treehouse

iOS 6
Foundations

Jesse Feiler

treehouse™

iOS 6 Foundations

treehouse™

iOS 6 Foundations

Jesse Feiler

WILEY
A John Wiley and Sons, Ltd, Publication

This edition first published 2013

©2013 John Wiley & Sons, Inc.

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. and/ or its affiliates in the United States and/or other countries, and may not be used without written permission. All trademarks are the property of their respective owners. John Wiley & Sons, Inc. 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 978-1-118-35657-9 (paperback); ISBN 978-1-118-57008-1 (ebook); ISBN 978-1-118-57010-4 (eMobi); ISBN 978-1-118-57009-8 (ePDF)

Set in Chaparral Pro Light 10/12.5 by Indianapolis Composition Services

Printed in the U.S. by Command Web Missouri

About the Author

JESSE FEILER is a developer, consultant, and author specializing in Apple technologies. He is the creator of Minutes Machine for iPad, the meeting management app available in Apple's App Store. He is also Software Architect for PlattInfo, the network of walk-up touch-screen kiosks in downtown Plattsburgh, New York. As a consultant, he has worked with small businesses and nonprofits on projects such as production control, publishing, and project management, usually involving FileMaker.

His books include:

- *iWork For Dummies*
- *Dashcode For Dummies*
- *FileMaker Pro in Depth*
- Sams *Teach Yourself Core Data in 24 Hours*
- Sams *Teach Yourself Objective-C in 24 Hours*
- *The Bento Book*

He is heard regularly on WAMC Public Radio for the Northeast's *The Roundtable*. He is a member of the City of Plattsburgh Planning Board and the Saranac River Trail Advisory Committee. A native of Washington DC, he has lived in New York City and currently lives in Plattsburgh, NY.

He can be reached at northcountryconsulting.com.

The photos in Chapter 17 show one of the City of Plattsburgh's PlattInfo kiosks. PlattInfo is a network of walk-up touch-screen kiosks powered by FileMaker. Jesse Feiler is Software Architect for PlattInfo. PlattInfo artwork by Kelly Chilton (`hey@kellychilton.com` or `www.kellychilton.com`). You can find out more about PlattInfo at PlattInfo.com.

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
Associate Commissioning Editor: Ellie Scott
Development Editor: Kezia Endsley
Copy Editor: Kezia Endsley
Technical Editor: Aaron Crabtree
Editorial Manager: Jodi Jensen
Senior Project Editor: Sara Shlaer
Editorial Assistant: Annie Sullivan

Marketing

Associate Marketing Director: Louise Breinholt
Marketing Manager: Lorna Mein
Senior Marketing Executive: Kate Parrett
Marketing Assistant: Tash Lee

Composition Services

Senior Project Coordinator: Kristie Rees
Compositor: Indianapolis Composition Services
Proofreader: Linda Seifert
Indexer: Potomac Indexing, LLC

Author's Acknowledgments

Many people have helped to make this book possible. At Treehouse and Wiley, Chris Webb and Kezia Endsley, brought the book from the initial idea to fruition. My agent, Carole Jelen, as always has been creatively supportive as the book has proceeded.

The tech editor, Aaron Crabtree, was great to work with, and I appreciate his help enormously. (You can find Aaron on Twitter at @aaron_crabtree and on the web at www.tapdezign.com.) Notwithstanding the help of so many people, any errors are mine. If you do find an error, please contact me through northcountryconsulting.com so that we can correct it in the next printing. And if you register on northcountryconsulting.com, we'll let you know of any updates.

Contents

About the Author v
Publisher's Acknowledgments. vi
Author's Acknowledgments. viii

Introduction. 1

Who Should Read This Book? 1
What You Will Learn 2
How to Use This Book 3
Using This Book with Treehouse 4

Part 1: Introducing iOS 6

CHAPTER ONE

Getting Started with iOS 6 7

Doing Your Homework 8
Getting Yourself Ready 8
Adopting a Developer's Point of View 8
Exploring the App Store 9
Reading Reviews 9
Understanding the App World—Past, Present, and Future. 9
Looking at the Master-Detail Application Template 10
Registering as a Developer. 15
Introducing Basic Programming Concepts 16
Object-Oriented Programming in Objective-C 17
Objects in Objective-C 17
Messaging in Objective-C 19
Frameworks 19
Graphical Coding 20
Model-View-Controller 20
Installing and Using Xcode 21
Summary 23

