
IOS DEVELOPMENT WORKSHOP

Insert Instructor Name

Title, Company

IOS DEVELOPMENT

LEARNING OBJECTIVES

- Explain relevant history and trends in iOS development.
- Identify key skills leveraged by iOS developers.
- Survey the common tools used within the iOS ecosystem.
- Apply key concepts and skills to build your own basic iOS application.
- Create a custom learning plan to help you continue to build fundamental iOS development skills after this workshop.

INTRODUCTION

WHAT IS IOS?

INTRODUCTION

WHAT IS IOS?

- iOS is the operating system that powers the iPhone and iPad.
- It's a close cousin of macOS (until recently "Mac OS X"), tvOS, and watchOS.
- Learning the basics of iOS is a way not only to write apps for the Apple App Store, but to learn about building apps for other Apple technologies and devices.
- Programming for iOS means learning about Xcode, building user interfaces, and writing code.

INTRODUCTION

WHAT IS SWIFT?

- Swift is the new language of choice for iOS development.
- While Objective-C is over 30 years old, Swift is more concise and approachable than Objective-C.
- Here are two code snippets as a comparison:

INTRODUCTION

Objective-C

```
@interface User : NSObject {  
    NSString* firstName;  
    NSString* lastName;  
}  
@property (retain) NSString* firstName;  
@property (retain) NSString* lastName;  
@end
```

```
@implementation User
```

```
@synthesize firstName;  
@synthesize lastName;  
  
- (void) dealloc  
{  
    [firstName release];  
    [lastName release];  
    [super dealloc];  
}  
@end
```

Swift

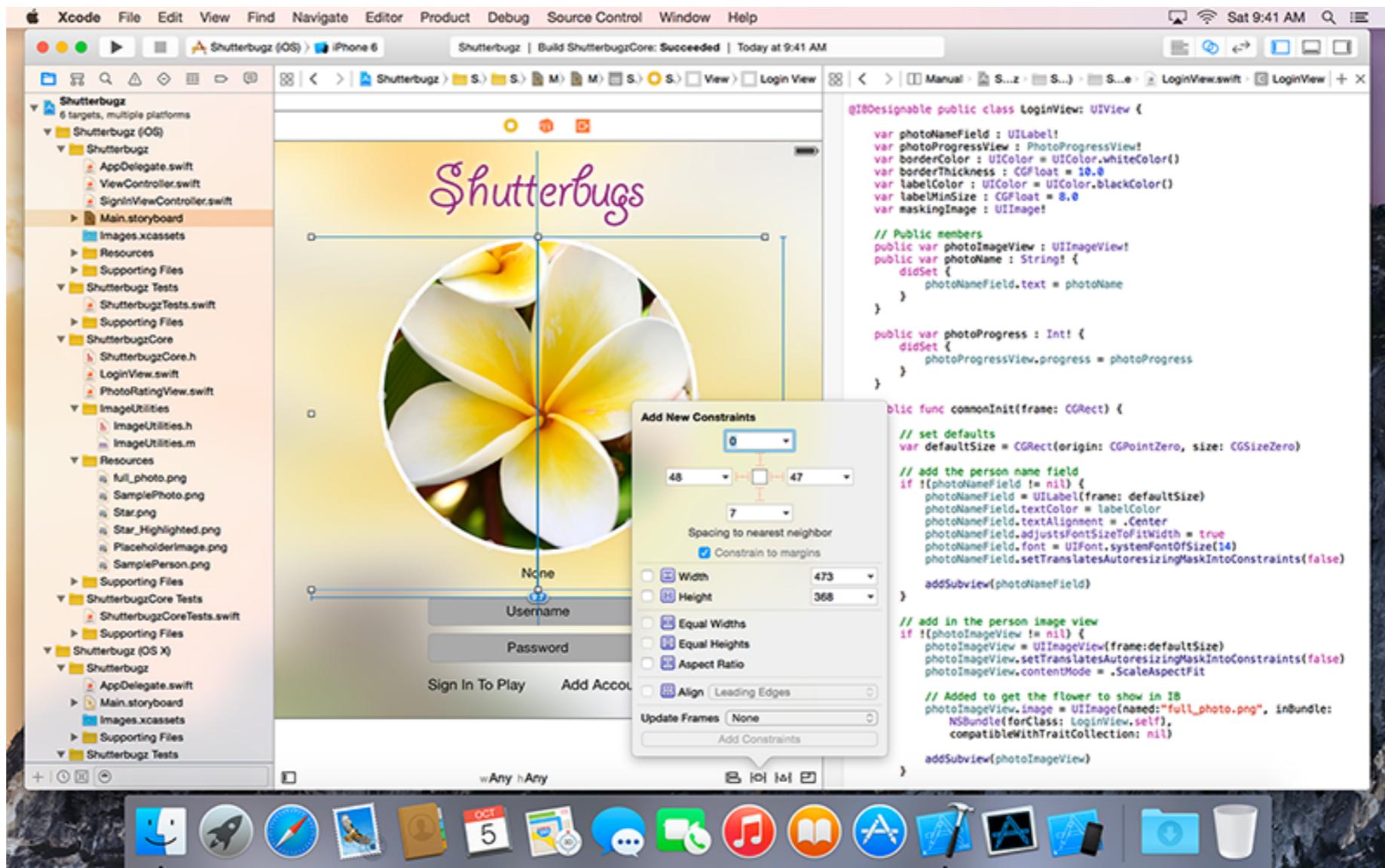
```
class User {  
    var firstName : String  
    var lastName : String  
}
```

