

Where to Next?

Well, wow! You're still with us, huh? Great! It sure has been a long journey since that very first iOS application we built together. You've certainly come a long way. We would love to tell you that you now know it all. But when it comes to technology, and especially when it comes to programming, you never know it all.

At its core, programming is about problem solving and figuring things out. It's fun, and it's rewarding. But, at times, you will run up against a puzzle that just seems insurmountable—a problem that appears to have no solution. Sometimes the answer will come to you if you just take a bit of time away from the problem. A good night's sleep or a few hours of doing something different can often be all that is needed to get you through it. Believe us—you can stare at the same problem for hours, overanalyzing and getting yourself so worked up that you miss an obvious solution. But sometimes even a change of scenery doesn't help. In those situations, it's good to have friends in high places. This appendix outlines some resources you can turn to when you're in a bind.

Apple's Documentation

Become one with Xcode's documentation browser, grasshopper. The documentation browser is a front end to a wealth of incredibly valuable sample source code, concept guides, API references, video tutorials, and a whole lot more.

There are few areas of iOS that you won't be able to learn more about by making your way through Apple's documentation. And if you get comfortable with Apple's documentation, navigating through uncharted territories and new technologies as Apple rolls them out will be easier.

Note Xcode's documentation browser takes you to the same information you can find by going to Apple's Developer web site at <http://developer.apple.com>.

Mailing Lists

You might want to sign up for these handy mailing lists:

Cocoa-dev: This moderately high-volume list, run by Apple, is primarily about Cocoa for Mac OS X. Because of the common heritage shared by Cocoa and Cocoa Touch, however, many of the people on this list may be able to help you. (Do make sure to search the list archives before asking your question.)

<http://lists.apple.com/mailman/listinfo/cocoa-dev>

Xcode-users: Another list maintained by Apple, this one is specific to questions and problems related to Xcode.

<http://lists.apple.com/mailman/listinfo/xcode-users>

Quartz-dev: This is an Apple-maintained mailing list for discussion of the Quartz 2D and Core Graphics technologies.

<http://lists.apple.com/mailman/listinfo/quartz-dev>

Cocoa-unbound: This list, intended for discussion of both Mac and iOS development, appeared in 2010 in response to the sometimes heavy-handed moderation of some of the Apple-run lists, particularly Cocoa-dev. The posting volume is lower here, and topics can run a bit further afield.

<http://groups.google.com/group/cocoa-unbound>

iPhone SDK Development: Another third-party list, this one is focused entirely on iOS development. You'll find a medium-sized community here, with a nice cast of regulars.

<http://groups.google.com/group/iphonesdkdevelopment>

Discussion Forums

These discussion forums allow you to post your questions to a wide range of forum readers:

Learn Cocoa Forum: As the official forum for this book, this features an active, vibrant community, full of people, such as you, with the wisdom and sensibility to buy our book. At least, that's what we hope! This forum was hosted somewhere else in the past, and not everyone has found their way to the new location yet.

<http://forum.learncocoa.org>

Apple Developer Forums: This is a web forum set up by Apple specifically for discussing iOS and Mac software development. Many iOS programmers, both new and experienced (including Apple engineers and evangelists), contribute to these

forums. It's also the only place where you can legally discuss issues with prerelease versions of the SDK that are under nondisclosure agreements. You'll need to sign in with your Apple ID to access this forum.

<http://devforums.apple.com>

Apple Discussions, Developer Forums: This link connects you to Apple's community forums for Mac and iOS software developers:

<http://discussions.apple.com/category.jspa?categoryID=164>

Apple Discussions, iPhone: This link connects to Apple's community forums for discussing the iPhone:

<http://discussions.apple.com/category.jspa?categoryID=201>

Web Sites

Visit these web sites for helpful coding advice:

CocoaHeads: This is the site of a group dedicated to peer support and promotion of Cocoa. It focuses on local groups with regular meetings where Cocoa developers can get together, help each other out, and even socialize a bit. There's nothing better than knowing a real person who can assist you, so if there's a CocoaHeads group in your area, check it out. If there's not, why not start one?

<http://cocoaheads.org>

NSCoder Night: NSCoder Nights are weekly, organized meetings where Cocoa programmers get together to code and socialize. Like CocoaHeads meetings, NSCoder Nights are independently organized local events.

<http://nscodernight.com>

Stack Overflow: This is a community Q&A site targeted at programmers. Many experienced iOS programmers hang out here and answer questions.

<http://stackoverflow.com>

Apart from accessing the main site, you may also want to use tags to browse topics relevant to iOS developers:

<http://stackoverflow.com/questions/tagged/objective-c>

<http://stackoverflow.com/questions/tagged/iphone>

<http://stackoverflow.com/questions/tagged/xcode>

<http://stackoverflow.com/questions/tagged/ios>

iDeveloper TV: This is a great resource for in-depth video training in iOS and Mac development, for a price. It also contains some nice, free video content, mostly from NSConference (listed in the “Conferences” section of this chapter), which is run by the same people behind iDeveloper TV.

<http://ideveloper.tv>

Cocoa Controls: Here, you’ll find a huge range of GUI components for both iOS and Mac OS X. Most of them are free and open source. These controls can be useful as is or as examples for further learning.

<http://cocoacontrols.com>

iphonedevbook.com: In addition to hosting the source code for this book’s example projects, this site aims to be a larger resource for iOS development knowledge. At the time of this writing, the newest incarnation of the site hasn’t launched yet, but we expect great things from it in the future.

