**Topic: Tab View Controller**  
(Previous topic: Multi-view apps)

Intro

* Look at due dates
* Reminder about code reviews

Review

* Multiple content views require multiple view controllers
* Switching between views can be done with a Navigation Controller (multiple types) or by your own code.
  + Textbook (Toply et al) did it with code
  + Xamarin Phoneword app used a Navigation Controller (UINavigationController) and Segues.
  + Your lab assignment had you do both

Overview

* The UITabBarController) is another kind of nav controller, but it inherits from UIViewController). <https://developer.apple.com/library/ios/documentation/UIKit/Reference/UITabBarController_Class/index.html>
* Has a tab bar. Clicking a tab launches a view (via it’s ViewController).
* Switching ViewControllers can be done in code, or with Segues.
  + Texbook shows a TabBarController with segues
  + Xamarin Guide shows a TabBarController with code that does the switching
* Storyboard arrangement
  + Source-less segue goes to TabBarController. It is the root controller
  + ctrl-dragging from the TabBarController to a ViewController adds a Segue and a TabBarItem. Change the title of the TabBarItem and add an icon.
* Picker
  + Give the instance of the Picker a name in the properties box (top item)
  + Swift supports multiple inheritance of classes, but C# only supports multiple inheritance of interfaces. Use IUIPickerViewDataSource.
  + Surprisingly, IUIPickerViewDelegate doesn’t have any methods on it! This is because it is mirroring the protocols in the Objective C delegate and the methods there are all optional. So, we need to use a class. We can use a nested class. In ViewDidLoad, we can create an object from it and assign it to the Picker’s Delegate property.
  + We could optionally create a nested class derived from UIPickerViewDataSource as well. Then, in ViewDidLoad, we would create an object from it and assign it to the Picker’s DataSource property.