CHAPTER TWO

Getting Up to Speed with Xcode. 25

Using the Workspace Window 26
Exploring the Jump Bar 27
Related Items Pop-up Menu 27
Back/Forward Navigation Buttons 30
Using the Jump Bar to Navigate 30

Exploring the Toolbar . . . 30
Building and Running Projects . . . 31
Choosing a Scheme . . . 31
Enabling and Disabling Breakpoints . . . 32
Activity Viewer . . . 32
Editor Buttons . . . 32
View Buttons . . . 33
Organizer . . . 35
Selector Bars . . . 35
Exploring the Tab Bar . . . 35
Using Projects . . . 36
Exploring the Editor Area . . . 37
Using Editing Preferences . . . 39
Using Code Completion . . . 39
Handling Indentation . . . 39
Using Fix-It . . . 40
Using Code Completion . . . 41
Exploring the Navigators . . . 42
Using the Project Navigator . . . 42
Using Groups . . . 43
Spotting Missing Files . . . 44
Using the Navigator Controls . . . 44
Using the Search Navigator . . . 47
Using the Other Navigators . . . 48
Exploring the Utilities . . . 49
Using the File Inspector . . . 49
Using Quick Help . . . 50
Using Inspectors . . . 50
Using the Libraries . . . 51
Summary . . . 51

CHAPTER THREE
Looking Ahead—Planning Your App . . . 53
Answering the Money Question . . . 54
Preparing Version 2 . . . 55
Submitting the App to the App Store . . . 55
Identifying Your App and Yourself . . . 56
Bundle Settings . . . 57
Settings for the App Store . . . 57

Setting Marketing Data (Discoverability) 59
Describing Your App's Requirements 61
Specifying Integration Features 62
Celebrating Learning iOS with Your App! 62
Summary 62

CHAPTER FOUR
Designing the Party Planner App 63
Planning the App: The Choices 63
Identifying Your App and Yourself 64
Setting Marketing Data (Discoverability) 64
Describing Your App's Requirements 64
Specifying Integration Features 65
Designing the App: The Conversation 65
What Kind of Data Do You Need to Track? 65
How Persistent Is the Data? 65
How Much Data Is There? 65
Is There Anything Else You Need to Consider? 66
Getting Started with the Template 67
Choosing the Right Template 67
Exploring Other Templates 67
Creating the Project 68
Getting Started with the Data 72
Introducing Core Data 72
Building Your Data Model 76
Summary 78

Part 2: Storyboards: The Building Blocks of iOS Apps

CHAPTER FIVE
Walking Through the iPhone Storyboard 81
Introducing Storyboards 81
Looking at the Storyboarding Process 82
Looking at Storyboarding for the Template 82
Introducing the iOS Simulator 83
Walking Through the Template and the Storyboard 88
Looking at a Scene 88
Considering View Controllers 88
Considering Views 90
Looking at a Segue 90

Looking at the Storyboard 90
Looking at the Scenes in the Storyboard 92
Looking at Segues and Relationships in the Storyboard 93
Exploring the Navigation Controller 93
Creating Your Own View Controllers 94
Adding Objects from the Library 98
Summary 100

CHAPTER SIX
Working with Storyboard Inspectors 101
Looking at the Party Planner App 102
Using Outlets and Actions 104
Looking at Outlets 104
Looking at Actions 105
Exploring the Storyboard Inspectors 105
Using the File Inspector 106
Renaming Files and Classes 107
Changing File Location 107
Using Auto Layout 108
Using the Identity Inspector 108
Setting the Class 109
Setting the Label 109
Using the Attributes Inspector 110
Using the Size Inspector 111
Using the Connections Inspector 113
Connecting a Button (Overview) 114
Exploring the Three Icons Below a View Controller 114
Summary 117

CHAPTER SEVEN
Laying Out Your Scenes and Views 119
Using Springs and Struts 120
Using Auto Layout 123
Understanding Intrinsic Content 124
Using Constraints 125
Working with Content Hugging 125
Using Content Compression Resistance 125