<http://iphonedevbook.com>

NSHipster: This site’s slogan says it all: It’s “a journal of the overlooked bits in Objective-C and Cocoa.” You can learn about a whole slew of underused pieces of our favorite frameworks here.

<http://nshipster.com>

Blogs

If you still haven’t found a solution to your coding dilemma, you might want to read these blogs:

Matt Gemmell’s blog: Matt has a lot of experience and a lot of strong opinions about development in Objective-C. We think these opinions are usually right, and always worth paying attention to.

<http://www.mattgimmell.com>

Cocoa is my Girlfriend: This is a group blog, spearheaded by Marcus Zarra, focusing on iOS and Cocoa development issues. These guys cover a wide range of topics useful to everyone working in this corner of the field.

<http://cimgf.com>

Ray Wenderlich’s blog: Ray’s site has expanded so much it can hardly be described as a blog at all anymore, but we’re putting it here anyway! This is a fantastic resource for iOS developers, full of tutorials on a wide range of topics. Much of this content is directed toward game developers, but there’s really something for everyone here.

<http://www.raywenderlich.com>

Wil Shipley's blog: Wil is one of the most experienced Objective-C programmers on the planet. His *Pimp My Code* series of blog postings should be required reading for any Objective-C programmer.

<http://www.wilshipley.com/blog>

Wolf Rentzsch's blog: Wolf is an experienced, independent Cocoa programmer and the founder of the (now defunct) C4 independent developers' conference.

<http://rentzsch.tumblr.com>

iDevBlogADay: This is a multiauthor blog, whose authorship rotates daily among several indie developers of iOS and Mac software. Follow this blog, and you'll be exposed to new insights from different developers every day.

<http://idevblogaday.com>

CocoaCast: This has a blog and podcast about various Cocoa programming topics, available in both English and French.

<http://cocoacast.com/>

@ObjectiveC on Twitter: The @objectiveC Twitter user posts about new Cocoa-related blog posts. It's worth a follow.

<http://twitter.com/objectivec>

Mike Ash's blog: Mike is "just this guy, you know?" This RSS feed presents Mike's collection of his ongoing iOS Friday Q&A.

<http://www.mikeash.com/pyblog/>

Conferences

Sometimes, books and web sites aren't enough. Attending an iOS-focused conference can be a great way to get new insights and meet other developers. Fortunately, this is an area that has really boomed over the past few years, and iOS developers have no shortage of interesting conferences to attend. Here are a few:

WWDC: Apple's World Wide Developer Conference is the annual event where Apple typically unleashes the next great new things for its developer community.

<http://developer.apple.com/wwdc>

MacTech: This is a conference for Mac and iOS programmers and IT professionals. It's hosted by the same people who publish *MacTech Magazine*.

<http://www.mactech.com/conference>

NSConference: This multiple-continent event has been held in both the United Kingdom and United States, so far. It's run and promoted by Steve "Scotty" Scott, perhaps the hardest working man in the Mac/iOS conference scene.

<http://nsconference.com>

360 iDev: This approximately once-a-year conference, which is hosted in either San Jose or Denver (flipping between the two year after year), began in 2009.

<http://www.360idev.com>

Çingleton: So far, there have been two instances of the Çingleton Symposium, so it's not a singleton any more. But it is a great conference that attendees rave about.

<http://www.cingleton.com>

CocoaConf: No one else seems to be operating as many iOS developer events in so many different cities as CocoaConf. These folks have events planned all over the U.S. through 2013.

<http://www.cocoaconf.com>

Follow the Authors

All of this book's authors are active Twitter users. You can follow Dave, Jack, Jeff, and Fredrik via @davemark, @jacknutting, @jeff_lamarche, and @peylow, respectively. Some of them have blogs, too:

- Jeff's iOS development blog contains a lot of great technical material. Be sure to check out the comprehensive series on OpenGL ES.

<http://iphonedevdevelopment.blogspot.com>

- Jack uses his blog, [nuthole.com](http://www.nuthole.com), to talk about what's going on in his career and his life (technically and otherwise). It's a blog like many others, but this one is Jack's.

<http://www.nuthole.com>

Tip Are you serious about diving more deeply into the iOS SDK, and especially interested in all the great new functionality introduced with the iOS 6 SDK (of which we only scratched the surface in this book)? If so, you should check out *More iOS 6 Development: Further Explorations of the iOS SDK* by Dave Mark, Alex Horovitz, Kevin Kim, and Jeff LaMarche (Apress 2012).

And if all else fails, drop us an e-mail at begin6errata@learncocoa.org. This is the perfect place to send messages about typos in the book or bugs in *our* code. We can't promise to respond to every e-mail message, but we will read all of them. Be sure to read the errata on the Apress site and the forums on <http://forum.learncocoa.org> before clicking *Send*. And please do write and tell us about the cool applications you develop.

Farewell

The programming language and frameworks we've worked with in this book are the end result of more than 20 years of evolution. And Apple engineers are feverishly working round the clock, thinking of that next cool new thing. The iOS platform has just begun to blossom. There is so much more to come.

By making it through this book, you've built yourself a sturdy foundation. You have a solid knowledge of Objective-C, Cocoa Touch, and the tools that bring these technologies together to create incredible new iPhone, iPod touch, and iPad applications. You understand the iOS software architecture—the design patterns that make Cocoa Touch sing. In short, you're ready to chart your own course. We are so proud!

We sure are glad you came along on this journey with us. We wish you the best of luck and hope that you enjoy programming iOS as much as we do.

