Design and develop your app from concept and vision to code

Beginning iOS Storyboarding

Using Xcode

Dr. Rory Lewis | Yulia McCarthy | Stephen M. Moraco

Apress®

Beginning iOS Storyboarding with Xcode

Easily Design and Develop Your App, from Concept and Vision to Code

Rory Lewis
Yulia McCarthy
Stephen M. Moraco

Apress®

Beginning iOS Storyboarding with Xcode

ISBN 978-1-4302-4272-7

ISBN 978-1-4302-4273-4 (eBook)

President and Publisher: Paul Manning
Lead Editor: Matthew Moodie
Technical Reviewer: Matthew Knott
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editor: Brigid Duffy
Copy Editor: Corbin Collins
Compositor: Bytheway Publishing Services
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`.

For information on translations, please e-mail `rights@apress.com`, or visit `www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `www.apress.com/info/bulksales`.

The source code for this book is available to readers at `www.apress.com`.

To my mother, Adeline. Thank you for those 13 hours! Love you.

—Rory

To my amazing mom—the most caring and supportive person I've ever known. Thank you for your endless love!

—Yulia

To Donna, my wife of 31 years, my best friend and travelling companion through this life and around this beautiful planet. Without your support and encouragement, many of my efforts throughout our time together would not have been possible, nor nearly as enjoyable. I look forward to our upcoming years together.

To my son Steve, for sharing in our many endeavors together, for your graphics contribution to our first joint iOS app, 9CardGolf in the App Store, but most importantly for being a shining example to me, and I hope to others, of constant self-motivation and constant learning, and for maintaining a youthful passion for learning about the universe in which we live. I look forward to seeing where you go with your photography passion and the life ahead of you.

—Stephen

Contents at a Glance

Contents

Foreword: About the Authors

The three authors have found a beautiful way to lead the beginner into Storyboarding and at the same time show the old school coders of Objective-C a new exquisite methodology for learning and debugging this incredible tool. Essentially, you have a guru of explaining complex Objective-C to beginners, a former Apple iOS intern, and a super-successful, old-school coder showing many people from many different walks of life the alpha and omega of Storyboard creation, debugging, and tweaking.

Dr. Rory Lewis

Stephen M. Moraco

Yulia McCarthy

Dr. Rory Lewis

Rory and I met in L.A. in 1983. He reminds me of one of my favorite film characters: Buckaroo Banzai—always going in six directions at once. If you stop him and ask what he's doing, he'll answer comprehensively and with amazing detail. Disciplined, colorful, and friendly, he has the uncanny ability to explain the highly abstract in simple, organic terms. He always accomplishes what he sets out to do, and he'll help you do the same.

Why You'll Relate to Dr. Lewis

While attending Syracuse University as a computer-engineering student, Rory scrambled to pass his classes and make enough money to support his wife and two young daughters. In 1990, he landed a choice, on-campus job as a proctor in the computer labs in the L.C. Smith College of Engineering. Even though he was struggling with subjects in the Electrical Engineering program, he was always there at the Help Desk. It was a daunting experience for Rory because his job was only to help his fellow students with computer lab *equipment* questions, yet he invariably found his classmates asking deeper and harder questions: "Dude, did you understand the calculus assignment? Can you help me?!"

These students assumed that, because Rory was the proctor, he knew the answers. Afraid and full of self-doubt, he sought a way to help them without revealing his inadequacies. Rory learned to start with: "Let's go back to the basics. Remember that last week when the professor presented us with an equation…?" By going back to the fundamentals, restating and rebranding them, Rory began to develop a technique that would, more often than not, lead to working solutions. By the time his senior year rolled around, there was often a line of students waiting at the Help Desk on the nights Rory worked.

Fast-Forward 17 Years

Picture a long-haired, wacky professor walking through the campus of the University of Colorado at Colorado Springs, dressed in a stunning contrast of old-school and dropout. As he walks into the Engineering Building, he's greeted by students and faculty who smile and say hearty hellos, all the while probably shaking their heads at his tweed jacket, Grateful Dead t-shirt, khaki pants, and flip-flops. As he walks down the hall of the Computer Science Department, there's a line of students standing outside his office. Reminiscent of the line of students that waited for him at the Help Desk in those early years as a proctor in the computer lab, they turn and greet him, "Good morning, Dr. Lewis!" Many of these UCCS students aren't even in his class, but they know Dr. Lewis will see them and help them anyway.

Past—Present—Future

Dr. Lewis holds three academic degrees. He earned a Bachelor of Science in Computer Engineering from Syracuse University. Syracuse's L.C. Smith College of Engineering is one of the country's top schools. It's there that Intel, AMD, and Microsoft send their top employees to study for their PhDs.

Upon completing his BS (with emphasis on the mathematics of electronic circuitry in microprocessors), he went across the quad to the Syracuse University School of Law. During his first summer at law school, Fulbright & Jaworski, the nation's most prolific law firm, recruited Rory to work in its Austin office, where some of the attorneys specialize in high-tech intellectual-property patent litigation. As part of his clerking experience, Lewis worked on the infamous *AMD v. Intel* case; he helped assess the algorithms of the mathematics of microprocessor electrical circuitry for the senior partners.

