iOS 4编程（影印版）

iPhone、iPad和iPod Touch开发基础

涵盖iOS 4.3和Xcode 4

Programming

iOS 4

O'REILLY®
東南大學出版社

Matt Neuburg 著

iOS 4编程（影印版）

Programming iOS 4

常州大学图书馆
藏书章

Matt Neuburg

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

O'Reilly Media, Inc.授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

iOS 4编程：英文/（美）诺伊贝格（Neuburg, M.）著. —影印本. —南京：东南大学出版社，2011.10
ISBN 978-7-5641-2941-5

I. ①i… II. ①诺… III. ①移动电话机－应用程序－程序设计－英文 IV. ①TN929.53

中国版本图书馆CIP数据核字（2011）第173915号

江苏省版权局著作权合同登记
图字：10-2010-416号

©2011 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2011. Authorized reprint of the original English edition, 2011 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2011。

英文影印版由东南大学出版社出版2011。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

iOS 4编程

出版发行：东南大学出版社
地　　址：南京四牌楼2号　　邮编：210096
出 版 人：江建中
网　　址：http://www.seupress.com
电子邮件：press@seupress.com
印　　刷：扬中市印刷有限公司
开　　本：787毫米×960毫米　16开本
印　　张：52.25
字　　数：1023千字
版　　次：2011年10月第1版
印　　次：2011年10月第1次印刷
书　　号：ISBN 978-7-5641-2941-5
定　　价：98.00元（册）

本社图书若有印装质量问题，请直接与读者服务部联系。电话（传真）：025-83792328

Preface

Aut lego vel scribo; doceo scrutorve sophian.

—Sedulius Scottus

With the advent of version 2 of the iPhone system, Apple proved they could do a remarkable thing — adapt their existing Cocoa computer application programming framework to make applications for a touch-based device with limited memory and speed and a dauntingly tiny display. The resulting Cocoa Touch framework, in fact, turned out to be in many ways better than the original Cocoa.

A programming framework has a kind of personality, an overall flavor that provides an insight into the goals and mindset of those who created it. When I first encountered Cocoa Touch, my assessment of its personality was: "Wow, the people who wrote this are really clever!" On the one hand, the number of built-in interface widgets was severely and deliberately limited; on the other hand, the power and flexibility of some of those widgets, especially such things as UITableView, was greatly enhanced over their Mac OS X counterparts. Even more important, Apple created a particularly brilliant way (UIViewController) to help the programmer make entire blocks of interface come and go and supplant one another in a controlled, hierarchical manner, thus allowing that tiny iPhone display to unfold virtually into multiple interface worlds within a single app without the user becoming lost or confused.

Even more impressive, Apple took the opportunity to recreate and rationalize Cocoa from the ground up as Cocoa Touch. Cocoa itself is very old, having begun life as NeXTStep before Mac OS X even existed. It has grown by accretion and with a certain conservatism in order to maintain something like backward compatibility. With Cocoa Touch, on the other hand, Apple had the opportunity to throw out the baby with the bath water, and they seized this opportunity with both hands.

So, although Cocoa Touch is conceptually based on Mac OS X Cocoa, it is very clearly *not* Mac OS X Cocoa, nor is it limited or defined by Mac OS X Cocoa. It's an independent creature, a leaner, meaner, smarter Cocoa. I could praise Cocoa Touch's deliberate use of systematization (and its healthy respect for Occam's Razor) through numerous examples. Where Mac OS X's animation layers are glommed onto views as a kind of afterthought, a Cocoa Touch view always has an animation layer counterpart.

Memory management policies, such as how top-level objects are managed when a nib loads, are simplified and clarified. And so on.

At the same time, Cocoa Touch is still a form of Cocoa. It still requires a knowledge of Objective-C. It is not a scripting language; it is certainly not aimed at nonprogrammers, like HyperCard's HyperTalk or Apple's AppleScript. It is still huge and complicated. In fact, it's rather difficult.

Meanwhile, Cocoa Touch itself evolves and changes. The iPhone System 2 matured into the iPhone System 3. Then there was a sudden sally in a new direction when the iPad introduced a larger screen and iPhone System 3.2. The iPhone 4 and its double-resolution Retina display also ran on a major system increment, now dubbed iOS 4. Every one of these changes has brought new complexities for the programmer to deal with. To give just one simple example, users rightly complained that switching between apps on the iPhone meant quitting one app and launching another. So Apple gave the iPhone 4 the power of multitasking; the user can switch away from an app and then return to it later to find it still running and in the state it was left previously. All well and good, but now programmers must scurry to make their apps compatible with multitasking, which is not at all trivial.

The popularity of the iPhone, with its largely free or very inexpensive apps, and the subsequent popularity of the iPad, have brought and will continue to bring into the fold many new programmers who see programming for these devices as worthwhile and doable, even though they may not have felt the same way about Mac OS X. Apple's own annual WWDC developer conventions have reflected this trend, with their emphasis shifted from Mac OS X to iOS instruction.

The widespread eagerness to program iOS, however, though delightful on the one hand, has also fostered a certain tendency to try to run without first learning to walk. iOS gives the programmer mighty powers that can seem as limitless as imagination itself, but it also has fundamentals. I often see questions online from programmers who are evidently deep into the creation of some interesting app, but who are stymied in a way that reveals quite clearly that they are unfamiliar with the basics of the very world in which they are so happily cavorting.

