iOS Foundations II
Day 4

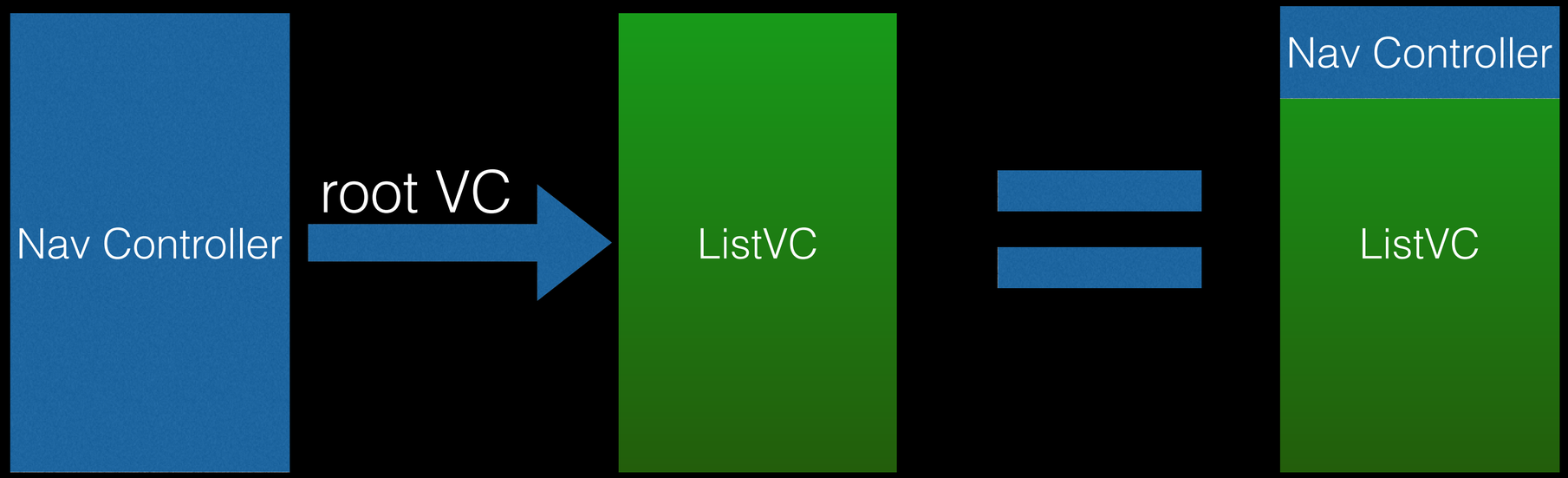
- Day 3 Lecture and Homework Review
- Segues
- Navigation Controller
- ‘Passing’ objects using protocols, delegates, and segues

NavigationControllers

- “A navigation controller manages a stack of view controllers to provide a drill-down interface for hierarchical content.”

NavigationControllers

- “The navigation controller’s primary responsibility is to respond to user actions by pushing new content view controllers onto the stack or popping content view controllers off of the stack”
- The first ViewController you push onto the stack (when the app launches) becomes the **rootViewController**. It never pops, to ensure a view is always on screen.
- Nav Controllers have properties:
  - The topViewController
  - An array of all the viewControllers on the stack.
- The Nav bar on top can be customized (e.g., widgets) or hidden.



A Navigation controller is always instantiated with a root view controller, which is the 1st view shown. A view controller contained in a navigation controller has a navigation bar at the top of the view.

NavigationControllers

- 2 ways to get navigation controllers into your app:
  - Via storyboard, select the view controller you want to set as the root view controller, then Editor> Embed In> Navigation Controller
  - In code, instantiate an instance of UINavigationController and give it a root view controller

Demo 4.1: Navigation with two views

Segues

- Pronounced “SAYG-WAY”, not “SEEG”
- Segues are provided by the **storyboard** to help you easily transition from one view controller to another.
- Storyboard role & Xcode restrictions:
  - You can **not** create a segue via code, but can **trigger** one in code (demo)
  - If you write an app without using storyboards, the app will not use segues.
  - Creating segues is just like creating IBOutlets and IBActions — lots of dragging and hoping you dragged to the right destination

