

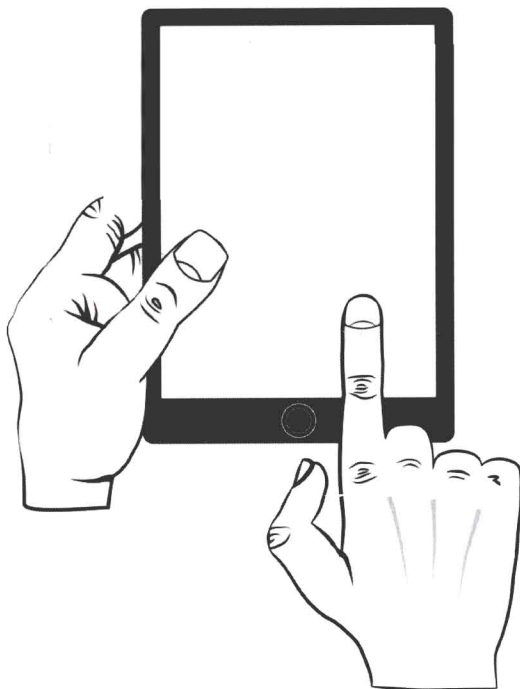
资深iOS开发工程师撰写，Amazon五星畅销书，国际Mac和iPhone开发者社区CocoaHeads联合创始人Mark Dalrymple等数位专家联袂推荐！

全书以一个经典的案例为驱动，全面而系统地讲解了开发iPad应用需要掌握的技术知识、方法和流程，可操作性强，是系统学习和实践iPad应用开发的经典之作！

Learning iPad Programming
A Hands-On Guide to Building iPad Apps with iOS 5

iPad应用开发实践指南

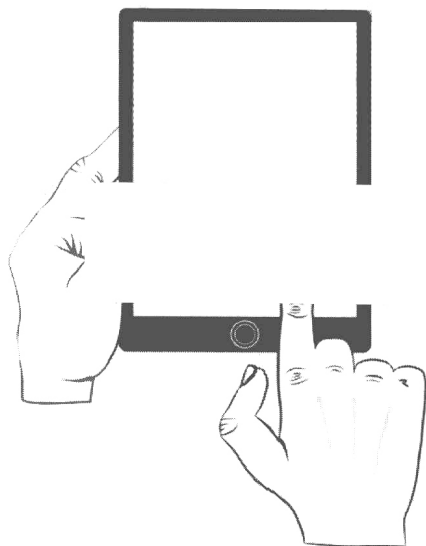
(美) Kirby Turner Tom Harrington 著
张菲 译



Learning iPad Programming
A Hands-On Guide to Building iPad Apps with iOS 5

iPad应用开发实践指南

(美) Kirby Turner Tom Harrington 著
张菲 译



机械工业出版社
China Machine Press

本书在国外 iOS 开发者社区内颇受推崇,由资深的 iOS 开发工程师撰写,国际 Mac 和 iPhone 开发者社区 CocoaHeads 联合创始人 Mark Dalrymple 等数位专家联袂推荐!相比同类书,它有两大特色:第一,全书以一个真实的 PhotoWheel 应用(可在 APP Store 上下载)为驱动,既以迭代的方式逐步讲解了整个应用的开发过程,又全面而系统地讲解了开发 iPad 应用所需要掌握的技术知识、方法、流程,可操作性强,是系统学习和实践 iPad 应用开发的经典著作;第二,本书根据 iPad 和 iPhone 在设备特性上的差异深刻地指出了 iPad 应用开发和 iPhone 应用开发之间的不同之处,对于深度的 iPad 应用开发者来说,本书是极为宝贵的!

全书有 27 章,分为三部分:第一部分(第 1~7 章)详细介绍了开发 iOS 应用应具备的基础知识,包括 Xcode、Interface Builder、Objective-C 和 Cocoa 等,以及如何为 iPad 配置信息和如何构建应用程序;第二部分(第 8~24 章)是本书的核心,详细讲解了 PhotoWheel 的完整开发过程以及所需的理论知识,具体包括创建主从复合应用程序、使用表格视图、用视图工作、使用触屏手势、添加照片、数据持久化、故事板、视图控制器、创建照片浏览器、支持设备旋转、用 AirPrint 打印、发送电子邮件、与 iCloud 同步、用 AirPlay 制作幻灯片和 Core Image 的视觉特效等;第三部分(第 25~27 章)介绍了应用程序的调试和发布。最后是一个附录,介绍了如何创建 iOS 开发账号、下载 iOS SDK,以及在 Mac 计算机上安装开发工具。

Authorized translation from the English language edition, entitled Learning iPad Programming: A Hands-On Guide to Building iPad Apps with iOS 5, 1E, 9780321750402 by Kirby Turner, Tom Harrington, published by Pearson Education, Inc., Copyright © 2012.

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 © 2013.

