中概念的细节，揭示如何向系统中添加你自己的文档。

第 25 章，Xcode 构建系统——该章将会介绍 Xcode 将源文件转化为可执行文件的过程背后所使用的工具和规则。

第 26 章，Instruments——使用 Apple 的 timeline profiler，你可以超越基本的性能和内存诊断，从整体上观察程序如何使用它的资源。

第 27 章，调试——如何使用断点和条件来减少内部代码诊断。你也可以找到一个关于 11db 调试器命令集的教程，以更进一步地掌控自己的代码。

第 28 章，零散的知识——综合介绍一些开发工具中会遇到的小技巧、陷阱以及特性。

附录

第五部分的附录包含帮助你掌握构建系统的参考信息，还有寻找帮助和支持的途径。

附录 A，一些构建变量——Xcode 构建系统中最重要的配置和环境变量。

附录 B，资源——介绍能在开发过程中助你一臂之力的书、工具和网络资源。

关于版本

本书基于 OS X 10.10、iOS 8.2 和 Xcode 6.2 编写而成。

关于代码

本书中有很多可执行的示例代码，Xcode 6 是一个创建并运行代码的系统。本书旨在教授工作流程。代码本身是学习工作流程时附带的内容。尤其要注意：本书中绝大部分代码在一开始都无法正常运行。本书是一本介绍开发流程的书，大部分内容都涉及寻找和解决 bug。如果不学着处理所碰到的问题，你就无法学习整个工作流程。

所以本书给出的都是一些有问题的代码。可能你会发现这些代码难以阅读，并且如果你尝试运行这些代码，也会感觉很痛苦。相信我，这么做肯定是有原因的。

本书的示例代码都可以在华章公司网站下载：www.hzbook.com。如果你在阅读的过程中使用示例代码，并用 Git（或者 Subversion）作为工作中的代码版本管理工具，那么需要将更改复制到自己的工作目录中。如果直接用示例文件夹替换自己的文件夹，将会导致自己的版本信息丢失。

约定

本书中命令行的部分内容可能会比一页书还要宽，所以我遵循在行末尾添加 \ 来换行的约定。大段的输出内容将会简单地将它们分成若干行，之后的每行都会缩进。当输出行包含很

长的文件路径的时候，我会用省略号（……）来替代不重要的部分，仅仅留下重要的部分。

Macintosh 在其最初的 20 年中就使用单按键的鼠标（不要笑，很多购买者都不知道这是什么样的鼠标，会在一个古老的 Mac 鼠标上按错误的按键）。现在它有 4 种方法可以实现鼠标点击的效果：在一个实际鼠标上点击右键（或者点击虚拟鼠标的某个代表右键点击的角落）；按住 Ctrl 键同时进行普通的点击操作；用两个手指敲击多点触摸屏（即便在台式机上这个操作也越来越常见）；或者点击多点触摸屏指定角落。在“系统偏好”（System Preferences）中有更多可以设置的选项。除非差别特别大，否则我会统称这些操作为“右击”（right-click），由你自己选择到底使用哪个操作。

致谢

当你拿到这本书的时候，其实文字编撰只占出版过程的一部分内容。我要感谢那些在背后默默奉献的人，没有他们本书就无法与读者见面。大部分人名都会出现在正式的制作人员名单中，但是他们应当收获比正式感谢更多的褒奖。

Trina MacDonald 面对有诸多更改的 Xcode，克服了巨大的困难，完成了本书的再版。这并不是一件保证能够完成的工作，但是她做到了。谢谢她。

Lori Hughes 凭借丰富的幽默细胞给我们开了个好头，为我们安排了严谨的日程表。Julie Nahil，我们的产品经理，很早就开始运作，让工作能够按照日程表的计划进行，而且在本书从手稿到成书的过程中还做了很多份外之事。

Olivia Basegio 确保了所有的合同、公文往来（包括提升薪水）的正常沟通。她还组织评审人员在本书还处于未装订阶段就开始了审阅工作。

审阅人 Chris Zahn、Gene Backlin 和 Josh Day 悉心审阅了本书，让整本书的质量与最开始相比提高了不少。当然可能还有一些错误，有一些是无意造成的，有一些是我自身水平导致的，无论怎样，都是我的问题。

