

# MOBILE DEVELOPMENT

*Kishin Manglani*

---

**RECAP**

---

**RECAP**

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

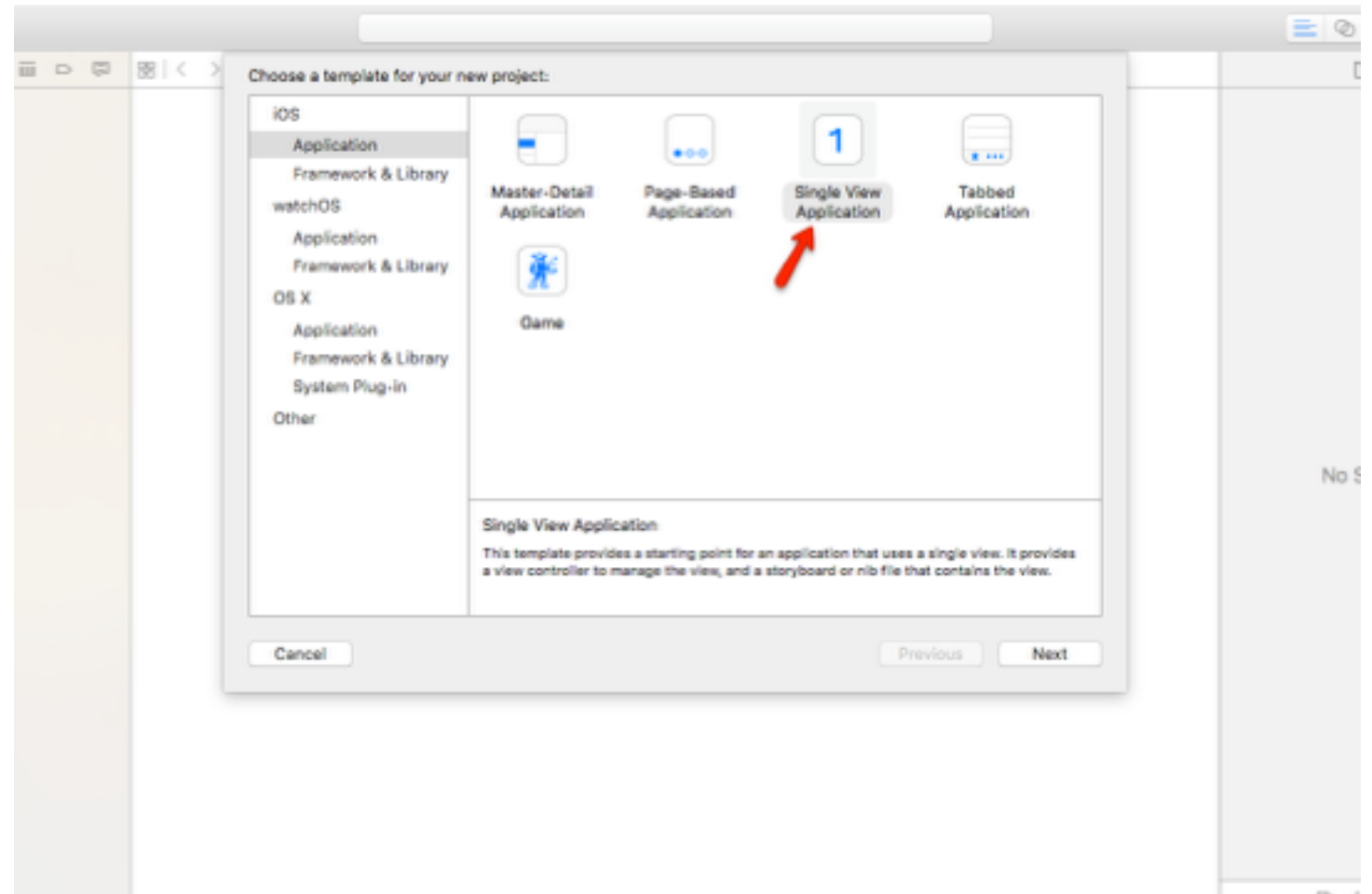
# OBJECTIVES

- Label Xcode and many of the buttons in it
- Deep dive into UIView
- Exploring UIViewController
- What's the difference between the two?
- Segues

# INTRO TO VIEWS AND VIEW CONTROLLERS

## NEW PROJECT

- ▶ When creating a new project (Command + Shift + N) we selected the “Single View Application” template
- ▶ This provides one view controller



# INTRO TO VIEWS AND VIEW CONTROLLERS

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

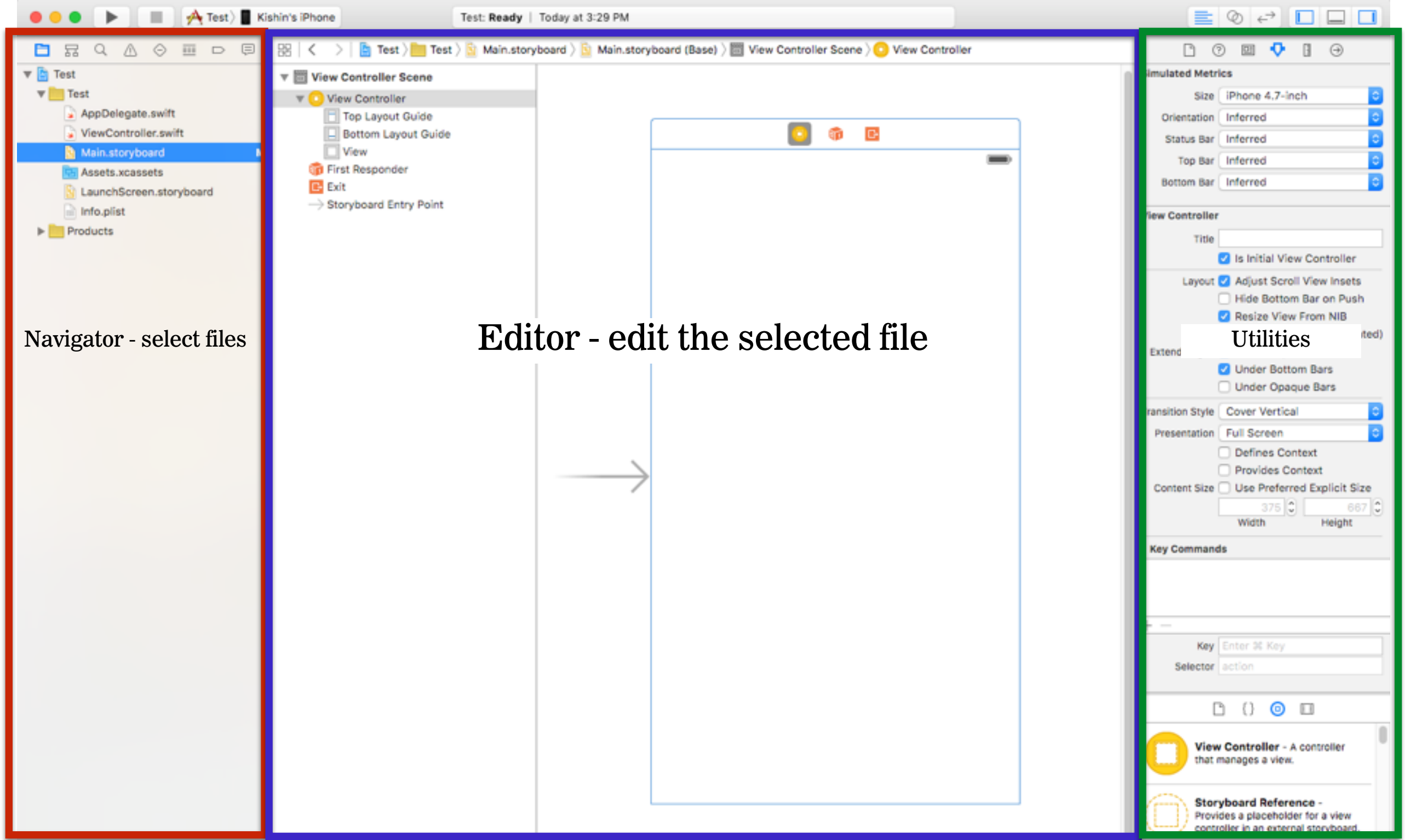
Devices:

☐ Use Core Data

☐ Include Unit Tests

☐ Include UI Tests

- Select Swift
- iPhone
- Uncheck all boxes



---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UIVIEW

# INTRO TO VIEWS AND VIEW CONTROLLERS

---

## UIVIEW

- A UIView defines a rectangular area that is drawn on the screen
  - To do so it needs an X and Y coordinate and a height and width
- It may also contain other views (subviews) that are inside it and move with it

x, y: 0, 0

height: 200



width: 100



---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UIVIEW

- UIKit framework includes widely used subclasses
- When we drag components into our storyboard these are all UIViews, many of them were subclasses of UIViews
- UIViews are “dumb”
- What does that mean?

