

ROB NAPIER | MUGUNTH KUMAR

iOS 7 PROGRAMMING

 **PUSHING THE LIMITS**

Develop Advanced Applications
for Apple iPhone®, iPad®, and iPod Touch®



WILEY

Pushing the Limits
iOS 7 Programming

Developing Extraordinary Mobile Apps for
Apple iPhone®, iPad®, and iPod touch®

Rob Napier and Mugunth Kumar



A John Wiley and Sons, Ltd, Publication

This edition first published 2014

© 2014 Rob Napier and Mugunth Kumar.

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

DESIGNATIONS USED BY COMPANIES TO DISTINGUISH THEIR PRODUCTS ARE OFTEN CLAIMED AS TRADEMARKS. ALL BRAND NAMES AND PRODUCT NAMES USED IN THIS BOOK ARE TRADE NAMES, SERVICE MARKS, TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS. THE PUBLISHER IS NOT ASSOCIATED WITH ANY PRODUCT OR VENDOR MENTIONED IN THIS BOOK. THIS PUBLICATION IS DESIGNED TO PROVIDE ACCURATE AND AUTHORITATIVE INFORMATION IN REGARD TO THE SUBJECT MATTER COVERED. IT IS SOLD ON THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING PROFESSIONAL SERVICES. IF PROFESSIONAL ADVICE OR OTHER EXPERT ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL SHOULD BE SOUGHT.

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. Ltd. and/ or its affiliates in the United States and/or other countries, and may not be used without written permission. iPhone, iPad and iPod touch are trademarks of Apple Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Ltd. is not associated with any product or vendor mentioned in the book.

A catalogue record for this book is available from the British Library.

ISBN 978-1-118-81834-3 (pbk); ISBN 978-1-118-81832-9 (ebk); ISBN 978-1-118-81833-6 (ebk)

Set in 9.5/12 MyriadPro-Regular by TCS/SPS, Chennai, India

Printed in United States by Bind-Rite

Dedication

To the Neverwood Five. We're getting the band back together.

Rob

To my mother who shaped the first twenty years of my life

Mugunth

Publisher's Acknowledgements

Some of the people who helped bring this book to market include the following:

Editorial and Production

VP Consumer and Technology Publishing Director: Michelle Leete

Associate Director–Book Content Management: Martin Tribe

Associate Publisher: Chris Webb

Executive Commissioning Editor: Craig Smith

Project Editor: Tom Dinse

Copy Editor: Chuck Hutchinson

Technical Editor: Jay Thrash

Editorial Manager: Jodi Jensen

Senior Project Editor: Sara Shlaer

Editorial Assistant: Annie Sullivan

Marketing

Marketing Manager: Lorna Mein

Marketing Assistant: Polly Thomas

About the Authors

Rob Napier is a builder of tree houses, hiker, and proud father. He began developing for the Mac in 2005, and picked up iPhone development when the first SDK was released, working on products such as The Daily, PandoraBoy, and Cisco Mobile. He is a major contributor to Stack Overflow and maintains the *Cocoaphony* blog (cocoaphony.com).

Mugunth Kumar is an independent iOS developer based in Singapore. He graduated in 2009 and holds a Masters degree from Nanyang Technological University, Singapore, majoring in Information Systems. He writes about mobile development, software usability, and iOS-related tutorials on his blog (blog.mugunthkumar.com). Prior to iOS development he worked for Fortune 500 companies GE and Honeywell as a software consultant on Windows and .NET platforms. His core areas of interest include programming methodologies (Object Oriented and Functional), mobile development and usability engineering. If he were not coding, he would probably be found at some exotic place capturing scenic photos of Mother Nature.

About the Technical Editor

Jay Thrash is a veteran software developer who has spent the past three years designing and developing iOS applications. During his career, he developed a keen interest in the areas of user interaction and interface design.

Prior to settling down as an iOS developer, Jay has worked on a variety of platforms and applications, including flight simulators and web application development. He has also spent over six years in the PC and console gaming industry.

Acknowledgments

One more time, Rob thanks his family for their patience. He also thanks the Triangle Cocoaheads for their great ideas, inspiration, and support. Mugunth thanks his parents and friends for their support while writing this book. Thanks to Wiley, especially Craig Smith, for the continued support, encouragement, and nudging that it takes to get a book out the door. Thanks to Jay Thrash, Tom Dinse, and Chuck Hutchinson for keeping our words intelligible, and for keeping everything on schedule.