INTRODUCTION

THE IOS ECOSYSTEM

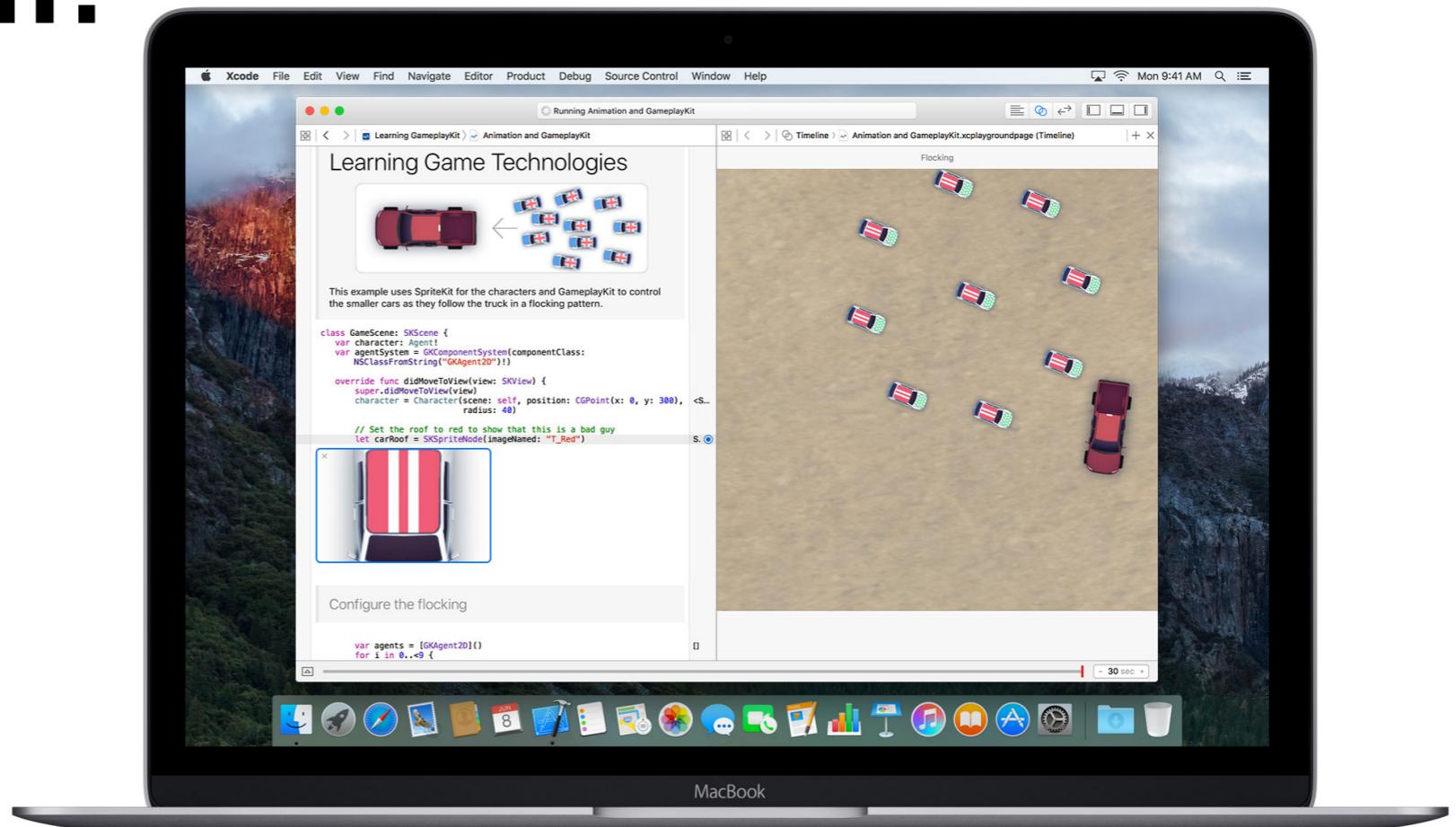
GETTING STARTED

XCODE



GETTING STARTED

INTERFACE BUILDERS, PLAYGROUNDS, SIMULATORS, OH MY!



INTRODUCTION

THE ECOSYSTEM

- Operating system (iOS) - Talks to the device. We build apps for the OS.
- Frameworks - Provided by Apple to make building apps easier. e.g. UIKit provides buttons, toolbars, ways to include images, etc.
- Third-party libraries and packages - Code from other developers that makes your life easier.