During his second summer in law school, Skjerven, Morrill, MacPherson, Franklin, & Friel—the other firm sharing the work on the *AMD v. Intel* case—recruited Rory to work with them at their Silicon Valley branches (San Jose and San Francisco). After immersing himself in law for several years and receiving his JD at Syracuse, Lewis realized his passion was for the *mathematics* of computers, not the legal ramifications of hardware and software. He preferred a nurturing and creative environment rather than the fighting and arguing intrinsic in law.

After three years away from academia, Rory Lewis moved south to pursue his PhD in Computer Science at the University of North Carolina at Charlotte. There, he studied under Dr. Zbigniew W. Ras, known worldwide for his innovations in data mining algorithms and methods, distributed data mining, ontologies, and multimedia databases. While studying for his PhD, Lewis taught computer science courses to computer engineering undergraduates, as well as e-commerce and programming courses to MBA students.

Upon receiving his PhD in Computer Science, Rory accepted a tenure-track position in Computer Science at the University of Colorado at Colorado Springs, where his research is in the computational mathematics of neurosciences. Most recently, he co-wrote a grant proposal on the mathematical analysis of the genesis of epilepsy with respect to the hypothalamus. However, with the advent of Apple's revolutionary iPhone and its uniquely flexible platform—*and market*—for mini-applications, games, and personal computing tools, he grew excited and began experimenting and programming for his own pleasure. Once his own fluency was established, Lewis figured he could teach a class on iPhone apps that would include *non*-engineers. With his insider knowledge as an iPhone beta tester, he began to integrate the parameters of the proposed iPad platform into his lesson plans—even before the official release in April 2010.

The class was a resounding success, and the feedback was overwhelmingly positive, from students and colleagues alike. When approached about the prospect of converting his course into a book to be published by Apress, Dr. Lewis jumped at the opportunity. He happily accepted an offer to convert his course outlines, class notes, and videos into the book you are now holding in your hands.

Why Write This Book?

The reasons Dr. Lewis wrote this book are the same reasons he originally decided to create a class for both engineering and non-engineering majors: the challenge and the fun! According to Lewis, the iPhone and iPad are "… some of the coolest, most powerful, and most technologically advanced tools ever made—period!"

He is fascinated by the fact that, just underneath the appealing touchscreen of high-resolution images and fun little icons, the iPhone and iPad are programmed in Objective-C, an incredibly difficult and advanced language. More and more, Lewis was approached by students and colleagues who wanted to program apps for the iPhone and would ask his opinion on their ideas. It seemed that with every new update of the iPhone, not to mention the advent of the expanded interface of the iPad, the floodgates of interest in programming apps were thrown open wider and wider. Wonderful and innovative ideas just needed the proper channel to flow into the appropriate format and then out to the world.

Generally speaking, however, the people who write books about Objective-C write for people who know Java, C#, or C++ at an advanced level. So, because there seemed to be no help for the average person who has no such knowledge but who has a great idea for an iPhone/iPad app, Dr. Lewis decided to launch such a class. He realized it would be wise to use his own notes for the first half of the course and then explore the best existing resources he could find.

As he forged ahead with this plan, Lewis was most impressed with *Beginning iPhone 3 Development: Exploring the iPhone SDK*. This best-selling instructional book from Apress was written by Dave Mark and Jeff Lamarche. Lewis concluded that their book would provide an excellent, high-level target for his lessons, a "stepping-stones" approach to comprehensive and fluent programming for all of Apple's multitouch devices.

After Dr. Lewis's course had been successfully presented, and during a subsequent conversation with a representative from Apress, Lewis happened to mention that he'd only started using that book about halfway through the semester, as he had to bring his non-engineering students up to speed first. The editor suggested converting his notes and outlines into a primer—an introductory book tuned to the less-technical programming crowd. At that point, it was only a matter of time and details—like organizing and revising Dr. Lewis's popular instructional videos to make them available to other non-engineers excited to program their own iPhone and/or iPad apps.

So, that's the story of how a wacky professor came to write this book. We hope you're inspired to take this home and begin. Arm yourself with this knowledge and begin now to change your life!

Ben Easton
Author, Teacher, Editor

Stephen M. Moraco

Stephen has more than 30 years of experience in software engineering. He's developed projects writing in high-level languages such as PL/I, RPG, ANSI C, C++, C#, Objective-C, and assembly languages for more microprocessors than he can count on two hands. Prior to joining Hewlett-Packard/Agilent Technologies 1989, he was an embedded-systems designer/developer. Stephen is a past member of the Large-scale Logic Analyzer Team, building system recovery media and writing triggering/capture drivers for multichannel custom data capture ASICs. As a software process engineer, he worked with medium-sized R&D teams developing techniques to improve the rate of release and initial release quality of software products. Stephen also designed and wrote an operating system for optical drives produced by Hewlett-Packard.

Stephen's profession is also his hobby. He is a strong believer in constant learning and of constantly practicing what he's learning. All during his career Stephen developed on non-work-related projects as a form of self-training. He enjoys designing and building his own hardware/software systems for home control and general experimentation. Stephen also developed firmware for key integration systems that fly aboard amateur radio satellites and developed hardware and software for testing these systems.

Stephen and his son Steve both enjoy building large LEGO models and working with LEGO Mindstorms robotics. Son Steve is studying photography, and together they've volunteered with Colorado First LEGO League for the past five years, with Dad refereeing the Mindstorms Robotics tournaments for 9–14 year-olds throughout Colorado while son Steve documents the excitement of the events though his photography.