Index

A

- Application settings, 397
 - bundle
 - application icon addition, 412–413
 - child setting view addition, 420–422
 - contextual menu, 418
 - finder window, 419
 - iOS device, 397–398
 - minimum and maximum value image, 419
 - multivalue field addition, 414–416
 - preferences, 398
 - property list, 398, 406, 408–409
 - Root.plist, 405
 - secure text field setting, 414
 - slider setting, 418
 - text field addition, 410–412
 - toggle switch setting, 417
 - Xcode, 405
 - child setting view, 401
 - default value
 - BIDFlipsideViewController, 429
 - extra code, 429
 - flipside view controller scene, 429
 - flipside view designing, 427
 - refreshFields method, 430
 - registration, 430–431
 - direct preference setting, 403
 - iOS 4, 431
 - main view, 402
 - notification, 432
 - primary setting view, 399
 - project creation
 - flipside view, 404
 - location, 404
 - utility application template, 405
 - reader settings
 - main view controller updation, 426–427
 - main view creation, 423–425
 - retrieving user setting, 422–423
 - simulator, 399
 - single preference item selection, 400
 - user defaults system, 432–434
 - application instance, 433
 - notification system, 434
 - synchronize method, 432–434
 - automatic reference counting, 61
- Automatic Reference Counting (ARC), 7, 15
- Autorotation
 - and autosizing, 115
 - designing an interface,
 - autosize, 121
 - button stretch, 127
 - button stuck, 123
 - dashed blue lines, 123
 - full width buttons, 126
 - labeled buttons, 122
 - leading space, 123
 - new constraints, 125
 - overriding constraints, 124
 - place, left button, 124
 - position, rotate, 125
 - single view application, 121
 - size inspector, 123
 - UR button, 127
 - width by constraint, 127
 - mechanics
 - added approaches, 117
 - approaches, 117
 - constraints, 117
 - interface builder, 117
 - pixels, 116
 - points, 116
 - status bar, 116
 - the retina display, 116
- mobile Safari, 115
- restructuring

Autorotation (*cont.*)

- autoconstraints, 129
- CGRect, 130
- CGRectMake, 130
- create and connect outlets, 129
- didLoad method, 131
- moving buttons, 130
- portrait layout, view, 128
- swapping views
 - bar buttons, 137
 - buttonTapped, 137
 - foo button, 138
 - implementing, 135
 - labeling views, 135
 - landscapeviews, 134
 - outlet changes, 137
 - outlet collection, 132
 - rotation transformation, 136
 - storage pop-up, 134
 - swap application, 133
 - transformation, 136
 - two views, 134
- view orientation
 - code sense, 120
 - C-style mask, 120
 - device orientation, 118
 - per controller rotation, 119
 - sense of gravity, 119
 - supported orientation at app level, 118
 - UIApplication, 120
 - UIInterf, 121
 - upside down, 118
- removing resources, 537
- saving the state, 539
- time request, 541
- UIApplication instance, 542
- UIBackgroundTaskInvalid, 543
- viewDidLoad, 537
- execution states
 - active to inactive, 531
 - applicationWillTerminate, 530
 - background to inactive, 532
 - inactive to active, 532
 - inactive to background, 532
 - message app, 530
 - thread 1 SIGKILL, 530
- inactive state
 - implementation block, 535
 - implicit animation, 534
 - label rotating magic, 534
 - methods, 533
 - override, initializer, 535
 - rotateLabelDown, 536
 - transform property, 534
 - view controller, 536
 - viewDidLoad, 533
- life cycle, application, 526
 - active, 526
 - background, 526
 - inactive, 526
 - not running, 526
 - suspend, 526
- state change notification, 527
 - applicationDidBecomeActive, 527
 - applicationDidEnterBackground, 527
 - applicationWillEnterForeground, 527
 - applicationWillResignActive, 527
 - delegate methods, 527
 - didFinishLaunchingWithOptions, 527
 - state lab creation, 528
 - VoIP, 525

B

- Background processing, 525
 - background state
 - applicationDidEnterBackground method, 540, 542
 - applicationWillEnterForeground, 541
 - beginBackgroundTaskWithExpirationHandler, 543
 - debugging terminate, 544
 - endBackgroundTask, 543
 - initWithNibName, 539
 - instance variable, 540
 - label-rotating magic, 538
 - Background processing
 - background state, 537, 539, 541
 - execution states, 531–532
 - Buttons, 100
 - control states, 112
 - edge insets, 112
 - stretchable images, 112
 - viewDidLoad Method, 111