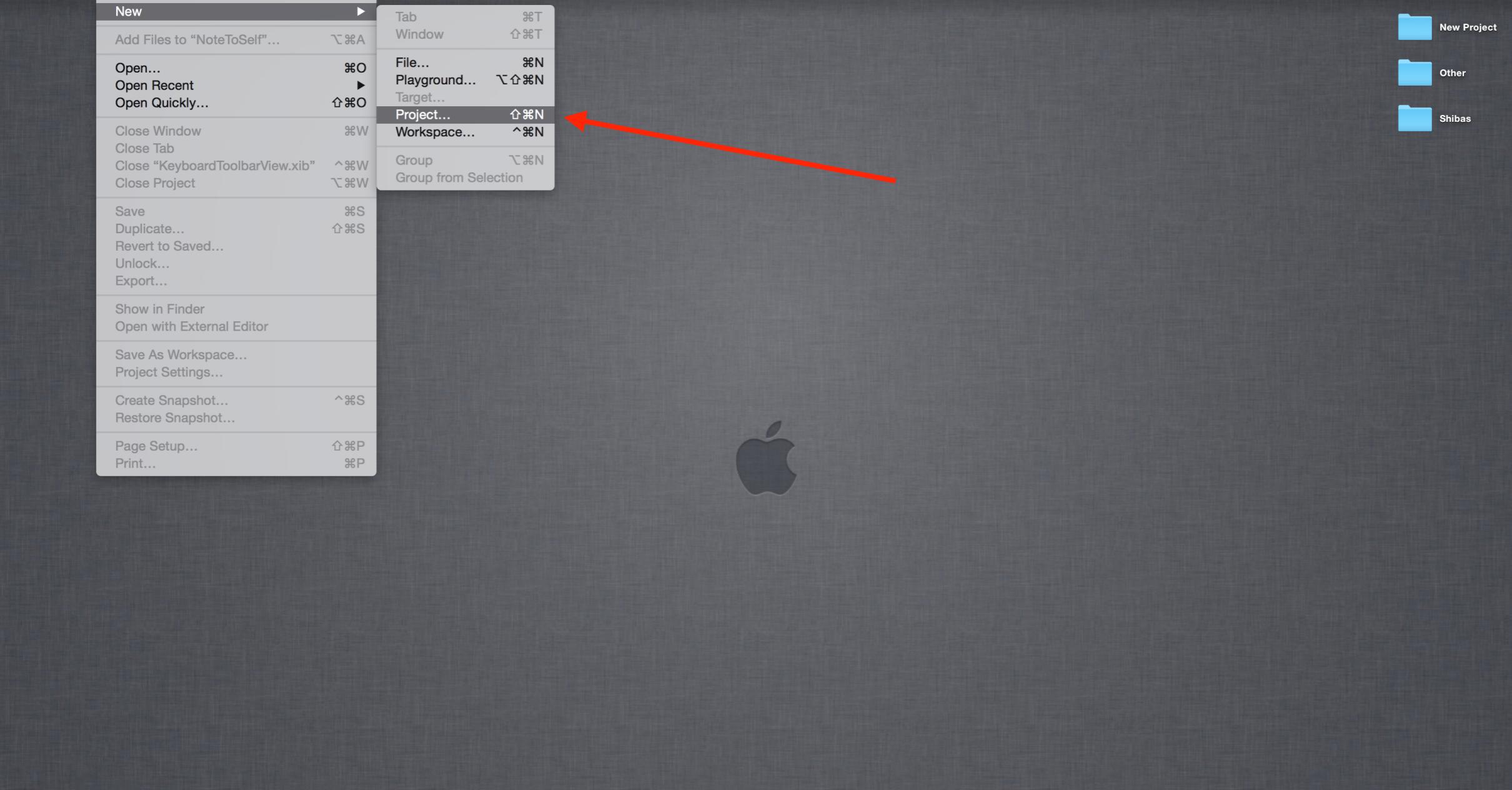
INTRODUCTION

WHAT GOES INTO AN APP?

