

Illustrations

Preface

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

I 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

xviii 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 .	69
4.7	Positioning the “ Tip ” and “ Total ” Text Fields .	70
4.8	Completed Tip Calculator user interface.	70
4.9	Adding a new Objective-C class file to your Xcode project.	71
4.10	Controller class for the Tip Calculator app.	72
4.11	Adding a Custom Object to a nib file.	74
4.12	Changing the class of a custom object via the Identity tab.	74
4.13	Controller’s Connections tab of the Inspector window.	74
4.14	Connecting an outlet to a Text Field .	75
4.15	Connecting an action to a Controller object.	76
4.16	Controller class for the Tip Calculator app.	76
4.17	Relational operators in Objective-C.	80
4.18	Arithmetic operators in Objective-C.	81

5 Favorite Twitter® Searches App

5.1	Favorite Twitter Searches app.	84
5.2	Running the Favorite Twitter Searches app.	85
5.3	Interface Builder Button styles.	86
5.4	App window with a Label and two Text Fields .	87
5.5	Adding the “ Save ” Button to the app window.	88
5.6	View with a colored background and a white Label at the top.	88
5.7	App window with a Round Rect Button at the bottom.	89
5.8	Adding a UIScrollView to the app window.	89
5.9	Controller class for the Favorite Twitter Searches app.	90
5.10	Adding a Custom Object to a nib file.	92
5.11	Controller class for the Favorite Twitter Searches app.	93
5.12	Method <code>awakeFromNib</code> of class <code>Controller</code> .	95
5.13	Method <code>refreshList</code> of class <code>Controller</code> .	95
5.14	Method <code>infoButtonTouched:</code> of class <code>Controller</code> .	97
5.15	Methods <code>addTag:</code> and <code>clearTags:</code> of class <code>Controller</code> .	97
5.16	Method <code>addNewButtonWithTitle:</code> of class <code>Controller</code> .	99
5.17	Method <code>buttonTouched:</code> of class <code>Controller</code> .	100
5.18	<code>UIButton</code> ’s sorting category.	101

6 Flag Quiz Game App

6.1	Flag Quiz Game app.	103
6.2	Correct answer in the Flag Quiz Game app.	104
6.3	Disabled incorrect answer in the Flag Quiz Game app.	104
6.4	Options screen of the Flag Quiz Game app.	105
6.5	Results alert after quiz completion.	105
6.6	Completed MainView GUI design.	107
6.7	<code>MainViewController</code> interface declaration.	108
6.8	Initializing the <code>MainViewController</code> class.	110
6.9	<code>MainViewController</code> ’s <code>viewDidLoad</code> method.	111
6.10	<code>MainViewController</code> ’s <code>loadNextFlag</code> method.	113

xx Illustrations

6.11	MainViewController's submitGuess: method.	116
6.12	Resetting the quiz.	118
6.13	showInfo and flipsideViewControllerDidFinish: 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 awakeFromNib 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