C

- Camera application, 671
 - BIDViewController implementation, 673
 - delegate methods, 676
 - interface design, 672
 - utility methods, 674
 - viewDidLoad and viewDidAppear
 - methods, 674
- Cell ID location, 619
- CheckPlease application
 - BIDCheckMarkRecognizer, 614
 - check-mark gesture, 612
 - CheckPlease project, 613
 - declarations, 614
 - instance variables, 614
- Cocoa Touch, 7
- Collection view
 - configuring, 682–683
 - controller class, 679
 - custom cells, 679–681
 - DialogViewer project, 677
 - header view, 687
 - keys, 682
 - snazzy look, 683
 - viewDidLoad, 683
 - layout flow, 685
 - providing, content cells, 683
 - BIDContentCell
 - BIDHeaderCell, 681
 - defaultFont method, 680
 - label property, 679
 - text property, 679
 - UILabel, 681
 - DialogViewer project, UICollection
 - View, 677–678
 - header view
 - cell class, 688
 - DialogViewer app, 689
 - UICollectionViewLayout, 688
 - layout flow
 - cramped, 687
 - flow and wrap, 686
 - paragraph flow, 686
 - UICollectionView, 685
 - providing, content cells
 - dequencing method, 684
 - format, 685
 - wordsInSection, 684
- Control fun application
 - image view
 - alpha value, 81
 - background property, 81
 - drawing checkboxes, 81
 - interaction checkboxes, 81
 - mode menu, 80
 - object library, 75
 - resize, 78
 - stretching, 82
 - tags, 80
 - outlets, creating and connecting, 89
 - text fields
 - attributes settings, 89
 - blue guidelines, 83
 - inspector settings, 88
- Core data framework, 463
 - entities and managed objects
 - application’s root controller, 478–479
 - attributes, 465
 - backing store, 466
 - data model designing, 468, 470–471, 473
 - data model editor, 464
 - fetches property, 465
 - key value coding, 465
 - managed object context, 466
 - new managed objects, 467–468
 - NSManagedObjectContext, 467
 - persistence view and controller
 - creation, 473–474, 476, 478
 - relationships, 465
 - new project creation, 463
 - persistence and supporting file
 - folders, 464
 - project templates, 463
- Core Location, 619
 - Cell ID location, 619
 - GPS, 619
 - location manager
 - CLLocation object, 622
 - distance filters, 621
 - error notifications, 624
 - level of accuracy, 620
 - location updates, 622
 - updates, 629
 - WhereAml application, 626
 - WPS option, 619

D, E, F

- Data persistence, 435
 - application's sandbox, 436–437
 - Documents directory, 437
 - globally unique identifiers, 436
 - iPhone simulator, 436
 - Library option, 436
 - NSUserDefaults, 436
 - temporary directory, 437
 - archiving model objects, 452
 - core data, 463
 - archiving model objects
 - BIDFourLines Class, 449–450
 - BIDViewController Class, 450–451
 - generic type, 446
 - NSCoding protocol, 446–447
 - NSCopying implementation, 447
 - property implementation, 446
 - and unarchiving data objects, 448–449
 - file-saving strategies, 438
 - multiple-file persistence, 438
 - single-file persistence, 438
 - SQLite3 database file, 438
 - nonvolatile storage, 435
 - property lists, 439
 - SQLite3 database, 452–457
 - advantages, 452
 - BIDViewController.h file, 456
 - BIDViewController.xib, 456
 - bind variables, 455
 - creation, 453–454
 - dynamic library, 456–457
 - object-relational mapping, 453
 - persistence view controller
 - modification, 457–458, 460
 - Structured Query Language, 453
- Delegate and data source picker, 175

G, H

- Gestures, 585
 - custom gesture
 - CheckPlease application, 612
 - CheckPlease touch methods, 615
 - definition, 585
 - events, 586
 - gesture recognizer, 586
 - handling process, 588

- pinch and rotation detection, 607
- responder chain
 - app delegate, 587
 - swipe gesture, 588
 - working principle, 586
- swipes application, 594
- taps, 586, 601
- touch, 586
- TouchExplorerApplication, 590
- touch notification methods, 589
- Globally unique identifiers (GUIDs), 436
- GPS, 619
- Grand central dispatch
 - access, blocks, 512
 - define, CPU speed, 511
 - low level queueing
 - activity indicator, 522
 - BIDViewController, 522
 - block access, 520
 - blockhead, 518
 - blocks, 518
 - closures, 518
 - concurrent blocks, 523
 - dispatch_async, 519
 - dispatch_get_global_queue()
 - function, 520
 - dispatch group, 523
 - doWork, 523
 - feedback, 521
 - improving slow worker, 519
 - lambdas, 518
 - main thread, 520
 - priority, 520
 - spinner, 522
 - startTime variable, 520
 - UIActivityIndicatorView, 521
 - wrapping, 518
 - over and out, 544
 - slow worker
 - BIDViewController, 513
 - device family, 513
 - GUI, 513
 - interface hide, 512
 - round rect button, 514
 - sleepForTimeInterval, 514
 - tapping application, 515
 - text view, 515

- threading basics
 - code safe, 516
 - main thread, 516
 - mutex, 516
- units of work, 517
- Gyro and accelerometer, 637
 - accelerometer results, 647
 - axes, graphic representation, 638
 - core motion and motion manager
 - alloc and init methods, 639
 - audio toolbox framework, 640
 - BIDViewController, 640
 - blocks, 639
 - event-based motion, 639
 - MotionMonitor, 640
 - supportedInterfaceOrientations method, 642
 - detecting shakes, 648, 650, 654
 - backed-in shaking, 650
 - shake and break, 650
 - viewDidLoad method, 654
 - directional controller, 657, 660, 663
 - ball movement calculation, 663
 - BIDBallView, 660
 - rolling marbles, 657
 - g-force measurement, 637
 - proactive motion access, 645
 - rotation value, 638

