Illustrations

Preface

1 iPhone for Programmers apps and the technologies they introduce. xxx

1 Introduction to iPhone App Development

1.1 Key online documentation for iPhone developers. 3
1.2 iPhone gestures. 4
1.3 iPhone hardware. 5
1.4 iPhone 3.x default apps. 5
1.5 New landscape keyboard. 6
1.6 Popular iPhone apps in the App Store. 7
1.7 New iPhone 3.x software features (www.apple.com/iphone/softwareupdate/). 8
1.8 Design patterns used in iPhone for Programmers. 10
1.9 Cocoa frameworks (developer.apple.com/iphone/library/navigation/Frameworks/index.html). 11
1.10 iPhone gestures on the simulator (developer.apple.com/IPhone/library/documentation/Xcode/Conceptual/iphone_development/125-Using_iPhone_Simulator/iphone_simulator_application.html). 15
1.11 Clicking the Build and Go button to run the Painter app. 18
1.12 Painter app with a blank canvas. 19
1.13 Painter app settings. 19
1.14 Drawing with a new brush color. 20
1.15 Changing the line color and line size to draw the stem and grass. 21
1.16 Changing the line color and line size to draw the rain. 21

2 iPhone App Store and App Business Issues

2.1 iPhone Development Team responsibilities. (iTunes Connect Developer Guide version 4.7, July 10, 2009.) 25
2.2 Points and suggestions from the iPhone Human Interface Guidelines. 31
2.3 iPhone functionality that is not available on the simulator. 32
2.4 Custom app icon design firms. 33
2.5 Languages available for localizing your iPhone apps. 34
2.6 Characteristics of great apps. 35
2.7 Reasons apps are rejected by Apple. 36
2.8 Ways to monetize your app. 36
2.9 Free iPhone apps that are building brand awareness. 37
Illustrations

2.10 iTunes Connect Overview page for adding an app. 38
2.11 App ratings. 39
2.12 App Stores worldwide. 40
2.13 Virtual goods. 41
2.14 Categorizing your products for sale using In App Purchase. 42
2.15 iTunes Connect modules. 43
2.16 Marketing Resources for iPhone app developers (developer.apple.com/iphone/). 43
2.17 Popular social media sites. 44
2.18 iPhone app review sites. 45
2.19 Internet public relations resources. 46
2.20 Mobile advertising networks. 47
2.21 iPhone development anecdotes, tips and humor. 48
2.22 Other popular platforms. 49
2.23 iPhone Reference Library documentation. 50

3 Welcome App
3.1 Welcome app. 52
3.2 Instruments capabilities. 53
3.3 Welcome to Xcode window. 54
3.4 Xcode templates. 54
3.5 Xcode toolbar. 55
3.6 Xcode toolbar elements. 55
3.7 Common Xcode keyboard shortcuts. 55
3.8 New Project window. 56
3.9 Naming your project. 57
3.10 WelcomeTest project window. 57
3.11 Adding a file to the Welcome project. 58
3.12 MainMenu.xib window. 58
3.13 Interface Builder’s app window. 59
3.14 Interface Builder’s Library window. 59
3.15 Adding an Image View via the Library Window. 60
3.16 Sizing handles on an Image View. 60
3.17 Image View Attributes tab in the Inspector window. 61
3.18 Adding a Label to a window. 62
3.19 Label Attributes tab of the Inspector window. 62
3.20 Completed Welcome app. 63
3.21 Xcode toolbar while an app is running. 63

4 Tip Calculator App
4.1 Entering the bill total and calculating the tip. 65
4.2 Adding a Text Field to the app window. 67
4.3 Adding a Label to the app window. 67
4.4 Placing a Label in the app window. 68
4.5 Adding a Slider to the app window. 69
4.6 App window after adding and sizing the Slider.
4.7 Positioning the “Tip” and “Total” Text Fields.
4.8 Completed Tip Calculator user interface.
4.9 Adding a new Objective-C class file to your Xcode project.
4.10 Controller class for the Tip Calculator app.
4.11 Adding a Custom Object to a nib file.
4.12 Changing the class of a custom object via the Identity tab.
4.13 Controller’s Connections tab of the Inspector window.
4.14 Connecting an outlet to a Text Field.
4.15 Connecting an action to a Controller object.
4.16 Controller class for the Tip Calculator app.
4.17 Relational operators in Objective-C.
4.18 Arithmetic operators in Objective-C.