---

**INTRO TO VIEWS AND VIEW CONTROLLERS**

---

# **UIVIEWCONTROLLER**

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UIVIEWCONTROLLER

- A UIViewController is NOT drawable on the screen
  - Don't technically see a UIViewController
- Instead, each UIViewController contains a UIView, which is what we see
- Each ViewController has a View property associated to it, which is the one you are seeing in interface
- UIViewControllers are “smart”: glue between your overall application and the UIViews/screen

Test > Test > Main.storyboard > Main.storyboard (Base) > View Controller Scene > View Controller

View Controller Scene

View Controller

Top Layout Guide

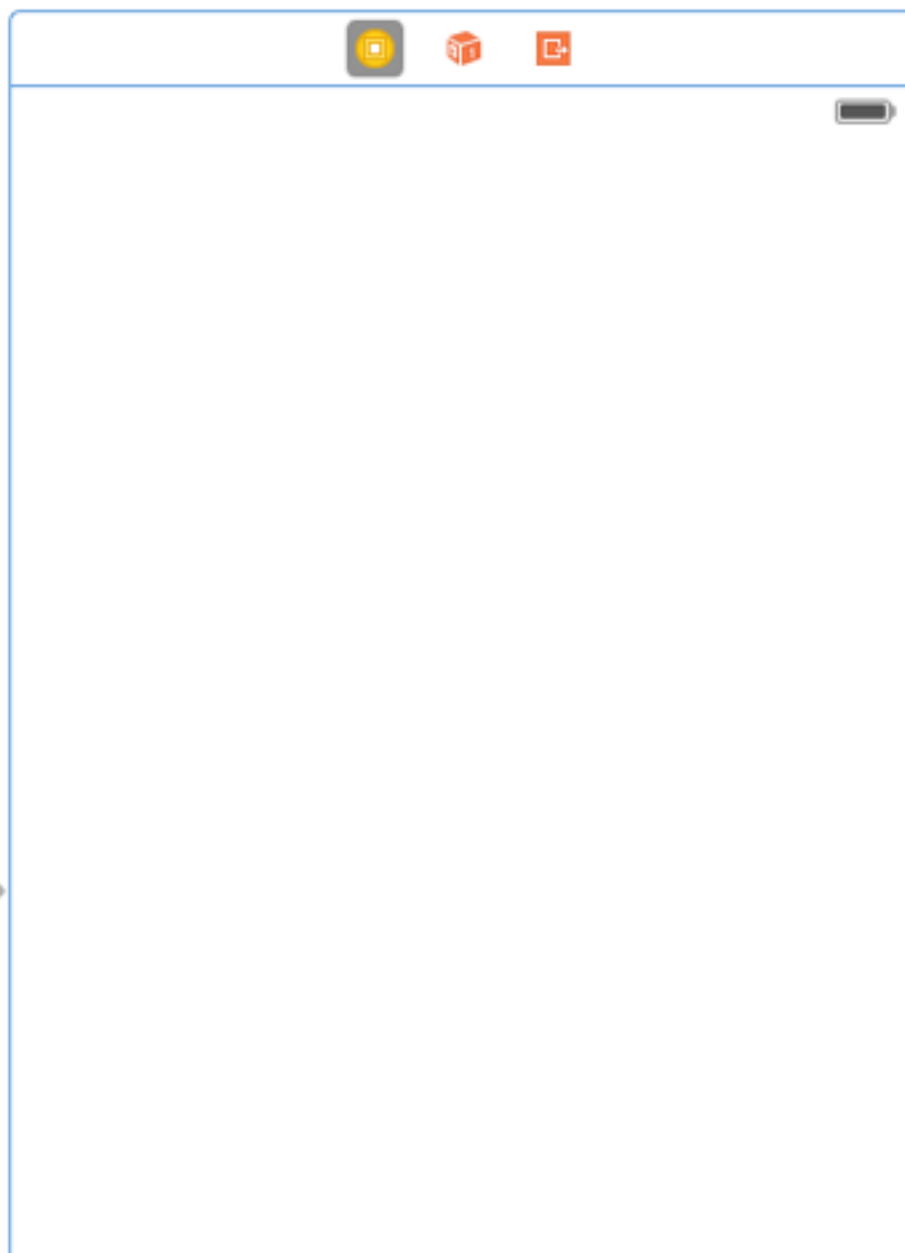
Bottom Layout Guide

View

First Responder

Exit

Storyboard Entry Point



Simulated Metrics

Size iPhone 4.7-inch

Orientation Inferred

Status Bar Inferred

Top Bar Inferred

Bottom Bar Inferred

View Controller

Title

☒ Is Initial View Controller

Layout ☒ Adjust Scroll View Insets

☐ Hide Bottom Bar on Push

☒ Resize View From NIB

☐ Use Full Screen (Deprecated)

Extend Edges ☒ Under Top Bars

☒ Under Bottom Bars

☐ Under Opaque Bars

Transition Style Cover Vertical

Presentation Full Screen

☐ Defines Context

☐ Provides Context

Content Size ☐ Use Preferred Explicit Size

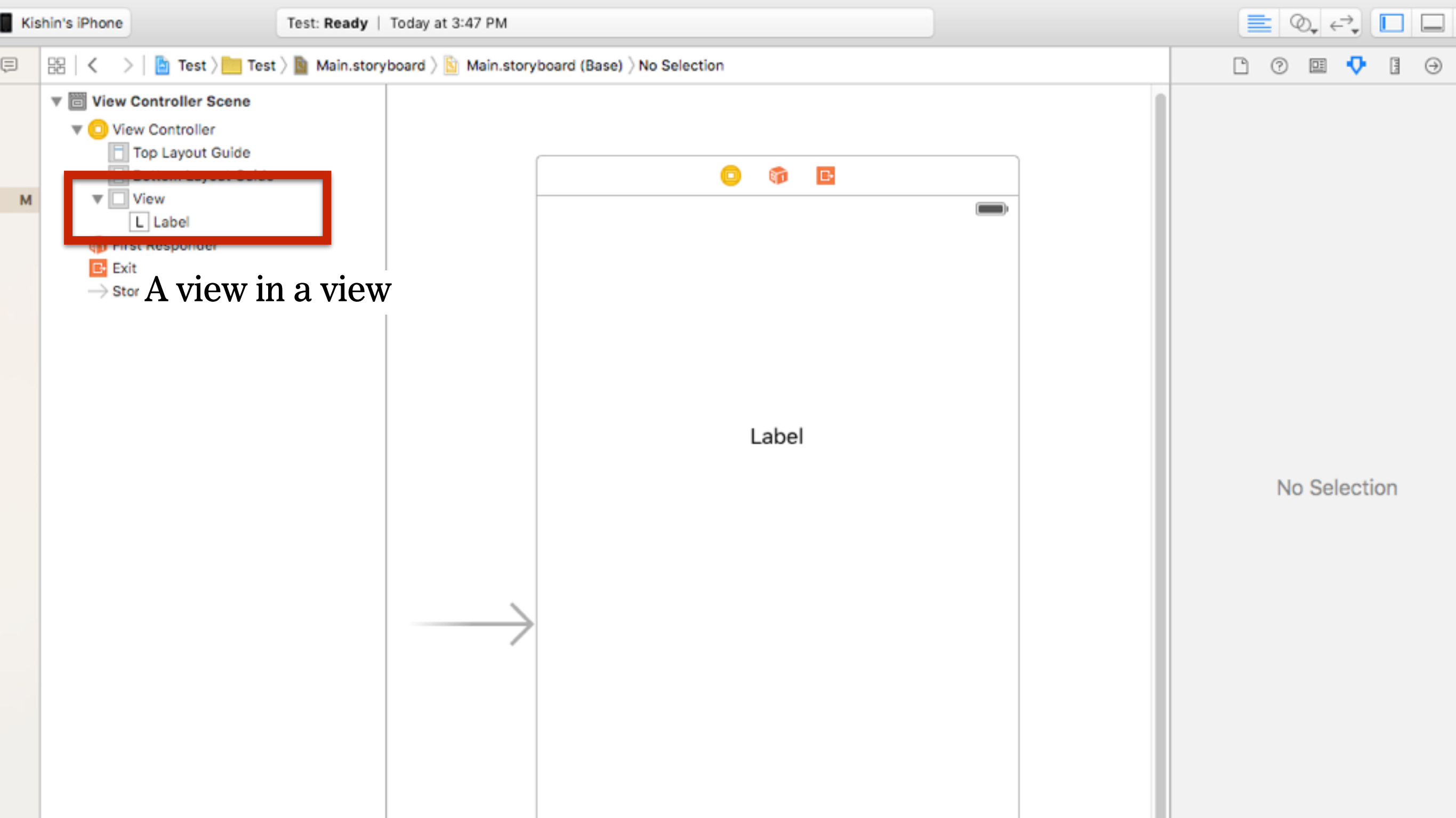
375

Width

667

Height

Key Commands



A view in a view

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UIVIEWCONTROLLER

- We can add a second view controller in the storyboard
- There are also many different types of view controllers, which we'll cover shortly
- The official storyboard terminology for a view controller is “scene”

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UIVIEW VS UIVIEWCONTROLLER

---

**INTRO TO VIEWS AND VIEW CONTROLLERS**

---

# STORYBOARDS



---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# STORYBOARDS

- A Storyboard is a what you see is what you get (WYSIWYG) editor
  - In other words, you can just drag and drop UIViews
- When we build and run our storyboard can serve as an app, it can run on a device and is functional
- Who's familiar with Flinto, Invisio, or something similar?
  - With what we've learned so far you've learned how to build a far more powerful prototype
- Composed of UIViews, UIViewController and Segues

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# SEGUE

---

## **INTRO TO VIEWS AND VIEW CONTROLLERS**

---

# **SEGUE**

- A transition from one View Controller to another View Controller



Label

Button

# Segue



View Controller



# INTRO TO VIEWS AND VIEW CONTROLLERS

---

## SEGUE

- How do we create a segue?
- Control + drag from a *button* in a view controller to a second view controller
- Can invoke/start the segue by pressing the a or in code
- Can use standard iOS transitions or even create custom ones

### Action Segue

show

show detail

present modally

popover presentation

custom

### Non-Adaptive Action Segue

push (deprecated)

modal (deprecated)

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# LET'S TRY IT

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# SEGUE PRACTICE

- Start a new project
- Create two view controllers
- Create a button with the text “Next”
- Create a segue between the two view controllers
- Make the background on the second view controller black
- How can you go back to the previous view controller?