文字编辑 Stephanie Geels 完成了文字、语法错误的相关检查，并为特定的文字找到了对应的字体，其中的过程很有意思。

全职工作对作者来说不是件好事（除了能拿全职工资以外）。Alan Takaoka，芝加哥大学 IT Services 中 Web Service 部门的负责人，在我写作期间，让我每周休息三四天。我保证在撰写本书的过程中参加所有的会议并在截止日期之前完成所有的工作。不过很少有工作会安排在周一或者周五。Cornelia Bailey 帮我管理了绝大多数项目，并合理地重新安排进度。

Bess 和 Kate 带给我的麻烦比我撰写本书时遇到的怀疑和沮丧更多，但却让我对本书将会写得更好充满了信心，这也许就是这两个小家伙的心意。

译者序
前 言

第一部分 起步

第1章 获取Xcode	2
1.1 开始之前	2
1.2 安装 Xcode	3
1.3 移除 Xcode	4
1.4 Apple 开发者计划	5
1.5 下载 Xcode	6
1.6 额外需要下载的内容	7
1.7 小结	8
第2章 初识Xcode	9
2.1 启动 Xcode	9
2.2 Hello World	11
2.2.1 新建项目	11
2.2.2 让 Xcode 安静下来	13
2.2.3 构建和运行	14
2.2.4 货真价实的程序	15

2.3 删除项目	15
2.4 小结	15
第3章 简单的工作流程和被动调试	16
3.1 创建项目	16
3.2 构建	19
3.3 运行	21
3.4 简单调试	22
3.5 小结	23
第4章 主动调试	24
4.1 一个简单的测试用例	24
4.2 开始主动调试	24
4.2.1 设置断点	25
4.2.2 变量窗格	26
4.2.3 单步调试	27
4.3 解决这个问题	28
4.3.1 行为	28
4.3.2 解决方法	31
4.4 小结	31
第5章 编译	32
5.1 编译	32

5.2 动态载入	38	7.5.2 User B	73
5.3 Xcode 和 Clang	38	7.5.3 返回 User A	75
5.3.1 本地分析	39	7.6 版本编辑器	78
5.3.2 跨函数分析	41	7.6.1 Comparison	78
5.3.3 索引	42	7.6.2 Blame	80
5.4 Swift	43	7.6.3 Log	81
5.5 编译产品	47	7.7 分支	81
5.5.1 中间产品	47	7.8 小结	83
5.5.2 预编译	49		
5.6 小结	51		
第6章 添加库target	52	第二部分 iOS 应用程序的生命周期	
6.1 添加 target	52	第8章 开始制作iOS应用程序	86
6.2 target 关系	54	8.1 规划应用程序	86
6.2.1 添加文件到 target 中	54	8.1.1 模型 – 视图 – 控制器	86
6.2.2 target 中的头文件	56	8.1.2 模型	87
6.3 从属 target	56	8.1.3 视图	87
6.3.1 添加库	57	8.1.4 控制器	88
6.3.2 调试依赖 target	59	8.2 开始制作一个全新的 iOS 项目	89
6.4 小结	59	8.3 项目中都包含什么	91
第7章 版本控制	60	8.4 小结	93
7.1 采取控制	61	第9章 iOS应用程序：模型	94
7.2 文件状态	63	9.1 实现模型	94
7.3 首次提交	65	9.1.1 实体	95
7.4 使用远程版本库	66	9.1.2 属性	95
7.4.1 本地建立一个“远程” 版本库	68	9.1.3 关系	97
7.4.2 将文件推送到远程服务器	69	9.2 托管对象类	100
7.5 合并与冲突	70	9.2.1 创建类——错误的方法	100
7.5.1 User A	70	9.2.2 为什么按照 Xcode 的方式 做是个错误	102