Setting and Editing Priorities 125
Working with Menus 126
Summary 128

Part 3: Building the Party Planner App

CHAPTER EIGHT
Building on the Data Model 131
Expanding the Data Model 132
Expanding the Interface with Entities 132
Filling in the Attributes for the Entities 134
Building Relationships 136
Creating a Basic Relationship 136
Refining a Relationship 139
Next Steps 140
Building the Detail View Controller 142
Creating the Party Class from the Data Model 144
Looking at the Existing KVC Code 144
Creating the Classes from the Data Model 145
Looking at the Code 148
Summary 151

CHAPTER NINE
Building the Detail Data View 153
Using the Party Class 154
Getting the Core Data Stack Info 155
Setting Up the Managed Object Model 157
Setting Up the Persistent Store Coordinator 158
Setting Up the Managed Object Context 159
Adding the Managed Object Context to the MasterViewController 159
Creating the New Managed Object 162
Setting Attributes of the New Managed Object 163
Saving the Managed Object Context 164
Connecting Interface Elements to Properties 164
Checking Existing Connections 164
Checking a Connection from the .h File 164
Checking a Connection from the Storyboard 166
Removing Existing Connections 166

Changing Existing Connections 167
Creating New Connections. 167
Laying Out the Detail View 170
Cleaning Up the Experiments 170
Adding a Field to the Storyboard. 174
Adding More Fields to the Storyboard 176
Creating and Connecting the Properties 178
Displaying the Data 180
Creating the iPad Interface 180
Summary 181

CHAPTER TEN
Saving and Restoring Data 183
Understanding the Editing Interface 185
Setting Up the Edit-Done Button 186
Handling Universal Apps 186
Setting Up the iPad Managed Object Context 187
Setting Up the iPhone Managed Object Context. 188
Adding the Button 188
Implementing setEditing 189
Adjusting the Interface for Editing. 189
Saving the Data 192
Moving the Data to the Party Instance 193
Saving the Data 194
Retrieving Data 198
Testing the App 198
Summary 200

CHAPTER ELEVEN
Testing the App with the Debugger. 201
Exploring the Debugger from a Basic Template 202
Setting Up the Debugger 204
Finding an Error. 204
Configuring Behaviors Preferences 205
Setting a Breakpoint 207
Inspecting Variables 208
Inspecting Objects 211
Writing a Console Message 212
Editing Breakpoints 214
Summary 215

Part 4: Using Table and Collection Views

CHAPTER TWELVE

Exploring the Table View in the Template 219

Introducing Table Views, Protocols, and Delegates 220
Looking at Table Views 220
Using Views on an iPhone 220
Using Views on iPad 225
Using Table Views for Data Display and Editing 228
UITableView High-Level Architecture 229
Introducing Protocols and Delegates 231
Exploring the Issue of Multiple Inheritance 232
Declaring a Delegate that Adopts a Protocol 233
Adopting a Protocol 233
Assigning an Object to a Delegate 233
Declaring a Protocol 234
Tracking Down the Protocol, Delegate, and Data Source Structure in UITableView . . . 235
Looking at the Master View Controller 238
Looking at the .h File 238
Looking at the .m File 238
Summary 244

CHAPTER THIRTEEN

Formatting Table Cells 245

Converting the Detail View to a Table View for iPhone 246
Clearing Out the Text Fields on iPhone 248
Adding the Table View on iPhone 252
Converting the Detail View to a Table View on iPad 257
Adding the Table View on iPad 257
Preparing the Prototype Cell in the Storyboard 262
Configuring and Returning a Single Table Cell 265
Setting the Detail Item 266
Using Accessors for Declared Properties 266
Segues: Exploring the Difference Between Relationship and Action/Manual Segues 268
Using a Custom Subclass of NSManagedObject for the Detail Item 269
Configuring the Detail Item 271
Removing configureView 271
Implementing the Data Source Protocol for the Detail View Controller 271
Implementing the Table View Delegate Protocol for the Detail View Controller 272
Summary 274

CHAPTER FOURTEEN
Editing Table Views 275
Modifying the Data Model to Store Row Sequence 276
Looking at the Fetched Results Controller Ordering 276
Adding a displayOrder Attribute 278
Refreshing the Data Store 282
Enabling the Table View Reordering Features 282
Moving the Rows and Saving the New Order 283
Rearranging the Elements in the Table View 283
Calculating and Saving the displayOrder Property for a Move 287
Adding a New Object 290
Deleting an Existing Object 292
Summary 293