---

**INTRO TO VIEWS AND VIEW CONTROLLERS**

---

# **UIVIEWCONTROLLERS**



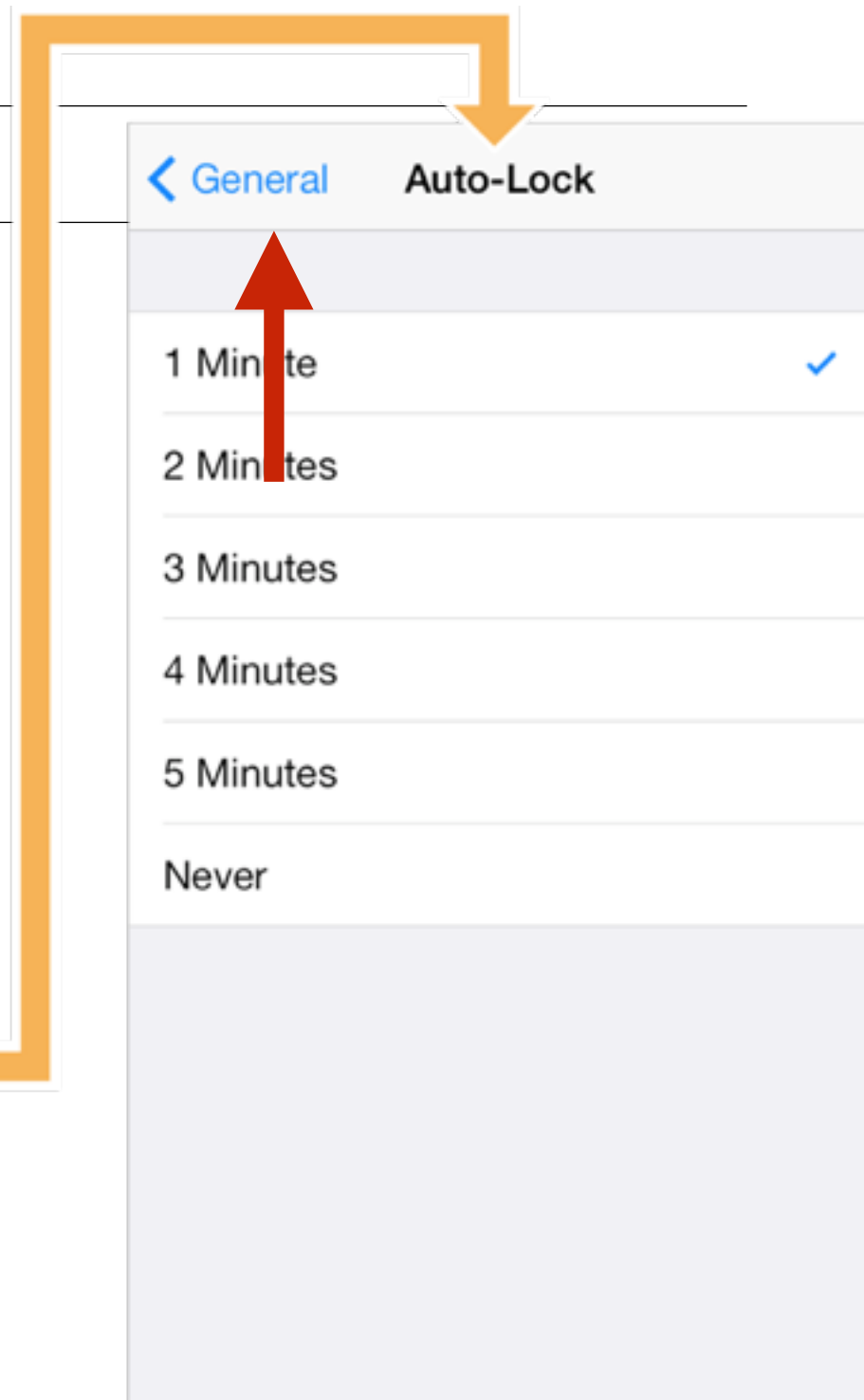
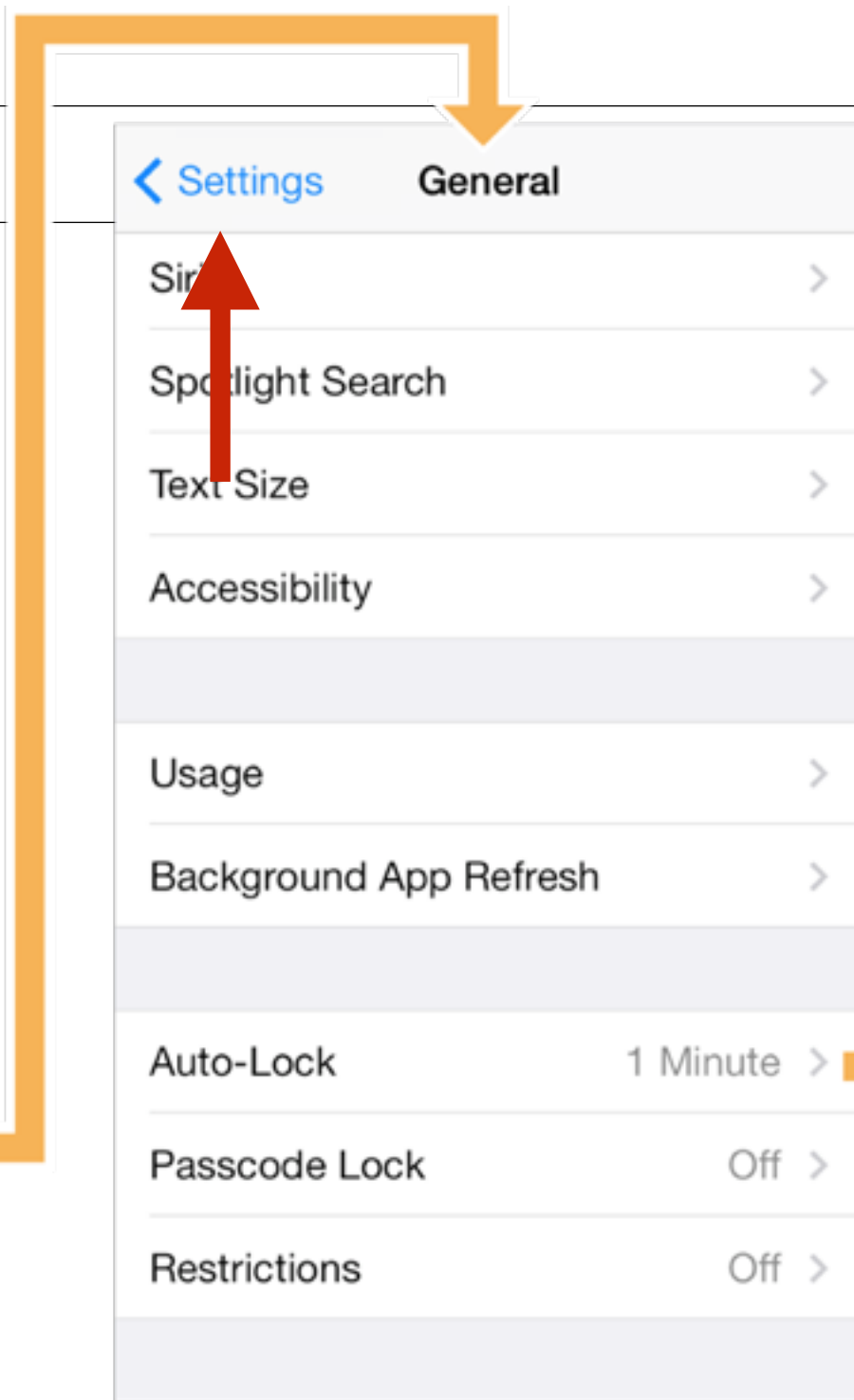
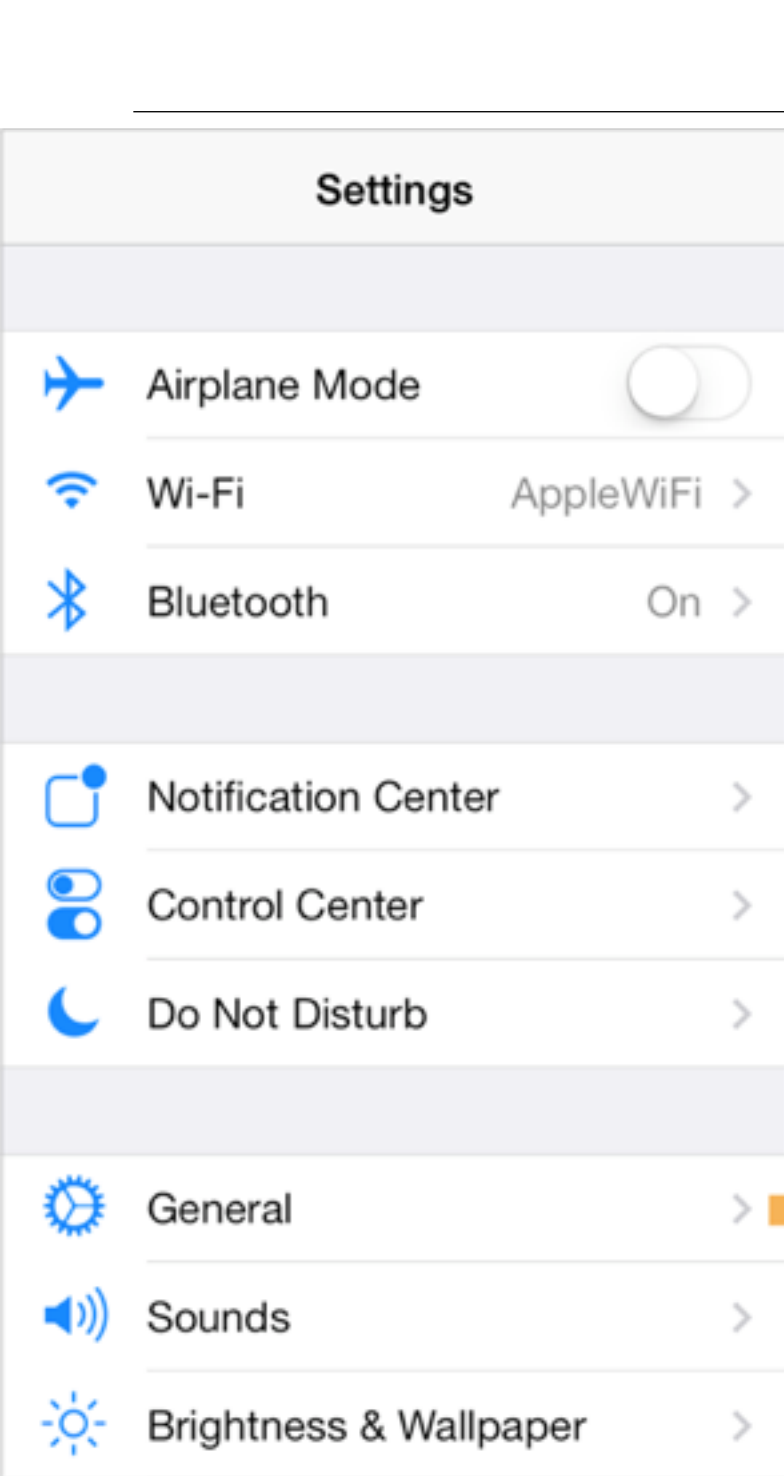
---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UINavigationController