9.3	Squiggle class implementation.	176
9.4	View for the frontside of the Painter app.	177
9.5	Method <code>initWithCoder:</code> of class <code>MainView</code> .	178
9.6	Methods <code>resetView</code> and <code>drawRect:</code> of class <code>MainView</code> .	178
9.7	Method <code>drawSquiggle:</code> of class <code>MainView</code> .	179
9.8	Touch-handling methods of class <code>MainView</code> .	180
9.9	Methods <code>motionEnded:withEvent:</code> , <code>alertView:clickedButtonAtIndex:</code> , <code>canBecomeFirstResponder</code> and <code>dealloc</code> of class <code>MainView</code> .	183
9.10	<code>MainViewController</code> interface.	184
9.11	Controller for the front side of the Painter app.	185
9.12	<code>FlipsideViewController</code> interface.	187
9.13	<code>FlipsideViewController</code> class.	188
9.14	The finished <code>flipside</code> 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 <code>viewDidLoad</code> of class <code>RootViewController</code> .	199
10.7	Method <code>addContact</code> of class <code>RootViewController</code> .	200
10.8	Method <code>addViewControllerDidFinishAdding:</code> of class <code>RootViewController</code> .	200
10.9	Methods <code>tableView:numberOfRowsInSection:</code> and <code>tableView:cellForRowAtIndexPath:</code> of class <code>RootViewController</code> .	201
10.10	Method <code>tableView:didSelectRowAtIndexPath:</code> of class <code>RootViewController</code> .	202
10.11	Methods <code>tableView:commitEditingStyle:forRowAtIndexPath:</code> , <code>shouldAutorotateToInterfaceOrientation:</code> and <code>dealloc</code> of class <code>RootViewController</code> .	203
10.12	<code>NSDictionary</code> 's sorting category.	204
10.13	<code>AddViewController.xib</code> in Interface Builder after placing the default TableView .	205
10.14	<code>AddViewController</code> 's interface declaration.	205
10.15	Methods <code>initWithNibName:bundle:</code> and <code>viewDidLoad</code> of class <code>AddViewController</code> .	207
10.16	Methods <code>doneAdding:</code> and values of class <code>AddViewController</code> .	208
10.17	Methods <code>editableCellDidBeginEditing:</code> , <code>editableCellDidEndEditing:</code> and <code>editableCellDidEndOnExit:</code> of class <code>AddViewController</code> .	209
10.18	Methods <code>numberOfSectionsInTableView:</code> , <code>tableView:numberOfRowsInSection:</code> and <code>tableView:titleForHeaderInSection:</code> of class <code>AddViewController</code> .	210
10.19	Method <code>tableView:cellForRowAtIndexPath:</code> of class <code>AddViewController</code> .	211

xxii 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 Text Field.	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

12.12	Methods <code>slideshowCellDidSelectEditButton:</code> and <code>slideshowCellDidSelectPlayButton:</code> of class <code>RootViewController</code> .	250
12.13	Method <code>tableView:commitEditingStyle:forRowAtIndexPath:</code> of class <code>RootViewController</code> .	251
12.14	Methods <code>tableView:moveRowAtIndexPath:toIndexPath:</code> and <code>tableView:canMoveRowAtIndexPath:</code> of class <code>RootViewController</code> .	252
12.15	<code>UITableViewCell</code> for previewing a slideshow.	253
12.16	<code>SlideshowCell</code> method <code>initWithFrame:reuseIdentifier:</code> .	254
12.17	Methods <code>editSlideshow</code> and <code>playSlideShow</code> of class <code>SlideshowCell</code> .	255
12.18	Controller for a View that shows a slideshow.	256
12.19	Methods <code>loadView</code> and <code>nextImageView</code> of class <code>SlideshowViewController</code> .	257
12.20	Methods <code>exitShow</code> and <code>timerFired</code> of class <code>SlideshowViewController</code> .	259
12.21	Methods <code>transitionFinished:finished:context:</code> , <code>viewWillAppear</code> and <code>viewDidDisappear</code> of class <code>SlideshowViewController</code> .	261
12.22	Methods <code>shouldAutorotateToInterfaceOrientation:</code> and <code>willRotateToInterfaceOrientation:</code> of class <code>SlideshowViewController</code> .	263
12.23	Scaling category of <code>UIImageView</code> .	263
12.24	Controls a View for naming a slideshow.	265
12.25	Implementation of <code>NameViewController</code> .	265
12.26	Manages the pictures, sounds and effects of a slideshow.	266
12.27	Method <code>viewDidLoad</code> of class <code>SlideshowDataViewController</code> .	268
12.28	Method <code>viewDidAppear:</code> and <code>viewDidDisappear:</code> of class <code>SlideshowDataViewController</code> .	269
12.29	Methods <code>addPhoto</code> and <code>imagePickerController:didFinishPickingImage:</code> of class <code>SlideshowDataViewController</code> .	271
12.30	Methods <code>addMusic</code> and <code>mediaPicker:didPickMediaItems:</code> of class <code>SlideshowDataViewController</code> .	272
12.31	Methods <code>addEffect</code> and <code>startSlideshow</code> of class <code>SlideshowDataViewController</code> .	273
12.32	Method <code>tableView:cellForRowAtIndexPath:</code> of class <code>SlideshowDataViewController</code> .	274
12.33	Methods <code>tableView:commitEditingStyle:forRowAtIndexPath:</code> , <code>tableView:moveRowAtIndexPath:toIndexPath:</code> and <code>tableView:canMoveRowAtIndexPath:</code> of class <code>SlideshowDataViewController</code> .	275

