



# Pace University - PCBT

Advanced iOS Class Spring 2015

# STORYBOARD INTRODUCTION

- + This lecture will provide a background and context for using the Storyboarding in your iOS Apps.
- + We will discuss the history, purpose, and features.
- + This lecture will also discuss many key topics such as navigation with segues and auto-layout.
- + The lecture should be accompanied by an XCode exercise creating several projects using Storyboards.
- + [https://developer.apple.com/library/ios/recipes/xcode\\_help-IB\\_storyboard/chapters/AboutStoryboards.html](https://developer.apple.com/library/ios/recipes/xcode_help-IB_storyboard/chapters/AboutStoryboards.html)

# What is a Storyboard?

- + A container for all your Scenes (View Controllers, Nav Controllers, TabBar Controllers, etc)
- + A manager of connections and transitions between these scenes (these are called Segues)
- + Storyboards give you a complete look at the flow of your application that you can never get from individual nib files floating around.
- + A reducer of all the "clutter" that happens when you have several controllers each with it's own nib file.

# Other Advantages of Storyboard?

- + Storyboards have better support for tableviews. That is you can use "Dynamic" and "Prototype" cells.
- + If your app supports multiple devices, it's a good way to organize different views.
- + If you are working on someone's code you can get the better understanding of the flow of the app.

# Disadvantages

- + StoryBoardSegues are kind of rigid and you may make use of prepareForSegue many times.
- + Can be difficult to work on in a team (not really if you know how to merge).
- + For storyboard you will need a big screen specially in case of iPad.
- + Difficulty while copying views from other apps to storyboard.

# What is a SCENE?

- + This is the fundamental unit of the Storyboard. It's really just a UIViewController.

# Can I use a Storyboard but program in a more traditional way?

- + Yes, it's very easy to load view controllers from a Storyboard just like you did XIB files.
- + Storyboards allow you to automatically setup the default Scene to be shown.
- + With the App Delegate you can use the traditional loading not through the PLIST file.

# Loading a View Controller in App Delegate

```
var window: UIWindow?  
  
var initialViewController: UIViewController?  
  
func application(application: UIApplication, didFinishLaunchingWithOptions  
launchOptions: [NSObject: AnyObject]?) -> Bool {  
  
    self.window = UIWindow(frame: UIScreen.mainScreen().bounds)  
  
    var storyboard = UIStoryboard(name: "Main", bundle: nil)  
  
    var initialViewController =  
storyboard.instantiateViewControllerWithIdentifier("MVC") as UIViewController  
  
    self.window?.rootViewController = initialViewController  
  
    self.window?.makeKeyAndVisible()  
  
    return true;  
}
```

# What is a Navigation Controller?

- + A Navigation Controller is a special kind of view controller that manages a stack of view controllers and their corresponding views. It's an ideal way to display hierarchical data. The Navigation Controller is always initialized with a root view controller; this will be the starting view at the bottom of the stack.



# What is a Page View Controller?

- + A page view controller has a single view in which it hosts your content. The UI provided by a page view controller is visible when the user is turning the page. The page view controller applies a page curl to its view controller's view, providing the visual appearance of a page being turned.
- + The navigation provided by a page view controller is a linear series of pages. It is well suited for presenting content that is accessed in a linear fashion, such as the text of a novel, and for presenting content that has natural page breaks, such as a calendar. For content that the user needs to access in a nonlinear fashion, such as a reference book, you are responsible for providing the navigation logic and UI.

# What is a SplitViewController?

- + The UISplitViewController class is a container view controller that manages two panes of information. The first pane has a fixed width of 320 points and a height that matches the visible window height. The second pane fills the remaining space. Figure 4-1 shows a split view controller interface.
- + <https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/SplitViewControllers.html>

# Links

- + <https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/NavigationControllers.html>
- + <https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/TabBarController.html>
- + <https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/PageViewController.html>

# What is a Tab Bar Controller?

- + You use tab bar controller to organize your app into one or more distinct modes of operation. The view hierarchy of a tab bar controller is self contained. It is composed of views that the tab bar controller manages directly and views that are managed by content view controllers you provide. Each content view controller manages a distinct view hierarchy, and the tab bar controller coordinates the navigation between the view hierarchies.

