Contents

List of Illustrations

Pushing the Limits
iOS 7 Programming

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

Rob Napier

Mugunth Kumar

WILEY

A John Wiley and Sons, Ltd, Publication

This edition first published 2014

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO198SQ, 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.

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

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 Acknowledgments

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 Author

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.

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 (`iosptl.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