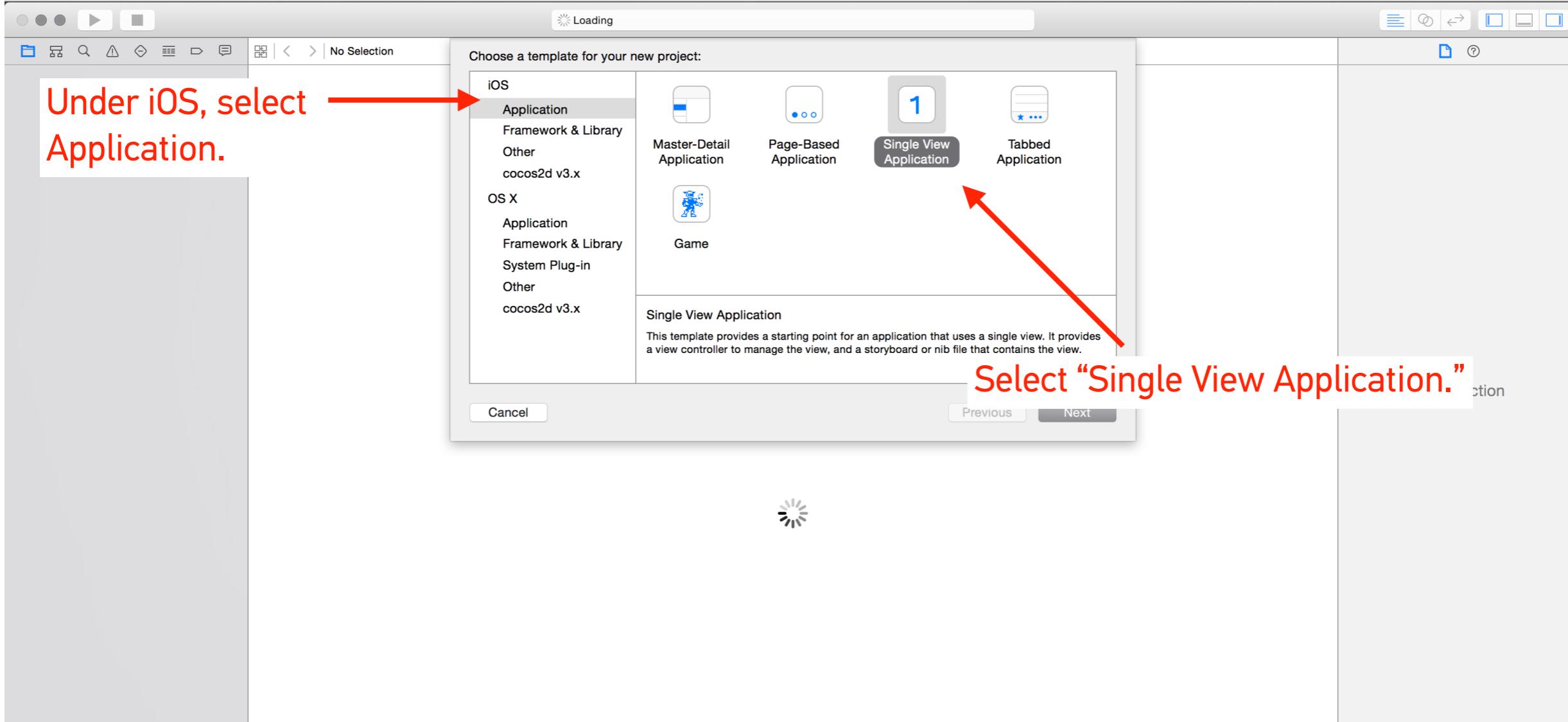
- User Interfaces - Built by composing "Views" together, provided by the UIKit framework.
- Code - Written in Swift.
- Tests - Code that ensures that the app code works well.