Contents

Introduction	1
Part I What's New?	7
Chapter 1 The Brand New Stuff	9
The New UI	9
UIKit Dynamics and Motion Effects	10
Custom Transitions	11
New Multitasking Modes	11
Text Kit	12
Dynamic Type	12
MapKit Integration	12
SpriteKit	12
LLVM 5	12
Enhancements to the Objective-C Language	13
<i>Modules</i>	13
<i>Improvements to ARC</i>	14
Xcode 5	14
NIB File Format Changes	14
Source Code Control Integration	14
Automatic Configuration	15
Refinements to the Debug Navigator	15
Documentation Browser	16
Asset Catalogs	16
Test Navigator	17
Continuous Integration	17
Auto Layout Improvements	17
iOS Simulator	18
Others	18
Summary	18
Further Reading	18
Apple Documentation	18
WWDC Sessions	19

Chapter 2 The World Is Flat: New UI Paradigms	21
Clarity, Deference, and Depth	21
Animations Animations Animations	22
UIKit Dynamics	22
UIMotionEvent	22
Tint Colors	23
Layering and Context through Translucency	23
Dynamic Type	25
Custom Transitions	25
Types of Custom Transitions	26
Transitioning (Migrating) Your App to iOS 7	26
UIKit Changes	27
Custom Artwork	27
Supporting iOS 6	27
<i>App Icon</i>	28
<i>Launch Images</i>	28
<i>Status Bar</i>	28
<i>Navigation Bar</i>	28
<i>Toolbars</i>	28
<i>View Controllers</i>	29
<i>Table View Controllers</i>	29
<i>Pan Gestures</i>	30
<i>Alert Views and Action Sheets</i>	30
Summary	30
Further Reading	30
Apple Documentation	30
WWDC Sessions	31
Part II Getting the Most Out of Everyday Tools	33
Chapter 3 You May Not Know	35
Naming Best Practices	35
Automatic Variables	35
Methods	35
Property and Ivar Best Practices	36
Categories	37
+load	38
Associative References	39
Weak Collections	41

NSCache	42
NSURLComponents	42
CFStringTransform	43
instancetype	44
Base64 and Percent Encoding	45
-[NSArray firstObject]	45
Summary	46
Further Reading	46
Apple Documentation	46
Other Resources	46
Chapter 4 Storyboards and Custom Transitions	47
Getting Started with Storyboards	47
Instantiating a Storyboard	48
Loading View Controllers Within a Storyboard	48
Segues	49
<i>Passing Data</i>	49
<i>Returning Data</i>	49
<i>Instantiating Other View Controllers</i>	50
<i>Performing Segues Manually</i>	50
<i>Unwinding Segues</i>	50
Building Table Views with Storyboards	50
<i>Static Tables</i>	50
<i>Prototype Cells</i>	52
Custom Transitions	52
Advantages	53
Disadvantages – Merge Conflict Hell	53
Summary	54
Further Reading	54
Apple Documentation	54
WWDC Sessions	54
Other Resources	55
Chapter 5 Get a Handle on Collection Views	57
Collection Views	57
Classes and Protocols	57
<i>UICollectionViewController</i>	57
<i>UICollectionViewCell</i>	58
<i>UICollectionViewDataSource</i>	58
<i>UICollectionViewDelegate</i>	58
<i>UICollectionViewDelegateFlowLayout</i>	58

Example	58
<i>Editing the Storyboard</i>	59
<i>Adding Your First Collection View Cell</i>	59
<i>Implementing Your Data Source</i>	60
<i>Using the Sample Photos</i>	60
<i>Preparing Your Data Source</i>	60
<i>Improving Performance</i>	61
<i>Supporting Landscape and Portrait Photos</i>	61
<i>Determining the Orientation</i>	62
<i>Adding a Selected State to the Cell</i>	63
<i>Handling Taps on Your Collection View Cell</i>	65
<i>Adding a “Header” and a “Footer”</i>	65
Advanced Customization with Collection View Custom Layout	66
Masonry Layout	67
Cover Flow Layout	73
Summary	74
Further Reading	74
Apple Documentation	74
WWDC Sessions	75
Chapter 6 Stay in Bounds with Auto Layout	77
Auto Layout in Xcode 4	77
Getting Started with Auto Layout	78
What’s New in Auto Layout in Xcode 5	78
Auto Layout in Xcode 5	80
Intrinsic Sizes	81
Intrinsic Size and Localization	82
Design Time and Runtime Layout	82
Updating Frames Automatically	84
Top and Bottom Layout Guides	84
Layout Previews in Assistant Editor	85
Debugging Auto Layout at Design Time	85
Working with Scroll Views in Auto Layout	86
Working with Frames and Auto Layout	86
Visual Format Language	86
Disadvantages of Visual Format Language	88
Debugging Layout Errors	88
Summary	89
Further Reading	90
Apple Documentation	90
WWDC Sessions	90