I, J

- iCloud, 481
 - adding support
 - bundle identifier, 505
 - bundle seed, 505
 - code signing identity, 505
 - entitlements, enable, 506
 - NSMetadataQuery, 507
 - ongoing query, 506
 - provisioning profile, 504
 - query, 506
 - reloadFiles method, 507
 - save file, 508
 - storing preferences, 509
 - urlForFilename, 508
 - viewWillAppear method, 509
 - computerless backup, 481
 - document based application, 510
 - document storage with UIDocument
 - alert view call, 490
 - BIDDetailViewController, 502
 - BIDMasterViewController, 486
 - BIDTinyPixDocument, 483
 - BIDTinyPixView, 496
 - _blockSize, 498
 - buttonIndex, 491
 - cellForRowAtIndexPath, 489
 - chooseColor method, 503
 - code master, 486
 - color segmented control, 487
 - completion Handler method, 491
 - configureView method, 503
 - constraints, 501
 - contentsForType, 486
 - control drag method, 495
 - C struct, 497
 - detailing storyboard, 500
 - diagonal pattern, 484
 - document class, 487
 - document directory, 487
 - _gapSize, 498
 - identifier to FileCell, 494
 - initWithFileURL method, 485
 - loadFromContents, 486
 - master detail application, 487
 - master view controller, 494
 - MVC principle, 486
 - navigation controller, 493
 - NSUserDefaults, 489
 - openWithCompletionHandler, 492
 - prepareWithInvocationTarget, 499
 - selectedIndex property, 503
 - setNeedsDisplay, 503
 - size inspector, 501
 - storyboarding, 493
 - table view data source methods, 489
 - text editor, 501
 - TinyPix document, 482, 495
 - toggle cell, 485
 - toggleSelectedBlock method, 499
 - touch events, 498
 - tr, 492
 - UIBarButtonItem, 490
 - UIViewdrawRect method, 498
 - UIView subclass, 497
 - undoManager, 499

- iCloud (*cont.*)
 - views, 500
 - viewWillAppear, 489
 - viewWillDisappear, 504
 - Image picker, 667
 - camera application
 - BIDViewController implementation, 673
 - delegate methods, 676
 - interface design, 672
 - pickMediaFromSource method, 674
 - shrinkImage method, 675
 - updateDisplay method, 674
 - utility methods, 674
 - viewDidLoad and viewWillAppear methods, 674
 - delegate method, 669
 - isSourceTypeAvailable, 669
 - photo library, 671
 - scaling or cropping, 668
 - UIImagePickerController, 668
 - Interaction handling, 43
 - application delegate
 - applicationWillTerminate, 65
 - BIDAppDelegate, 66
 - cocoa touch, 64
 - declared link, 66
 - didFinishLaunchingWithOptions, 68
 - option clicked, 66
 - option key, Xcode, 65
 - referenced link, 66
 - UIApplication, 64
 - view controller
 - action methods, 48, 62
 - actions, 50
 - alignment button, 58
 - arguments field, 56
 - attribute inspector, 59
 - BIDViewController, 47
 - button fun group, 46
 - buttonPressed, 57
 - buttons and actions, 52
 - connection type, 55
 - control dragging, 54, 60
 - drag to action, 57
 - edit in Xcode, 51
 - editor toggle button, 53
 - empty class extension, 51
 - event field, 56
 - floating pop-up, 55
 - HIG, 53
 - IBOutlet, 48
 - interface builder, 49
 - label and outlet, 58
 - labeled storage, 60
 - low level virtual machine (LLVM), 49
 - malloc(), 61
 - myButton, 48
 - navigator, class files, 47
 - NeXTSTEP, 49
 - nib file, 50, 56
 - object library, 52, 58
 - outlets and action, 47, 48
 - owner icon, 54
 - sender argument, 50
 - statusLabel, 60
 - string creation, 62
 - stub method, 56
 - type field, 56
 - user interface, 51
 - model view controller
 - cocoa touch applications, 45
 - NSObject subclass, 45
 - patterns, 44
 - reusability, 45
 - NSAttributedString, 43
 - project creation
 - ARC, 45
 - git repository, 46
 - naming and selection, 46
 - two button application, 44
 - view controller
 - attributeText property, 63
 - NSMutableAttributedString, 63
 - practicing, 62
 - styling, 63
 - view controller, clean up, 50
- iOS applications development, 1
- desktop application development
 - garbage collection, not support, 7
 - limited access, 5
 - limited response time, 6
 - limited system resources, 6
 - one active application, 5
 - one window, 5
 - screen size limitation, 6

- developer options
 - general-purpose computer, 8
 - programming knowledge, 4
 - restrictions, 3
 - simulator, 3
- functionality, 7
- Objective-C, 4
- requirements, 1
- SDK versions and source code, 3
- iPad considerations
 - split view
 - App delegate, 374–376
 - awakeFromNib method, 377
 - detail view controller, 378–379
 - in landscape mode, 368
 - Master View Controller, 376–377
 - in portrait mode, 369
 - project creation, 370
 - storyboard, 373
 - viewDidLoad method, 377
 - Xcode, 374
 - popovers
 - instance methods, 393
 - retain cycle, 390
 - setDetailItem method, 393
 - setLanguageString, 393
 - size, definition, 391
 - viewDidLoad method, 393
 - presidents
 - BIDMasterViewController, 382
 - configureView method, 386
 - tableView, 384
 - viewDidLoad method, 382
 - split views
 - class extension, 379
 - detail view controller, 379–381
 - setDetailItem method, 380

K

- Keyboard, closing, 91
 - action method, 99
 - actions and outlets, 99
 - adding slider and label, 96
 - backgroundTap action, 94
 - compile and run app, 96
 - constraints, 98
 - done button, taps, 92
 - identity inspector, 95

- using interface builder, 94
- view property, 94

L

- LLVM (Low Level Virtual Machine) compiler, 25
- Location manager, 620
 - CLLocation object, 622
 - distance filters, 621
 - error notifications, 624
 - level of accuracy, 620
 - location updates, 622
 - updates, 629
 - WhereAml application, 626