In the fall of 2009, father Stephen and son Steve took an Objective-C, iOS programming class together at the University of Colorado at Colorado Springs. Shortly thereafter Stephen started his company Iron Sheep Productions LLC, the name under which he sells the hardware and software he's developed. After a successful 22-year career with Hewlett-Packard/Agilent Technologies, Stephen is now a retired professional software engineer and … a successful iPhone and iPad app programmer who sells his apps on the iTunes store.

Yulia McCarthy

Yulia is a Senior iOS Developer at InspireSmart Solutions, Inc., a local Denver firm specializing in innovative mobile business solutions. After graduating from one of the best classic universities in Russia with a BS in Mathematics, she went on to conquer the snowy peaks of Colorado, pursuing her dream of snowboarding and adventure. Soon she decided to pursue a graduate career in Computer Science at University of Colorado at Denver where, after taking an iPhone development class with Dr. Lewis, she quickly converted into a Mac user and transferred all her passion and incredible ability to program and solve complex problems into developing iPhone and iPad apps, which has been her new passion ever since. Her amazing talent soon attracted iOS recruiters at Apple, and now Yulia is even more inspired and devoted to Cocoa Touch programming after her invaluable experience as an iOS Apps and Frameworks intern at Apple's headquarters in Cupertino, California during the summer of 2011. She believes that life is all about constantly reaching for new horizons and challenging yourself. As a programmer, this concept is very close to Yulia's heart.

From Russia to UC Denver to Apple's iOS Division at Cupertino, Yulia believes that everything is possible if we follow our dreams.

About the Contributing Author

Ben Easton is a graduate of Washington & Lee University and has a BA in Philosophy. His eclectic background includes music, banking, sailing, hang gliding, and retail. Most of his work has involved education in one form or another. Ben taught school for 17 years, mostly middle-school mathematics. More recently, his experience as a software trainer and implementer reawakened his long-time affinity for technical subjects. As a freelance writer, he has written several science fiction stories and screenplays, as well as feature articles for magazines and newsletters. Ben resides in Austin, Texas, and is currently working on his first novel.

About the Technical Reviewer

Matthew Knott is a Learning Platform developer and SharePoint expert. He has been programming since a young age and hasn't stopped learning since. An experienced C and C# developer, Matthew has recently started developing iOS apps to mobilize the Learning Platform. He lives in Wales, United Kingdom, with his wife and two children and likes to write on his blog (`mattknott.com`) from time to time.

Introduction

In editions of Rory's previous book *iPhone and iPad Apps for Absolute Beginners* (Apress), there were only two ways to teach the reader how to make an iOS app user interface. The first was to write everything in code, and the other was to use Interface Builder to compose a Windows-based app. But things have changed with Storyboarding … boy, have they!

Storyboarding first appeared with Xcode version 4.2. When we first saw the scenes that made up an app, we thought Storyboard was fantastic. It was wonderful how Storyboard allowed us to navigate a path through out app in a visual way. Almost immediately Rory found freshmen students coming into his office, knee-deep in trouble using Storyboards. Meanwhile Xcode experts were pooh-poohing Storyboards. This book helps the novice understand the power of Storyboards and can help even experts in Xcode to unleash it.

In this book you'll discover how Xcode's Interface Builder's support for Storyboarding in iOS 5 makes designing your iOS apps so much easier. Storyboarding lets you graphically arrange all your views within a single design canvas, where you can then define the app's logical flow and even assign transition animations. You'll be able to learn how to use Storyboards to quickly go from concept to a fully functional iOS application.

First, we go over the fundamental concepts of Storyboarding and the technology behind it. We then walk you through building seven complete projects that advance you through using various Storyboarding features, covering the most important aspects you need to know to successfully create your own apps from start to finish. By the end of this book, you'll eventually see how to use Storyboarding with almost every application template offered by Xcode and you'll learn which Storyboarding techniques are most suitable in certain scenarios.

Working with Storyboarding involves much more than simply dragging and dropping View Controllers onto a canvas. In this book we show how to start from scratch and build complete apps using Storyboarding. Along the way we demonstrate using common iOS technologies as Map Views, Page View Controllers, Split View Controllers, Core Data, Table Views, and more—and we tell you how they all fit together with the new Storyboarding feature.

What You'll Learn

In Chapter 1, we help you to get started in iOS development by walking you through Apple's iOS Developer Program registration process and installing Xcode and other tools you'll be using throughout this book.

Chapter 2 talks about the basics of Storyboard structure and introduces the main Storyboarding concepts, including standard view transitions, passing information around, and creating custom transitions between the views.

Chapter 3 explains how to create a map-driven app using Storyboarding and how to transition to other scenes from a Map View. It also demonstrates several important Storyboarding concepts, such as triggering manual segues and instantiating View Controllers designed in the Storyboard

from within the code. Additionally, you'll learn how to easily parse JSON data from a remote server (such as Flickr) using nothing but the new iOS 5 API.

In Chapter 4, you'll find out how to develop a fun utility app using Storyboarding targeted for the iPad. You'll learn the foundation of many apps, which is how to place controls on the settings screen of a utility application and to return those settings to the Main View of the app via the Settings View delegate protocol. You'll get a good grasp of the main Storyboarding specifics of the iPad environment, including Split View Controllers, Popover View, and iPad-specific segues. As a part of building this chapter's project, you'll also demonstrate how you can use a build-it media framework to enable your app to play audio files.