Chapter 7 Better Custom Drawing	91
iOS's Many Drawing Systems	91
UIKit and the View Drawing Cycle	92
View Drawing Versus View Layout	94
Custom View Drawing	94
Drawing with UIKit	95
Paths	95
Understanding Coordinates	97
Resizing and contentMode	100
Transforms	101
Drawing with Core Graphics	104
Mixing UIKit and Core Graphics	106
Managing Graphics Contexts	107
Optimizing UIView Drawing	109
Avoid Drawing	109
Caching and Background Drawing	109
Custom Drawing Versus Prerendering	110
Pixel Alignment and Blurry Text	110
Alpha, Opaque, Hidden	111
Summary	112
Further Reading	112
Apple Documentation	112
Other Resources	113
Chapter 8 Layers Like an Onion: Core Animation	115
View Animations	115
Managing User Interaction	117
Drawing with Layers	118
Setting Contents Directly	120
Implementing Display	120
Custom Drawing	121
Drawing in Your Own Context	122
Moving Things Around	122
Implicit Animations	124
Explicit Animations	124
Model and Presentation	125
A Few Words on Timings	127
Into the Third Dimension	129
Decorating Your Layers	132
Auto-Animating with Actions	133

Animating Custom Properties	134
Core Animation and Threads	135
Summary	136
Further Reading.....	136
Apple Documentation.....	136
Other Resources.....	136
Chapter 9 Two Things at Once: Multitasking	137
Introduction to Multitasking and Run Loops	137
Developing Operation-Centric Multitasking	138
Setting Maximum Concurrent Operations	142
Multitasking with Grand Central Dispatch.....	143
Introduction to Dispatch Queues	143
<i>Queue Targets and Priority</i>	143
Creating Synchronization Points with Dispatch Barriers	144
Dispatch Groups	145
Summary	146
Further Reading.....	146
Apple Documentation.....	146
WWDC Sessions	147
Other Resources.....	147
Part III The Right Tool for the Job	149
Chapter 10 Building a (Core) Foundation.....	151
Core Foundation Types.....	151
Naming and Memory Management.....	152
Allocators.....	153
Introspection	154
Strings and Data	155
Constant Strings.....	155
Creating Strings	155
Converting to C Strings.....	156
Other String Operations	158
Backing Storage for Strings	159
CFData	160
Collections.....	160
CFArray.....	160
CFDictionary	161
CFSet, CFBag	161

Other Collections	161
Callbacks	162
Toll-Free Bridging	163
Summary	167
Further Reading	167
Apple Documentation	167
Other Resources	168
Chapter 11 Behind the Scenes: Background Processing	169
Best Practices for Backgrounding: With Great Power Comes	
Great Responsibility	169
Important Backgrounding Changes in iOS 7	171
Network Access with NSURLSession	171
Session Configuration	173
Tasks	173
Background Transfers	174
Periodic Fetching and Adaptive Multitasking	176
Waking Up in the Background	177
When We Left Our Heroes: State Restoration	177
Testing State Restoration	178
Opting In	178
Startup Changes	179
Restoration Identifiers	179
State Encoders and Decoders	180
Table Views and Collection Views	183
<i>Advanced Restoration</i>	<i>184</i>
Summary	185
Further Reading	185
Apple Documentation	185
WWDC Sessions	185
Chapter 12 REST for the Weary	187
The REST Philosophy	188
Choosing Your Data Exchange Format	188
Parsing XML on iOS	188
Parsing JSON on iOS	190
XML Versus JSON	190
<i>Designing the Data Exchange Format</i>	<i>190</i>
Model Versioning	191
A Hypothetical Web Service	191
Important Reminders	192

RESTfulEngine Architecture (iHotelApp Sample Code)	192
NSURLConnection Versus Third-Party Frameworks	193
Creating the RESTfulEngine	193
<i>Adding Authentication to the RESTfulEngine</i>	194
<i>Adding Blocks to the RESTfulEngine</i>	195
Authenticating Your API Calls with Access Tokens	196
Overriding Methods to Add Custom Authentication	
Headers in RESTfulEngine.m	197
Canceling Requests	197
Request Responses	198
Key Value Coding JSONs	198
List Versus Detail JSON Objects	201
Nested JSON Objects	202
Less Is More	203
Error Handling	203
Localization	205
Handling Additional Formats Using Categories	206
Tips to Improve Performance on iOS	207
Caching	207
Reasons for Going Offline	207
Strategies for Caching	208
Storing the Cache	209
<i>Implementing Data Model Caching</i>	209
<i>Core Data</i>	210
<i>Using Core Data for On-Demand Caching</i>	210
<i>Raw SQLite</i>	211
<i>Which Caching Technique Should You Use?</i>	211
<i>Data Model Cache Versus URL Cache</i>	211
Cache Versioning and Invalidation	211
Creating a Data Model Cache	212
<i>Refactoring</i>	214
Cache Versioning	215
Invalidating the Cache	216
Creating an In-Memory Cache	217
Designing the In-Memory Cache for AppCache	218
Handling Memory Warnings	219
Handling Termination and Entering Background Notifications	220
Creating a URL Cache	221
Expiration Model	221
Validation Model	221

