Extracted from:

iPad Programming

A Quick-Start Guide for iPhone Developers

The Pragmatic Programmers

iPad Programming

A Quick-Start Guide for iPhone Developers

Daniel H Steinberg
Eric T Freeman

Edited by Colleen Toporek

Pragmatic Bookshelf

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at

http://www.pragprog.com

Printed in the United States of America.

ISBN-10: 1-934356-57-3

ISBN-13: 978-1-934356-57-9

Printed on acid-free paper.

B1.0 printing, April 3, 2010

Version: 2010-4-14

Chapter 1

From iPhone to iPad

Before you write a line of code for your iPad app, you need to hold the device in your hands. You need to feel the device melt away as you become immersed in an app. The app you are using gets out of your way as you get involved in an activity it enables.

That’s the kind of app we want you to write.

We don’t merely want you to write apps that run on an iPad, we want you to write apps that are perfect for the platform. People are going to tell their friends about your app because it delights them when they notice it, and it disappears as they use it.

How can that be? Isn’t the iPad just an oversized iPod touch that runs on the same iPhone OS we already know and love? Have we become caught up in the marketing hype generated by Steve Jobs’ famous reality distortion field?

No. There is something fundamentally different about this high resolution device with its large display that fits in your hands but not your pocket—a device that you navigate with multi-touch and gestures.

Throughout this book, we show you how to build experiences for this new device. We’ll explore the similarities and differences between developing for the iPad and what you’re used to in developing for the iPhone. In fact, we’ll start by creating an iPhone app in this chapter and transform it into an app that runs on your iPad.

But first let’s figure out how the iPad fits in your end users’ lives—somewhere between the laptop and the iPhone.

1.1 The iPad and the Laptop

Once you've checked email or surfed the web on an iPad, your laptop is going to seem bulkier and less portable than it was the minute before you picked up the iPad. Suddenly, you'll be aware of the amount of space required to house the keyboard, trackpad, hard drive, and battery.

On the other hand, your iPad is more constrained. It is not meant to live as a standalone device. It must be synced to a laptop or desktop machine once in a while. It doesn't have the processing power, memory, or storage of a laptop. It is designed to belong to a single user, with a single app occupying your attention at any given time. The iPad is not meant to be the digital hub that a desktop or laptop becomes. For now, it is a node on the hub, not the center of your digital universe.

Most importantly, there is something very different about interacting with the screen by touching it as opposed to having to use a mouse or a trackpad. On the iPad, you use your finger to land planes in Flight Control, paint beautiful images with Brushes, and mark email messages as deleted. If you want to move or guide an object, you put your finger on it and move your finger. The object comes along for the ride.

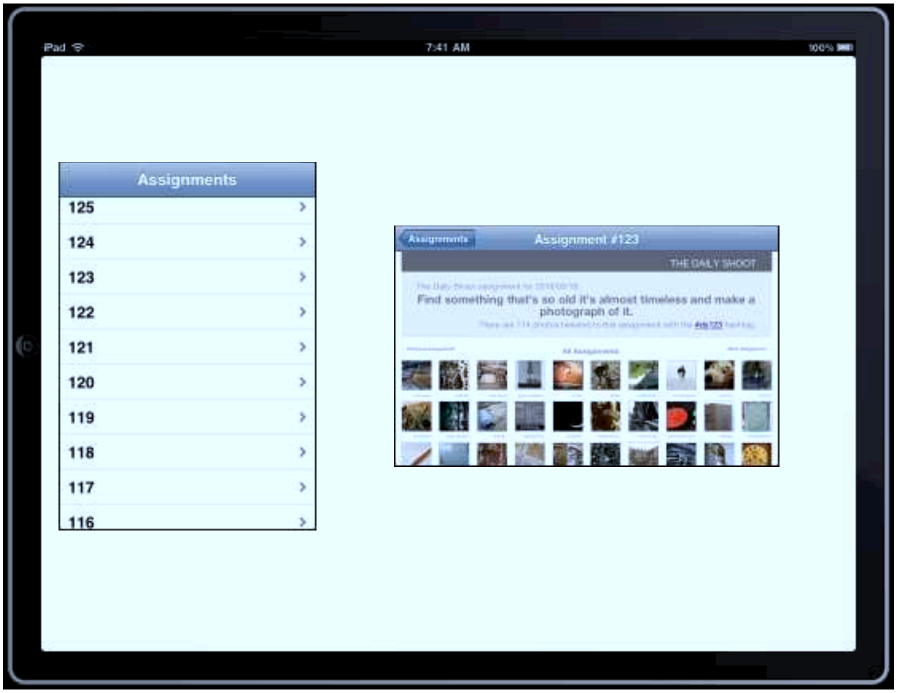
If you want to resize an object, you use two fingers to pinch or zoom. You use your finger to resize, reposition, and crop an image. You zoom in on a web page to better read it, and so that your finger can more easily select the right text box to type into. These gestures have become so natural that it feels strange to use a mouse to select an object and grab handles to resize and reposition. Reaching for the mouse or the trackpad takes us out of the moment.