In Chapter 5, we explore a very special Xcode template: the Page-Based Application template. Unlike other templates, it doesn't let you opt out of using Storyboarding. In this chapter, you'll learn the powerful tools that let you create Page View Controller transitions. We dig deep into the ins and outs of how to use the `UIPageViewController` to build an iPad brochure with beautiful, built-in page-curl animations and custom layout.

Chapters 6–8 bring to you a whole new world of Storyboarding features that dramatically change the way to program Table Views. We walk you through a more advanced Table Views-based project that utilizes Core Data in the back end. You'll learn critical Storyboarding techniques such as Dynamic Cell Prototyping and designing Static Table Views. Most importantly, in this chapter we show you how to design your entire app workflow entirely in the Storyboard before doing any coding at all.

In Chapters 9–11 you'll learn to develop a cool game app that stretches your knowledge of how segues can be used to provide much more complex navigation paths between screens.

Who Should Read This Book?

This book is for readers of Rory Lewis's last book, *iPhone and iPad Apps for Absolute Beginners*, but it's also for the beginner who's never programmed but who can use the Storyboarding tool in Xcode to get up and running fast. This book is also for experienced iOS developers who want to learn Storyboarding to quickly cut down on app development and debugging time.

For the beginner who has never programmed, *Beginning iOS Storyboarding with Xcode* shows how to extract those cool and innovative app ideas you have in your head into a working app ready for sale on the App Store. Even if you're an intermediate or pro-level Objective-C developer, you can still learn the ins and outs of Xcode's new Storyboarding feature—and find new ways of building and debugging your new Storyboarding app. Yup: This book is for you, too.

Regardless of your skill level, we're extremely happy to have you on board and hope you enjoy the ride. Let's get to Storyboarding!

Chapter 1

Preliminaries

This introductory chapter will make sure that you have all the required tools and accessories to proceed fully and confidently. Three types of readers are likely reading this book. One group can skip to Chapter 2 immediately without reading Chapter 1. Another group may only need to read one small section in Chapter 1 and then move on to Chapter. The third group should read Chapter 1 very carefully before moving on.

- *Group 1*: You own a Mac. You have experience coding with Xcode on your Mac. You have an up-to-date iOS SDK and an up-to-date version of Xcode. You also have experience with DemoMonkey, and it's installed on your machine. If all this is true, meet me in Chapter 2.
- *Group 2*: You own a Mac. You have experience coding with Xcode on your Mac. You have an up-to-date iOS SDK and an up-to-date version of Xcode. However, you don't have experience with DemoMonkey or it's not installed on your machine. Please check out the section "Installing DemoMonkey" in this chapter and then meet me in Chapter 2.
- *Group 3*: You are a seeker of knowledge and have begun travelling down a wonderful road. We need to check your backpack and make sure you have all the tools you'll need for your journey. So let's start right here.

Necessities and Accessories

In order to program for the iPhone and/or iPad, and to follow along with the exercises, tutorials, and examples presented in this book, you'll need to have 6

minimal requirements which you may not completely understand right now but that's OK just roll with me for a second, I'll explain everything as we go through these steps.

> **NOTE:** Whenever we say *iPhone* or *iPad*, we're referring to any iPhone or iPad OS device, including the iPod touch. Also when we say *Macintosh HD*, yours may be named something different.

Briefly, you will need six things:

- An Intel-based Macintosh
- The correct operating system for your Mac (OS X 10.7.4 Lion or later)
- Be a registered developer or be simulator-based (discussed in detail later in this chapter)
- To have the correct operating system for your iPhone (iOS 5 or above)
- To have the correct Software Development Kit (SDK) for your iPhone that runs a program called Xcode (version 4.3 and above)
- To install and run DemoMonkey

Let's go into each of these in a bit more detail.

Getting a Mac

If your Mac was manufactured after 2006, you're okay. One of the authors purposefully programs everything on a MacBook bought in 2008. All the videos on the net are screencast from Dr. Lewis's MacBook from 2006; or if he broadcasts from his 2010 iMac, he first runs it on his MacBook bought in 2006.

- You don't need the latest revved-up Mac. If you haven't bought one yet, we suggest you get a basic, no-frills MacBook Air.

- If you do own an older Mac, you may be able to add some RAM. Make a free appointment at the Genius Bar at an Apple Store and ask whether they can increase the RAM on your older model Mac, and if so, ask about the maximum the RAM can be increased. Then ask explicitly: "Can this old computer run Lion, *at least* 10.7.1, and Xcode 4.3 or later?" Note that some of the apps in this book will work using Xcode 4.3 on Snow Leopard. But if possible, try to get Lion (at least Mac OS X 10.7.4) and iOS SDK 4.3.
- If you don't have a Mac, you'll need to buy one if you want to follow along with this book and or program Objective-C to make iPhone apps. Keep in mind that, as mentioned, we have made a point to code and run every program in this book on Apple's smallest and cheapest model, the MacBook. Apple has discontinued the MacBook; it now sells the MacBook Air for $999, which is more advanced than the Author's MacBook. You can purchase a MacBook on eBay and other such sites. See Figure 1-1.