5 Favorite Twitter® Searches App
5.1 Favorite Twitter Searches app.
5.2 Running the Favorite Twitter Searches app.
5.3 Interface Builder Button styles.
5.4 App window with a Label and two Text Fields.
5.5 Adding the “Save” Button to the app window.
5.6 View with a colored background and a white Label at the top.
5.7 App window with a Round Rect Button at the bottom.
5.8 Adding a UIScrollView to the app window.
5.9 Controller class for the Favorite Twitter Searches app.
5.10 Adding a Custom Object to a nib file.
5.11 Controller class for the Favorite Twitter Searches app.
5.12 Method awakeFromNib of class Controller.
5.13 Method refreshList of class Controller.
5.14 Method infoButtonTouched: of class Controller.
5.15 Methods addTag: and clearTags: of class Controller.
5.16 Method addNewButtonWithTitle: of class Controller.
5.17 Method buttonTouched: of class Controller.
5.18 UIButton’s sorting category.

6 Flag Quiz Game App
6.1 Flag Quiz Game app.
6.2 Correct answer in the Flag Quiz Game app.
6.3 Disabled incorrect answer in the Flag Quiz Game app.
6.4 Options screen of the Flag Quiz Game app.
6.5 Results alert after quiz completion.
6.6 Completed MainView GUI design.
6.7 MainViewController interface declaration.
6.8 Initializing the MainViewController class.
6.9 MainViewController’s viewDidLoad method.
6.10 MainViewController’s loadNextFlag method.
Illustrations

6.11 MainViewController’s submitGuess: method.  116
6.12 Resetting the quiz.  118
6.13 showInfo and flipsideViewControllerDidLoadFinish: methods.  119
6.14 NSString’s displayName category.  121
6.15 Segmented Control in the FlipsideView.  122
6.16 Switches in the FlipsideView.  123
6.17 FlipsideViewController interface declaration.  123
6.18 FlipsideViewController class implementation.  125

7 Spot-On Game App

7.1 Spot-On Game app.  130
7.2 Spot-On Game with a touched spot.  131
7.3 Game Over alert.  131
7.4 SpotOnViewController interface declaration.  132
7.5 SpotOnViewController’s user interface.  134
7.6 SpotOnViewController’s viewDidLoad method.  135
7.7 SpotOnViewController’s resetGame method.  136
7.8 SpotOnViewController’s addNewSpot method.  137
7.9 SpotOnViewController’s beginSpotAnimation: method.  139
7.10 SpotOnViewController’s touchesBegan:withEvent: method.  140
7.11 SpotOnViewController’s touchedSpot: method.  142
7.12 SpotOnViewController’s beginSpotEndAnimation method.  143
7.13 SpotOnViewController’s finishedAnimation:finished:context: method.  144
7.14 SpotOnViewController’s methods alertView:clickedButtonAtIndex:, shouldAutoRotateToInterfaceOrientation: and dealloc.  146

8 Cannon Game App

8.1 Completed Cannon Game app.  155
8.2 Cannon Game app with cannonball in flight.  156
8.3 CannonView’s interface declaration.  157
8.4 Global variable declarations in CannonView.m.  159
8.5 CannonView’s initWithCoder: and awakenFromNib methods.  160
8.6 CannonView’s newGame method.  161
8.7 CannonView’s timerFired: method.  162
8.8 CannonView’s showAlertWithTitle:message: and alertView:clickedButtonAtIndex: methods.  165
8.9 CannonView’s drawRect: method.  166
8.10 CannonView’s touchesBegan:withEvent:, touchesMoved:withEvent: and processTouch: methods.  170

9 Painter App

9.1 Painter app and its control panel.  174
9.2 Class Squiggle represents the points, color and width of one line.  175
Illustrations

9.3 Squiggle class implementation. 176
9.4 View for the frontside of the Painter app. 177
9.5 Method initWithCoder: of class MainView. 178
9.6 Methods resetView and drawRect: of class MainView. 178
9.7 Method drawSquiggle: of class MainView. 179
9.8 Touch-handling methods of class MainView. 180
9.9 Methods motionEnded:withEvent:, alertView:clickedButtonAtIndex:, canBecomeFirstResponder and dealloc of class MainView. 183
9.10 MainViewController interface. 184
9.11 Controller for the front side of the Painter app. 185
9.12 FlipsideViewController interface. 187
9.13 FlipsideViewController class. 188
9.14 The finished flipside interface. 191

10 Address Book App
10.1 List of contacts. 194
10.2 Viewing a single contact's details. 195
10.3 Add Contact screen. 195
10.4 Deleting a contact. 196
10.5 Controller for the main table of the Address Book app. 197
10.6 Method viewDidLoad of class RootViewController. 199
10.7 Method addContact of class RootViewController. 200
10.8 Method addViewControllerDidFinishAdding: of class RootViewController. 200
10.9 Methods and tableView: numberOfRowsInSection: and tableView:cellForRowAtIndexPath: of class RootViewController. 201
10.10 Method tableView:didSelectRowAtIndexPath: of class RootViewController. 202
10.11 Methods tableView:commitEditingStyle:forRowAtIndexPath:, shouldAutorotateToInterfaceOrientation: and dealloc of class RootViewController. 203
10.12 NSDictionary's sorting category. 204
10.13 AddViewController.xib in Interface Builder after placing the default TableView. 205
10.14 AddViewController's interface declaration. 205
10.15 Methods initWithNibName:bundle: and viewDidLoad of class AddViewController. 207
10.16 Methods doneAdding: and values of class AddViewController. 208
10.17 MethodseditableCellDidBeginEditing:, editableCellDidEndEditing: and editableCellDidEndOnExit: of class AddViewController. 209
10.18 Methods numberOfSectionsInTableView:, tableView:numberOfRowsInSection: and tableView:titleForHeaderInSection: of class AddViewController. 210
10.19 Method tableView:cellForRowAtIndexPath: of class AddViewController. 211
Illustrations