The last important difference between an iPad and a laptop is that you can easily rotate the iPad ninety degrees and get a whole different view of your application. In fact, as an iPad developer, you are asked to consider how best to present your app in both landscape and portrait orientations.

1.2 The iPad and the iPod Touch

The iPad runs iPhone OS. You will use many of the same APIs to write your iPad apps that you use to write apps for the iPhone. So isn't the iPad *just* a big iPod touch?

It’s certainly true that an iPad is much bigger than an iPhone. An iPad is about the size of the printed version of this book—about half an inch taller and a little thinner, but roughly the same size. The iPhone is much smaller. Here’s a picture that shows the relative size of two iPhone screens to the iPad screen.



Find something that's so old it's almost timeless and make a photograph of it.

So the iPad *is* a big iPod touch—it’s just not *just* a big iPod touch. Being bigger is the key to what makes it special.

Much of the GUI that you know and love on Mac OS X and later on Windows originated at Xerox PARC with Alan Kay’s team. Kay has consistently complained that the way we interact with a computer has not changed enough from the work his team was doing decades ago. Alan Kay told Gigaom’s Janko Roettgers this story about Kay’s reaction to the unveiling of the iPhone in 2007.

“When the Mac first came out, Newsweek asked me what I [thought] of it. I said: Well, it’s the first personal computer worth criticizing. So at the end of the presentation, Steve came up to me and said: Is the iPhone worth criticizing? And I said: Make the screen five inches by eight inches, and you'll rule the world."[1]

The size is an important part of the story. On the iPad, you can browse web pages at full resolution, sort your pictures into piles with a flick of your finger, and fill out forms using a keyboard you can fit all of your fingers on.

You will be missing the point if you think of the iPad from the developer's point of view. The APIs are similar enough to those for the iPhone that apps for one device can easily be run on the other. When you start to create an app for the iPad, take the time to understand what is different and special about this device.

Your favorite iPhone apps are the ones that take advantage of all of the capabilities of the iPhone in natural ways.[2] They incorporate the accelerometer, the compass, the GPS, the camera, and the phone itself when it makes sense. In the remainder of the book we will show you *how* to program the iPad, but it's up to you to think about *what* to program.

1.3 Getting Started

In this book, we assume you know how to develop iPhone apps. Maybe you've written an iPhone app or two for fun or profit. You are certainly familiar with Cocoa, Objective-C, and the iPhone APIs. You are comfortable with the most recent versions of Xcode and Interface Builder. You know how to work with view controllers and delegates and can use reference counting to manage memory.

So that's where we'll start.

We'll create an iPhone app and quickly review some of the key concepts you'll need to know in this book. These techniques are common to iPhone and iPad development. You'll see this at the end of the chapter when we convert this from a project that only works on the iPhone to one that works on both the iPhone and the iPad.