Figure 1-1. *The authors use the cheapest 2006 Mac on the market, the MacBook, to perform all the coding and compiling in this book. Many of the authors' students purchase the MacBook Air for $999 as illustrated here.*

Getting OS X

You will need the correct version of OS X. At the time of this writing, that version is OS X 10.7.4. We need to make sure that you have the latest greatest operating system inside your Mac. We see a lot of emails and forum questions revealing that many of you will think: "*Ah, my code probably did not compile correctly because Dr. Lewis has a different version of OS X or/and iOS on his machine…*"

> **NOTE:** Even if you think everything is up to date, we suggest you follow along with the procedure in this section and make sure your system has the latest OS X and the latest iOS inside it. We say this because as you follow along in this book and tackle all the programs, there will be times when your code doesn't work the first time you run it.

To make sure your system is recent enough to follow along with the book, please do the following:

1. Close every program running on your Mac so that only the Finder is running.
2. Click the little apple in the upper left-hand corner of your screen and select About This Mac. You'll see the window shown in Figure 1-2. Make sure it says OS X 10.7.4.

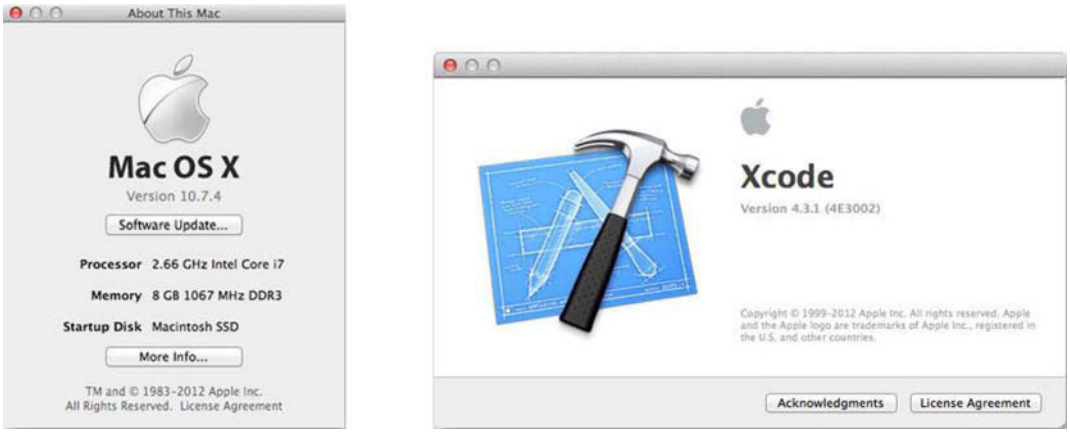

Figure 1-2. *Here you can see that Dr. Lewis's MacBook is using is OS X 10.7.4.*

Now to make sure you have the latest software on your Mac:

1. With all your programs closed except for the Finder, click the apple in the upper left-hand corner again and select Software Update… as illustrated in Figure 1-3.
2. If updates are available, click Continue and follow the instructions and four screen prompts, as shown in Figure 1-3.