iOS6 Tab Bar

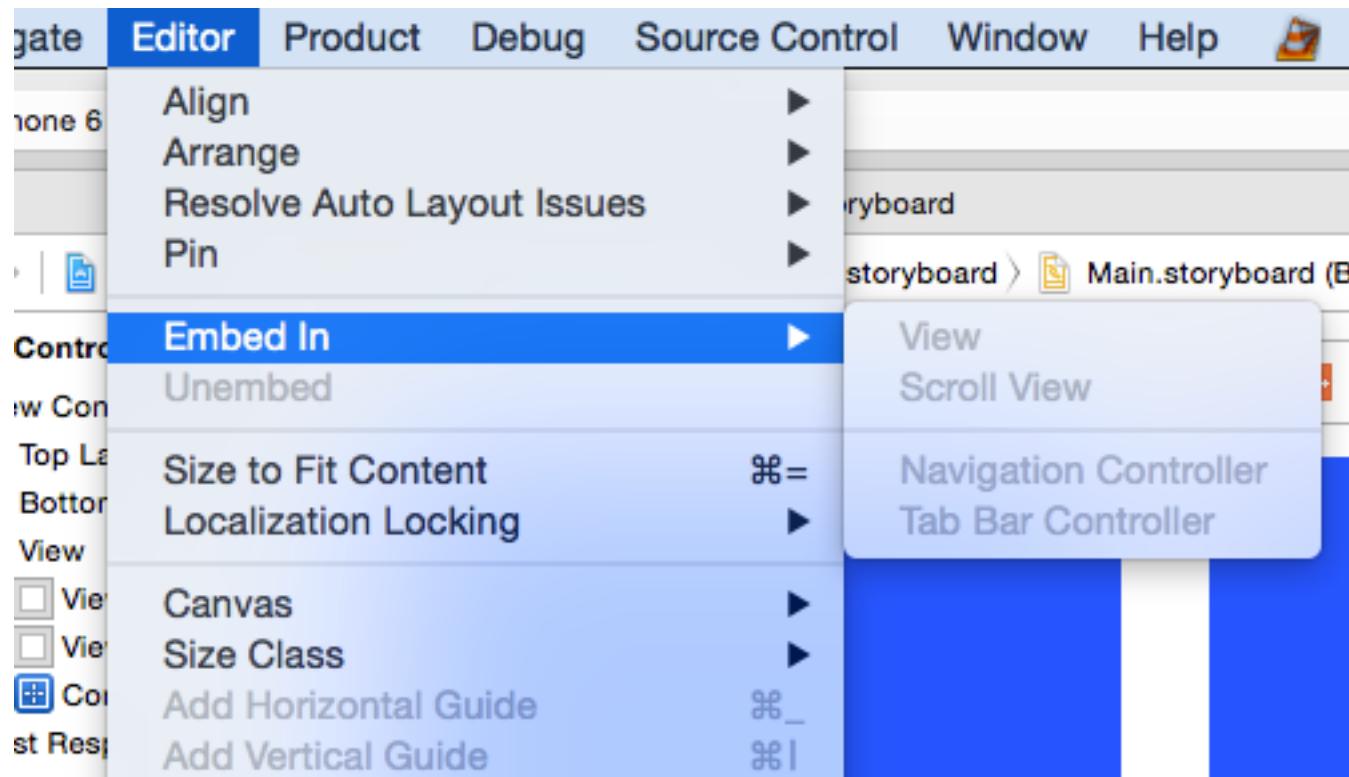


iOS7 Style



# Using the Embed In Trick

- + You can take a basic Scene and embed it in the navigation.



# Life of a SEGUE

- + To understand how custom segues work, you need to understand the life cycle of a segue object. Segue objects are instances of UIStoryboardSegue or one of its subclasses. Your app never creates segue objects directly; they are always created on your behalf by iOS when a segue is triggered. Here's what happens:
  - + The destination controller is created and initialized.
  - + The segue object is created and its initWithIdentifier:source:destination: method is called. The identifier is the unique string you provided for the segue in Interface Builder, and the two other parameters represent the two controller objects in the transition.
  - + The source view controller's prepareForSegue:sender: method is called. See Configuring the Destination Controller When a Segue is Triggered.
  - + The segue object's perform method is called. This method performs a transition to bring the destination view controller on-screen.
  - + The reference to the segue object is released, causing it to be deallocated.