- UINavigationController
- A “special” view controller that manages hierarchical content
- When we want to view the next view controller, it will “slide” in and give us a back button
- Has a root view controller, which is just a fancy way of saying the first view controller
- Navigation Bar at the top



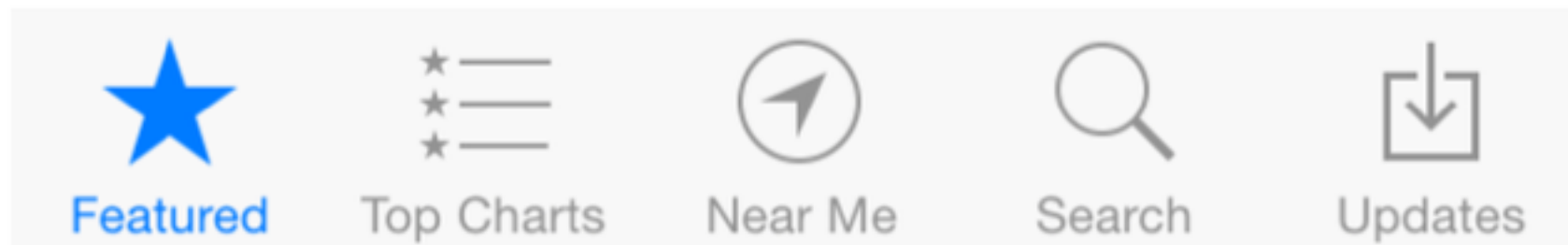
---

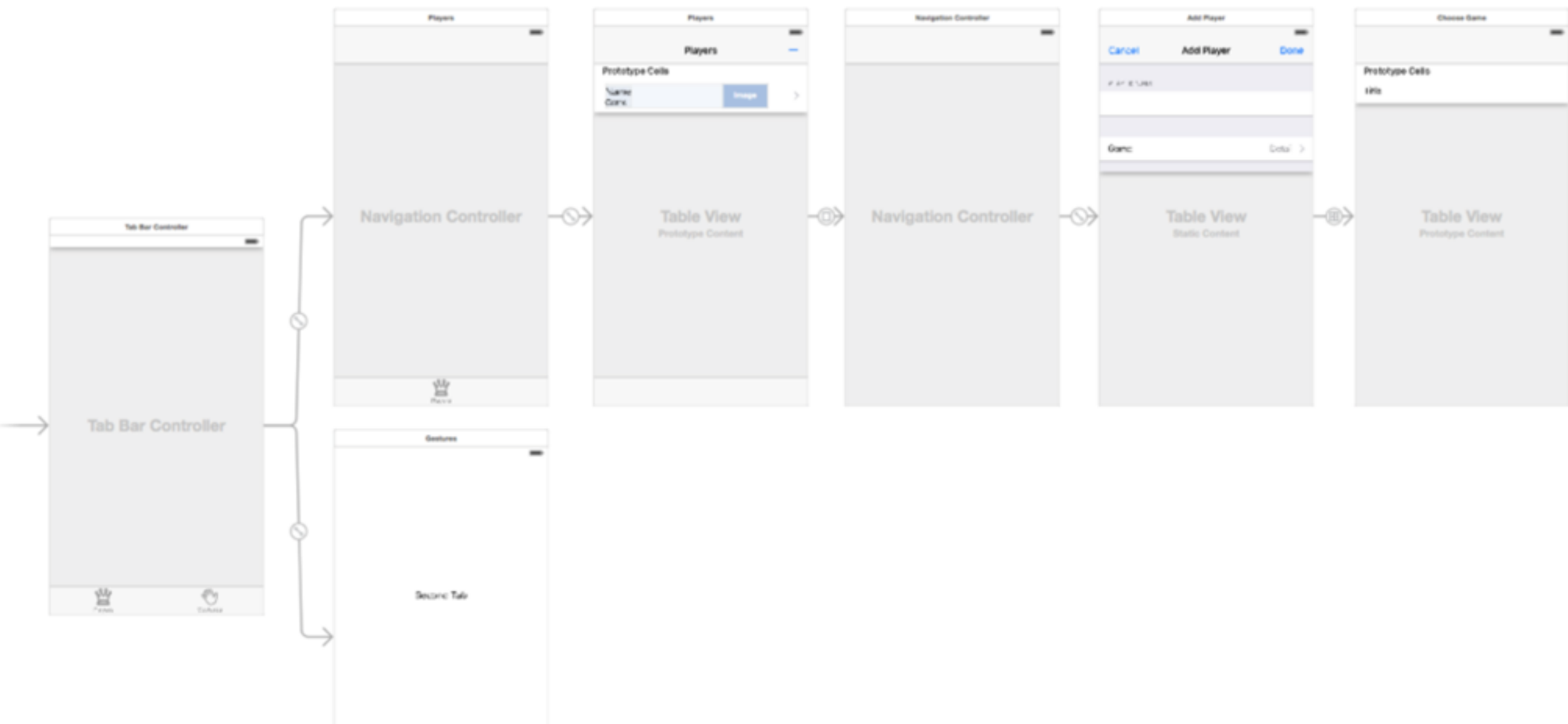
## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UITABBARCONTROLLER

- UITabBarController
- Provides tabs at the bottom of the screen
- Similar to what you see in the Facebook, Twitter, or Instagram apps
- Each tab represents a view controller
- UITabBar





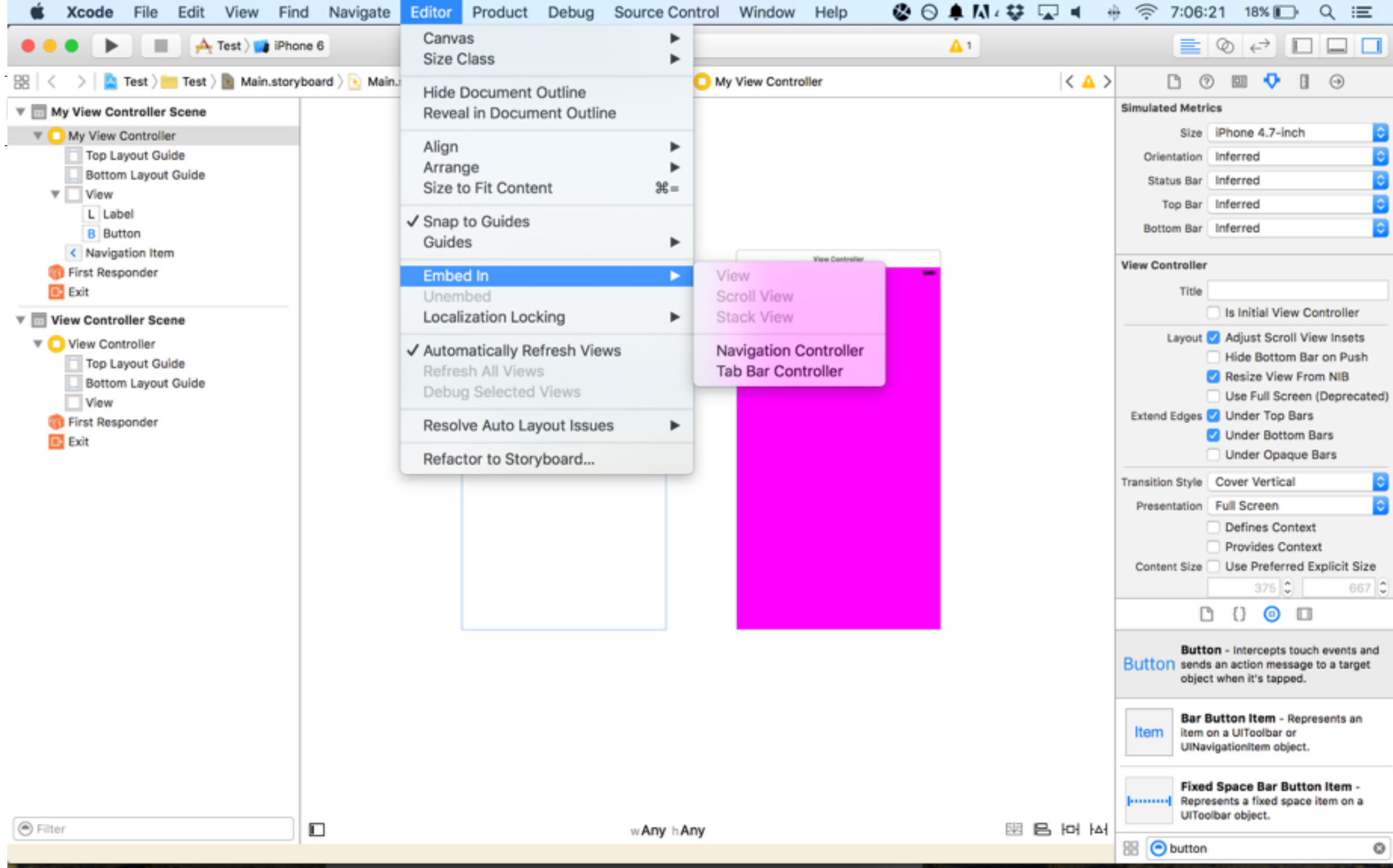
---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