Example	221
Caching Images with a URL Cache	222
Summary	222
Further Reading	223
Apple Documentation	223
Books	223
Other Resources	223
Chapter 13 Getting More Out of Your Bluetooth Devices	225
History of Bluetooth	225
Why Bluetooth Low Energy?	226
Bluetooth SDK in iOS	226
Server	226
Client	227
Classes and Protocols	227
Working with a Bluetooth Device	227
Scanning for a Service	228
Connecting to a Device	229
Retrieving Peripherals Directly	230
Discovering Services	230
Discovering Characteristics	230
<i>Getting Update Values</i>	232
Creating Your Own Peripherals	232
Advertising Your Service	233
Common Scenarios	236
Running in the Background	236
Background Modes	236
Power Considerations	237
State Preservation and Restoration	237
Summary	238
Further Reading	238
Apple Documentation	238
WWDC Sessions	238
Other Resources	238
Chapter 14 Batten the Hatches with Security Services	239
Understanding the iOS Sandbox	239
Securing Network Communications	241
How Certificates Work	241
Checking Certificate Validity	243
Determining Certificate Trust	247

Employing File Protection	248
Using Keychains	250
Sharing Data with Access Groups	251
Using Encryption	253
Overview of AES	254
Converting Passwords to Keys with PBKDF2	254
AES Mode and Padding	256
The Initialization Vector (IV)	258
Authentication with HMAC	259
Bad Passwords	259
Combining Encryption and Compression	260
Summary	260
Further Reading	260
Apple Documentation	260
WWDC Sessions	260
Other Resources	261
Chapter 15 Running on Multiple iPlatforms, iDevices, and 64-bit Architectures	263
Developing for Multiple Platforms	264
Configurable Target Settings: Base SDK versus Deployment Target	264
<i>Configuring the Base SDK Setting</i>	264
<i>Configuring the Deployment Target Setting</i>	264
Considerations for Multiple SDK Support: Frameworks, Classes, and Methods	265
<i>Framework Availability</i>	265
<i>Class Availability</i>	265
<i>Method Availability</i>	266
Checking the Availability of Frameworks, Classes, and Methods	266
<i>Developer Documentation</i>	267
<i>Macros in iOS Header Files</i>	267
Detecting Device Capabilities	268
Detecting Devices and Assuming Capabilities	268
Detecting Hardware and Sensors	268
<i>Detecting Camera Types</i>	269
<i>Detecting Whether a Photo Library Is Empty</i>	270
<i>Detecting the Presence of a Camera Flash</i>	270
<i>Detecting a Gyroscope</i>	271
<i>Detecting a Compass or Magnetometer</i>	271
<i>Detecting a Retina Display</i>	271

<i>Detecting Alert Vibration Capability</i>	272
<i>Detecting Remote Control Capability</i>	272
<i>Detecting Phone Call Capability</i>	273
In App Email and SMS	273
Obtaining the <code>UIDevice+Additions</code> Category	273
Supporting the New Class of 4-inch Devices	
(iPhone 5 and 5s)	274
Cocoa Auto Layout	275
Hard-Coded Screen Size	275
iPhone 5s and the New 64-bit Instruction Set	275
<i>iPhones, Instruction Sets, and Fat Binaries</i>	276
Transitioning to iOS 7	276
Auto Layout	276
Supporting iOS 6	277
Application Icon	277
Borderless Buttons	277
Tint Color	277
Artwork Updates	278
Transitioning to 64-bit Architecture	278
Data Overflows	279
Serialized Data	279
Conditional Compilation for 64-bit Architectures	279
UIRequiredDeviceCapabilities	280
Summary	281
Further Reading	281
Apple Documentation	281
Other Resources	281
Chapter 16 Reach the World: Internationalization and Localization	283
What Is Localization?	283
Localizing Strings	284
Auditing for Nonlocalized Strings	285
Formatting Numbers and Dates	286
Nib Files and Base Internationalization	290
Localizing Complex Strings	290
Summary	292
Further Reading	292
Apple Documentation	292
WWDC Sessions	292

Chapter 17 Those Pesky Bugs: Debugging	293
LLDB	293
Debugging with LLDB	293
Debug Information File (dSYM file).....	294
Symbolication.....	295
<i>Xcode’s Symbolication</i>	295
<i>Inside the xcarchive</i>	295
<i>Committing Your dSYMs to Your Version Control</i>	296
Breakpoints	296
The Breakpoint Navigator.....	296
<i>Exception Breakpoints</i>	296
<i>Symbolic Breakpoints</i>	298
<i>Editing Breakpoints</i>	299
<i>Sharing Breakpoints</i>	300
Watchpoints	300
The LLDB Console	300
<i>Printing Scalar Variables</i>	301
<i>Printing Registers</i>	301
<i>Scripting Your Debugger</i>	304
NSZombieEnabled Flag	304
Different Types of Crashes	305
EXC_BAD_ACCESS.....	305
SIGSEGV.....	306
SIGBUS.....	306
SIGTRAP.....	306
EXC_ARITHMETIC.....	306
SIGILL.....	306
SIGABRT.....	306
Watchdog Timeout.....	307
Custom Error Handling for Signals.....	307
Assertions	308
Exceptions	310
Collecting Crash Reports	311
iTunes Connect.....	311
<i>Collecting Crash Reports</i>	311
<i>Symbolicating iTunes Connect Crash Reports Using Xcode</i>	311
<i>Symbolicating iTunes Connect Crash Reports Manually</i>	311
Third-Party Crash Reporting Services	312
Advantages of TestFlight or HockeyApp over iTunes Connect.....	313
Summary	313

