iPhone/iPad/iOS development

Overview of iOS development

- **Language:** Objective C and Swift
- **IDE:** XCode on an intel Mac ONLY
- **Simulator**
  - quite responsive
  - easy to use
- **Device**
  - limited devices, two pixel densities
  - requires "profiles", developer keys
  - academic program vs commercial development
- **Similar object oriented libraries, callbacks, views**
- **Primary design:** Model-View-Controller
- **Objective C has "Protocols" rather than "Interfaces"**
- **Idea of a delegate**
  - A class responsible for implementing a protocol
iOS design (page 2)

Primary Classes

- UIWindow -- area in which to display content
- UIView -- drawing content and touch events
- UIViewController -- manages a set of views and their interactions

Again, can build UIs with either

- Interface Builder (graphic UI editor, .xib or .storyboard files)
- In code (I prefer this way)

Several different kinds of controllers

- View Controller -- primary UI class
- Navigation Controller -- manages view controllers
- Start with a "root view controller"

ViewController life cycle

- (id) initWithNibName: bundle: -- create from a nib
- (id) init

Standard call:

    TheClass *newObject = [[TheClass alloc] init];
view property (class instance variable) -- primary view for controller
   -(void) loadView -- default => builds view, override to build view yourself
   -(void) viewDidLoad -- Other "load time code"
   -(void) viewWillUnload -- save stuff before unload
   -(void) viewDidUnload -- view is gone, cleanup
   -(void) dealloc -- deallocation cleanup
   -(void) didReceiveMemoryWarning -- app received a memory warning
Use sample code "TheElements" (old and new) as example.

- main.m -- Main program ... same for all
- Application Delegate -- first code run after main runs
  - applicationDidFinishLaunching:
    - Initialize the application
    - Set up the root ViewController (UITabBarController)
    - Set up the Navigation Controller (for use with TabBar)
    - Add the navigationController’s view to the "window"
    - Make the window visible ...
  - bunch of other possible things in the app delegate
    - applicationDidEnterBackground:
    - applicationWillEnterForeground:
    - applicationWillResignActive:
    - applicationWillTerminate:
    - dealloc
ElementsTableViewController and the TableView

- ElementsTableViewController
  - view allocated in application delegate
- UITableViewDataSource delegate
  - tableView: didSelectRowAtIndexPath: -- click happened
- has an associated Data Source
- Data source implements UITableViewDataSource delegate (similar to an adapter)
  - numberOfSectionsInTableView:
  - tableView: numberOfRowsInSection:
  - tableView: cellForRowAtIndexPath:
  - tableView: willSelectRowAtIndexPath:
- TableView similar to a ListView ...
- Asks for views only for visible elements
- Highly configurable
Other things View Controller do

- Rotation events
  - 
  - -(void) willRotateToInterfaceOrientation: (UIInterfaceOrientation)
    toInterfaceOrientation duration:(NSTimeInterval) duration
  - viewWillAppear: and viewDidAppear:
  - viewWillDisappear: and viewDidDisappear:
  - viewWillLayoutSubviews: and viewDidLayoutSubviews:

Navigation Controller

- Provides for the top "navigation bar" and
- Push view controllers ...
  - [[self navigationController] pushViewController:target animated:YES]
Let's look at "UICatalog" and "NavBar" to show ideas

- MainViewController.h/m -- initially run code
  - NSMutableArray & NSDictionary (very much like JSONArray/Object)
  - -(NSInteger)tableView:numberOfRowsInSection: -- callbacks for the tableview
  - -(UITableViewCell *)tableView:cellForRowAtIndexPath:
  - -(void)tableView:didSelectRowAtIndexPath:

ButtonViewController.m

- Static factory method -- for button creation
  + (UIButton *)buttonWithTitle:(NSString *)title
    target: (id)target
    selector: (SEL)selector
    frame: (CGRect)frame
    image: (UIImage *)image
    imagePressed: (UIImage *)imagePressed
    darkTextColor: (BOOL)darkTextColor

- typical viewDidUnload
SegmentViewController.h/m

- Typical view controller that builds the UI in code

WebViewController.h/m

- How you start a browser on iOS
- Browser/WebView delegate issues
  - viewWillAppear -- UIViewController delegate
  - viewWillDisappear

Navbar has a variety of navigation bar styles.
iOS version 7 and beyond

iOS 7 -- Major UI redesign of look, "flat look", 2014
☐ use of storyboards is now pushed
☐ Example code all rewritten!
  ☐ UICatalog as an example: "old" vs "current"
☐ Many other changes is "preferred style" of programming

iOS 8 -- 2014
☐ Major change: Continuity -- hand off between products
☐ Added Health apps and APIs
☐ Swift released for Xcode v6. Uses the Objective-C runtime library

iOS 9 -- 2015
☐ No huge changes, just incremental, many in Apple apps
☐ New Music API allowed developers to integrate Apple Music into their apps
☐ Swift version 2.0

iOS 10 -- 2016
☐ Updates to many Apple apps, Swift version 3.0
☐ APIs now available to developers: Siri, iMessage and Maps
☐ Allowed apps to produce RAW camera formats.
☐ CallKit allowed VoIP apps to act similar to Phone and Face Time
iOS 11 -- 2017
- Siri could translate, new lock screen capabilities (notifications)
- Added augmented reality via ARKit
- Core ML framework for ML work.
- Core NFC -- near field communication
- Swift version 4

iOS 12 -- 2018
- Battery improvements, Screen Time feature
- Many Standard apps improved with new features
- Swift 4.2 (5.0 released in March 2019)