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。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2012-1281

图书在版编目(CIP)数据

iPad 应用开发实践指南 / (美)特纳(Turner, K.), (美)哈林顿(Harrington, T.)著;张菲译. —北京:机械工业出版社, 2013.1

(华章程序员书库)

书名原文: Learning iPad Programming: A Hands-On Guide to Building iPad Apps with iOS 5

ISBN 978-7-111-40617-4

I. i… II. ①特… ②哈… ③张… III. 便携式计算机—基本知识 IV. TP368.32

中国版本图书馆 CIP 数据核字(2012)第 287269 号

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:谢晓芳

北京市荣盛彩色印刷有限公司印刷

2013 年 1 月第 1 版第 1 次印刷

186mm×240mm·33 印张

标准书号: ISBN 978-7-111-40617-4

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) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

译者序

2001年，苹果公司推出了 iPhone、iPad 系列产品。这些产品一面市就引起巨大的轰动，甚至引发了抢购热潮。最近有消息传出，苹果公司超越微软公司，成为全球市值最高的公司。追逐潮流的年轻人都以拥有 iPhone、iPad 为荣。iPad 平板电脑的出现，标志着后 PC 时代的到来，即台式机与嵌入式系统正往一起融合，而不像我们先前认为的，各自是独立的电子产品。如今的嵌入式系统，其操作系统和硬件资源的强大程度都直逼台式机，甚至超越后者。这种趋势使 PC 程序员和嵌入式系统固件程序员都面临着更复杂的技术和平台，也带来了前所未有的整合机会，因为先前独立的 PC 软件开发和嵌入式系统固件开发变得不再那么泾渭分明了。

iPhone、iPad 的流行促使不少程序员转向这些移动应用程序的开发。其中要涉及不少苹果公司特有的内容，例如，配置设备信息、发布。这些都是未涉足此领域的程序员所不曾有过的经历。Kirby Turner 与 Tom Harrington 合著的这本书不仅为我们提供了 iPhone、iPad 上（主要是 iPad）开发应用程序（开发环境是 Xcode 4.2，运行环境是 iOS 5）的基础知识，介绍了软件应用的许多技巧和怎样充分利用开发工具和操作系统的最新特性，还详细讲述了开发苹果公司产品所运行的应用程序的独到之处。后一部分内容倒不讨论什么技术难题，但它对那些有着丰富编程经验，却未接触过苹果公司产品开发的高级程序员来说，能够马上用来使应用程序运行起来，因此显得尤其珍贵。

更难得的是，本书采用一种实战的做法，饶有兴趣地提供了一个生动的照片浏览应用程序 PhotoWheel 项目。你将在前面章节建立这个项目，随着后续章节内容的不断深入，陆续添加各种代码来实现某项功能。等读完本书后，你会惊喜地发现，你亲手制作的 PhotoWheel 项目已可编译生成功能完整、强大的应用程序！你可以向周围的同学、同事、家人炫耀，得到他们的赞许和钦佩，让他们实际使用它。这是何等令人兴奋！这样的做法能让你随着阅读本书而逐步积累成就感和自信心。通过本书的引导，你能够很快成长为一名 iOS 程序员，开发出自己的 iPad 应用程序。这在同行业的软件指导书中是不多见的，也是效果很好的做法，由此也体现了作者的良苦用心。

原书中出现单词 need to、your 的地方比比皆是，翻译为使中文流畅，我尽量交替使用各种词汇；而许多时候为避免啰唆，在不发生歧义的前提下 your 一般都没有译出。此外，由于英语中大量用到从句，有别于中文习惯。在先前译作中，我经常要用“后者”指代前句提到的最后一个代词或名词，有读者认为这样表达不明确，容易糊涂。因此在本书的译文中，基本上没有出现“后者”，而是明确指出为相应的代词或名词。作者在书中提及一些参考书，如有中文翻译版出版的，我以中文书名为准，并给出了译者、出版社、出版日期、书号等信息；尚无中文翻译版书籍的，我则自行翻译。倘若与日后可能出版的中文翻译版书名有出入，请读者在检索时以英文书名为准。

机械工业出版社的编辑关敏女士作为本书的联络人，在翻译本书的过程中为我提供了周到细致的服务。其他编辑对译稿做了高效率、全方位、细致的审校工作，指出了不少问题。这些我都一一接受，对提高本书的翻译质量起了很大的作用。细心的编辑体现了机械工业出版社员工崇高的职业

素养，在此我向各位编辑致以崇高的谢意！

最后也是最重要的，我想感谢选择阅读本书的读者。市面上讲述 iPad 编程的著作尽管不多，但并非仅此一部，而且每个人的时间和精力都很宝贵，您愿意研读本书，愿意为它投入时间和精力，表明了您对它的信任和期望。我希望本书能帮助您达到目标。祝您成功！

译文力争以通俗通畅的汉语再现原著的知识。由于译者水平有限，可能存在某些疏漏之处，请读者不吝赐教。您的意见、建议能够帮助我们改善本书的质量。也欢迎发邮件到 zhangfei97@163.com，与我交流本书相关的信息，再次感谢！