Further Reading	313
Apple Documentation	313
WWDC Session	314
Other Resources	314
Chapter 18 Performance Tuning Until It Flies	315
The Performance Mindset	315
Rule 1: The App Exists to Delight the User	315
Rule 2: The Device Exists for the Benefit of the User	315
Rule 3: Go to Extremes	315
Rule 4: Perception Is Reality	316
Rule 5: Focus on the Big Wins	316
Welcome to Instruments	316
Finding Memory Problems	318
Finding CPU Problems	322
The Accelerate Framework	325
GLKit	327
Compiler Optimizations	327
Linker Optimizations	328
Drawing Performance	328
Optimizing Disk and Network Access	330
Summary	331
Further Reading	331
Apple Documentation	331
WWDC Sessions	331
Other Resources	332
Part IV Pushing the Limits	333
Chapter 19 Almost Physics: UIKit Dynamics	335
Animators, Behaviors, and Dynamic Items	335
UIKit “Physics”	336
Built-in Behaviors	337
Snap	337
Attachment	338
Push	339
Gravity	340
Collision	340
Dynamic Item	340
Behavior Hierarchies	341

Custom Actions	341
Putting It Together: A Tear-Off View	341
Dragging a View.....	342
Tearing Off the View.....	343
Adding Extra Effects	345
Multiple Dynamic Animators	347
Interacting with UICollectionView	348
Summary	351
Further Reading.....	351
Apple Documentation.....	351
WWDC Sessions	351
Chapter 20 Fantastic Custom Transitions.....	353
Custom Transitions in iOS 7.....	353
Transition Coordinators	354
Collection View and Layout Transitions	355
Custom View Controller Transitions Using Storyboards and Custom Segues.....	355
Custom View Controller Transition: The iOS 7 Way.....	356
Interactive Custom Transitions Using the iOS 7 SDK	357
Interaction Controller.....	358
Summary	360
Further Reading.....	361
Apple Documentation.....	361
WWDC Sessions	361
Chapter 21 Fancy Text Layout.....	363
Understanding Rich Text	363
Characters versus Glyphs	363
Understanding Fonts	365
Paragraph Styles.....	366
Attributed Strings	366
Selecting Fonts with Font Descriptors	367
Assigning Paragraph Styles	368
HTML.....	369
Easier Attributed Strings	370
Dynamic Type	371
Text Kit	373
Text Kit Components	373
Multicontainer Layout	374
Exclusion Paths.....	376

Subclassing the Text Container	377
Subclassing the Text Storage	378
Subclassing the Layout Manager	382
Per-Glyph Layout	385
Core Text	388
Simple Layout with CTFramesetter	388
Creating Frames for Noncontiguous Paths	390
Typesetters, Lines, Runs, and Glyphs	392
Summary	392
Further Reading	393
Apple Documentation	393
WWDC Sessions	393
Other Resources	393
Chapter 22 Cocoa’s Biggest Trick: Key-Value Coding and Observing	395
Key-Value Coding	395
Setting Values with KVC	398
Traversing Properties with Key Paths	398
KVC and Collections	398
KVC and Dictionaries	400
KVC and Nonobjects	400
Higher Order Messaging with KVC	401
Collection Operators	401
Key-Value Observing	402
KVO and Collections	404
How Is KVO Implemented?	404
KVO Tradeoffs	405
Summary	407
Further Reading	407
Apple Documentation	407
Other Resources	407
Chapter 23 Beyond Queues: Advanced GCD	409
Semaphores	409
Dispatch Sources	411
Dispatch Timer Sources	413
Dispatch Once	414
Queue-Specific Data	415
The Myth of the “Current Queue”	415
Dispatch Data and Dispatch I/O	417

Summary	423
Further Reading.....	423
Apple Documentation.....	423
Chapter 24 Deep Objective-C.....	425
Understanding Classes and Objects.....	425
Working with Methods and Properties.....	428
Method Signatures and Invocations	430
Using Trampolines	432
How Message Passing Really Works.....	435
Dynamic Implementations	436
Fast Forwarding	438
Normal Forwarding.....	441
Forwarding Failure	442
The Flavors of objc_msgSend.....	442
Method Swizzling.....	443
ISA Swizzling.....	445
Method Swizzling Versus ISA Swizzling	447
Summary	447
Further Reading.....	447
Apple Documentation.....	447
Other Resources.....	448
Index.....	449

Introduction

In some ways, iOS 7 is the most radical change to iOS since the software development kit (SDK) was released in iPhone OS 2. The press and blogosphere have discussed every aspect of the new “flat” user interface and what it means for app developers and users. Suffice to say, no iOS upgrade has required so many developers to redesign their UI so much.