- Assets - Images, videos, sound files, etc.

CREATE AN XCODE PROJECT

XCODE PROJECTS

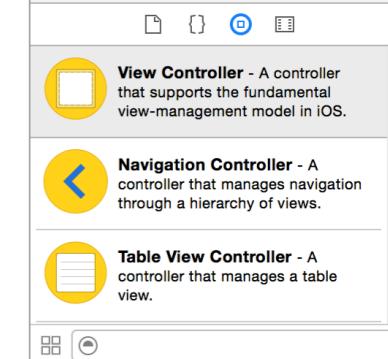


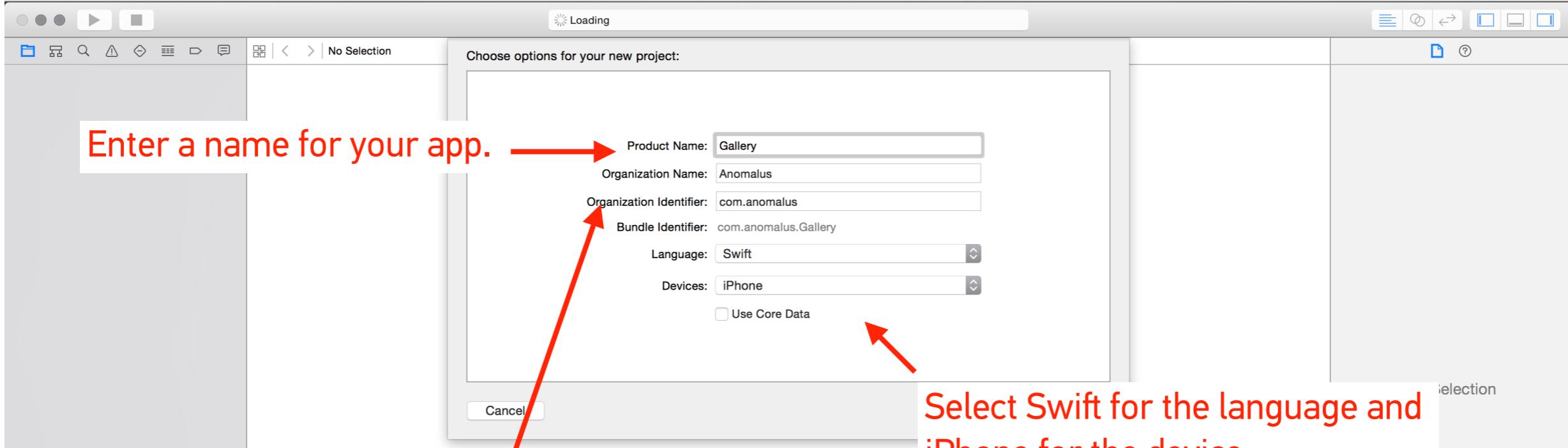
Start creating a project by opening Xcode and going to File > New > Project...