13 Enhanced Slideshow App

13.1	iPhone photo library.	279
13.2	Viewing a video.	280
13.3	Setting the image transition effect.	280
13.4	Flip effect—rotates slide horizontally revealing the next slide underneath.	281
13.5	<code>MediaItem</code> class represents an image or a video.	283

xxiv 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 <code>init</code> and <code>initWithCoder:</code> of class Slideshow.	288
13.11	Methods <code>encodeWithCoder:</code> and <code>firstImage</code> of class Slideshow.	289
13.12	NSCoding category For UIImage.	290
13.13	Method <code>viewDidLoad</code> of class RootViewController.	291
13.14	Method <code>nameViewController:didGetName:</code> of class RootViewController.	292
13.15	Method <code>slideshowCellDidSelectEditButton:</code> of class RootViewController.	293
13.16	Method <code>tableView:cellForRowAtIndexPath:</code> of class RootViewController.	294
13.17	Method <code>initWithSlideshow:</code> of class SlideshowDataViewController.	295
13.18	Method <code>addPhoto</code> of class SlideshowDataViewController.	295
13.19	Method <code>imagePickerController:didFinishPickingMediaWithInfo:</code> of class SlideshowDataViewController.	296
13.20	Method <code>mediaItemCreator:didCreateMediaItem:</code> of class SlideshowDataViewController.	298
13.21	Methods <code>mediaPicker:didPickMediaItems:</code> and <code>addEffect</code> of class SlideshowDataViewController.	299
13.22	Methods <code>startSlideshow</code> and <code>actionSheet:clickedButtonAtIndex:</code> of class SlideshowDataViewController.	299
13.23	Method <code>tableView:cellForRowAtIndexPath:</code> of class SlideshowDataViewController.	300
13.24	Method <code>tableView:moveRowAtIndexPath:toIndexPath:</code> of class SlideshowDataViewController.	302
13.25	Method <code>applicationWillTerminate:</code> of class EnhancedSlideshowAppDelegate.	303
13.26	Controller for a view that shows a slideshow.	303
13.27	Method <code>nextImageViewWithMedia:</code> of class SlideshowViewController.	305
13.28	Method <code>changeSlide</code> of class SlideshowViewController.	306
13.29	Method <code>displayNewImage:</code> of class SlideshowViewController.	306
13.30	Method <code>displayNewVideo:</code> 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
14.9	Method <code>record:</code> of class VoiceRecorderViewController.	317

14.8	Methods <code>initWithNibName:bundle:</code> and <code>viewDidLoad</code> of class <code>VoiceRecorderViewController</code> .	317
14.10	Method <code>nameRecordingViewController:didGetName:</code> of class <code>VoiceRecorderViewController</code> .	320
14.11	Method <code>flip:</code> of class <code>VoiceRecorderViewController</code> .	321
14.12	Methods <code>playbackViewControllerDidFinish:</code> and <code>timerFired:</code> of class <code>VoiceRecorderViewController</code> .	322
14.13	Controls a View for naming a recording.	322
14.14	Finished layout of <code>NameRecordingViewController</code> 's view.	323
14.15	Implementation of <code>NameRecordingViewController</code> .	324
14.16	View that displays a visualization of a recording in progress.	326
14.17	Method <code>initWithCoder:</code> of class <code>Visualizer</code> .	326
14.18	Methods <code>setPower:</code> and <code>clear</code> of class <code>Visualizer</code> .	327
14.19	Method <code>drawRect:</code> of class <code>Visualizer</code> .	327
14.20	Controls the View where the user plays existing sound files.	329
14.21	Completed <code>PlaybackViewController</code> GUI.	330
14.22	Implementation for <code>PlaybackViewController</code> .	331
14.23	Methods <code>sliderMoved:</code> , <code>togglePlay</code> and <code>updateVolume:</code> of class <code>PlaybackViewController</code> .	332
14.24	<code>PlaybackViewController</code> methods <code>timerFired:</code> and <code>record:</code> .	333
14.25	Method <code>playSound</code> of class <code>PlaybackViewController</code> .	334
14.26	Methods <code>stopSound</code> , <code>numberOfSectionsInTableView:</code> and <code>tableView:numberOfRowsInSection:</code> of class <code>PlaybackViewController</code> .	335
14.27	Method <code>tableView:cellForRowAtIndexPath:</code> of class <code>PlaybackViewController</code> .	336
14.28	Method <code>tableView:commitEditingStyle:forRowAtIndexPath:</code> of class <code>PlaybackViewController</code> .	337
14.29	Method <code>tableView:didSelectRowAtIndexPath:</code> of class <code>PlaybackViewController</code> .	338
14.30	Methods <code>tableView:accessoryButtonTappedForRowWithIndexPath:</code> and <code>mailComposeController:didFinishWithResult:error:</code> of class <code>PlaybackViewController</code> .	339