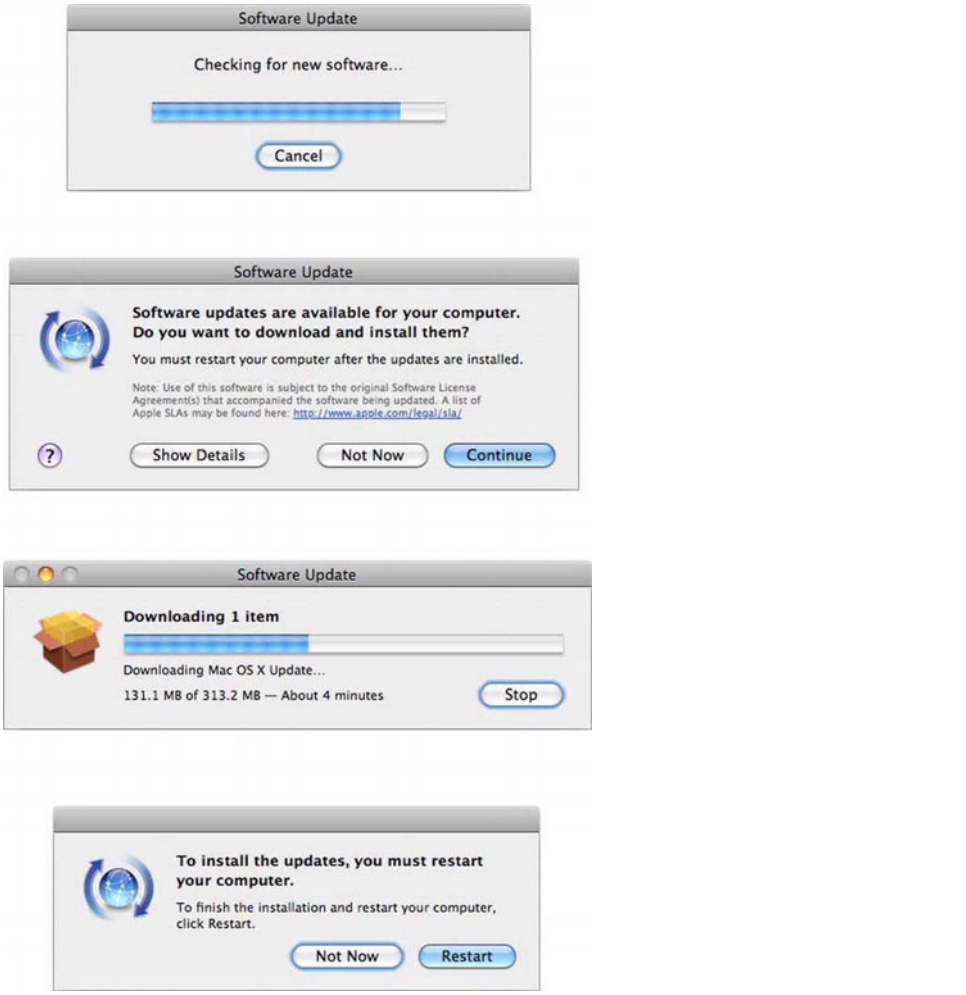

Figure 1-3. *Top: Checking for new software. Second from top: Download the new software. Second from bottom: Wait for software to download. Bottom: Click Restart to have your Mac properly install the new software.*

If by the time you are reading this book, you realize that your version of OS X or iOS makes my pictures seemed dated, don't freak out. We have an online forum where we and volunteers love to help others. We always update the forum with news about recent updates of OS X and iOS. You can visit the forum here:

```
www.rorylewis.com/ipad_forum/
http://bit.ly/oLVwpY
```

Become a Developer

You will need to become a registered developer via the iPhone/iPad Software Development Kit (SDK) for $99. Or you can pay $0 for an introductory set of bells and whistles.

Making Your Choice

If you are a student, it's likely that your professor has already taken care of this, and you may already be registered under your professor's name. If you are not a student, you need to decide which type of developer you would like to be. Here are your options:

- *$0 option*: You can go to the App Store and download Xcode for free. This is fine, but bear in mind that unless you become a developer ($99), you will only be able to see the apps you code and program in this book running on the iPhone or iPad *Simulator*. That means you can't run them on a real physical iPad or iPhone. You also won't be able to sell your apps on the iTunes store. Lastly, you won't be able to log in to the developer site to view code snippets and updates, beta-test new products, or be a part of the Apple online community. This may be a very good choice for the person who isn't sure whether they want to continue with Xcode and programming. If that's the case, then download the latest version of Xcode from `https://developer.apple.com/xcode/` and meet me at Figure 1-13.
- *$99 option*: If you do want to run your apps on a physical device such as a real iPad or iPhone, sell apps on the iTunes store, and be a part of the developer group at Apple—simply continue reading.

Installing Xcode

Let's get started installing Xcode.

1. Go to `http://developer.apple.com/programs/ios/` or `http://bit.ly/rrrdjc`. You'll see a page similar to the one shown in Figure 1-4. Click the Enroll Now button.

Figure 1-4. *Click the Enroll Now button.*

2. Click the Continue button, as illustrated in Figure 1-5.

Figure 1-5. *Click the Continue button.*

3. Most people reading this book will select the "I need to create a new account for…" option (arrow 1 in Figure 1-6). Next, click the Continue button (arrow 2). (If you already have an existing account, then you have been through this process before; go ahead with the process beginning with the "I currently have an Apple ID..." option, and I'll meet you at step 6, where you'll log in to the iPhone/iPad development page and download the SDK.)

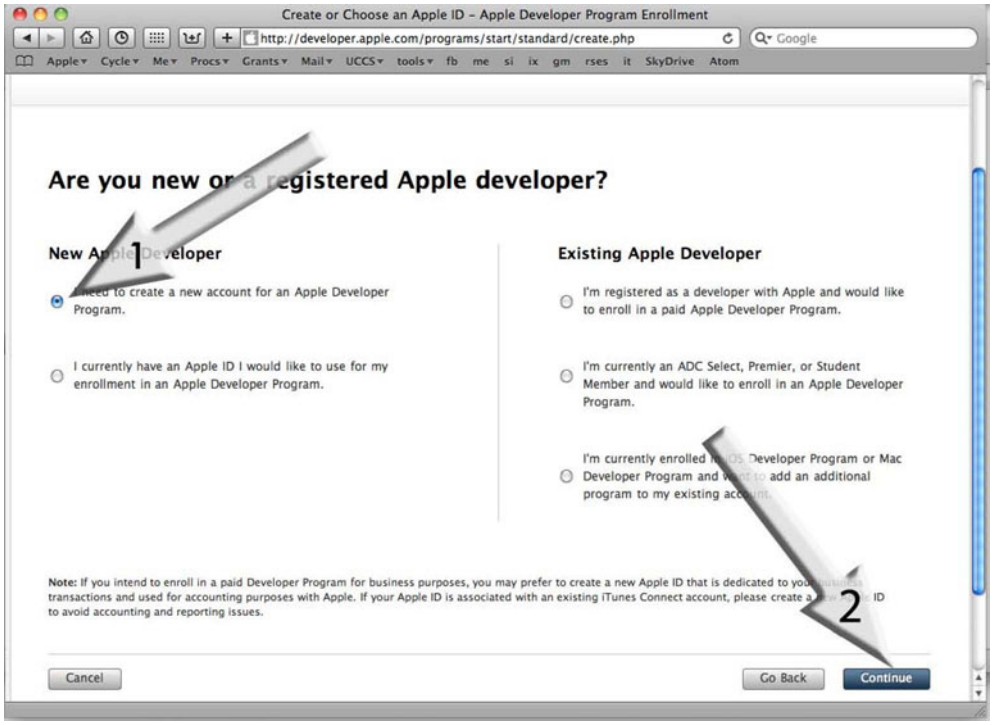

Figure 1-6. *Click the "I need to create a new account …"option to proceed.*

4. You are probably going to be enrolling as an individual, so click the Individual link shown in Figure 1-7. If you are enrolling as a company, click the Company option to the right and follow the appropriate steps; Skip to step 6.