10.20 ContactViewController's interface declaration.  212
10.21 ContactViewController class displays information for a contact.  213
10.22 Interface for a UITableViewCell that contains a Label and a TextField.  215
10.23 EditableCell's class definition.  216

11 Route Tracker App

11.1 Approximate user location on world map.  221
11.2 Map just after the user presses Start Tracking.  222
11.3 User's route displayed on the map with arrows showing the user's direction.  222
11.4 Satellite and hybrid map views.  223
11.5 Statistics for a completed route.  223
11.6 MainWindow.xib in Interface Builder.  225
11.7 TrackingMapView interface declaration.  226
11.8 Method initWithFrame: of class TrackingMapView.  226
11.9 Method drawRect: of class TrackingMapView.  227
11.10 Methods addPoint: and reset of class TrackingMapView.  230
11.11 Methods mapView:regionWillChangeAnimated: and mapView:regionDidChangeAnimated: of class TrackingMapView.  231
11.12 Controller class for the Route Tracker app interface declaration.  232
11.13 Method viewDidLoad of class Controller.  233
11.14 Method toggleTracking of class Controller.  234
11.15 Methods resetMap, selectMapMode: and mapView:viewForAnnotation: of class Controller.  236
11.16 Method locationManager:didUpdateToLocation:fromLocation: of class Controller.  237
11.17 Methods locationManager:didUpdateHeading: and locationManager:didFailWithError: of class Controller.  238

12 Slideshow App

12.1 List of saved slideshows.  241
12.2 Slideshow playing in portrait and landscape orientations.  242
12.3 Editing the list of slideshows.  242
12.4 Creating a new slideshow.  243
12.5 Photo library.  243
12.6 Picking a photo.  244
12.7 RootViewController class controls the main list of slideshows.  246
12.8 Methods viewDidLoad of class RootViewController.  246
12.9 Method viewWillAppear: and addSlideshow of class RootViewController.  247
12.10 RootViewController methods nameViewController:didGetName:, and tableView:numberOfRowsInSection:.  248
12.11 Method tableView:cellForRowAtIndexPath: of class RootViewController.  249
13 Enhanced Slideshow App

13.1 iPhone photo library.  
13.2 Viewing a video.  
13.3 Setting the image transition effect.  
13.4 Flip effect—rotates slide horizontally revealing the next slide underneath.  
13.5 MediaItem class represents an image or a video.
Illustrations

13.6 MediaItem class implementation. 283
13.7 Presents the user with an interface for creating a MediaItem. 285
13.8 MediaItemCreator class implementation. 286
13.9 Interface for class Slideshow which represents a slideshow. 287
13.10 Methods init and initWithCoder: of class Slideshow. 288
13.11 Methods encodeWithCoder: and firstImage of class Slideshow. 289
13.12 NSCoding category For UIImage. 290
13.13 Method viewDidLoad of class RootViewController. 291
13.14 Method nameViewController:didGetName: of class RootViewController. 292
13.15 Method slideshowCellDidSelectEditButton: of class RootViewController. 293
13.16 Method tableView:cellForRowAtIndexPath: of class RootViewController. 294
13.17 Method initWithSlideshow: of class SlideshowDataViewController. 295
13.18 Method addPhoto of class SlideshowDataViewController. 295
13.19 Method imagePickerController:didFinishPickingMediaWithInfo: of class SlideshowDataViewController. 296
13.20 Method mediaItemCreator:didCreateMediaItem: of class SlideshowDataViewController. 298
13.21 Methods mediaPicker:didPickMediaItems: and addEffect of class SlideshowDataViewController. 299
13.22 Methods startSlide and actionSheet:clickedButtonAtIndex: of class SlideshowDataViewController. 299
13.23 Method tableView:cellForRowAtIndexPath: of class SlideshowDataViewController. 300
13.24 Method tableView:moveRowAtIndexPath:toIndexPath: of class SlideshowDataViewController. 302
13.25 Method applicationWillTerminate: of class EnhancedSlideshowAppDelegate. 303
13.26 Controller for a view that shows a slideshow. 303
13.27 Method nextImageViewWithMedia: of class SlideshowViewController. 305
13.28 Method changeSlide of class SlideshowViewController. 306
13.29 Method displayNewImage: of class SlideshowViewController. 306
13.30 Method displayNewVideo: of class SlideshowViewController. 308