张菲

序

为什么许多书不够好？

知道我的人，不管认识有多久，都会觉得我是个彻头彻尾的书生。我喜欢看书。精心编写的书是用来自学的最便宜、最快速的工具。记得在我个人的职业发展过程中，曾有一些书对我有着非同凡响的意义——像 Bertrand Meyer 写的《Object Oriented Software Construction》、Scott Knaster 和 Stephen Chernicoff 早期的 Mac 计算机编程书、Dave Mark 的 C 语言编程书籍、Robert C. Martin 的那本《Designing Object Oriented C++ Applications Using the Booch Method》，虽然书名有些吓人，但充满了让人喜出望外的亮点；当然还有 W. Richard Stevens 写的比较近的 UNIX 与网络编程书籍。我至今还记得这些经典书卷中的课程，即使有些是在 25 年前学的。

遗憾的是，并非所有书籍都一样。我也曾看过有些糟糕至极的书。当我最初从 Mac 编程转向 iPhone 编程时，我得到的有些书很棒，有些书却很烂。真的是烂，就好像有人把《Instant Visual Basic Programming Guide for Complete Dummies》的序列号揭下换成 iPhone 的图片一样。当我在书店翻阅书时，有一本早期的 iPad 书几乎每页都有错误。有些只是录入错误；有些是细微错误，如果你还没有用 Cocoa 工作过几年，这是可以理解的；有些却是完全错误的建议，显然是并不知道自己在做什么事的人提出来的。当你把辛苦挣来的钱花到一本书上，肯定带有某些信任的期望，辜负这种信任是无法饶恕的。

所以本书物有所值吗？它能归到我的第一类书（出类拔萃），还是第二类（凑合还行）？问得好，很高兴你提到这个问题。

首先，好书应当涵盖其主题，而且涵盖得恰到好处。本书仅从其分量来看，就知道囊括了大量内容。当然，这是假定你手里已经拿到此书的印刷版。《战争与和平》的分量和《小王子》在电子书上不分伯仲，所以难以说清。单是扫一眼目录内容，你就能看到此书讲述了很多东西。还有个度量是材料的相关性，而它全是相关的内容。它涉及的基础知识包括安装开发工具、模型-视图-控制器（Model-View-Controller）、表格视图、UIViewControllor、导航视图、对设备旋转模式的处理。也有更高级的话题，例如，使用 Web 服务、媒体库、触摸手势、数据持久化和原始的复杂做法，即为苹果设备配置信息。此外还有些尖端的内容，如故事板、AirPrint、AirPlay、iCloud 和 Core Image。Kirby 和 Tom 花费了几个月时间来处理预发布软件的古怪问题，使你免受这些问题的困扰。

写得非常好的书应当及时而不浪费读者的时间。我是在 iPad 设备面世后三个月看到第一本 iPad 编程书籍的。该书不能向读者传递 iPad 的完整信息，因为还没有人那么快拿到设备。它是抢先进入了市场，却是匆忙进入的。本书的核心距我写这篇文章已经一年有余了。好书是要花时间来达到高水平的美。

卓越的书籍会超越其主题。顾名思义，从书名上不难猜想本书只是要以简单的方式来介绍 iPad 编程。“视图看着很酷！”“嘿，点一下按钮！”但不限于这些。没有多少书能在贯穿全书的叙述中使

用一个生动的项目，而且不断随着叙述而进化。很多书不这么做的理由，是很难做好。重要的工具功能被蹩脚地塞到怪异的地方，因为作者没有预先规划好设计。而本书带你从设计开始，实现一次性的原型，接着又实现真实的应用程序。

然后它还走得更远。没有多少书会讨论设计的内在思路，本书却做到了。涉及调试内在原理的书更是凤毛麟角。调试是程序员日常生活的基本组成部分，鲜有书籍花上一两段以上的篇幅来讲述调试。本书则对这个话题采用了一整章，它远不只是讲述用调试器单步调试。在我阅读这些章的初稿以便在此序中写点什么时，当我翻到第 25 章时不由发出了赞叹。我喜欢调试，乐于看到这么一个重要的话题在这本看似入门级的书籍里详细提及。正如你能看出的那样，我喜欢了解事物，并从第 25 章有所收获。

最后，那些创作卓越书籍的人也会超出常规。Mac 和 iPhone 社区相当小，但沟通得很好。跟着信任的人，你会学得很快。我先前提到的许多糟糕书籍是我从未听说过的一些人写的，也从此再也未听说过。他们没有博客，在会议上也没有露面，在开发社区上也没有留下足迹。他们只是匆匆过客。

Kirby 和 Tom 则不同。他们是确实存在的。他们有博客。Tom 是一本 Core Data 书的作者。他们已经发布了产品。他们有满意的客户。他们组织会议，并在会议上发表讲话。他们还组织了 CocoaHeads 分会。他们为改善社区投入了大量的时间，在他们请我写此序时，我感到很荣幸，原因正在于此。

也许你已经感觉到，我对这本书非常看好。有许多介绍 iOS 编程的优秀书籍。我之所以建议阅读它们（至少是其中的一些好书），是因为 iOS 是个广泛的话题，即使 Kirby 和 Tom 也不能在一本书中涵盖你要知道的所有内容。但如果你专门针对 iPad，本书将是很好的选择。我能够感觉到它会成为对你们有些人有影响的一部书。