9.2.3 正确的方法——使用 mogenerator	102	11.1.1 如果想要添加新视图 控制器	129
9.3 准备	105	11.1.2 storyboard、scene 和 segue	130
9.3.1 Utilities	105	11.2 创建视图	133
9.3.2 扩展类	105	11.2.1 Passing 中的 Outlet 和 Assistant	134
9.3.3 passer_rating	107	11.2.2 Billboard 视图	136
9.4 定制 Core Data 类	108	11.2.3 链接视图与视图控制器	137
9.4.1 在项目中添加 Game 数据	108	11.2.4 特定场合下的自动布局	138
9.4.2 在项目中添加 Passer 数据	109	11.2.5 各种标签	140
9.4.3 一些测试数据	110	11.2.6 清理	142
9.4.4 源代码控制和产品文件	112	11.3 表视图	144
9.5 让模型更容易调试	114	11.4 outlet	145
9.6 小结	115	11.4.1 连接 outlet	146
第10章 iOS应用程序：控制器	116	11.4.2 检查连接	147
10.1 Objective-C 中的符号重命名	116	11.4.3 连接 GameListController	148
10.1.1 重构 Objective-C 方法名	117	11.4.4 代码补全和片段	149
10.1.2 重构类名	117	11.4.5 代码片段	150
10.2 在 Swift 中重命名类	118	11.5 测试 Billboard View	152
10.3 编辑 View Controller	119	11.6 小结	153
10.3.1 表视图	120	第12章 新视图的自动布局	154
10.3.2 构建 Passer List	121	12.1 为什么选择自动布局	154
10.3.3 新建 Passer	121	12.1.1 Autoresizing 的局限	155
10.4 生命周期问题及处理方法	122	12.1.2 Auto Layout	155
10.5 真正的 Passer Rating	123	12.1.3 需要牢记的问题	155
10.5.1 另外一个 bug	123	12.2 重新制作 Player Billboard	156
10.5.2 运行 Passer Rating	127	12.3 重构子视图布局	157
10.6 小结	128	12.3.1 Playground	158
第11章 构建新视图	129	12.3.2 StatView	160
11.1 下一个视图控制器	129		

12.3.3	安装 StatView	164	14.2	添加模型场景	189
12.4	设计约束	165	14.2.1	嵌入的 View Controller	191
12.5	两行、两个标签	167	14.2.2	将 Passer List 与 Editor 相连	193
12.6	实际的约束	169	14.2.3	Static Table Cells	194
12.6.1	默认 (Any/Any)	170	14.3	编辑器视图控制器	195
12.6.2	Any Height (非 Compact)	170	14.3.1	编辑器表	195
12.6.3	Landscape (wAny/hCompact)	171	14.3.2	将数据传递给编辑器	197
12.6.4	解决问题	171	14.3.3	从编辑器中获得数据	198
12.6.5	细节修改	171	14.4	segue	200
12.7	小结	172	14.5	小结	201
第13章	添加表格单元格	173	第15章	单元测试	202
13.1	Game 表	173	15.1	Test Navigator	203
13.1.1	Table View 中的 outlet	173	15.2	测试 CSV Reader	204
13.1.2	添加所需的 Protocol 方法	174	15.2.1	CSV 测试代码	205
13.1.3	添加 Model-to-View 的支持	175	15.2.2	测试数据	209
13.1.4	原型元素	176	15.2.3	运行测试程序	210
13.2	Game Table: 第一次运行	177	15.3	测试和调试器	211
13.3	自定义表格单元格	179	15.4	添加测试类	213
13.4	添加一些图形	182	15.5	异步测试	216
13.4.1	带有图片的单元格	182	15.5.1	测试异步代码	217
13.4.2	将 Image 视图与图片 联系在一起	182	15.5.2	文档	217
13.4.3	Assets 目录	183	15.6	XCTest 断言	218
13.4.4	向 Assets Catalog 中 添加图片	184	15.6.1	简单测试	218
13.4.5	图标和启动图片	186	15.6.2	等价	219
13.5	小结	188	15.6.3	异常	219
第14章	添加编辑器	189	15.7	小结	220
14.1	计划	189	第16章	度量和分析	221
			16.1	速度	221