M

- Map Kit, 619
 - didUpdateLocations method, 633
 - MKCoordinateRegion, 633
 - property declaration, 631
 - viewDidLoad method, 633
- Multiple-file persistence, 438
- Multiview applications, 139
 - content views, 139
 - Music application, 143
 - navigation bar, 141
 - split view, 142
 - stocks application, 139
 - tab bar, 140
 - toolbar, 143
 - utility application, 139
- View Switcher
 - add controller, 154
 - alerts, 147
 - animate, transition, 166
 - animation block, 167
 - app delegate, 152
 - bar button, 158
 - BIDBlueViewController, 162–163, 165
 - BIDSwitchViewController, 151, 154
 - BIDYellowController, 165
 - blueButtonPressed method, 164
 - blue view flips, 146
 - button and toolbar, 145
 - content view implement, 163
 - content views, 148
 - controller and nib files, 150
 - controllers, 144

Multiview applications (*cont.*)

- curve, animation, [167](#)
- didReceiveMemoryWarning
 - method, [160](#), [162](#)
- empty application project, [148](#)
- features, toolbar, [158](#)
- flip style, [167](#)
- IDAppDelegate, [153](#)
- identity inspector, [155](#)
- lazy loading, [161](#)
- metrics section, [164](#)
- navigation controller, [148](#)
- navigator, [149](#)
- nib files, [147](#), [157](#)
- project template, [149](#)
- received actions, [155–156](#)
- root controller, [147](#)
- root view controller, [159](#)
- subclass, view controller, [150](#)
- SwapAppDelegate, [153](#)
- switch off, [168](#)
- switchView method, [159](#)
- tab bar controller, [148](#)
- toolbar, build, [156](#)
- transition methods, [168](#)
- UI section, [152](#)
- viewDidLoad method, [160](#)
- yellowViewController, [161](#)

NNavigation controller, [269](#)

fundamentals

- computer stack, [270](#)
- navigation, [271](#)
- navigation controller, [271](#)
- root view controller, [270](#)
- stack, [270](#)
- subcontrollers, [270](#)

nav

- abstract class, [281](#)
- accessory icons, [272](#)
- accessory view, [275](#), [305](#), [309](#)
- action method, [329](#)
- add, edit view, [336](#)
- add images, [286](#)
- application skeleton, [278](#), [286](#)
- BIDAppDelegate, [280](#)

- BIDCheckListViewController, [297](#)
- BIDDeleteMeViewController, [315](#)
- BIDFirstLevelViewController, [279](#)
- BIDPresident, [322](#)
- BIDPresidentDetailViewController, [324](#)
- BIDSecondLevelViewController, [282](#)
- bundle method, [299](#), [317](#)
- buttonType property, [306](#)
- button view, disclosure, [273](#)
- button views, [294](#)
- checklist, [296](#)
- checklist controller, [302](#)
- checklist view, [274](#)
- code deletion, [338](#)
- contents view, [330](#)
- controller instance, [292](#)
- controls on table rows, [302](#)
- create, movable rows, [310](#)
- data model object, [322](#)
- data source method, [330](#)
- decode, [335](#)
- deletable rows, [276](#), [315](#)
- delete me instance, [318](#)
- detail disclosure, [274](#), [295](#)
- detail tapping, [273](#)
- detail view, [287](#), [296](#)
- detail view controller, [323](#)
- didFinishLaunchingWithOptions, [280](#)
- didSelectRowAtIndexPath, [299](#)
- disclosure button view, [286](#)
- disclosure indicator, [272](#)
- editable detail pane, [321](#)
- editable detail view, [277](#)
- EditingStyleForRowAtIndexPath, [312](#)
- editing styles, [317](#)
- edit mode, [277](#)
- edit mode, tapping, [320](#)
- features, [337](#)
- fieldLabels, [325](#)
- final subcontroller, [337](#)
- grouped table, [278](#)
- grouped table view, [321](#)
- implement detail view list, [333](#)
- initWithNibName, [291](#)
- initWithStyle method, [281](#), [313](#), [329](#)
- instance, checklist controller, [300](#)
- instances, [282](#)
- label property, [289](#)