——Mark Dalrymple，国际 Mac 和 iPhone 程序员社区 CocoaHeads 的协同创办人、
《Advanced Mac OS X Programming: The Big Nerd Ranch Guide》的作者
2011 年 11 月 12 日

目 录

译者序

序

前言

第一部分 入门

第 1 章 你的第一个应用程序	1
1.1 创建“Hello World”项目	1
1.2 在屏幕上显示文字	5
1.3 说“Hello”	6
1.4 小结	10
第 2 章 Xcode 入门	11
2.1 集成开发环境	11
2.2 工作区窗口	12
2.2.1 工具栏	12
2.2.2 导航区	12
2.2.3 编辑区	13
2.2.4 工具区	14
2.2.5 调试区	15
2.3 首选项	15
2.3.1 字体与颜色	16
2.3.2 文本编辑	16
2.3.3 按键绑定首选项	18
2.3.4 代码补全	19
2.4 开发者文档	20
2.5 编辑器	21
2.6 项目设置	22
2.7 方案	23
2.8 组织者	24

2.9 其他 Xcode 工具	25
2.10 小结	25

第 3 章 Interface Builder 入门

3.1 Interface Builder	26
3.2 IB 怎样工作	27
3.3 着手使用 IB	27
3.3.1 选取与复制对象	29
3.3.2 对齐对象	29
3.3.3 布局矩形框	31
3.3.4 修改状态	31
3.4 将 NIB 与代码关联起来	33
3.4.1 在代码中定义插座变量	33
3.4.2 使用辅助编辑器	36
3.5 故事板	37
3.6 小结	37

第 4 章 Objective-C 入门

4.1 什么是 Objective-C	38
4.2 玩转 Objective-C	39
4.2.1 动手写代码	40
4.2.2 对象	41
4.2.3 类	42
4.2.4 NSObject	43
4.2.5 接口	43
4.2.6 实例变量	44
4.2.7 声明属性	45
4.2.8 方法	47
4.2.9 实现	47
4.2.10 合成	48
4.2.11 init	49

4.2.12	super	49	5.3.22	UINavigationController	69
4.2.13	flip	49	5.3.23	UIToolbar	69
4.2.14	选择器	50	5.3.24	UITabBar	69
4.2.15	点语法	51	5.3.25	UIBarButtonItem	69
4.3	CoinTosser 类的用法	51	5.3.26	UISegmentedControl	70
4.4	内存管理	52	5.4	Cocoa 中常见的设计模式	70
4.5	小结	53	5.4.1	模型-视图-控制器模式	70
第 5 章	Cocoa 入门	54	5.4.2	目标-动作模式	71
5.1	Cocoa 栈	54	5.5	小结	71
5.2	Foundation	55	第 6 章	为 iPad 配置信息	72
5.2.1	数据类型	56	6.1	关于 iOS 配置门户	72
5.2.2	集合类	60	6.2	提供信息的过程: 概括说明	73
5.2.3	工具类与函数	61	6.2.1	设备 ID 是什么	73
5.3	UIKit	65	6.2.2	什么是应用程序 ID	74
5.3.1	UIApplication	65	6.2.3	什么是开发信息提供概述	75
5.3.2	UIWindow	65	6.3	设立开发用的机器	76
5.3.3	UIScreen	65	6.3.1	请求开发许可证	76
5.3.4	UIView	65	6.3.2	提交 CSR 以求批准	78
5.3.5	UIViewController	65	6.3.3	下载并安装许可证	79
5.3.6	UIWebView	65	6.4	配置设备	80
5.3.7	UILabel	66	6.5	使用 iOS 配置门户	81
5.3.8	UITextField	66	6.5.1	添加设备 ID	82
5.3.9	UITextView	66	6.5.2	添加应用程序 ID	83
5.3.10	UIButton	66	6.5.3	创建开发信息提供概述	84
5.3.11	UITableView 与 UITableViewCell	67	6.5.4	下载开发信息提供概述	85
5.3.12	UIScrollView	67	6.5.5	安装开发信息提供概述	86
5.3.13	UIPageControl	67	6.6	小结	86
5.3.14	UIPickerView	67	第 7 章	应用程序设计	87
5.3.15	UIDatePicker	67	7.1	定义应用程序	87
5.3.16	UISwitch	67	7.1.1	应用程序名称	88
5.3.17	UISlider	68	7.1.2	应用程序概述	88
5.3.18	UIMenuController 和 UIMenuItem	68	7.1.3	功能清单	88
5.3.19	UIImage	68	7.1.4	目标用户	89
5.3.20	UIImageView	68	7.1.5	再度审视功能清单	90
5.3.21	UINavigationController	68	7.1.6	有竞争关系的产品	90
			7.1.7	App Charter 示例	90

7.2 用户界面构思	92
7.2.1 阅读 HIG (iOS 人机 界面指南)	92
7.2.2 使你的应用程序 “令人心动”	92
7.2.3 为设备设计	93
7.2.4 人们使用 iOS 设备与使用 网页或桌面计算机的方式 不一样	93
7.2.5 戴上工业设计师的帽子	94
7.2.6 比喻	94
7.2.7 声音效果	94
7.2.8 定制现有的控件	95
7.2.9 聘请设计师	96
7.3 原型图	96
7.3.1 什么是原型图	96
7.3.2 要对什么设计原型图	97
7.3.3 使用的工具	97
7.4 建立原型	100
7.4.1 什么是原型	100
7.4.2 怎样创建原型应用程序	101
7.5 小结	101

