

iPhone Application Programming with Cocoa Touch

By Jon Kessler

The Application Delegate

- UIApplication
 - Don't subclass it! That's silly!
- Application delegate receives messages regarding the application
- Adopt UIApplicationDelegate protocol
- Methods include
applicationDidFinishLaunching: and
applicationWillTerminate:

Design Patterns

- Key in Cocoa
- Model-View-Controller
- Delegation
- Target-Action
- Managed Memory Model

Application Life

- `applicationDidFinishLaunching:`
- `applicationWillResignActive:`
- `applicationDidBecomeActive:`
- `applicationWillTerminate:`
- `applicationDidReceiveMemoryWarning:`
- `didReceiveMemoryWarning:` in `UIViewController` subclasses

Customizing Application Behavior

- `UIInterfaceOrientationLandscape(Left/Right)` key in `Info.plist`
- `shouldAutorotateToInterfaceOrientation:` in `ViewController`
- Custom URL schemes
- Preferences
- Internationalization
- Using memory efficiently

UIWindow and UIView

- One main window per app, a UIWindow
 - Create manually, or from NIB/XIB
- UIView represents a view on which things can be drawn; a UIWindow is a UIView too!
- Lots of views and controls to use in your applications

Types of Views

- Containers—UIScrollView, UITableView
- Controls—UIButton, UITextField, UISlider, etc.
- Display views—UIImageView, UILabel, etc.
- Text/web views—UITextView, UIWebView
- Alert views and action sheets—UIAlertView, UIActionSheet
- Navigation views—UITabBar, UINavigationController
- The window

What does a UIViewController do?

- Creates and manages UIViews
- UINavigationController—Push and pop UIViewControllers
- UITabBarController—Set which UIViewControllers are contained
- See View Controller Programming Guide for more detail

UIView basics

- drawRect is the drawing method
- Can use CoreAnimation (CALayers) and CoreGraphics for custom drawing
- UIViews handle events
 - touchesBegan:withEvent:
 - touchesMoved:withEvent:
 - touchesEnded:withEvent:
 - touchesCancelled:withEvent:
- frame, bounds, and center
- Autoresizing when frame changes

Creating Views, Managing View Hierarchy

- Can be created using IB (preferred) or programmatically
- Add and remove subviews with `addSubview:` and `removeFromSuperview`; other methods exist for changing order, etc.
- `setNeedsDisplay` to request a redraw for a `UIView`
- Can hide views as well—`hidden` property

Custom Views

- Subclass UIView
- Override drawRect
- Override touch response methods
- Implement dealloc

Handling Events

- UIResponder, superclass to most UIKit classes
 - An object that can respond to UIEvents
- Responder chain
 - First responder passes event up, and it continues up until it is handled, like an Exception, in a way
- Override event handling method (touchesBegan:withEvent:, for example)
 - Handle event or forward
- Handle multi-touch events with customized event handling methods

Graphics and Drawing

- We will go over this at a later point
- Basically, use CoreGraphics, OpenGL ES, and/or CoreAnimation

Text and Web Support

- UILabel—Static text
- UITextField—Single-line input
- UITextView—Multi-line input
- UIWebView—Add Safari to your app!

Using the Keyboard

- May pop up automatically
- Handle notifications:
 - `UIKeyboardWillShowNotification`
 - `UIKeyboardDidShowNotification`
 - `UIKeyboardWillHideNotification`
 - `UIKeyboardDidHideNotification`
- `becomeFirstResponder` and `resignFirstResponder`

Files and Networking

- Application has a home directory (\$HOME) containing:
 - App bundle
 - Documents directory for storing data
 - Library directory containing saved preferences and caches
 - tmp directory for temporary storage (\$TMPDIR)
- Access using NSHomeDirectory, NSTempDirectory, others
- Read/write using Foundation or POSIX calls
 - Classes exist for writing plists and objects to files

Networking tips

- Send data all at once rather than in short bursts
- Be economical with the amount of data you send

Accessing Hardware Features

- accelerometer:didAccelerate: to access accelerometer values
 - Set update interval
 - Set your object as the delegate
 - Filter to obtain certain portions of movement
- Get device orientation with the UIDeviceOrientationDidChangeNotification notification
 - Tell shared UIDevice to beginGeneratingDeviceOrientationNotifications

Using CoreLocation

- Create an instance of CLLocationManager
- Assign a delegate
- Check if locationServicesEnabled is YES
- Call startUpdatingLocation

Getting Images

- Use UIImagePickerController
- May not be available on all devices (i.e. not iPod Touches)
- Use isSourceTypeAvailable: method

Preferences

- Make a Settings.bundle
- Specify using plist files
- Can localize with .strings file