### EMBED

- Putting a view controller inside a UINavigationController or a UITabBarController is easy using storyboards
- Select a View Controller in Interface Builder -> Click Editor in the menu bar -> Click Embed In



---

**INTRO TO VIEWS AND VIEW CONTROLLERS**

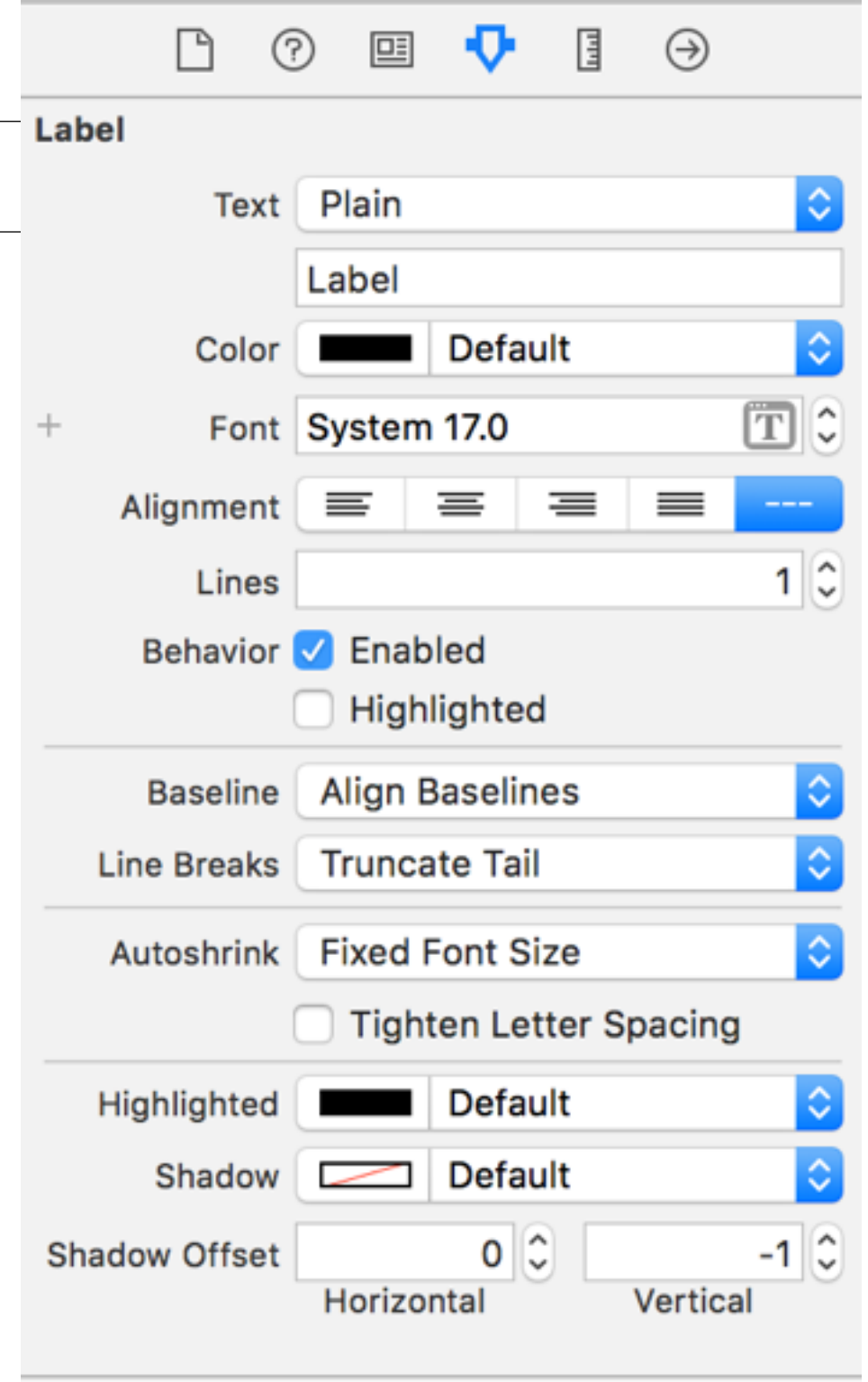
---

# DEEPER DIVE INTO UIVIEWS

## INTRO TO VIEWS AND VIEW CONTROLLERS

# UILabel

- A Label or UILabel displays static text on the screen
- This is one of the most commonly used elements