第二部分 构建 PhotoWheel

第 8 章 创建主从复合应用程序	103
8.1 构建原型应用程序	103
8.1.1 什么是分割视图控制器	104
8.1.2 创建新的项目	105
8.1.3 使用模拟器	105
8.2 更进一步的观察	106
8.2.1 项目结构	107
8.2.2 应用程序委派	107
8.2.3 启动选项	111
8.2.4 其他 UIApplicationDelegate 方法	112
8.3 UISplitViewController 详解	112
8.3.1 指定分割视图控制器的委派	115

8.3.2 细节视图控制器	116
8.3.3 主视图控制器	116
8.4 小结	117
8.5 习题	117
第 9 章 使用表格视图	118
9.1 首要的事情	118
9.2 深层剖析	120
9.2.1 UITableView	120
9.2.2 UITableViewCell	121
9.2.3 UITableViewDelegate	121
9.2.4 UITableViewDataSource	121
9.2.5 UITableViewController	121
9.3 操作表格视图	121
9.3.1 一个简单的模型	121
9.3.2 显示数据	123
9.3.3 添加数据	127
9.3.4 编辑数据	139
9.3.5 删除数据	143
9.3.6 重排数据	144
9.3.7 选取数据	145
9.4 小结	147
9.5 习题	148

第 10 章 用视图工作

10.1 定制视图	149
10.2 轮状视图	150
10.3 旋转木马视图	156
10.4 Photo Wheel 视图单元	161
10.5 小结	164
10.6 习题	164

第 11 章 使用触屏手势

11.1 触摸手势简介	165
11.1.1 预定义的触屏手势	166
11.1.2 手势种类	166
11.1.3 怎样使用手势识别类	166
11.2 定制触屏手势	169
11.2.1 创建拨动手势识别类	170

11.2.2 拨动手势识别类的用法	172	13.4.1 Core Data 的模型编辑器	206
11.3 小结	176	13.4.2 添加实体	206
11.4 习题	176	13.4.3 创建 NSManagedObject 子类	209
第 12 章 添加照片	177	13.5 向模型对象添加定制代码	211
12.1 两种途径	177	13.5.1 用 Core Data 读取和保存 相册	215
12.1.1 Assets Library	177	13.5.2 使用 Core Data 向相册 添加新照片	218
12.1.2 图片捡拾控制器	178	13.5.3 用 Core Data 显示相册中 的照片	220
12.2 图片捡拾控制器的用法	179	13.6 直接使用 SQLite	221
12.2.1 使用动作单	181	13.7 小结	221
12.2.2 UIImagePickerController 的 用法	184	13.8 习题	221
12.2.3 保存至摄像头胶卷	187	第 14 章 Xcode 中的故事板	222
12.3 小结	188	14.1 什么是故事板	222
12.4 习题	188	14.1.1 故事板的用法	223
第 13 章 数据持久化	189	14.1.2 场景	224
13.1 数据模型	189	14.1.3 过渡	224
13.1.1 照片	189	14.2 对 PhotoWheel 创作故事板	225
13.1.2 相册	189	14.2.1 工作区	225
13.1.3 前瞻性考虑	189	14.2.2 添加主故事板	226
13.2 使用属性清单来构建模型	190	14.2.3 设置 UIMainStoryboardFile	227
13.2.1 什么是属性清单	190	14.2.4 更新 AppDelegate	228
13.2.2 建立数据模型	190	14.2.5 添加图片	228
13.2.3 读取和保存相册	191	14.2.6 应用程序图标	229
13.2.4 向相册添加新照片	195	14.2.7 初始的视图控制器	229
13.2.5 显示相册中的照片	198	14.2.8 另一个场景	230
13.3 使用 Core Data 构建模型	200	14.2.9 创建过渡	232
13.3.1 什么是 Core Data	200	14.3 小结	233
13.3.2 托管对象和实体描述	200	14.4 习题	233
13.3.3 托管对象语境	201	第 15 章 视图控制器详解	234
13.3.4 持久存储和持久存储 协调器	202	15.1 实现视图控制器	234
13.3.5 向 PhotoWheelPrototype 添加 Core Data	202	15.2 过渡	236
13.3.6 添加 Core Data 框架	202	15.2.1 创建定制过渡	237
13.3.7 建立 Core Data 栈	203	15.2.2 设置过渡	237
13.4 在 PhotoWheel 中 使用 Core Data	206		