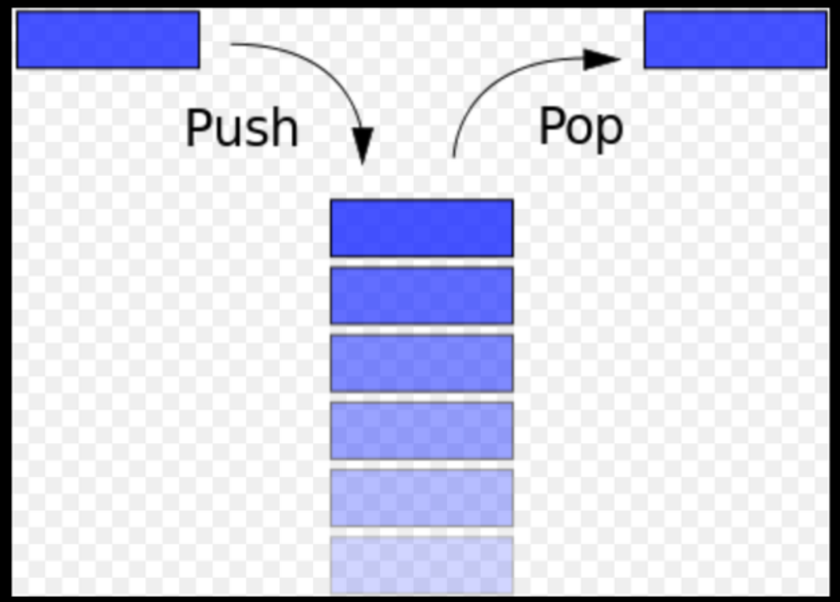
Creating Segues

- Storyboard role & Xcode restrictions:
  - You can **not** create a segue via code, but can **trigger** one in code (demo)
  - If you write an app without using storyboards, the app will not use segues.
  - Creating segues is just like creating IBOutlets and IBActions — lots of dragging and hoping you dragged to the right destination
  - In code, a segue is referred to as a *transition*.

Demo 4.2:
Simple Segues

Stack Data Structure

- “A stack is particular kind of abstract data type or collection in which the only operations on the collection is adding (push) or removal (pop).” - Wikipedia
- LIFO : Last-In-First-Out. The last item added is the first to be removed.



NavigationControllers

- Storyboards make navigation controllers extremely easy to install into your app. Here's all the methods you need to do it in code without the storyboard:
- `init(rootViewController:)` UINavigationController is initialized with a rootViewController.
- `pushViewController(animated:)` To add or 'push' a view controller onto the stack.
- `popViewController(animated:)` To remove or 'pop' a view controller from the stack.

Demo 4.3: Navigation With Four Views

Segues

- Two primary segues: “Show” and “Present”
  - **Show** pushes a view controller onto the navigation stack, it will slide in, right to left
  - **Present** refers to modally presenting a view controller, it usually slides up from the bottom. Modal presentations are usually ‘one-off’ screens, like settings or quick uploads.
- Customized segues — change their behavior to match your needs.

Segues in Code

- 2 primary methods for dealing with segues in code:
    1. performSegueWithIdentifier: triggers a segue manually in code
    2. prepareForSegue:sender: allows you to run some code before the segue actually fires. You don’t call this method yourself, you just implement it in your view controller, it is called by the system.

Triggering a Segue

- There are 2 ways a segue is triggered:
    1. The user presses an interface object that is hooked up to a segue
    2. Code is run that triggers the segue ‘manually’
- To fire the segue manually, you call the method performSegueWithIdentifier

Passing data to View Controllers

- Many patterns to choose from to pass around data in our app:
  - Delegation
  - Notification Center
  - Singletons
  - Persistence
- Demo 4.4 uses **delegation**, which we discussed on Day 3.

`prepareForSegue(arg1, arg2)`

- Called on the **source** view controller of the segue, right before the segue is actually performed.
- Arg1 is the segue object itself (an instance of the UISegue class)
- The most important property of a **UISegue** instance is the **destinationViewController** attribute. It's a reference to the **destination** view controller (the one you're about to segue to).
- Now we will segue to the next slide of this lecture. Ha!

Demo 4.3: Navigation With Four Views