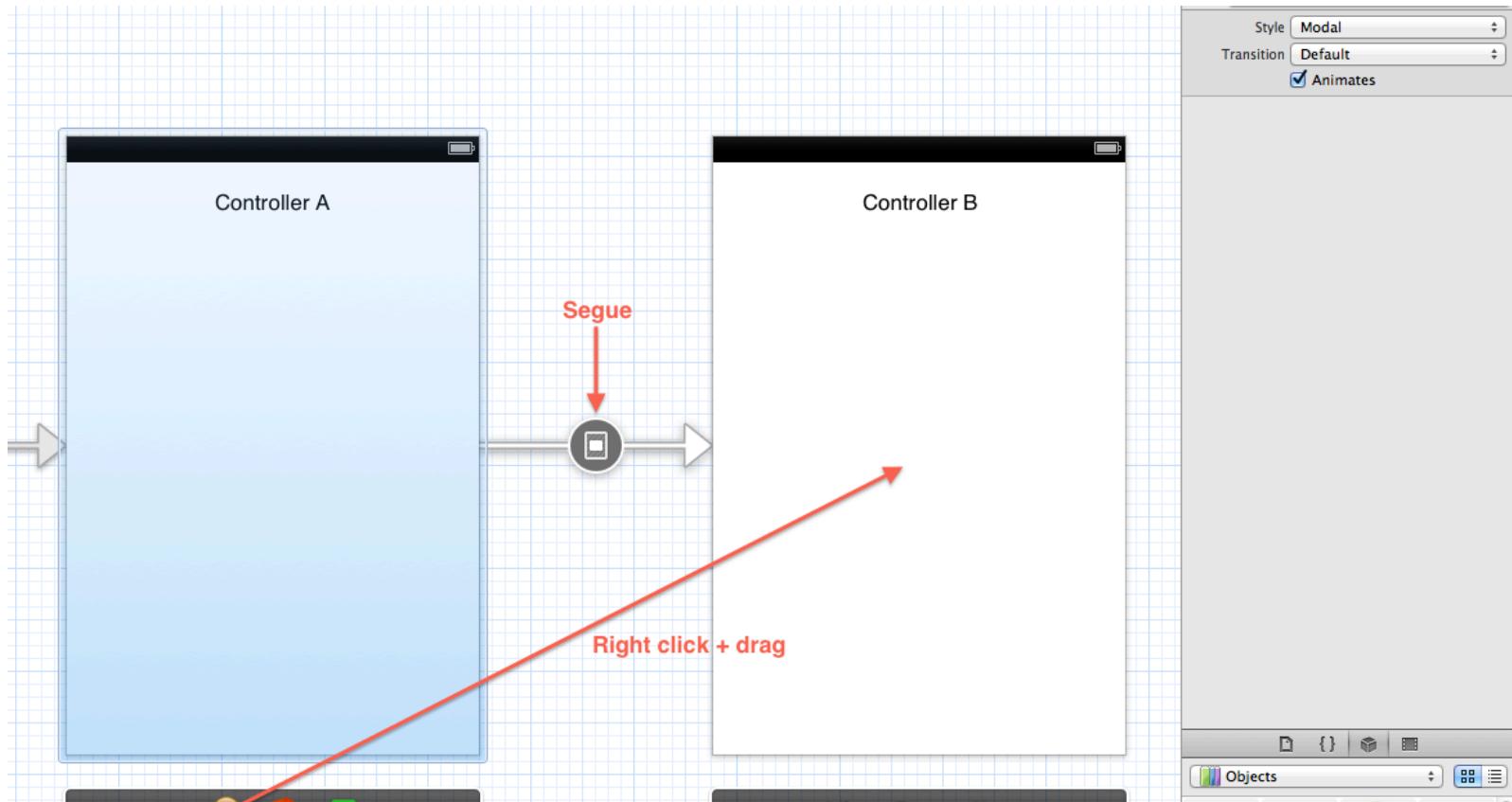
# Custom Segues

- + <http://blog.dadabeatnik.com/2013/10/13/custom-segues/>
- + A Push segue is used when you're using a Navigation Controller. The first VC is placed on the navigation stack and the Second View Controller slides into view. When you press the "Back" button the Second VC is dismissed and the first VC is "popped" off the stack and is presented again.
- + A "Modal" segue creates a relationship between the VC that did the presenting and the VC that was presented. Put simply, when a modal segue occurs, the target VC is presented in front of the source VC and it's the responsibility of the presented VC to dismiss itself at some point, perhaps when triggered by pressing a "Done" or "Cancel" button.

# Creating a Segue with a button



# Create a Segue without a Button



# What is Auto Layout?

- + Auto Layout is a system that lets you lay out your app's user interface by creating a mathematical description of the relationships between the elements. You define these relationships in terms of constraints either on individual elements, or between sets of elements. Using Auto Layout, you can create a dynamic and versatile interface that responds appropriately to changes in screen size, device orientation, and localization.
- + <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/AutolayoutPG/Introduction/Introduction.html>

# Storyboard Tutorials

- + <http://www.raywenderlich.com/81879/storyboards-tutorial-swift-part-1>

# All About Segues

- + [https://developer.apple.com/library/ios/recipes/xcode\\_help-IB\\_storyboard/chapters/StoryboardSegue.html](https://developer.apple.com/library/ios/recipes/xcode_help-IB_storyboard/chapters/StoryboardSegue.html)

# Auto Layout Tutorial

- + <http://www.raywenderlich.com/83129/beginning-auto-layout-tutorial-swift-part-1>
- + <http://makeapppie.com/2015/01/08/basic-auto-layout-a-practical-view/>
- + <http://makeapppie.com/tag/xib-in-swift/>

# End of Lecture

[rpascazio@pace.edu](mailto:rpascazio@pace.edu)