15.2.3 实现定制过渡	239	17.5 删除照片	342
15.2.4 编译之前的工作	241	17.6 小结	348
15.3 定制弹出转换	243	17.7 习题	348
15.4 容器视图控制器	245	第 18 章 支持设备旋转	349
15.4.1 创建容器视图控制器	246	18.1 怎样支持旋转功能	349
15.4.2 添加子场景	246	18.1.1 所支持的设备放置模式	350
15.4.3 添加子视图控制器	248	18.1.2 使用自动尺寸调整功能	350
15.4.4 修正定制的推式过渡	251	18.2 自定义旋转	351
15.5 小结	251	18.2.1 支持旋转相册	355
15.6 习题	252	18.2.2 旋转相册场景	355
第 16 章 构造主屏幕	253	18.2.3 调整 WheelView 类的代码	356
16.1 重用原型代码	253	18.2.4 旋转 About 视图	358
16.1.1 复制文件	254	18.2.5 旋转照片浏览器	358
16.1.2 Core Data 模型	255	18.3 修正麻烦点	358
16.1.3 WheelView 要做的修改	260	18.3.1 修正照片浏览器	358
16.2 显示相册	270	18.3.2 修正主屏幕	363
16.2.1 实现相册集视图控制器	271	18.4 启动画面	365
16.2.2 设置托管对象语境	277	18.5 小结	367
16.2.3 添加相册	278	18.6 习题	367
16.3 管理相册	279	第 19 章 用 AirPrint 打印	368
16.3.1 选取相册	279	19.1 打印功能如何工作	368
16.3.2 对相册起名	283	19.1.1 打印中心 Print Center	368
16.3.3 修正工具栏的显示	288	19.1.2 能打印的前提	369
16.3.4 删除相册	289	19.1.3 打印 API	369
16.4 更美观的相册缩略图	291	19.2 向 PhotoWheel 添加打印功能	369
16.5 添加照片	294	19.3 小结	372
16.6 显示照片	299	19.4 习题	372
16.6.1 使用 GridView 类	308	第 20 章 发送电子邮件	373
16.6.2 构建图片栅格视图单元	312	20.1 工作原理	373
16.7 小结	315	20.2 SendEmailController 类	374
16.8 习题	315	20.2.1 引入 SendEmailController 类	375
第 17 章 创建照片浏览器	316	20.2.2 SendEmailController 的 用法	378
17.1 使用滚动视图	316	20.3 小结	383
17.2 启动照片浏览器	324	20.4 习题	383
17.3 添加 Chrome 效果	332		
17.4 放大缩小	336		

第 21 章 Web 服务	384	22.7 小结	430
21.1 基础知识	384	22.8 习题	431
21.2 Flickr	385	第 23 章 用 AirPlay 放映幻灯片	432
21.2.1 向 PhotoWheel 加入 Flickr 支持	386	23.1 外部显示选项	432
21.2.2 更新 Flickr View Controller Scene	388	23.2 应用程序对外部显示器的要求	432
21.2.3 显示 Flickr 场景	389	23.3 外部显示器的应用编程接口	433
21.2.4 包装 Flickr 应用编程接口	391	23.4 向 PhotoWheel 添加幻灯片放映	434
21.2.5 异步下载照片	397	23.4.1 更新故事板	434
21.2.6 FlickrViewController 的实现	402	23.4.2 添加幻灯片放映显示器	435
21.3 其他事宜	410	23.5 管理外部显示器	437
21.4 还缺什么	411	23.6 推进到下一张照片	440
21.5 小结	411	23.7 添加幻灯片放映的用户界面控件	442
21.6 习题	411	23.8 更新照片浏览器	444
第 22 章 与 iCloud 同步	412	23.9 测试和调试时的注意事项	445
22.1 同步使事情变得简单	412	23.10 添加对 AirPlay 设备的支持	446
22.2 iCloud 的各种概念	412	23.11 AirPlay 的运用	447
22.2.1 文件协调器和表现器	413	23.12 小结	448
22.2.2 UIDocument 与 UIManagedDocument	413	23.13 习题	448
22.2.3 无所不在持久存储	414	第 24 章 Core Image 的视觉特效	449
22.3 为设备提供信息的回顾	414	24.1 Core Image 的概念	449
22.3.1 配置应用程序 ID	414	24.2 CIFilter	450
22.3.2 为 iCloud 提供信息	415	24.2.1 滤镜类型	451
22.3.3 配置 iCloud 权利	416	24.2.2 CIFilter 的使用	451
22.4 对 PhotoWheel 的 iCloud 考虑	418	24.3 图片分析	452
22.4.1 同步要做得恰到好处	418	24.3.1 自动增强	453
22.4.2 使用瞬态 Core Data 特性	419	24.3.2 脸部识别	454
22.5 更新 PhotoWheel 以支持 iCloud	419	24.4 为 PhotoWheel 添加 Core Image 特效	454
22.6 用 iCloud 来同步照片	423	24.4.1 新建委派方法	455
22.6.1 使持久存储协调器无所不在	423	24.4.2 滤镜管理的实例变量	456
22.6.2 从 iCloud 接收变更	427	24.4.3 用户界面的添加	456
		24.4.4 创建 CIFilter 特效	461
		24.4.5 滤镜的运用	465

24.4.6	图片自动增强功能的实现	466
24.4.7	实现脸部缩放	466
24.4.8	其他必要的方法	468
24.5	小结	469
24.6	习题	469