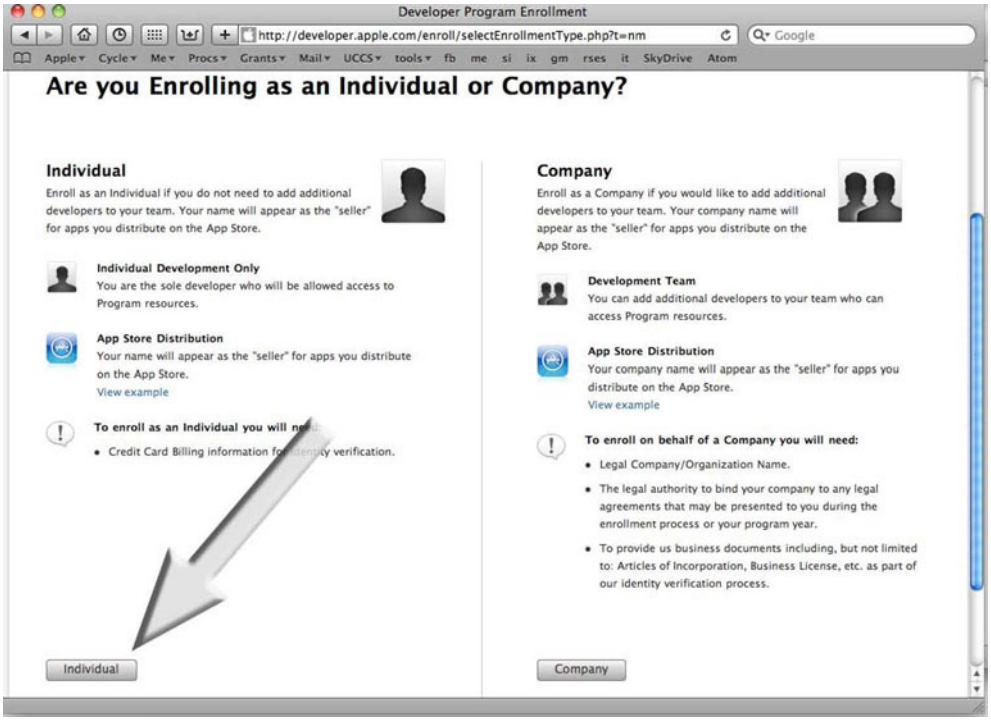

Figure 1-7. *Click the Individual option.*

5. Enter all your information as shown in Figure 1-8 and pay your fee of $99 for the Standard Program. This provides all the tools, resources, and technical support you will need. (If you're reading this book, you really don't want to buy the Enterprise program at $299—it's for commercial in-house applications.) After paying, save your Apple ID and username; then receive and interact with your confirmation email appropriately.

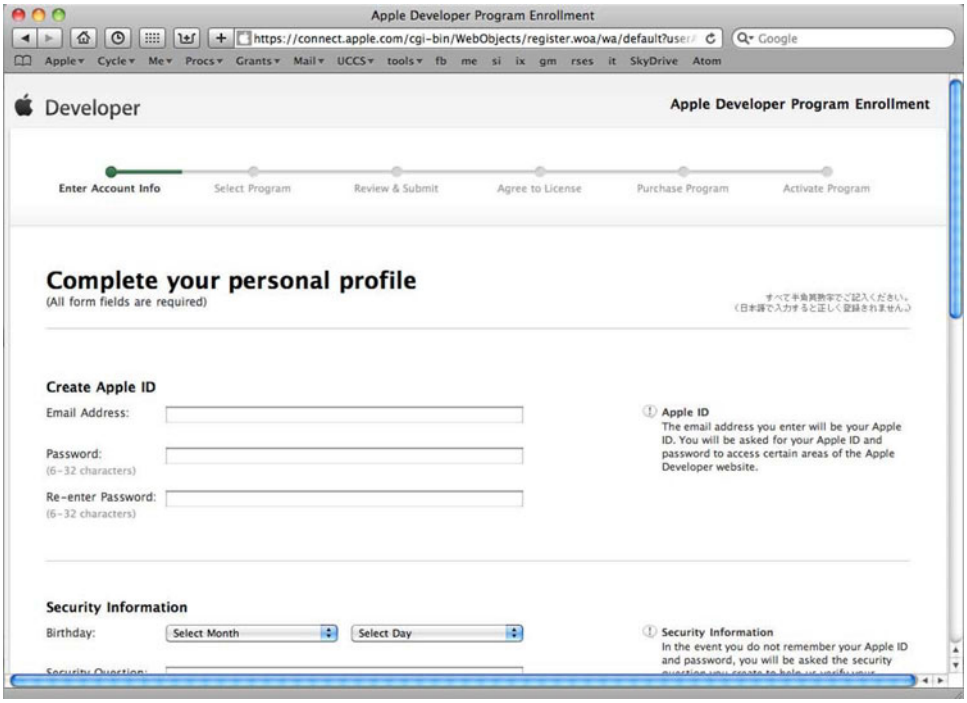

Figure 1-8. *Enter all your information accordingly.*

> **NOTE:** Before you move on to step 6, make sure you have received your confirmation email and chosen a password to complete the last step of setting you up as a bona fide registered Apple developer. Congratulations!