15 Enhanced Address Book App

15.1	Viewing a contact.	343
15.2	Requesting a connection.	344
15.3	Getting a Bluetooth Connection Request .	344
15.4	Data model editor.	346
15.5	Controller that displays the contact information for a created contact.	347
15.6	Methods <code>viewDidLoad</code> and <code>send</code> of class <code>ContactViewController</code> .	347
15.7	Method <code>peerPickerController:didConnectPeer:toSession:</code> of class <code>ContactViewController</code> .	348
15.8	Methods <code>peerPickerControllerDidCancel:</code> , <code>updateTitle</code> and <code>tableView:numberOfRowsInSection:</code> of class <code>ContactViewController</code> .	349
15.9	Method <code>tableView:cellForRowAtIndexPath:</code> of class <code>ContactViewController</code> .	350



xxvi Illustrations

15.10	Controls the main view of the Enhanced Address Book app.	351
15.11	Method <code>viewDidLoad</code> of class <code>RootViewController</code> .	352
15.12	Method <code>session:didReceiveConnectionRequestFromPeer:</code> of class <code>RootViewController</code> .	353
15.13	Methods <code>alertView:clickedButtonAtIndex:</code> and <code>alertViewCancel:</code> of class <code>RootViewController</code> .	354
15.14	Method <code>receiveData:fromPeer:inSession:context:</code> of class <code>RootViewController</code> .	355
15.15	Methods <code>insertNewObject</code> and <code>addViewControllerDidFinishAdding:</code> of class <code>RootViewController</code> .	356
15.16	Methods <code>numberOfSectionsInTableView:</code> and <code>tableView:numberOfRowsInSection:</code> of class <code>RootViewController</code> .	358
15.17	Methods <code>tableView:cellForRowAtIndexPath:</code> and <code>tableView:didSelectRowAtIndexPath:</code> of class <code>RootViewController</code> .	359
15.18	Methods <code>tableView:commitEditingStyle:forRowAtIndexPath:</code> and <code>tableView:canMoveRowAtIndexPath:</code> of class <code>RootViewController</code> .	360
15.19	Method <code>fetchedResultsController</code> of class <code>RootViewController</code> .	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 <code>Airfare</code> .	367
16.4	Controller for the root View of the Twitter Discount Airfares app.	367
16.5	<code>RootViewController</code> class implementation.	368
16.6	<code>UITableView</code> delegate and data source methods of class <code>RootViewController</code> .	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 <code>TwitterConnection</code> .	374
16.11	Class that gets tweets and parses them for information.	376
16.12	<code>AirfareFinder</code> class implementation.	377
16.13	XML containing information about a single tweet.	378
16.14	<code>NSXMLParser</code> delegate methods.	379
16.15	Method <code>parseCost</code> of category parsing of class <code>NSString</code> .	381
16.16	Method <code>parseLocation</code> of category parsing of class <code>NSString</code> .	382
16.17	Method <code>removeLink</code> of category parsing of class <code>NSString</code> .	383
16.18	<code>UITableViewCell</code> that displays information about an airfare.	383
16.19	<code>AirfareCell</code> class implementation.	384