16.1.1	Debug 调试器	222	18.5	Gatekeeper 和 Developer ID	262
16.1.2	Instruments	224	18.5.1	获取 Developer ID	262
16.2	XCTest 和性能	231	18.5.2	使用 Developer ID	263
16.3	内存	232	18.5.3	限制	264
16.4	小结	233	18.6	发布构建	264
第17章	iOS扩展	234	18.6.1	基本的构建设置	265
17.1	添加 Today Target	235	18.6.2	调整构建设置	265
17.2	设计挂件	235	18.6.3	构建	267
17.3	框架内的共享库	239	18.7	小结	268
17.4	Today 扩展	244			
17.5	构建依赖	247	第三部分	Mac OSX 开发中的 Xcode	
17.6	结果	248	第19章	开始制作OS X应用程序	270
17.7	小结	248	19.1	目标	270
第18章	配置	249	19.2	开始	271
18.1	Apple 开发者计划	249	19.3	模型	274
18.1.1	通用 (App Store) 计划	250	19.3.1	从 iOS 移植	274
18.1.2	企业计划	250	19.3.2	添加实体	275
18.2	iOS 的配置	251	19.4	制作菜单	278
18.2.1	你会看到什么	251	19.4.1	Target/Action	279
18.2.2	注册你的 App	253	19.4.2	First Responder	280
18.2.3	保护资源	254	19.4.3	将数据载入 LeagueDocument	281
18.2.4	提交 iOS 应用程序	255	19.4.4	适应托管文档	281
18.3	功能编辑器	257	19.4.5	测试命令	283
18.3.1	OS X 独有的功能	257	19.4.6	为 League 数据确定 一个类型	284
18.3.2	iOS 和 OS X 中都有的功能	257	19.4.7	确定 App 如何处理 League 文件	285
18.3.3	iOS 功能	258	19.4.8	应用程序和文档图标	286
18.4	OS X 沙箱	259	19.5	小结	287
18.4.1	为什么要选择沙箱	261			
18.4.2	沙箱的缺点	261			

第20章 绑定：连接OS X 应用程序··· 288

- 20.1 OS X 中的 Storyboard Segue··· 288
- 20.2 构建文档窗口····· 290
 - 20.2.1 载入窗口····· 290
 - 20.2.2 表视图····· 291
- 20.3 填充表——绑定····· 294
 - 20.3.1 对象控制器····· 294
 - 20.3.2 将表绑定到 Team····· 296
 - 20.3.3 将列绑定到 Team 属性··· 297
- 20.4 League 文档数据的结构····· 299
 - 20.4.1 从 League 表到 Source List·· 299
 - 20.4.2 捕获 Team 的选择····· 301
 - 20.4.3 从 Team 到表····· 303
 - 20.4.4 Passer 部分····· 304
- 20.5 小结····· 311

第21章 本地化····· 312

- 21.1 本地化的工作原理····· 312
- 21.2 添加本地化····· 313
 - 21.2.1 Base Localization····· 313
 - 21.2.2 为什么需要
Base Localization····· 314
- 21.3 哪些需要本地化····· 315
 - 21.3.1 游戏细节视图：布局····· 315
 - 21.3.2 游戏细节视图：代码····· 317
 - 21.3.3 模块和命名空间····· 319
- 21.4 法语本地化····· 320
 - 21.4.1 添加语言环境····· 320
 - 21.4.2 由简入繁：Credits.rtf····· 323
 - 21.4.3 本地化 Main.storyboard··· 325
 - 21.4.4 本地化资源····· 328

21.4.5 本地化程序字符串····· 329

21.4.6 genstrings····· 330

21.4.7 xdiff 文件····· 331

21.4.8 其他····· 332

21.5 本地化系统字符串····· 332

21.6 小结····· 334

第22章 程序包····· 335

- 22.1 简单的包：RTFD····· 335
- 22.2 bundle····· 336
- 22.3 应用程序的 bundle····· 337
- 22.4 Info.plist File····· 338
- 22.5 Info.plist 中用于应用程序的键··· 340
 - 22.5.1 用于 iOS 和 OS X 的键··· 340
 - 22.5.2 OS X 中使用的键····· 342
 - 22.5.3 iOS 中使用的键····· 345
- 22.6 小结····· 347