Part 5: Interacting with Users

CHAPTER FIFTEEN
Telling Users the News: Alerts and NSError 297
Reviewing User Interaction on iOS 298
Analyzing an Alert 299
Thinking About a Save Alert 299
Planning to Handle the Error 300
Getting the User's Perspective 300
Adding the Error's Perspective 300
Adding Your Perspective 301
What You Must Do to Handle Errors 302
Implementing a Data Store Error Alert 303
Handling a Non-Error Error 303
Posting the Alert 305
Adding a Log Message 307
Summary 308

CHAPTER SIXTEEN
Getting Input from Users: Alerts and Action Sheets 309
Using Alerts with Multiple Buttons 312
Adding the Buttons 312
Handling the Buttons 313
Using Action Sheets 315
Looking at Action Sheets 315
Managing Action Sheets 316
Summary 317

CHAPTER SEVENTEEN

Back to the Storyboard: Enhancing the Interface 319

Cleaning Up Some Loose Ends 319
Setting Up New Objects 320
Using Storyboards Today 320
Using the Utility Application Template 321
Using the Tabbed Application Template 325
Editing Basic Party Data 327
Reusing the Basic Detail Data Code 327
Editing the Storyboard 330
Adding the Segue to the Code 334
Adding the Navigation Bar and Buttons 335
Handling the Date Field 338
Converting the Text Field String to a Date 338
Converting a Date to a Text Field String 339
Handling Relationships to Guests and Food 339
Handling Relationships with Static Fields 340
Handling Relationships with a Table View 340
Grouping the Detail View 340
Implementing the Guest View Controller 344
GuestViewController.h 345
GuestViewController.m 346
Connect the Guest View Controller to the Accessory View 346
Summary 347

Index 349

Introduction

GETTING STARTED WITH iOS 6 is easier than ever. Long-time iOS programmers who started programming with iPhone OS all those years ago (in 2007) might scarcely recognize the tools at their disposal. Some people thought that programming iPhone—and later, iPad—was just too hard. And maybe it was, but the engineers at Apple were working feverishly to transfer major aspects of app development from external developers to in-house Apple engineers. The process accelerated with iOS 5 and, with iOS 6, newcomers to iOS development have a wealth of riches in the frameworks and tools at your command.

Who Should Read This Book?

This book is for people who want to learn about developing iOS apps. It provides a hands-on tutorial for you to develop your first app. Some people will use the book to launch themselves on a career as an app developer. For others, the book will serve to introduce the basics of iOS. This means that managers, clients, marketers, and others who need to work with iOS can get up to speed.

The assumption in this book is that you know a programming language and the basics of computer programming and software development. You don't need an in-depth knowledge of a programming language, and, in some cases, that may actually be a disadvantage. It doesn't particularly matter which programming language you're familiar with, although if it is a modern object-oriented programming language such as C++,

C#, Java Python, and Ruby, that's great. If you are familiar with the object-oriented features of Perl and PHP, that knowledge will help you along the way.

What about "the basics of computer programming and software development"? Many people (including many people in the technology world) don't understand how software is developed today. Unfortunately, you can still find many books and courses that begin by teaching you how to develop a basic program to do something like balance a checkbook. Leaving aside for the moment the fact that most people don't balance a checkbook manually any more (online banking has changed all that), if your goal is to build the next killer app in the music world or to manage a recycling center or whatever, that checkbook-balancing app may not be relevant. However, if you want to write innovative apps for the 21st Century and the great iOS operating system, this book is for you.

What You Will Learn

The first thing that you'll learn is right here in this paragraph. iOS is the operating system of iPhone, iPad, and iPod touch. It is written in Objective-C. The iOS software is developed with the Xcode integrated development environment (IDE). iOS (as well as OS X) is a product of Apple, as is Xcode. Although Objective-C is not an Apple product (there are several implementations), most people refer to Apple's documentation for the last word on Objective-C and its features. (There is no single published standard other than the Apple documentation.)

As a result of these three points, it is sometimes hard to discern where the operating system and its frameworks end and the language begins, not to mention which features are implemented in Xcode and which features are part of the framework or even the language. They all work together in a seamless fashion. Don't try to tear them apart and learn the language separately from the frameworks or Xcode. Just remember that they are all part of an extraordinary whole. As you work through the book, you'll see how things fit together.