It is this state of affairs that has motivated me to write this book, which is intended to ground the reader in the fundamentals of iOS. I love Cocoa and have long wished to write about it, but it is iOS and its popularity that has given me a proximate excuse to do so. Indeed, my working title was "Fundamentals of Cocoa Touch Programming." Here I have attempted to marshal and expound, in what I hope is a pedagogically helpful and instructive yet ruthlessly Euclidean and logical order, the principles on which sound iOS programming rests, including a good basic knowledge of Objective-C (starting with C itself) and the nature of object-oriented programming, advice on the use of the tools, the full story on how Cocoa objects are instantiated, referred to, put in communication with one another, and managed over their lifetimes, and a survey of the primary interface widgets and other common tasks. My hope, as with my previous

此为试读，需要完整PDF请访问：www.ertongbook.com

books, is that you will both read this book cover to cover (learning something new often enough to keep you turning the pages) and keep it by you as a handy reference.

This book is not intended to disparage Apple's own documentation and example projects. They are wonderful resources and have become more wonderful as time goes on. I have depended heavily on them in the preparation of this book. But I also find that they don't fulfill the same function as a reasoned, ordered presentation of the facts. The online documentation must make assumptions as to how much you already know; it can't guarantee that you'll approach it in a given order. And online documentation is more suitable to reference than to instruction. A fully written example, no matter how well commented, is difficult to follow; it demonstrates, but it does not teach.

A book, on the other hand, has numbered chapters and sequential pages; I can assume you know C before you know Objective-C for the simple reason that Chapter 1 precedes Chapter 2. And along with facts, I also bring to the table a degree of experience, which I try to communicate to you. Throughout this book you'll see me referring to "common beginner mistakes"; in most cases, these are mistakes that I have made myself, in addition to seeing others make them. I try to tell you what the pitfalls are because I assume that, in the course of things, you will otherwise fall into them just as naturally as I did as I was learning. You'll also see me construct many examples piece by piece or extract and explain just one tiny portion of a larger app. It is not a massive finished program that teaches programming, but an exposition of the thought process that developed that program. It is this thought process, more than anything else, that I hope you will gain from reading this book.

iOS is huge, massive, immense. It's far too big to be encompassed in a book even of this size. And in any case, that would be inappropriate and unnecessary. There are entire areas of Cocoa Touch that I have ruthlessly avoided discussing. Some of them would require an entire book of their own. Others you can pick up well enough, when the time comes, from the documentation. This book is only a beginning — the fundamentals. But I hope that it will be the firm foundation that will make it easier for you to tackle whatever lies beyond, in your own fun and rewarding iOS programming future.

In closing, some version numbers, so that you know what assumptions I am making. At the time I started writing this book, system versions 3.1.3 (on the iPhone) and 3.2 (on the iPad) were most recent. As I was working on the book, iOS 4 and the iPhone 4 came into being, but it didn't yet run on the iPad. Subsequently iOS 4.2 emerged: the first system able to run on both the iPhone and the iPad. At the same time, Xcode was improved up to 3.2.5.

Then, just in time for my final revisions, Xcode 3.2.6 and iOS 4.3 were released, along with the first public version of the long-awaited Xcode 4. Xcode 4 is a thorough overhaul of the IDE: menus, windows, and preferences are quite different from Xcode 3.2.x. At the same time, both Xcode 4 and Xcode 3.2.x can coexist on the same machine and can be used to work on the same project; moreover, Xcode 3.2.x has some specialized

capabilities that Xcode 4 lacks, so some long-standing developers may well continue to use it. This situation presents a dilemma for an author describing the development process. However, for iOS programming, I recommend adoption of Xcode 4, and this book assumes that you have adopted it.

Conventions Used in This Book

The following typographical conventions are used in this book:

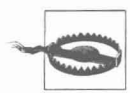
Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Programming iOS 4* by Matt Neuburg (O'Reilly). Copyright 2011 Matt Neuburg, 978-1-449-38843-0."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at *http://my.safaribooksonline.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

http://oreilly.com/catalog/0636920010258/

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

It's a poor craftsman who blames his tools. No blame attaches to the really great tools by which I have been assisted in the writing of this book. I am particularly grateful to the Unicomp Model M keyboard (*http://pckeyboard.com*), without which I could not have produced so large a book so painlessly. I was also aided by wonderful software, including TextMate (*http://macromates.com*) and AsciiDoc (*http://www.methods.co.nz/asciidoc*). BBEdit (*http://www.barebones.com*) helped with its `diff` display. Screenshots were created with Snapz Pro X (*http://www.ambrosiasw.com*) and GraphicConverter (*http://www.lemkesoft.com*); diagrams were drawn with OmniGraffle (*http://www.omnigroup.com*).

The splendid O'Reilly production process converted my AsciiDoc text files into PDF while I worked, allowing me to proofread in simulated book format. Were it not for this, and the Early Release program that permitted me to provide my readers with periodic updates of the book as it grew, I would never have agreed to undertake this project in the first place. I would like particularly to thank Tools maven Abby Fox for her constant assistance.

I have taken advice from two tech reviewers, Dave Smith and David Rowland, and have been assisted materially and spiritually by many readers who submitted errata and encouragement. I was particularly fortunate in having Brian Jepson as editor; he provided enthusiasm for the O'Reilly tools and the electronic book formats, a watchful eye, and a trusting attitude; he also endured the role of communications pipeline when I needed to prod various parts of the O'Reilly machine. I have never written an O'Reilly book without the help of Nancy Kotary, and I didn't intend to start now; her sharp eye has smoothed the bristles of my punctuation-laden style. For errors that remain, I take responsibility, of course.

Table of Contents

此为试读,需要完整PDF请访问: www.ertongbook.com

Part IV. Views

Part V. Interface

此为试读,需要完整PDF请访问: www.ertongbook.com