But in other ways, iOS 7 is a nearly seamless transition from iOS 6. Compared to the multitasking changes in iOS 4, iOS 7 may require very little change to your apps, particularly if you have either a very standard UI or a completely custom UI. For those on the extremes, the UI changes are either nearly free or irrelevant.

For all developers, though, iOS brings changes. There are more ways to manage background operations, but the rules for running in the background are even stricter than before. UIKit Dynamics means even more dynamic animations, but they can be challenging to implement well. TextKit brings incredible features to text layout, coupled with maddening limitations and bugs. iOS 7 is a mixed bag, both wonderful and frustrating. But you need to learn it. Users are upgrading quickly.

If you’re ready to take on the newest Apple release and push your application to the limits, this book will get you there.

Who This Book Is For

This is not an introductory book. Many other books out there cover Objective-C and walk you step by step through Interface Builder. However, this book assumes that you have a little experience with iOS. Maybe you’re self-taught, or maybe you’ve taken a class. Perhaps you’ve written at least most of an application, even if you haven’t submitted it yet. If you’re ready to move beyond the basics, to learn the best practices and the secrets that the authors have gleaned from practical experience writing real applications, this is the book for you.

This book also is not just a list of recipes. It contains plenty of sample code, but the focus is on discovering how to design, code, and maintain great iOS apps. Much of this book is about *why* rather than just *how*. You find out as much about design patterns and writing reusable code as about syntax and new frameworks.

What This Book Covers

The iOS platforms always move forward, and so does this book. Most of the examples here require at least iOS 6, and many require iOS 7. All examples use Automatic Reference Counting (ARC), automatic property synthesis, and object literals. Except in a very few places, this book does not cover backward compatibility. If you’ve been shipping code long enough to need backward compatibility, you probably know how to deal with it. This book is about writing the best possible apps using the best features available.

This book focuses on iPhone 5, iPad 3, and newer models. Most topics here are applicable to other iOS devices. Chapter 15 is devoted to dealing with the differences between the platforms.

What Is New in This Edition

This edition covers most of the newest additions to iOS 7, including the new background operations (Chapter 11), Core Bluetooth (Chapter 13), UIKit Dynamics (Chapter 19), and TextKit (Chapter 21). We provide guidance on how to best deal with the new flat UI (Chapter 2) and have added a chapter of “tips and tricks” you may not be aware of (Chapter 3).

We wanted to keep this book focused on the most valuable information you need for iOS 7. Some chapters from earlier editions have been moved to our website (iospt1.com). There, you can find chapters on common Objective-C practices, Location Services, error handling, and more.

How This Book Is Structured

iOS has a very rich set of tools, from high-level frameworks such as UIKit to very low-level tools such as Core Text. Often, you can achieve a goal in several ways. How do you, as a developer, pick the right tool for the job?

This book separates the everyday from the special purpose, helping you pick the right solution to each problem. You discover why each framework exists, how the frameworks relate to each other, and when to choose one over another. Then you learn how to make the most of each framework for solving its type of problem.

There are four parts to this book, moving from the most common tools to the most powerful. Chapters that are new in this edition or have been extensively updated are indicated.

Part I: What’s New?

If you’re familiar with iOS 6, this part quickly introduces you to the new features of iOS 7.

- **(Updated) Chapter 1: “The Brand New Stuff”**—iOS 7 adds a lot of new features, and here you get a quick overview of what’s available.
- **(New) Chapter 2: “The World Is Flat: New UI Paradigms”**—iOS 7 dramatically changes what it means to look and act like an iOS app. In this chapter, you learn the new patterns and design language you need to make the transition.

Part II: Getting the Most Out of Everyday Tools

As an iOS developer, you have encountered a wide variety of common tools, from notifications to table views to animation layers. But are you using these tools to their full potential? In this part, you find the best practices from seasoned developers in Cocoa development.

- **(New) Chapter 3: “You May Not Know...”**—Even if you’re an experienced developer, you may not be familiar with many small parts of Cocoa. This chapter introduces you to best practices refined over years of experience, along with some lesser-known parts of Cocoa.
- **(Updated) Chapter 4: “Storyboards and Custom Transitions”**—Storyboards can still be confusing and a bit intimidating for developers familiar with nib files. In this chapter, you learn how to use storyboards to your advantage and how to push them beyond the basics.