In Part I, "Introducing iOS 6," you'll see how the key components of your development environment fit together. You'll learn about the structure of iOS 6, and you'll see how to use Xcode. You'll walk through the process of thinking about an app and see how to begin defining it.

In Part II, "Storyboards: The Building Blocks of iOS Apps," you get to work designing your app's interface. Some people think of the interface as an add-on, thinking that the code you write is the real thing. Don't fall into that trap; the interface *is* your app. It's what people see and use. The interface comes first, and the code is used to support it. This is particularly important with iOS because, as noted previously, the functionality can be implemented in the iOS framework itself, in the Objective-C language, in Xcode, or in some combination of

them. But the storyboard—a step-by-step walk through the interface—brings them all together.

And, yes, if you're wondering if these storyboards are anything like storyboards for movies or games, you're right. Today's storyboards can be traced to the Walt Disney studios in the 1930s. Look up storyboards in Wikipedia and you'll see that long before iOS, they were used to plan *Gone with the Wind* (1939). You'll also find earlier references such as Constantin Stanislavski's use of storyboarding in theatrical productions in the 1890s.

In Part III, "Building the Party Planner App," you'll use the Core Data Model editor in Xcode to build your data store using graphical tools. From there, you'll move on to customize the Xcode template that will become your app. In this section, you also learn how to save and restore data and how to use the debugger.

In Part IV, "Using Tables and Collection Views," you'll see how to use a critical component of iOS. Structuring data and allowing users to edit it is a common task for developers and users. With the built-in table functionality, much of your work is already done for you.

Finally, in Part V, "Interacting with Users," you circle back to the world of storyboards. There are a number of specific user interface elements that need to be covered so that you can complete your app. Here is where you find them.

How to Use This Book

There are no "reading police;" you can read this (or any) book when, where, and how you want to. (Actually, there is one generally accepted taboo with regard to reading a book — do not look at the last page of a murder mystery until you've read everything that comes before.)

That said, it's important to note that the practical example in this book—the Party Planner app— is built, chapter-by-chapter, as you read through the book. However, if you spot something that you want to explore out of sequence, it's easy to do so. The example code in each chapter is posted at `wiley.com/go/treehouse/ios6foundations` as well as on my website at `northcountryconsulting.com`. If you want to jump into Chapter 12, for example, you can download the code from Chapter 11 and modify it as you read on. (Note that the code posted on the web for each chapter represents the code as it is at the *end* of the chapter.)

This book describes iOS 6. Many of the concepts have been introduced in previous versions, but there also are new features that make their debut in iOS 6. This badge identifies those new features.

Using This Book with Treehouse

Just to be clear, you don't have to be a Treehouse member to use this book. However, the online videos at `teamtreehouse.com` do supplement the content quite nicely. When there is a video that covers the same content that is being covered in the book, you will see the Video icon in the margin and a link to the relevant video. Viewing all the videos and completing badges is a good way of testing what you have learned in the book (and of showing off your new skills to others).

If you ever get stuck on a concept in the book, Treehouse has a great community of members who would be more than happy to help you. You can find them in the official Treehouse members group on Facebook.

Ready to go? Let's get started.