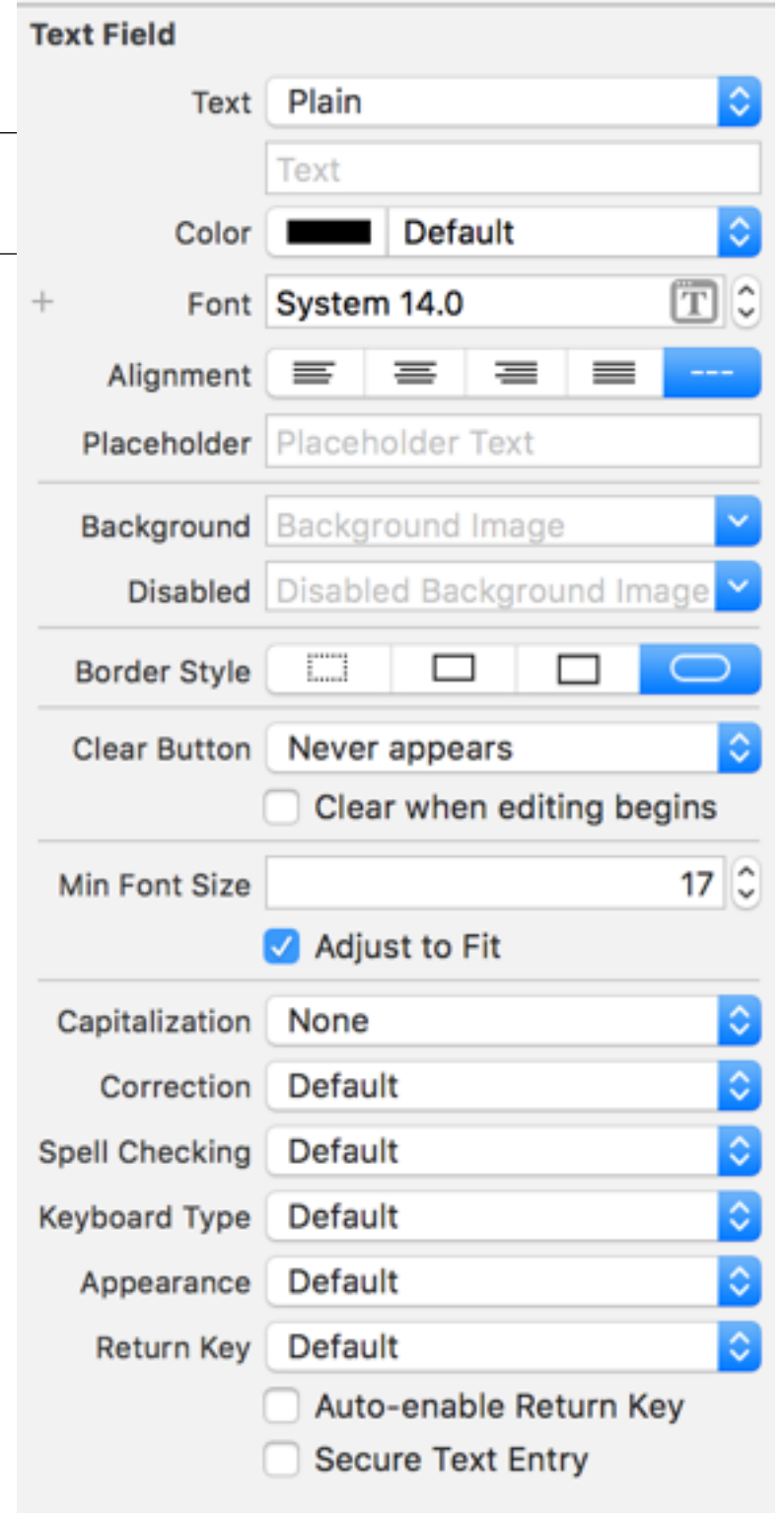




## INTRO TO VIEWS AND VIEW CONTROLLERS

# UITextField

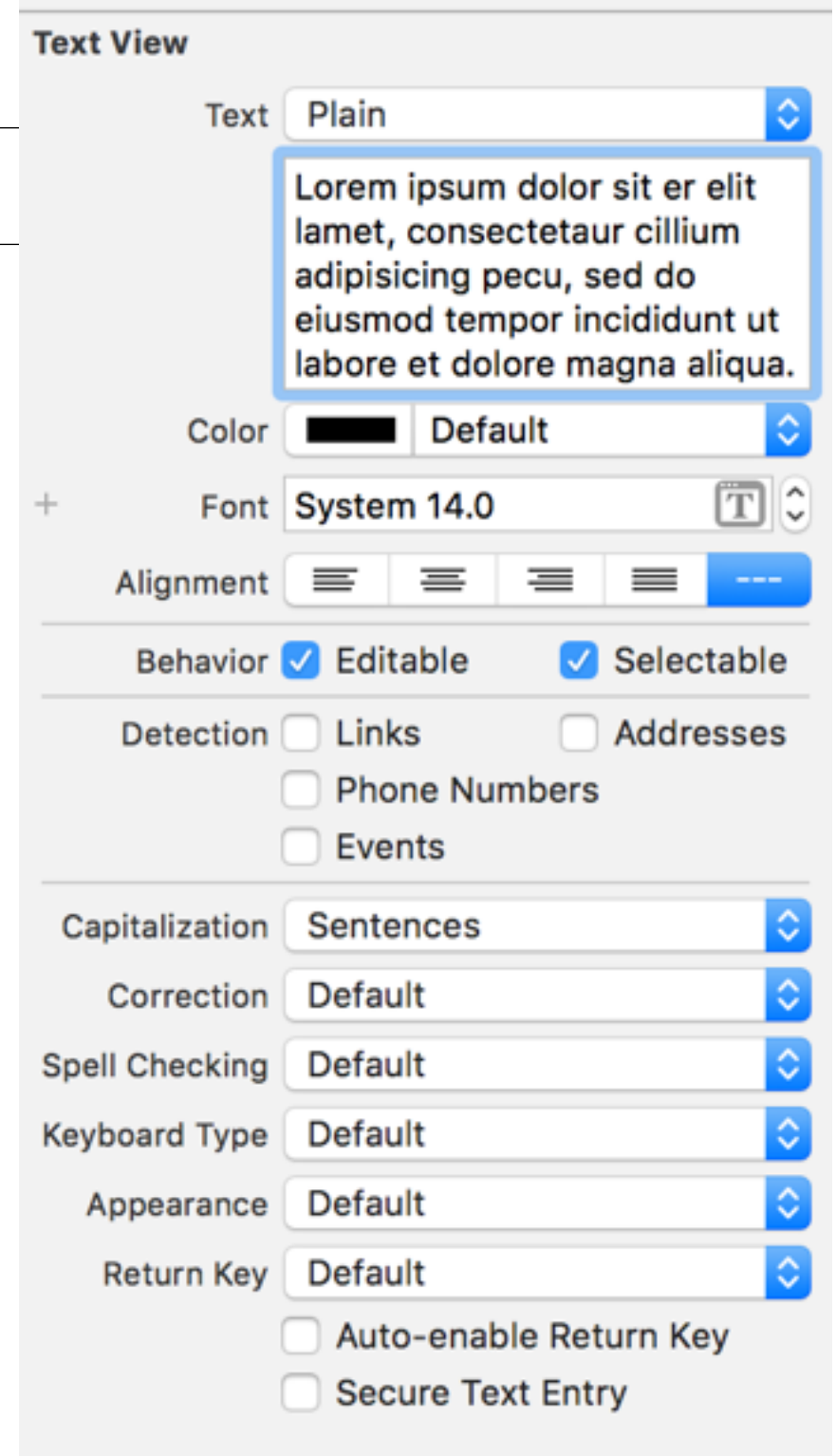
- A Text Field or UITextField provides a view for user input
- Can set the “Keyboard Type” to get a numbers only keyboard, for example



# INTRO TO VIEWS AND VIEW CONTROLLERS

## UITextView

- A text view or UITextView is a multiline editable text field
- Can also be non editable



## INTRO TO VIEWS AND VIEW CONTROLLERS

# UIImageView

- An image view or a UIImageView is a view used to display an image
- Mode is very important!

### Image View

Image

Image

Highlighted

Highlighted Image

State ☐ Highlighted

### View

Mode

Scale To Fill

Semantic

Unspecified

Tag

0

Interaction

☐

User Interaction Enabled

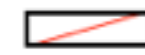
☐

Multiple Touch

Alpha

1

Background



Default

Tint



Default

Drawing ☒ Opaque

☐ Hidden

- TipCalculator M
  - TipCalculator
    - AppDelegate.swift
    - ViewController.swift M
    - SecondViewController.swift A
    - Main.storyboard M
    - Assets.xcassets M**
    - LaunchScreen.storyboard M
    - Info.plist
  - Products

- AppIcon
  - emoji-image**
  - LaunchImage



## INTRO TO VIEWS AND VIEW CONTROLLERS

# UIButton

- A button or UIButton allows for user interaction
- This can trigger segues or functions, which we'll cover next week

**Button**

Type System

State Config Default

Title Plain

Button

+ Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background Default Background Image

Shadow Offset 0 0  
Width Height

☐ Reverses On Highlight

Drawing ☐ Shows Touch On Highlight

☒ Highlighted Adjusts Image

☒ Disabled Adjusts Image

Line Break Truncate Middle

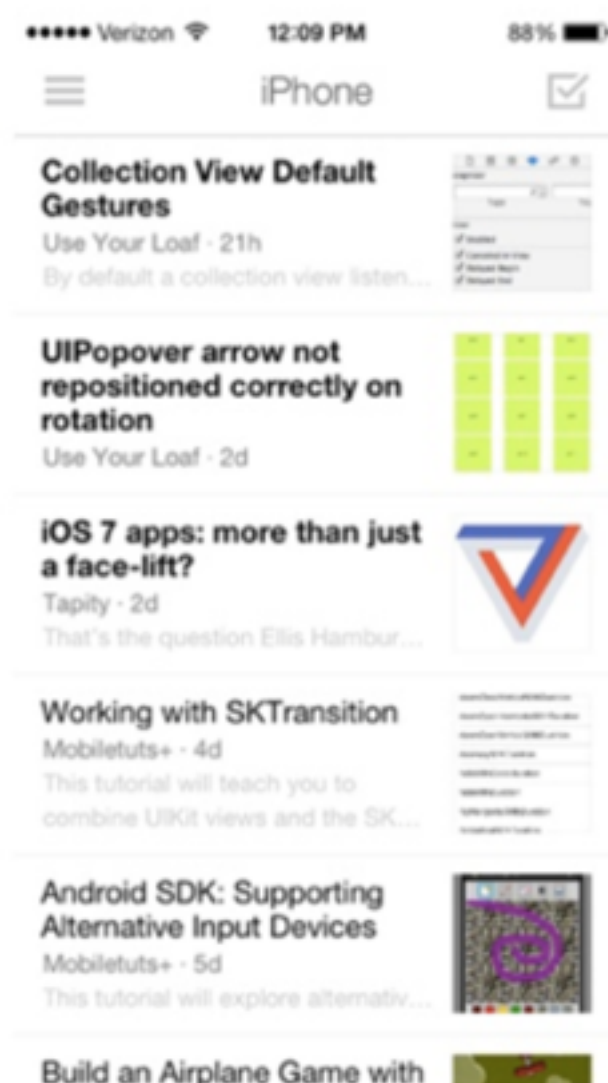
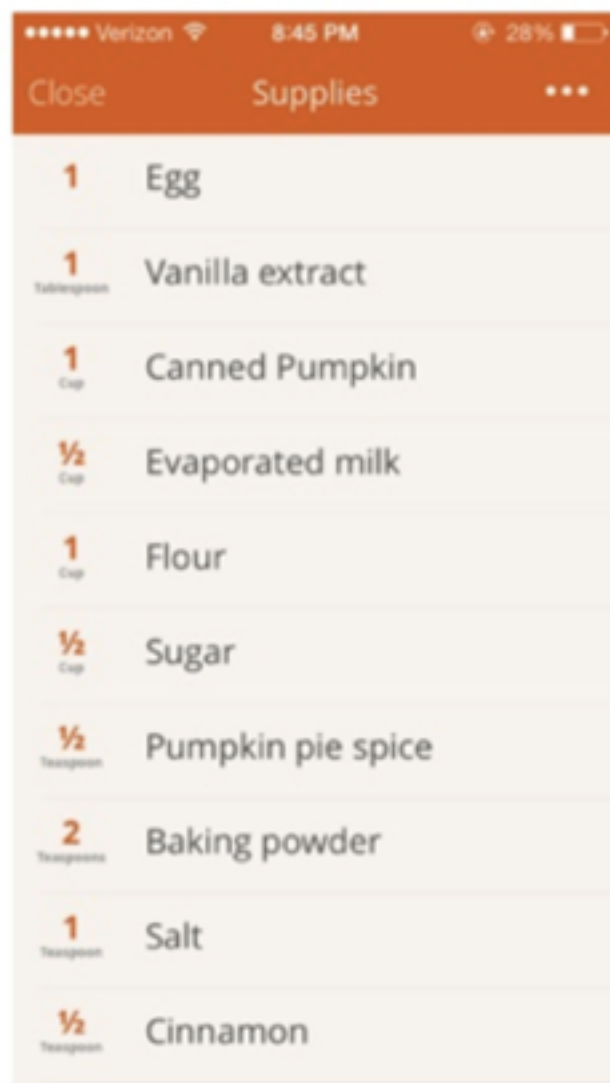
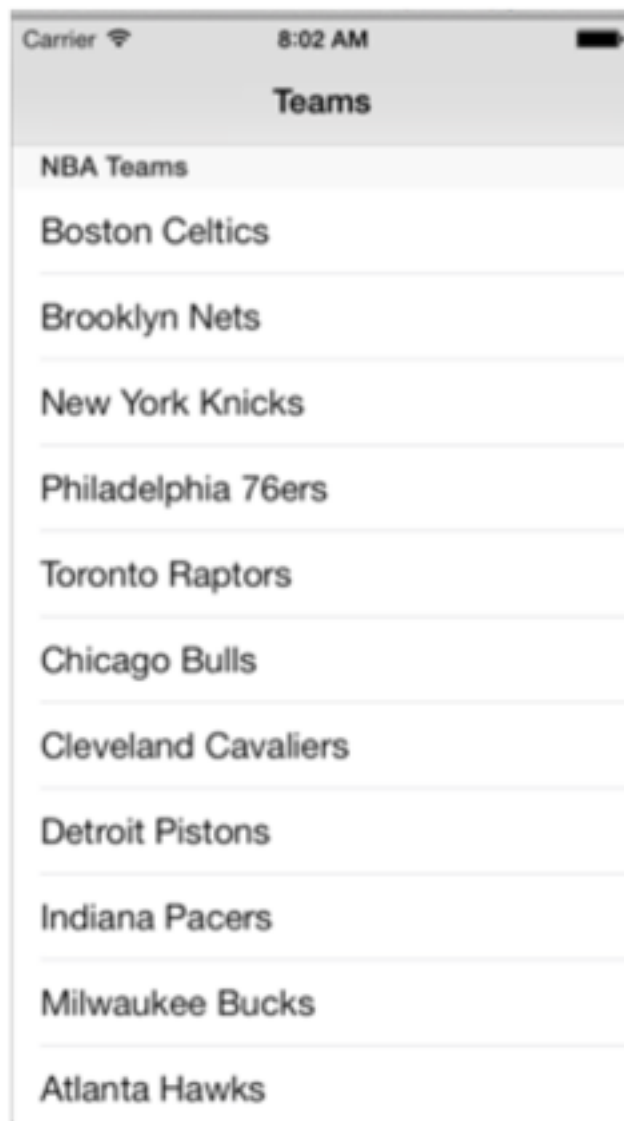
---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UITABLEVIEW