- launch, delete me, 319
 - modify disclosure button, 289
 - movable rows, 275, 310
 - move me controller, 313
 - mutable strings, 288
 - navigationController property, 285
 - navigation controller set up, 280
 - NSKeyedUnarchiver method, 335
 - NSMutableString, 288
 - numberOfRowsInSection
 - method, 284
 - protocols, 325
 - return key type, 330
 - root level controller, 308
 - row controller instance, 307
 - row controls view, 303
 - row selection, 274
 - save method, 329
 - second level controllers, 293, 301
 - selected Snack method, 299
 - setEditing method, 310
 - subcontrollers, 272
 - on table, move me row, 314
 - tag property, 305
 - textFieldDone, 338
 - top-level view controller, 279
 - touch and drag, 276
 - UIImage property, 281
 - UITableViewRowAnimation
 - Automatic, 318
 - view, checklist, 297
 - view, row control, 275
 - views, deletable rows, 315
 - viewWillAppear, 287
 - wrap rows, 339
 - NSString method, 437, 453
 - NSZone parameter, 448
- O**
- Object-relational mapping (ORM), 453
- P**
- Photo library, 671
 - Pickers, 169
 - in clock application, 170
 - date picker, 171, 188
 - delegates and data sources, 175
 - dependent components
 - bundle, 206
 - compile and run app, 208
 - content view, 203
 - controller class, 203
 - delegate and data source
 - methods, 205
 - delegate methods, 207
 - statedictionary.plist file, 203
 - states array, 207
 - viewDidLoad method, 205
 - game creation
 - audio toolbox framework, 218
 - controller header file, writing the, 209
 - controller implementation, 211
 - fonts, 211
 - image files, 211
 - playerWon method, 217
 - sound files, 215
 - spin method, 213
 - view building, 210
 - viewDidLoad method, 212, 214
 - multicomponent picker
 - buttonPressed method, 200
 - controller implementation, 199
 - delegate method, 201
 - implementation of, 198
 - outlets and actions, 198
 - view building, 198
 - root view controller, 176
 - single-component picker
 - BIDSingleComponentPickerView
 - Controller.m, 194
 - data source methods, 196
 - delegate methods, 197
 - with dependent components, 173
 - with images, 174
 - implementation of, 191
 - outlets and actions, 192
 - #pragma, 197
 - view building, 192
 - Pinch and rotation detection
 - action methods, 610
 - image transformation, 610
 - instance variables, 609
 - PinchMe application, 611
 - PinchMe project, 607
 - Swipes project, 595

Pinch and rotation detection (*cont.*)
 two-finger rotation, 607
 UIImageView, 609
 UIPinchGestureRecognizer, 607

#PRAGMA, 197

Property lists

persistence application
 creation, 441
 designing, 442
 editing, 443, 445
 first version, 440
 serialization, 439–440

Q

Quartz and OpenGL

2D approach

CGContextStrokePath(), 547
 CGFloat, 549
 color theory, 549
 convenience method, color, 551
 coordinate system, 548
 drawing a line, 548
 drawing shapes, 552
 fill color, 547
 gamut, 550
 graphic libraries, 549
 graphics contexts, 546
 images in context, draw, 551
 models, color, 551
 parameters, 548
 primary colors, 550
 QuartzDemo, 553
 RGB color model, 550
 RYB color model, 550
 specifying colors, 549
 strike color, 547
 tool sampler, 552

GLFun application

BIDGLFunView, 575
 BIDViewController update, 582
 CGPoint structs, 579
 completion, 583
 coordinate, 575
 ellipse, Quartz, 580
 ES based drawing, 575, 580
 glEnable(), 579
 glVertexPointer, 580

initWith, 578
 nib update, 582
 objectm Texture2D, 578
 QuartzFun, 574
 Quartz version, 579
 RGBA color space, 579
 set up, 574
 snapshot, 574
 texture, 575
 vertex array, 581

QuartzFun application, 570–571

BIDQuartzFunView, 571
 2D drawing code, 564
 action methods, 563
 application in action, 554
 CGColor property, 565
 CGPoint, 570
 CGRect, 568
 changeShape, 564
 color control, 558
 color drawing, 567
 constants, define, 556
 currentColor property, 565, 568
 drawing the image, 569
 drawing the line, 565
 drawRect method, 565, 567
 image segment, 564
 initialization code, 558
 initWithCoder, 559
 lastTouch, 566
 navigation bar, 560–561
 optimizing, 570
 outlets and action, 560
 QuartzFunView skeleton, 557
 random color create, 555
 rectangle and ellipse, 567
 redDrawRect, 571
 reference change, 571
 segmented control, 562
 setNeedsDisplayInRect, 570
 set up, 554
 touchesMoved, 570
 UIImage property, 557
 UIView, 559

views of graphical world, 546
 2D drawing, 546
 painter's model, 546
 state machine, 546

R

Resources

- Apple's documentation, 717
- blogs, 720
- conferences, 721
- farewell, 723
- forums, 718
- mailing list, 718
- web sites, 719

Retain cycles, 390

S

Simple property lists, 440

Single-file persistence, 438

Storyboards, 341

- features, 341

- prototype cells

- attentionCell, 347
- cell types, 350
- editing, 346
- fields, 347
- flags, 350
- load, 350
- queueReusableCellWithIdentifier, 349
- table content, 345
- table view data source, 348
- task grabbing, 349
- view controller, 346

- segue

- BIDTasDetailController, 361
- blank slate, 356
- change of title, 357
- cocoa touch section, 359
- controller model, 358
- dragging a navigation, 356
- editedSelection property, 365
- first transition, 358
- handle tasks, 363
- key value coding, 362
- list receive the details, 364
- mutableCopy, 365
- navigator, 355
- passing a task, 361
- passing back details, 364
- plainCell to attentionCell, 366
- prepareForSegue, 362
- scene set up, 361

- seg nav, 358

- selected section, 358

- task list, 359

- text editing, 360

- triggering, 358

- UINavigationController, 356

- viewing task details, 360

- a simple storyboard

- BIDViewController, 344

- didFinishLaunchingWithArguments, 344

- empty methods, 344

- project navigator, 342

- responder, 343

- targets, 344

- view controller, 343

- static cells

- attribute inspector, 352

- clock application, 354

- from dynamic, 351

- grouped view, 352

- modes, 351

- in new table, 352

- table view data source, 353

- UITableViewController, 351

- UITableViewDataSource
- methods, 351

- transition, 366

Structured Query Language (SQL), 453

Swap file, 7

Swipes application, 594

- automatic gesture recognition, 598

- fabsf() function, 597

- multiple-finger swipe, 601

- multiple swipes implementation, 599

- performSelector, 598

- single view application template, 595

- swipe-reporting methods, 601

- swipes detection, 595

- touchesBegan, 597

- touchesMoved, 597

Switches, 100

- doSomethingButton, 105

- labeled switches, 102

- on and off, 100

- outlets and actions, 103

- round rect button, 104

- segmented control, 101

- switchChanged method, 104

T

Tab bar application

- compile and run app, 186
- empty application template, 176
- icons, 178
- identifier, 185
- root view controller, 178
- TabBarController.xib, 179