part 1

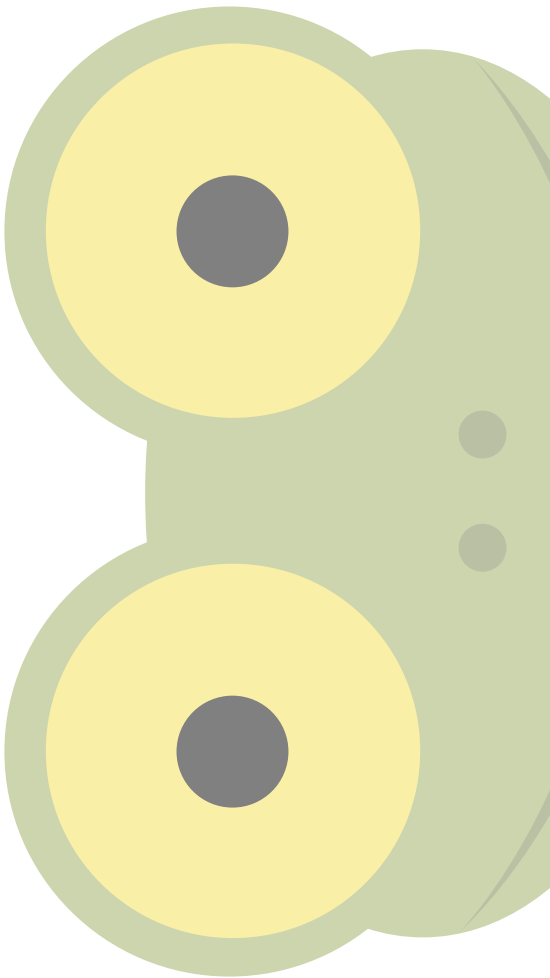
Introducing iOS 6

chapter one Getting Started with iOS 6

chapter two Getting Up to Speed with Xcode

chapter three Looking Ahead—Planning Your App

chapter four Designing the Party Planner App



chapter one

Getting Started with iOS 6

WELCOME! IT'S GREAT to welcome new developers to the world of iOS 6, and I'm happy to help you get started. For most developers with experience on other platforms, iOS is unlike the development environments they are used to. For starters, it lets you build apps for some of the most exciting products today (and, indeed, for many, many days). When you build an app for iPhone, iPod touch, or iPad you become part of the exciting ecosystem centered on Apple's extraordinary technologies and designs. You can find many books, articles, and media stories about Apple, its products, and their designs. There is analysis and prognostication; there are books and training materials for users. And there are books and training materials for that special cadre of people who extend Apple's handiwork: the developers.

This chapter gets you started as quickly as possible. You'll see how to register as a developer. After that, you'll be able to download tools and documentation from `developer.apple.com`. In this chapter you learn the basics of the Objective-C programming language and the highlights of the history of iOS 6—how we got here. Then you'll find a high-level overview of the Xcode integrated development environment (IDE). Before you know it, you'll be following the steps at the end of the chapter to build your first iOS 6 app.

Doing Your Homework

How did you decide to start developing for iOS? Some people use iOS devices and just want to find out more about what makes them tick. Other people have an idea for a great app and would like to build it themselves. Still others are IT professionals who want to expand their skills to this new platform. And others are IT professionals who have been asked to find out how to port an existing or planned project to iOS devices.

Getting Yourself Ready

No matter which category you find yourself in, you probably need to do a bit of homework before you start. You should have some background in programming. It can be long ago or recent, and it can be in advanced languages derived from C or in scripting languages such as PHP. (As noted in the Introduction, some experience with object-oriented programming can definitely help.)

You should be familiar with iOS from a user's perspective. If you plan to develop for only one of the devices (iPhone or iPad, perhaps), you can just explore that device and its features. However, to fully understand the iOS ecosystem, it's good to have both devices and to share data between them using iCloud.

Apple has fairly aggressively pushed out new versions of its devices on roughly an annual basis. It has followed a pattern of dramatically lowering the prices on the previous version of each device as a new one becomes available. You may be able to find a model that is several years old (you may even know someone who can give you one) that you can use for testing. As long as you can install iOS 6 on it, you'll have a test device and not have to worry about mixing up your actual data with test data.

Adopting a Developer's Point of View

When people use computers, they usually focus on a task that they need to accomplish. As a developer, you need to learn a secondary focus: watch *how* people do things rather than *what* they do. Develop this skill and use it to observe how people behave with iOS devices. You have a perfect test subject: yourself.

When something goes wrong or doesn't work the way you expect it to, don't just push on to try it another way. Take a moment to think back not so much about *what* you did wrong but *why* you did it. Did you mistake one icon for another? (Perhaps the icon's meaning wasn't clear.) Did you assume that an action would be carried out differently than it actually was? It doesn't matter if you made a mistake or if the app has a bug in it; in either case, something broke the chain of logic in the user interface and the app. Get used to spotting and analyzing these little glitches. Each one is a learning experience if you just pay attention to it before moving on with the task at hand.

Exploring the App Store

As of this writing, Apple's App Store has surpassed three quarters of a million apps. There are all kinds of apps for all kinds of purposes. Explore the App Store to see what people are writing. If you have an idea for an app, look to see how other people are approaching the topic.