- A Table View or UITableView displays a scrollable list of data
- Almost every app I've worked on has used a UITableView
- Twitter, Facebook, Instagram, Settings, Messages, etc.
- Composed of sections that have rows
- Each row has a cell or a UITableViewCell, ONLY one cell per row
- Static content vs. Dynamic Content
  - Dynamic content is provided by code and can be any variable amount provided by an API, the user, etc.
  - Static content is predetermined





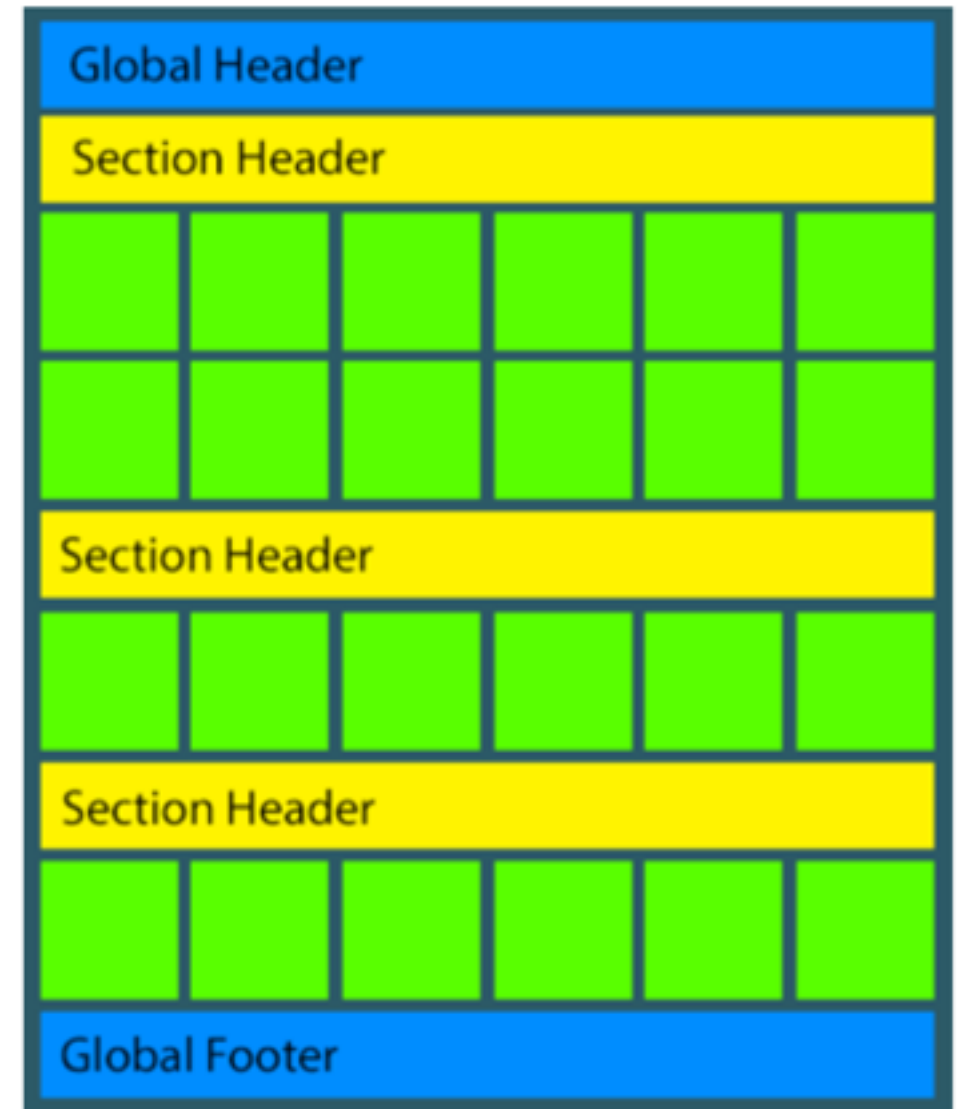
---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

# UICollectionView

- A Collection View or UICollectionView is similar to a UITableView (it can display a variable amount of data)
- However, unlike a UITableView a UICollectionView can have multiple items on a single line
- It can be in a grid form or really any kind of layout you want
- This is much more complex than a UITableView