第三部分 最后的润色

第 25 章	调试	471
25.1	了解问题所在	471
25.1.1	问题出在哪里	471
25.1.2	重现缺陷	471
25.2	调试的概念	472
25.3	在 Xcode 中调试	472
25.3.1	设置和管理断点	473
25.3.2	定制断点	473
25.3.3	到达断点	474
25.3.4	检查变量	475
25.4	调试示例：外部显示器代码	477
25.5	什么时候真的需要 NSLog	480
25.6	使用 Instruments 剖析代码	481
25.7	小结	485
第 26 章	发布应用程序	486
26.1	发布方法	486

26.2	构建非正式的发布	486
26.2.1	设置非正式发布的配置信息	487
26.2.2	准备构建非正式发布的版本	487
26.3	构建 App Store 的发布版本	488
26.3.1	设置 App Store 条款	489
26.3.2	准备为 App Store 发布的版本	489
26.3.3	接下来怎么做	490
26.4	App Store 流程	490
26.5	为 App Store 提供应用程序信息	491
26.6	App Store 媒体材料	493
26.7	iTunes Connect 的运用	493
26.7.1	用户角色	493
26.7.2	管理应用程序	494
26.8	发送应用程序	494
26.9	更进一步	495
26.10	小结	495
第 27 章	结语	496
附录 A	安装开发工具	498

第一部分 入门

第①章

你的第一个应用程序

在学习时，实际操作出来比别的任何办法都管用。所以我们就入手写一个简单的 iPad 应用程序。要写的首个应用程序就是“Hello World”。没错，这个“Hello World”应用程序超级简单。不过别担心，本书后面会构建复杂得多的应用程序。现在，重要的是你能对相关代码和工具有所了解。

本章主要是让你对用来构建 iPad 应用程序的工具有个大致的认识。如果你已经熟悉了 Xcode，则可以直接跳至第 4 章或第 6 章；倘若你还对 Xcode 比较陌生，请继续阅读。

本章接下来的部分将引导你了解创建 iPad 应用程序所需的步骤。本章不会涉及详细的 Xcode 代码，随后的第 2 章和第 3 章会讲到 Xcode 代码。

注意：在开始前，必须在 Mac 计算机上安装了 Xcode 和 iOS SDK。如果尚未安装这些软件，则请先翻到附录 A 查看如何在 Mac 计算机上创建 iPad 编程环境。当然了，你还需要有一台 Mac 计算机。

1.1 创建“Hello World”项目

从运行 Xcode 开始。如图 1-1 所示，如果你从 Mac App Store 安装了 Launchpad，则 Launchpad 里面就会有 Xcode。否则 Xcode 在 Dock 中可以找到。单击 Xcode 图标运行它。

注意：倘若 Launchpad 或 Dock 上没有 Xcode，则应当添加它。Xcode 可以从硬盘目录/Developer/Applications/下找到（假定默认安装的话）。要想把 Xcode 加到 Dock 上，可以运行 Xcode。在它运行期间，右击（或按住 Control 键单击）Dock 上出现的 Xcode 图标，选择 Options→Keep in Dock 命令。这样就会在程序不运行时 Dock 上仍显示有 Xcode 图标，从而在下次需要时可以方便地运行 Xcode。

执行 Xcode 后弹出的第一个窗口就是 Welcome to Xcode，如图 1-2 所示。在此窗口内可以做几

件事情。可以创建新项目，连接到源代码库，进入 Xcode 4 用户指南（使用 Xcode 的辅导教程），或者访问苹果公司的网站（developer.apple.com）。倘若过去打开过项目，则还会在右侧看到最近项目的清单。选择其中某个项目，单击 **Open** 按钮，就可以打开这个项目。

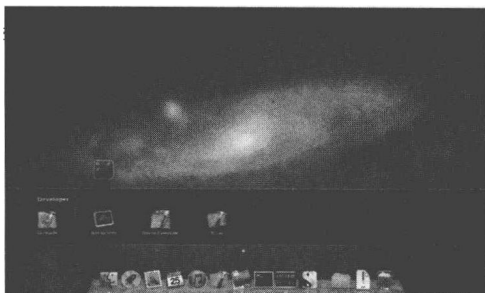


图 1-1 从 Launchpad 里的 Xcode 图标来运行 Xcode。Mac App Store 安装程序将 Xcode 放在 Developer 群组里

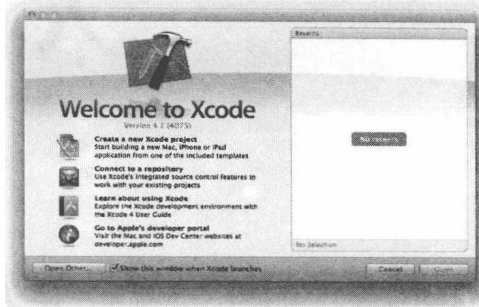


图 1-2 Welcome to Xcode 窗口

窗口左下角是 **Open Other...**按钮，可以单击它来打开硬盘上某个现有的项目。该按钮的右边是个复选框，指定 Welcome to Xcode 窗口是否在执行 Xcode 时显示出来。