14 Voice Recorder App

14.1 Voice Recorder app ready to record. 311
14.2 Visualizer during a recording. 312
14.3 Naming a recording. 312
14.4 Playing a saved recording. 313
14.5 E-mailing a recording. 313
14.6 VoiceRecorderViewController Interface declaration. 315
14.7 VoiceRecorderViewController’s finished view. 316
14.8 Method record: of class VoiceRecorderViewController. 317
14.8 Methods initWithNibName:bundle: and viewDidLoad of class VoiceRecorderViewController.

14.10 Method nameRecordingViewController:didGetName: of class VoiceRecorderViewController.


14.13 Controls a View for naming a recording.

14.14 Finished layout of NameRecordingViewController's view.

14.15 Implementation of NameRecordingViewController.

14.16 View that displays a visualization of a recording in progress.


14.18 Methods setPower: and clear of class Visualizer.


14.20 Controls the View where the user plays existing sound files.

14.21 Completed PlaybackViewController GUI.

14.22 Implementation for PlaybackViewController.

14.23 Methods sliderMoved:, togglePlay and updateVolume: of class PlaybackViewController.

14.24 PlaybackViewController methods timerFired: and record:.

14.25 Method playSound of class PlaybackViewController.

14.26 Methods stopSound, numberOfSectionsInTableView: and tableView:numberOfRowsInSection: of class PlaybackViewController.

14.27 Method tableView:cellForRowAtIndexPath: of class PlaybackViewController.

14.28 Method tableView:commitEditingStyle:forRowAtIndexPath: of class PlaybackViewController.

14.29 Method tableView:didSelectRowAtIndexPath: of class PlaybackViewController.

14.30 Methods tableView:accessoryButtonTappedForRowWithIndexPath: and mailComposeController:didFinishWithResult:error: of class PlaybackViewController.

15 Enhanced Address Book App

15.1 Viewing a contact.

15.2 Requesting a connection.

15.3 Getting a Bluetooth Connection Request.

15.4 Data model editor.

15.5 Controller that displays the contact information for a created contact.

15.6 Methods viewDidLoad and send of class ContactViewController.

15.7 Method peerPickerController:didConnectPeer:toSession: of class ContactViewController.

15.8 Methods peerPickerControllerDidCancel:, updateTitle and tableView:numberOfRowsInSection: of class ContactViewController.

15.9 Method tableView:cellForRowAtIndexPath: of class ContactViewController.
xxvi Illustrations

15.10 Controls the main view of the Enhanced Address Book app. 351
15.11 Method viewDidLoad of class RootViewController. 352
15.12 Method session:didReceiveConnectionRequestFromPeer: of class RootViewController. 353
15.13 Methods alertController:clickedButtonAtIndex: and alertControllerCancel: of class RootViewController. 354
15.14 Method receiveData:fromPeer:inSession:context: of class RootViewController. 355
15.15 Methods insertNewObject and addObjectControllerDidFinishAdding: of class RootViewController. 356
15.16 Methods numberOfSectionsInTableView: and tableView:numberOfRowsInSection: of class RootViewController. 358
15.17 Methods tableView:cellForRowAtIndexPath: and tableView:canMoveRowAtIndexPath: of class RootViewController. 359
15.18 Methods tableView:commitEditingStyle:forRowAtIndexPath: and tableView:canMoveRowAtIndexPath: of class RootViewController. 360
15.19 Method fetchedResultsController of class RootViewController. 361

16 Twitter® Discount Airfares App
16.1 Twitter Discount Airfares app showing several discount airfares. 365
16.2 Class that represents an airfare. 366
16.3 Implementation of class Airfare. 367
16.4 Controller for the root View of the Twitter Discount Airfares app. 367
16.5 RootViewController class implementation. 368
16.6 UITableViewDelegate delegate and data source methods of class RootViewController. 370
16.7 View that displays the website for purchasing flight tickets. 371
16.8 View that displays a website for purchasing flight tickets. 372
16.9 Class that connects with Twitter web services and returns data. 373
16.10 Implementation of TwitterConnection. 374
16.11 Class that gets tweets and parses them for information. 376
16.12 AirfareFinder class implementation. 377
16.13 XML containing information about a single tweet. 378
16.14 NSXMLParser delegate methods. 379
16.15 Method parseCost of category parsing of class NSString. 381
16.16 Method parseLocation of category parsing of class NSString. 382
16.17 Method removeLink of category parsing of class NSString. 383
16.18 UITableViewDelegateCell that displays information about an airfare. 383
16.19 AirfareCell class implementation. 384