Even if you have an idea for your own app, continue browsing in the App Store in other genres. Many apps are free, so download and install any that seem interesting in any way. If you are planning to build an app for people to use for keeping track of livestock breeding, you may spot an interface element in a game or other app that would be useful in your own app. You can't see every app in the App Store, but keep yourself up to date.

Reading Reviews

Read reviews on the App Store as well as reviews in the media, including blogs. Remember at this point that you're looking for points that reviewers pick up on, both good and bad. Listen to friends as they point out what they like and dislike about the apps they use.

Understanding the App World—Past, Present, and Future

For most people, the app world began in the summer of 2008. On July 10, the App Store opened (it's part of iTunes which received an update). The next day, July 11, the iPhone 3G went on sale. It ran iOS 2.0.1. The phrase "there's an app for that" was a key part of the marketing of the new iPhone 3G. Before long, people around the world understood the basics of apps that could be downloaded from iTunes directly onto an iPhone.

The app world is just a few years old. Every day, new people join it as they get their first iOS device or, as in your case, they decide to start developing apps. As you explore this world, keep a few critical milestones in mind to help you to make sense of information that you find in your studies:

- As noted, the first release of iOS to developers was iOS 2 in July of 2008.
- iOS 3 in June of 2009 added new features such as copy-and-paste. (Yes, in case you didn't know or have forgotten, you didn't have them at the start.)
- iOS 4 was released in June of 2010. iOS 4.2.1 in November supported the iPad.
- iOS 5, released in October of 2011 was the first unified release for iPhone, iPad, and iPod touch.
- iOS 6 (the subject of this book) was released in September 2012.

Along with new versions of the iOS operating system, the engineers at Apple were updating OS X as well as Xcode, the tool for developers of both operating systems as well as third-party apps. (Xcode is discussed later in this chapter and in Chapter 2, "Getting Up to Speed with Xcode.")

Xcode changes less frequently than the operating systems, but there have been very significant changes accompanying the unification of iOS for all the iOS devices, as well as major changes to the structure of Xcode itself.

Looking at the Master-Detail Application Template

All of this background information matters because, as you browse the web and discussion groups, it's important that you know what version you're looking at. Here's an example of how the evolution has taken shape. Xcode contains a number of templates that you can use as the basis for your apps. One of the commonly used templates today is the Master-Detail Application template for iOS (discussed more later in this chapter). It often serves as the basis of apps, and it will serve as the foundation of the app that you will build throughout this book. When you build the app, you can run it on the iPad simulator. Figure 1-1 shows it running in the simulator in landscape mode.

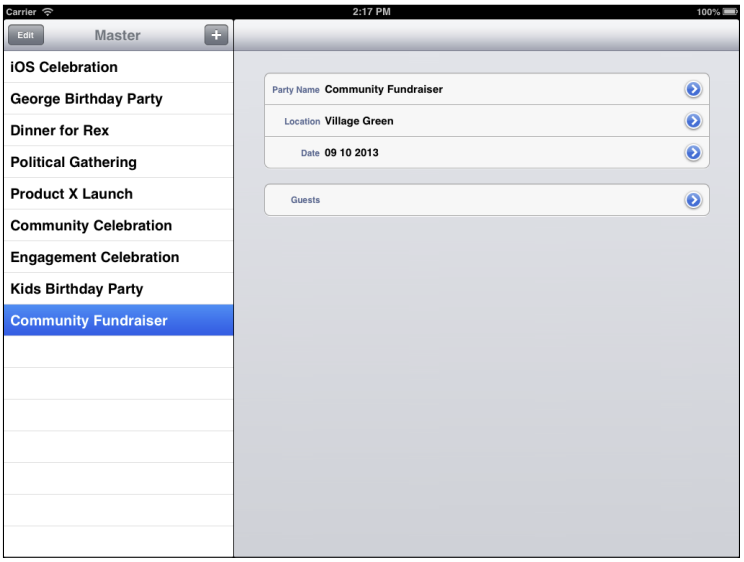


FIGURE 1-1 Master-Detail Application template running on the iPad simulator in landscape mode.

What you see in Figure 1-1 is an iPad feature called a *split view*. It combines two views in one. (Settings uses this architecture so you may recognize the bare bones of the design.) On the