6. Use your Apple ID to log in to the main iOS development page at `http://developer.apple.com`. This page has three icons for the three types of Apple programmers. As shown on Figure 1-9, click the iOS Dev Center icon, which leads to the download page for iOS development software.

Figure 1-9. *For now click on the iOS Dev Center icon as indicated by the arrow. Later you may want to also program apps for the Mac Computer or the Safari Web Browser.*

7. After logging in with your username and password as described in step 6, you will see a screen similar to Figure 1-10. The iOS Dev Center contains all the tools necessary to build iOS apps. Later on you will spend time here, but for now just go to the Developer Page of the latest build of the iOS SDK. Click the icon indicated by the arrow.

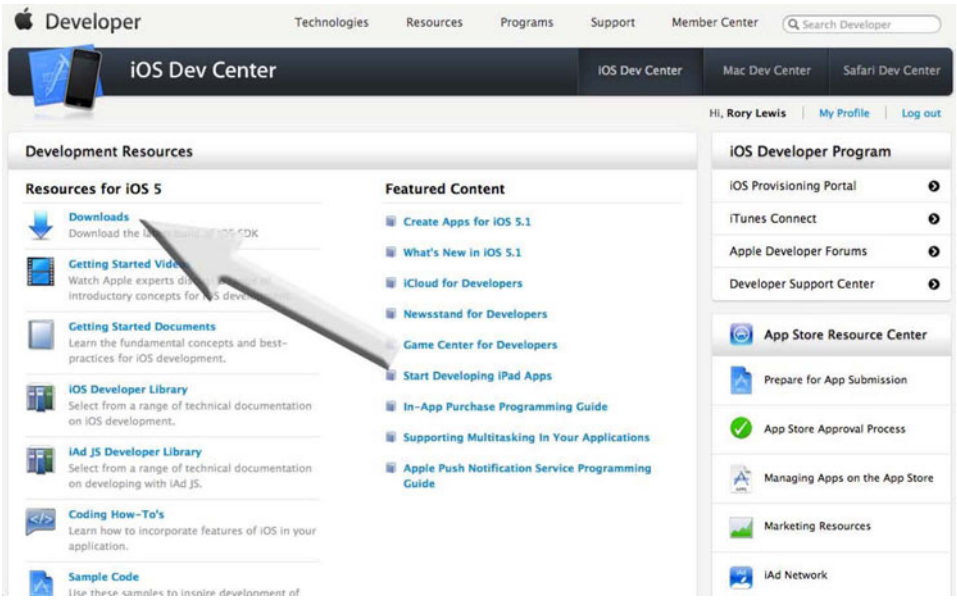

Figure 1-10. *The Downloads link takes you to the bottom of the page as shown in Figure 1-11.*

> **NOTE:** At the time of writing, Xcode 4.3 and iOS SDK 5 are the latest environments. There is a great chance that by the time you read this book these may have larger numbers. This is not a problem—just go on to step 8. If by chance there is something that has really thrown us a curve ball, it will be discussed and solved for you in our forum located at `www.rorylewis.com/ipad_forum/` or `http://bit.ly/oLVwpY`.

8. For now we want you to click on the latest version. The figures in this section show the latest version at the time of print. These *will* be different by the time you read this. Right now the latest version is Xcode 4.3 for Lion, so click the link indicated by the arrow in Figure 1-11.

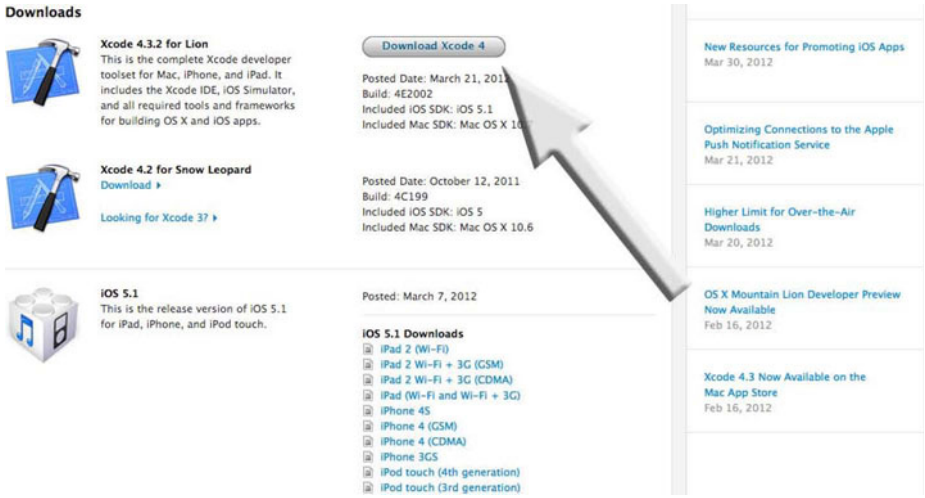

Figure 1-11. *Clicking the Download Xcode 4 button takes you to the Xcode 4 Developer page.*

9. Click the View in Mac App Store button. Remember that if it's a later version than shown in Figure 1- 12 things may look slightly different, but we have confidence in you.