- **(Updated) Chapter 5: “Get a Handle on Collection Views”**—Collection views are steadily replacing table views as the preferred layout controller. Even for table-like layouts, collection views offer incredible flexibility that you need to understand to make the most engaging apps. This chapter shows you how to master this important tool.
- **(New) Chapter 6: “Stay in Bounds with Auto Layout”**—If there was one consistent message from WWDC 2013, it was this: use Auto Layout. There was hardly a UIKit session during the conference that didn’t stress this point repeatedly. You may have shied away from Auto Layout due to its many Interface Builder problems in Xcode 4. Xcode 5 has dramatically better support for Auto Layout. Whether you’re a constraints convert or longing to return to springs and struts, you should check out what’s new in Auto Layout.
- **Chapter 7: “Better Custom Drawing”**—Custom drawing is intimidating to many new developers, but it’s a key part of building beautiful and fast user interfaces. Here, you discover the available drawing options from UIKit to Core Graphics and how to optimize them to look their best while keeping them fast.
- **Chapter 8: “Layers Like an Onion: Core Animation”**—iOS devices have incredible facilities for animation. With a powerful GPU and the highly optimized Core Animation, you can build engaging, exciting, and intuitive interfaces. In this chapter, you go beyond the basics and discover the secrets of animation.
- **(Updated) Chapter 9: “Two Things at Once: Multitasking”**—Multitasking is an important part of many applications, and you discover how to do multiple things at once with operations and Grand Central Dispatch.

Part III: The Right Tool for the Job

Some tools are part of nearly every application, and some tools you need only from time to time. In this part, you learn about the tools and techniques that are a little more specialized.

- **Chapter 10: “Building a (Core) Foundation”**—When you want the most powerful frameworks available on iOS, you’re going to want the Core frameworks such as Core Graphics, Core Animation, and Core Text. All of them rely on Core Foundation. In this chapter, you discover how to work Core Foundation data types so you can leverage everything iOS has to offer.
- **(Updated) Chapter 11: “Behind the Scenes: Background Processing”**—iOS 7 adds a lot more flexibility for background processing, but there are new rules you need to follow to get the most out of the changes. In this chapter, you discover the new `NSURLSession` and learn how to best implement state restoration.
- **Chapter 12: “REST for the Weary”**—REST-based services are a mainstay of modern applications, and you learn how to best implement them in iOS.
- **(New) Chapter 13: “Getting More Out of Your Bluetooth Devices”**—Apple keeps expanding iOS’s capability to form ad hoc networks with other devices. This makes it possible to develop entirely new kinds of applications, from better games to micro-location services to easier file sharing. Get up to speed on what’s new and jump into whole new markets.
- **Chapter 14: “Batten the Hatches with Security Services”**—User security and privacy are paramount today, and you find out how to protect your application and user data from attackers with the keychain, certificates, and encryption.
- **(Updated) Chapter 15: “Running on Multiple iPlatforms, iDevices, and 64-bit Architectures”**—The iOS landscape gets more complex every year with iPod touch, iPhone, iPad, Apple TV, and a steady stream of new editions. It’s not enough just to write once, run everywhere. You need your applications to be their *best* everywhere. You learn how to adapt your apps to the hardware and get the most out of every platform.

- **Chapter 16: “Reach the World: Internationalization and Localization”**—Although you may want to focus on a single market today, you can do small things to ease the transition to a global market tomorrow. Save money and headaches later, without interrupting today’s development.
- **Chapter 17: “Those Pesky Bugs: Debugging”**—If only every application were perfect the first time. Luckily, Xcode and LLDB provide many tools to help you track down even the trickiest of bugs. You go beyond the basics and find out how to deal with errors in development and in the field.
- **(Updated) Chapter 18: “Performance Tuning Until It Flies”**—Performance separates the “okay” app from the exceptional app. It’s critical to optimizing CPU and memory performance, but you also need to optimize battery and network usage. Apple provides an incredible tool for this task in Instruments. You discover how to use Instruments to find the bottlenecks and then how to improve performance after you find the problems.

Part IV: Pushing the Limits

This part is what this book is all about. You’ve learned the basics. You’ve learned the everyday. Now push the limits with the most advanced tools available. You discover the ins and outs of deep iOS.