注意：如果你是个 Xcode 新手，应当花些时间阅读 Xcode 4 用户指南。此用户指南对 Xcode 工具集有完整的说明。从本书可以学习 Xcode，不过看看苹果公司的官方指南文档还是很有好处的。

因为这里想创建一个新的 iPad 应用程序，所以单击 **Create a new Xcode project** 按钮。将弹出新项目窗口，如图 1-3 所示。在继续下一步前先了解一下这个窗口。正如图 1-3 所示的那样，新项目有三个主要部分：目标类型、项目模板及模板详细信息。

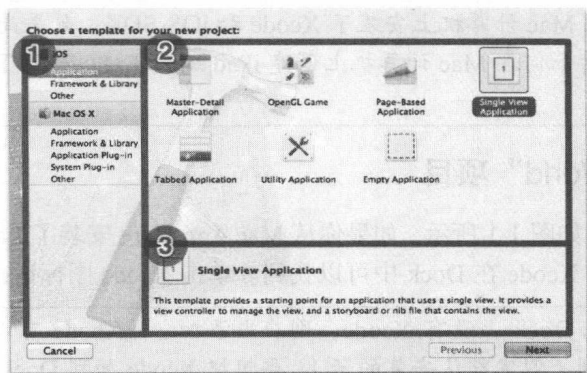


图 1-3 Xcode 的新项目窗口，包含有 Xcode 用户界面的若干部分
(1. 目标类型；2. 项目模板；3. 模板详细信息)

在图 1-3 的第 1 部分中，可以选择目标类型是 iOS 还是 Mac OS X。由于 iPad 应用程序运行在 iOS 上，因此目前可以忽略 Mac OS X 目标类型。在 iOS 下可以构建两种类型的目标：Application 或者 Framework & Library。Application 类型正如其名字所述，用于构建 iPhone 和 iPad 应用程序。

而 Framework & Library 目标类型则用于建立可重用的静态库，目前也可以不用管它。

因为要创建的“Hello World”正是一个应用程序，所以在第 1 部分的 iOS 下选择 Application。然后你会发现第 2 部分的内容随之变化了，目前显示为所选目标类型的模板清单。模板用来生成某个 Xcode 项目的初始文件。

如果你曾花时间玩过 iPad，就可能会注意到有一些共同的应用程序类型，或者说是“风格”。第 2 部分列出的模板清单正是用来快速生成特定风格的应用程序的。例如，要创建一个外观类似于 iPad 上 Mail 的应用程序，可以在这里选择 Master-Detail Application。

应用程序模板

在选择 iOS 作为目标后，可以使用下列类型的应用程序模板。

- ☐ Master-Detail Application: 在有主从复合风格的应用程序，并希望显示使用分割视图控制器时，选此模板。
 - ☐ OpenGL Game: 倘若想利用 OpenGL ES 创建一个游戏，就选择此模板。该模板提供带有 OpenGL 场景的视图，并有个计数器用来使视图产生动画效果。
 - ☐ Page-Based Application: 选择此模板来创建书本或杂志风格的应用程序，后者采用页视图控制器。
 - ☐ Single View Application: 选择此模板来创建只有一个视图的应用程序。
 - ☐ Tabbed Application: 选择此模板来创建通过页签定义各自区域的应用程序。这一模板为首个页签提供页签栏控制器，及视图控制器。
 - ☐ Utility Application: 选择此模板来创建有主视图和另一个可选视图的应用程序。
 - ☐ Empty Application: 该模板提供一个起始点，用以创建任意类型的应用程序。当你想从一个空白项目框架开始做项目时，选择此模板。
-

“Hello World”应用程序只有一个视图，所以选择 Single View Application 模板。此后，你会看到模板细节部分的内容变化了。这个部分显示在项目模板部分中所选模板的简要说明。

单击 Next 按钮将会打开项目选项的屏幕，如图 1-4 所示。项目选项根据模板不同而稍有差异。每个模板都有 Product Name、Company Identifier、Bundle Identifier（基于输入的公司标识会自动生成），以及 Device Family。应用程序模板还可能有 Use Storyboard 复选框、Use Automatic Reference Counting 复选框、Use Core Data 复选框及 Include Unit Tests 复选框等选项。

对于“Hello World”应用程序，在 Product Name 文本框中输入“Hello World”，Company Identifier 框中输入你或公司的名字，使用反向域名格式。例如我个人名字为“com.kirbyturner”，公司名字为“com.whitepeaksoftware”。第 6 章将会解释公司标识与应用程序包标识符之间的关系，以及如何用它们生成应用程序 ID。

下一步，选择 iPad 作为设备系列名。这里有三种设备系列类型可选：iPad 型、iPhone 型和通用型。iPad 型指定本应用程序仅为 iPad 设计，并只运行于其上；iPhone 型指定应用程序仅为 iPhone 设计；通用型则指定应用程序可同时运行于 iPad 和 iPhone。

在这个“Hello World”应用程序中，用不着故事板和单元测试，所以不要选中这两个选项。但要选中 Use Automatic Reference Counting 选项。该选项将在第 4 章的存储管理部分讲解。单击 Next 按钮，指定一个路径来保存 Xcode 项目，然后单击 Create 按钮（如图 1-5 所示）。