第23章 属性列表····· 348

- 23.1 属性列表数据类型····· 348
- 23.2 编辑属性列表····· 350
 - 23.2.1 属性列表编辑器····· 352
 - 23.2.2 为什么不使用属性列表
编辑器····· 356
- 23.3 其他格式····· 358
 - 23.3.1 文本属性列表····· 358
 - 23.3.2 二进制属性列表····· 359
 - 23.3.3 JSON····· 360
- 23.4 特殊的属性列表····· 361
- 23.5 小结····· 362

第四部分 Xcode 任务集

第24章 Xcode中的文档 364

- 24.1 Quick Help 364
 - 24.1.1 Inspector 364
 - 24.1.2 弹出窗口 365
- 24.2 快速打开 366
- 24.3 帮助 367
- 24.4 文档窗口 368
 - 24.4.1 侧边导航栏 368
 - 24.4.2 侧边栏的内容表 368
 - 24.4.3 类信息 368
 - 24.4.4 查找和导航 369
- 24.5 保持同步 371
- 24.6 自定义的 Quick Help 373
 - 24.6.1 C 语言系列的文档 373
 - 24.6.2 Doxygen 375
 - 24.6.3 运行 Doxygen 380
 - 24.6.4 安装文档集合 380
- 24.7 Swift 和 reStructuredText 381
- 24.8 小结 383

第25章 Xcode构建系统 384

- 25.1 Xcode 如何组织构建过程 384
- 25.2 构建变量 387
- 25.3 设置的层级 388
- 25.4 编辑构建变量 389
- 25.5 配置 390
- 25.6 配置文件 392
 - 25.6.1 创建配置文件 392
 - 25.6.2 SDK 和指定的架构设置 393

- 25.6.3 预处理 xcconfig 文件 393

25.7 命令行工具 394

- 25.7.1 xcodebuild 394
- 25.7.2 xcode-select 395
- 25.7.3 xcrun 396

25.8 自定义构建规则 397

25.9 构建日志 398

25.10 简单的构建记录 400

25.11 小结 408

第26章 Instruments 409

- 26.1 Instruments 是什么 409
- 26.2 运行 Instruments 410
- 26.3 记录文档窗口 411
 - 26.3.1 工具栏 412
 - 26.3.2 记录区域 414
 - 26.3.3 Detail 区域 415
 - 26.3.4 Extended Detail 区域 415
 - 26.3.5 库 418
- 26.4 追踪 419
 - 26.4.1 记录 419
 - 26.4.2 保存和重新打开 422
 - 26.4.3 不启动 instrument 记录 422
- 26.5 Instruments 422
 - 26.5.1 Behavior 423
 - 26.5.2 Core Data 423
 - 26.5.3 Dispatch 423
 - 26.5.4 Filesystem 423
 - 26.5.5 Graphics 424
 - 26.5.6 Input/Output 424
 - 26.5.7 Master Tracks 424

26.5.8	Memory	425
26.5.9	系统	425
26.5.10	System-iOS Energy Instruments	427
26.5.11	线程 / 锁	427
26.5.12	记录	428
26.5.13	UI Automation	428
26.5.14	用户界面	428
26.6	自定义 Instruments	429
26.7	模板	430
26.7.1	全部平台	430
26.7.2	仅限 iOS	430
26.7.3	仅限 Mac	431
26.8	小结	431
第27章 调试		432
27.1	Scheme 选项	432
27.1.1	Info	432
27.1.2	参数	433
27.1.3	选项	433
27.1.4	诊断	435

27.2	使用断点做更多的工作	435
27.3	视图层级	437
27.4	lldb 命令行	439
27.5	小技巧	441
27.6	小结	444
第28章 零散的知识		445
28.1	技巧	445
28.1.1	一般技巧	445
28.1.2	代码折叠	449
28.1.3	Assistant 编辑器	450
28.1.4	Instruments 和调试	452
28.1.5	构建	452
28.2	陷阱	454

第五部分 附录

附录A 一些构建变量 460

附录B 资源 474