By the end of this chapter, you'll know whether you have the right amount of background for this book. Everything should be familiar,

[1] http://gigaom.com/2010/01/26/alan-kay-with-the-tablet-apple-will-rule-the-world/

[2] It will get tedious to say "iPhone and iPod touch" every time. We'll often use one or the other to refer to the common platform.

What You Need for This Book

You will need to register as an iPhone developer for free at http://developer.apple.com/iphone. There is an additional yearly charge (currently $99) if you want to deploy your applications to your iPad or iPhone or sell your applications on the iTunes App Store.

Once you've registered, download the latest tools and SDKs. Currently, iPad development requires that you are running Mac OS X 10.6.2 on an Intel machine. The examples in this book were created using Xcode version 3.2.2 and iPhone OS 3.2.

Also, you can download all of the projects for this book from the book's homepage at http://pragprog.com/titles/sfipad.

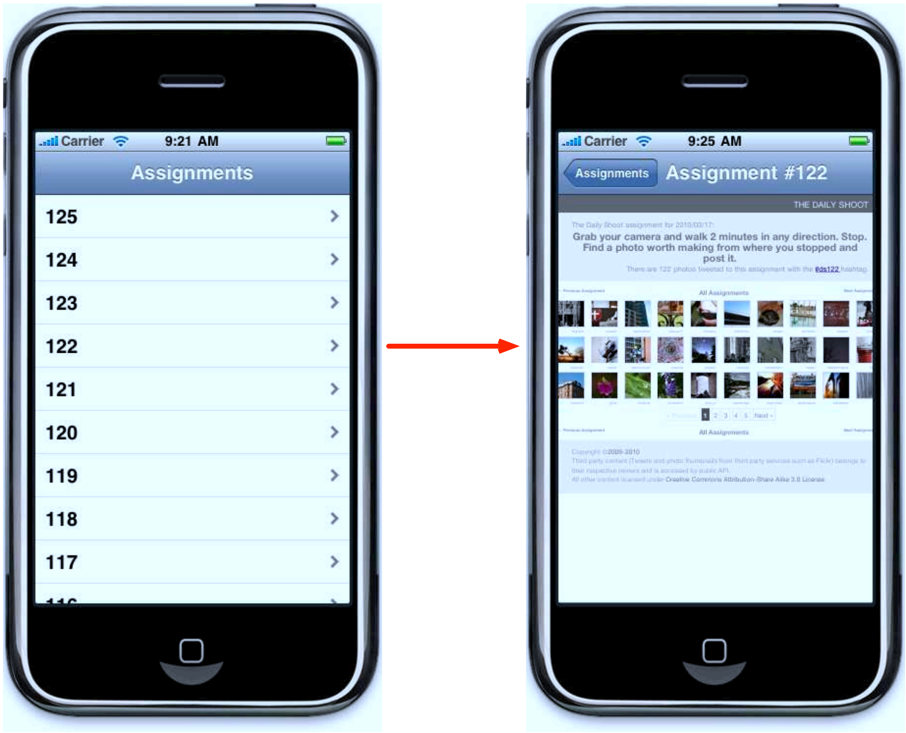
and the quick review should be enough to bring you up to speed for the remainder of the book. If you find yourself confused or think that we're moving too fast, then you may need to start with a book on Cocoa or iPhone development.[3]

If you think we're moving too slowly and you want to get right to the iPad material, start with the project in the FromTo/DailyShoot1 directory in the code download, and skip ahead to Section 1.8, *Creating an iPad Version*.

We'll build a simple navigation-based application around "The Daily Shoot" (http://dailyshoot.com) website created by Mike Clark and James Duncan Davidson. Their site has suggested a daily assignment for photographers since November 2009. In this type of app, users navigate through some number of levels of table views until they select an individual item and navigate to its detail view.