---

**INTRO TO VIEWS AND VIEW CONTROLLERS**

---

# ASSESSMENT 1

---

## INTRO TO VIEWS AND VIEW CONTROLLERS

---

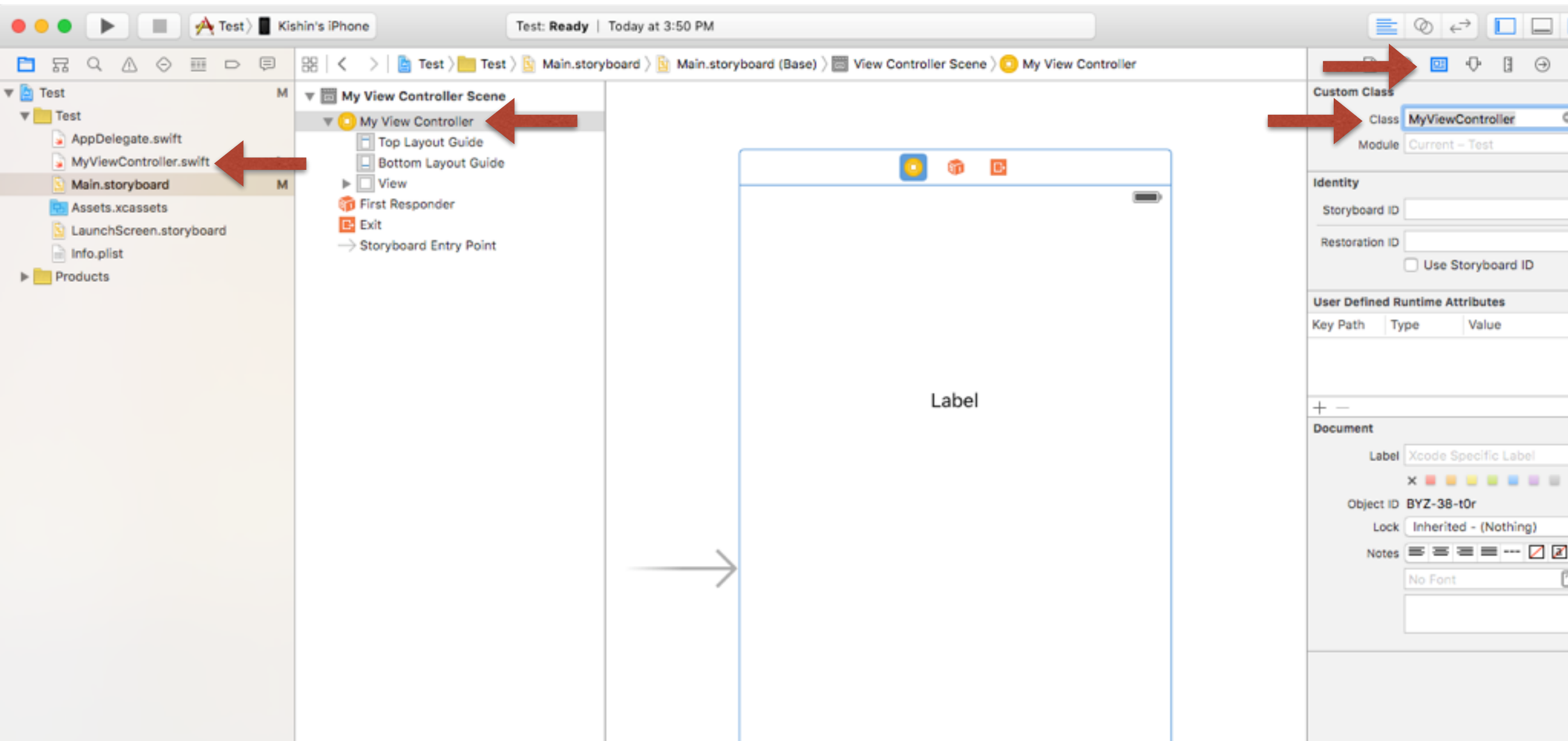
# Q&A

## INTRO TO VIEWS AND VIEW CONTROLLERS

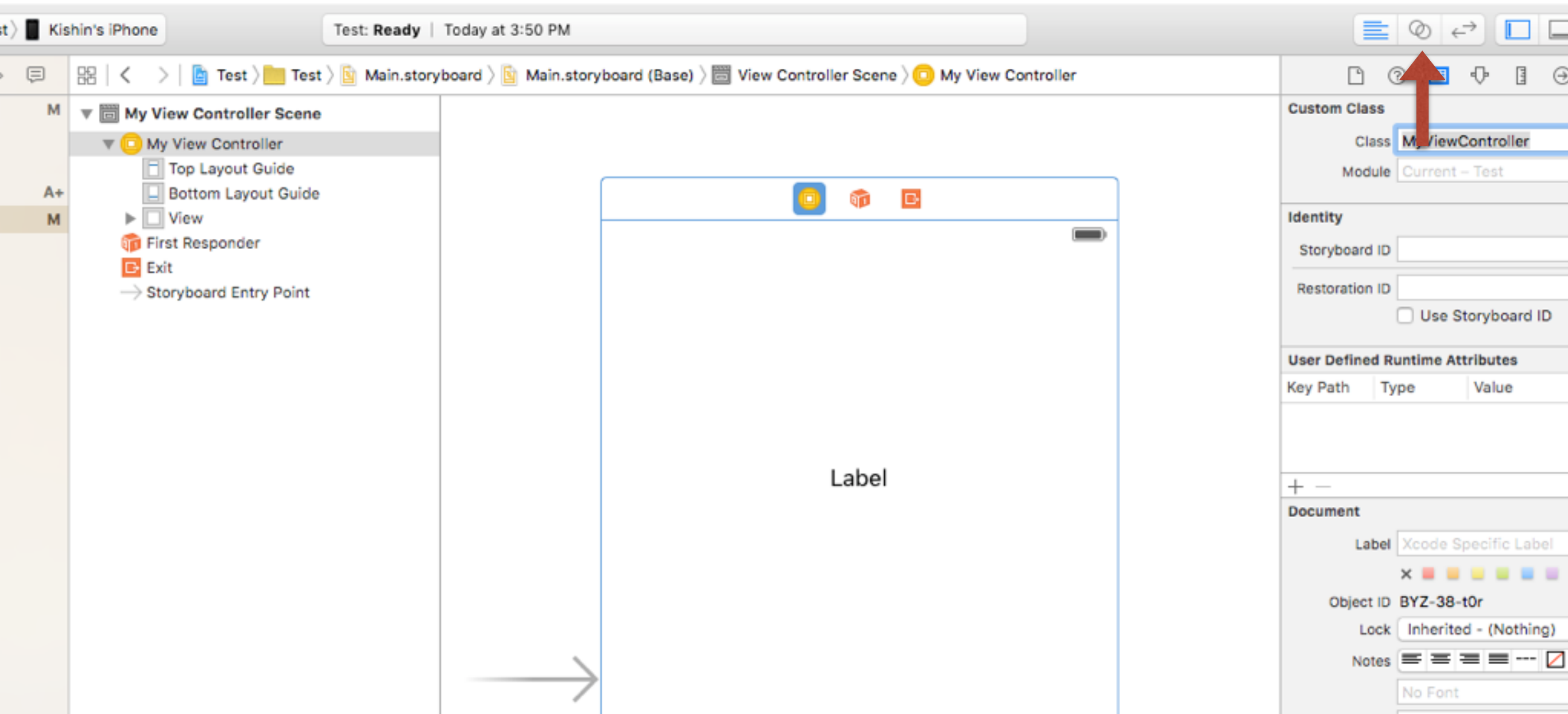
---

**“SMART”  
VIEWCONTROLLERS**

# This is no coincidence

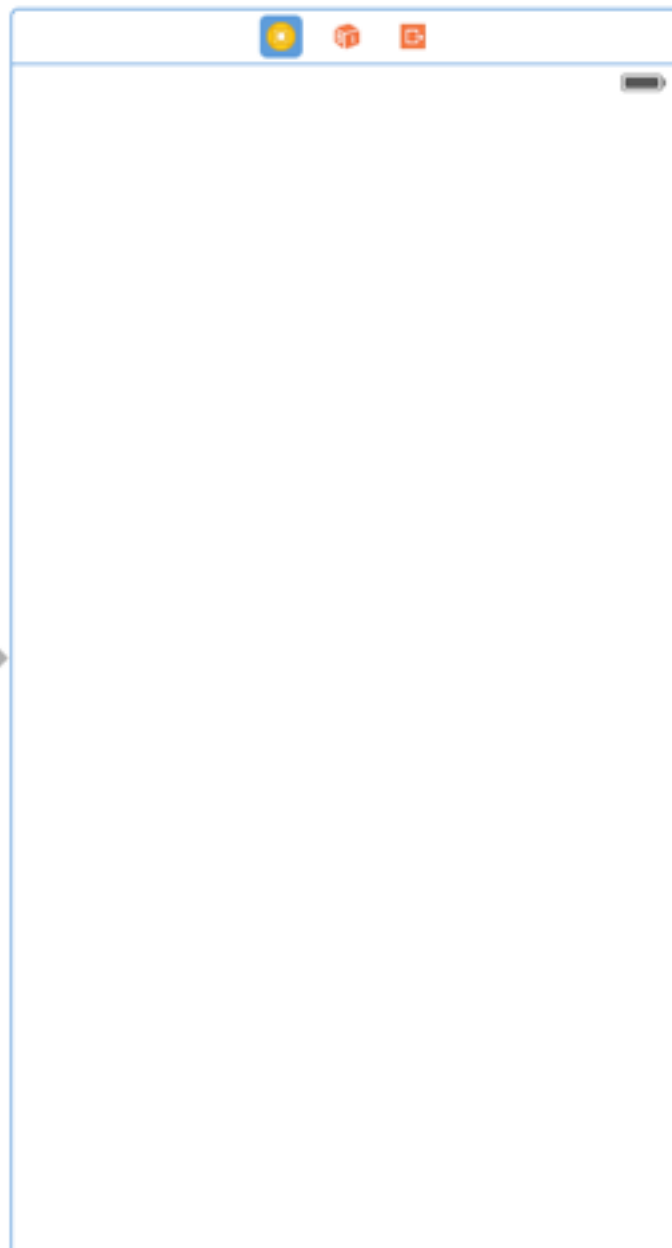


# Assistant Editor



View Controller Scene

- View Controller
  - Top Layout Guide
  - Bottom Layout Guide
  - View
  - First Responder
  - Exit
  - Storyboard Entry Point



```

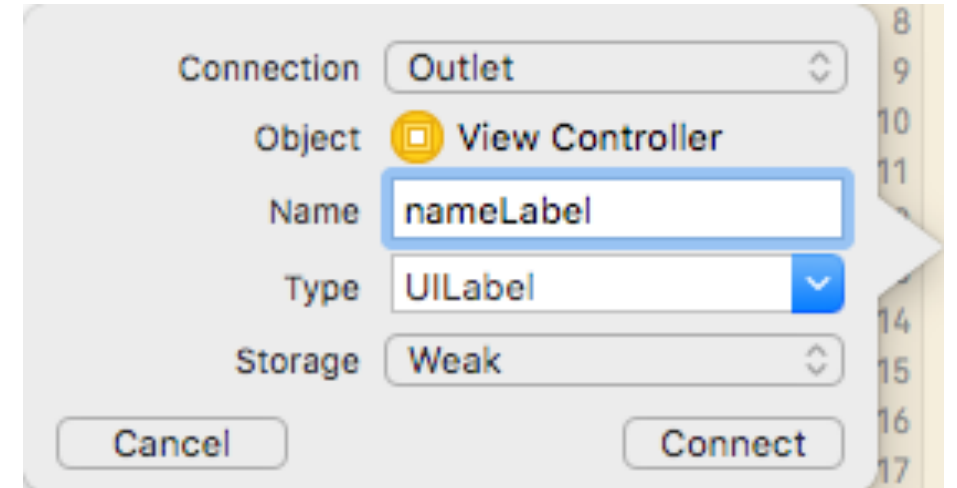
1 //
2 // ViewController.swift
3 // Test
4 //
5 // Created by Kishin Manglani on 10/5/15.
6 // Copyright © 2015 KM. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view, typically from a nib.
16     }
17
18     override func didReceiveMemoryWarning() {
19         super.didReceiveMemoryWarning()
20         // Dispose of any resources that can be recreated.
21     }
22
23 }
24
25
26

```

# INTRO TO VIEWS AND VIEW CONTROLLERS

## IBOUTLET

- IBOutlet or Interface Builder Outlet
- Create by control + dragging from our storyboard to our view controller file in the assistant editor
- This allows us to reference UIViews in interface builder in our code
- Best practice tip\*\*



# INTRO TO VIEWS AND VIEW CONTROLLERS

## IBACTION

- IBAction or Interface Builder Action
- Create by control + dragging from our storyboard to our view controller file in the assistant editor
- Typically used for UIButtons or Views that can trigger actions