Table views

- customize cells
 - add subviews, 241
 - attributes inspector, 249
 - BIDNameAndColorCell, 242, 247
 - BIDViewController, 245
 - cell addition, 242
 - cellForRowWithIndexPath, 246
 - CellTableIdentifier, 248–249
 - contentView, 243
 - controller code, 244
 - CustomCell, 251
 - delegate methods, 251
 - dequenceReusableCellWith Identifier, 246, 251
 - dragged labels, 250
 - group and target, 242
 - identifier field, 249
 - interface builder, 248
 - labels, static text, 244
 - multiline rows, subviews, 241
 - new table view cells, 251
 - nib editor, 248
 - NSMutableString, 242
 - resized views, 250
 - reuseIdentifier method, 243
 - row properties, 246
 - UITableViewCell subclass, 242
 - UITableViewCell, nib, 247
 - user interface, 247
 - viewDidLoad, 246
- grouped and indexed sections
 - adding indexes, 258
 - attribute inspector, 252
 - cellForRowAtIndexPath, 256
 - controller implementation, 253
 - data import, 252
 - data source methods, 255

- dictionary, 253
- NSDictionary, 255
- titleForHeaderInSection, 256
- view building, 252
- with an index, 259
- with multiple sections, 257
- without indexes, 258
- iOS tables, 222–223
 - and cells, 222
 - indexed table, 224
 - plain and grouped, 224
 - UITableView, 223
 - UITableViewCell, 223
 - UITableViewDelegate protocol, 223
 - UITableViewSource protocol, 223
- search bar
 - application, 260
 - bar to a table, 265
 - data source methods, 262
 - delegate method, 264
 - didFinishLaunchingWithOptions, 266
 - didLoadSearchResultsControllerView, 261
 - filtered array, 264
 - navigation bar, 265
 - shouldReloadTableForSearchString, 263
 - tapping, 267
 - UISearchDisplayController, 260
- simple table
 - array values, 229
 - BIDViewController, 226, 236
 - cellForRowAtIndexPath, 230, 237
 - cell styles, 232
 - change size and height, 237
 - coding, 227
 - controller writing, 226
 - delegate methods, 237
 - design, view, 225
 - dwarven glory, 229
 - full size view, 225
 - image addition, 230
 - indent level set, 234–235
 - indexPath method, 236

- justification, 233
- modified size and height, 239
- modify text label, 233
- numberOfRowsInSection, 227
- picker view, 226
- property, image, 231
- row change, delegate, 240
- row selection handling, 235
- SimpleTableIdentifier, 228
- string instance, 228
- style in a row, 233
- subtitle, styles, 233
- textLabel, 229
- UITableViewCellStyleDefault, 228

Taps, 585

- doDoubleTap method, 602
- doSingleTap method, 602
- eraseMe method, 605
- gesture recognizers, 606
- labels, 603
- multiple taps detection, 601
- TapTaps application, 603
- UITapGestureRecognizer, 602

Touches, 585

- TouchExplorer application, 590
- Touch notification methods, 589

U

- UIImagePickerController, 667

User interface, 69

- action sheets and alerts, 106–109
 - buttonPressed method, 107
 - delegate method, 107
 - initializer method, 108
 - view and self.view, 109
- active and passive controls, 73
- buttons (see Buttons)
- control fun application, 69, 74
- iOS vs. Mac controls, 74
- segmented control action, 106
- static controls, 73
- switches (see Switches)

V

- View switcher, 144
- Voice over IP (VoIP), 525

W

- WhereAml application, 626
- Wi-Fi Positioning Service (WPS), 619
- Workspace window, Xcode, 16
 - interface builder, 25
 - jump bar, 24
 - LLVM compiler, 25
 - navigator view
 - breakpoint navigator, 22
 - debug navigator, 21
 - issues navigator, 20
 - log navigator, 23
 - project navigator, 18
 - search navigator, 19
 - symbol navigator, 18
- toolbar, 17
 - activity view, 17
 - assistant view, 17
 - scheme, 17
 - standard view, 17
- utility pane, 25

X, Y, Z

- Xcode, 11
 - changing attributes, 35
 - groups, 26
 - Hello World project, 26
 - inspector pane, 35
 - interface builder
 - dock, 29
 - history of, 28
 - .nib extension, 29
 - .xib extension, 29
 - in iPhone application, 38
 - keyboard shortcuts, 25
 - label, 32
 - library
 - code snippet library, 32
 - file template library, 32
 - media library, 32
 - object library, 32
 - project creation
 - ARC, 15
 - bundle identifier, 14
 - class prefix, 14
 - device family, 15
 - name and company identifier, 14

Xcode (*cont.*)

- project template selection sheet, 13
- single view application, 14
- source control repository, 16
- storyboard option, 15
- unit tests, 15
- welcome window, 12

workspace window

- compiler and debugger, 25
- interface builder, 25
- jump bar, 24
- navigator view, 18
- toolbar, 17
- utility pane, 25