Our app will launch with a simple table that is filled with the numbers of the first 125 assignments. When a user selects a number, we will display the landing page for that assignment.

[3] You can start with *Cocoa Programming: A Quick-Start Guide for Developers* [Ste09] and/or *iPhone SDK Development* [DA09].



The Daily Shoot assignment for 2010/03/17:

Grab your camera and walk 2 minutes in any direction. Stop. Find a photo worth making from where you stopped and post it.

There are 122 photos tweeted to this assignment with the #ds122 hashtag.

Copyright ©2009-2010

Third party content (Tweets and photo thumbnails from third party services such as Flickr) belongs to their respective owners and is accessed by public API.

All other content licensed under Creative Commons Attribution-Share Alike 3.0 License

There's nothing fancy about this example, but it lets us set up all of the machinery we need for a nav-based application, and it sets the stage for the Split View-based application in the next chapter.

Start up Xcode and create a new Navigation-based Application that does *not* use Core Data for storage. Name the project "DailyShoot".

By the way, you may have noticed some important differences in the templates you had available. The Split View-based template is brand new, and can only be used for iPad-based projects. The Navigation-based Application and Utility Application templates can only be used for new iPhone projects. The Window-based template can create an iPhone project, an iPad project, or a Universal project that targets both platforms at once. The remaining templates can be used as the starting point for either iPhone or iPad projects, but you have to specify which using a drop-down.

We're not going to do anything with the app delegate or with the MainWindow nib. If this were a production app, we would take the time to change the name of the RootViewController class and the corresponding nib to something that has more meaning in the current setting—

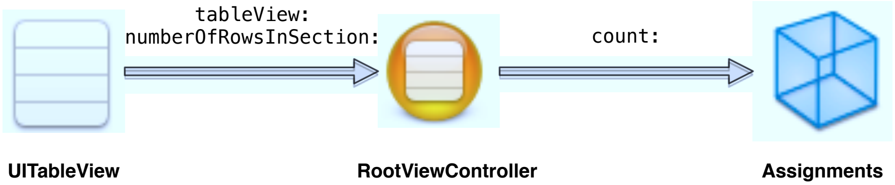
something like AssignmentViewController. Because it's easy to make an error making these changes and because the changes have nothing to do with what we're working on in this chapter, we'll leave the names as they are.

1.4 Adding a Model Object

A table view needs a data source that can answer three basic questions:

- How many sections are in the table?
- For each section, how many rows are in this section?
- For each row, what goes in this row?

In the project created for us from the template, the RootViewController is the table view's data source. We're going to add a model object to our project. This is the classic MVC (Model View Controller) decoupling. The view talks to the controller and the controller talks to the model.



The table view sends the message tableView:numberOfRowsInSection: to the controller. The controller in turn sends the message count to the Assignments object, our model object. The model object responds to the controller with the number of assignments, which the controller then relays on to the table view so that it knows how many rows will be displayed.

To implement this, in Xcode add a new Objective-C class that is a subclass of NSObject and name it Assignments.

An Assignments object will need to let the controller know how many total assignments there are and be able to tell the controller which assignment number goes in a given row. It will keep the collection of assignments internally in an NSArray named assignmentArray. This is all captured in the header file.

Download FromTo/DailyShoot1/Classes/Assignments.h

```
#import <Foundation/Foundation.h>
```

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Home page for iPad Programming
http://pragprog.com/titles/sfipad/ipad-programming
Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates
http://pragprog.com/updates
Be notified when updates and new books become available.

Join the Community
http://pragprog.com/community
Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy
http://pragprog.com/news
Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/sfipad/ipad-programming.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	`support@pragprog.com`
Non-English Versions:	`translations@pragprog.com`
Pragmatic Teaching:	`academic@pragprog.com`
Author Proposals:	`proposals@pragprog.com`
Contact us:	1-800-699-PROG (+1 919 847 3884)