- **(New) Chapter 19: “Almost Physics: UIKit Dynamics”**—Apple constantly tries to make it easier to build dynamic, animated interfaces. UIKit Dynamics is its latest offering, giving a “physics-like” engine to UIKit. This tool is powerful but also can be challenging to use well. Learn to make the most of it.
- **(New) Chapter 20: “Fantastic Custom Transitions”**—One of the favorite demos at WWDC 2013 was about custom transitions. Move beyond the “push” and learn how to make transitions dynamic and interactive.
- **(Updated) Chapter 21: “Fancy Text Layout”**—The text-centric UI in iOS 7 demands incredible attention to detail in your font handling and text layout. TextKit brings many new features, from dynamic type to exclusion paths. It also brings bugs and frustrating limitations. No matter how you handle text, you need to master attributed strings. In this chapter, you learn the ins-and-outs of these powerful data structures and discover how to best use them with TextKit.
- **Chapter 22: “Cocoa’s Biggest Trick: Key-Value Coding and Observing”**—Many of Apple’s most powerful frameworks rely on KVO for their performance and flexibility. You find out how to leverage the flexibility and speed of KVO, as well as the trick that makes it so transparent.
- **(New) Chapter 23: “Beyond Queues: Advanced GCD”**—Dispatch queues are incredibly powerful tools that have become a key part of many applications. But there is more to Grand Central Dispatch than just queues. Learn about the tools such as semaphores, dispatch groups, and the incredibly powerful dispatch data and dispatch I/O.
- **Chapter 24: “Deep Objective-C”**—When you’re ready to pull back the curtain on how Objective-C really works, this is the chapter for you. You find out how to use the Objective-C runtime directly to dynamically modify classes and methods. You also learn how Objective-C method calls are dispatched to C function calls and how you can take control of the system to extend your programs in incredible ways.

You can skip around in this book to focus on the topics you need most. Each chapter stands alone, except for those that require Core Foundation data objects (particularly Core Graphics and Core Animation). Those chapters direct you to Chapter 10, “Building a (Core) Foundation,” when you need that information.

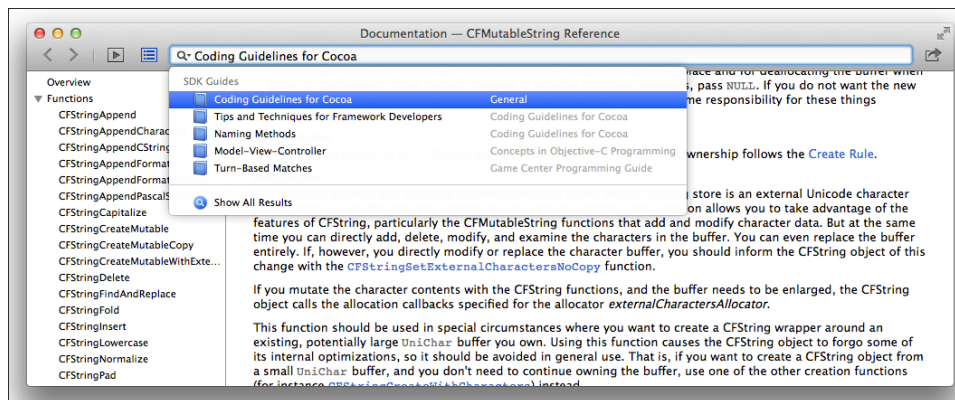
What You Need to Use This Book

All examples in this book were developed with Xcode 5 on Mac OS X 10.8 and iOS 7. You need an Apple developer account to access most of the tools and documentation, and you need a developer license to run applications on your iOS device. Visit developer.apple.com/programs/ios to sign up.

Most of the examples in this book will run in the iOS Simulator that comes with Xcode 5. You can use the iOS Simulator without an Apple developer license.

Finding Apple Documentation

Apple provides extensive documentation at its website and within Xcode. The URLs change frequently and are often very long. This book refers to Apple documents by title rather than by URL. To find documents in Xcode, press `Cmd-Option-?` or click `Help` → `Documentation and API Reference`. In the Documentation Organizer, click the Search icon, type the name of the document, and then select the document from the search results. The following Figure shows an example of how to search for the *Coding Guidelines for Cocoa*.



To find documents at the Apple developer site, visit developer.apple.com, click Member Center, and log in. Select the iOS Dev Center and enter the document title in the Search Developer search box.

The online documentation is generally identical to the Xcode documentation. You may receive results for both iOS and Mac. Be sure to choose the iOS version. Many iOS documents are copies of their Mac counterparts and occasionally include function calls or constants that are not available on iOS. This book tells you which features are available on iOS.

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All the source code used in this book is available for download at either <http://iosptl.com/code> or www.wiley.com/go/ptl.ios7programming.

For example, you can find the following sample code online in the Chapter 21 folder, in the SimpleLayout project, and the `CoreTextLabel.m` file:

CoreTextLabel.m (SimpleLayout)

```
- (id)initWithFrame:(CGRect)frame {
    if ((self = [super initWithFrame:frame])) {
        CGAffineTransform
        transform = CGAffineTransformMakeScale(1, -1);
        CGAffineTransformTranslate(transform,
                                   0, -self.bounds.size.height);
        self.transform = transform;
        self.backgroundColor = [UIColor whiteColor];
    }
    return self;
}
```

Some source code snippets shown in the book are not comprehensive and are meant to help you understand the chapter. For those instances, you can refer to the files available on the website for the complete source code.

Errata

We try to get everything right, but sometimes things change, sometimes we mistype, and sometimes we're just mistaken. See <http://iospt1.com/code> for the latest updates, as well as blog posts on topics that haven't made the book yet. Send any mistakes you find to robnapier@gmail.com and contact@mk.sg.