Under iOS, select Application. **Select “Single View Application.”**

There are several Application templates available. While they're intended to be convenient starting points for various kinds of apps, they turn out to be more trouble than they're worth. We'll always start with the Single View Application template.



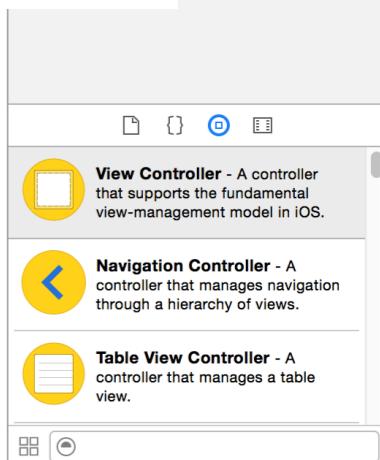


Enter a name for your app.

Enter an “Organization Name” and “Identifier.”
This helps Apple reference your app with a unique
name. The Identifier is of the form:
“com.OrganizationName”.

Select Swift for the language and
iPhone for the device.

Uncheck Use Core Data unless
you’re working on a project that
explicitly needs it.



Loading

No Selection

Working

Search

Favorites

- Recent
- Applications
- awmartin
- Dropbox
- General Assembly
- Google Drive
- iCloud Drive
- Desktop
- Documents
- Downloads
- Movies
- projects
- Projects - Current
- Working
- All My Files

_Scratch MOB 4

DogGallery

DogGallery App Originals

DogGallery App Slides

StreetEasyEmbed

No Selection

Source Control: Create Git repository on My Mac
Xcode will place your project under version control

Add to: Don't add to any project or workspace

New Folder Cancel Create

View Controller - A controller that supports the fundamental view-management model in iOS.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table view.

Pick a place to save your app.

Gallery > iPhone 6

Gallery: Ready | Today at 5:08 PM

General Capabilities Info Build Settings Build Phases Build Rules

Identity

Bundle Identifier: com.anomalous.Gallery
Version: 1.0
Build: 1
Team: None

Deployment Info

Deployment Target: 8.4
Devices: iPhone
Main Interface: Main
Device Orientation: Portrait
 Upside Down
 Landscape Left
 Landscape Right
Status Bar Style: Default
 Hide status bar

App Icons and Launch Images

App Icons Source: AppIcon
Launch Images Source: Use Asset Catalog
Launch Screen File: LaunchScreen

Embedded Binaries

Add embedded binaries here

Linked Frameworks and Libraries

Name Status

Add frameworks & libraries here

Identity and Type

Name: Gallery
Location: Absolute
Full Path: /Users/awmartin/Working/General Assembly/Working/Gallery/Gallery.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible
Organization: Anomalous
Class Prefix:

Text Settings

Indent Using: Spaces
Widths: Tab 4 Indent 4
 Wrap lines

Source Control

Repository: --
Type: --
Current Branch: --
Version: --
Status: No changes
Location:

View Controller - A controller that supports the fundamental view-management model in iOS.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table view.

You should be presented with a screen like this.

GETTING STARTED

XCODE TOUR

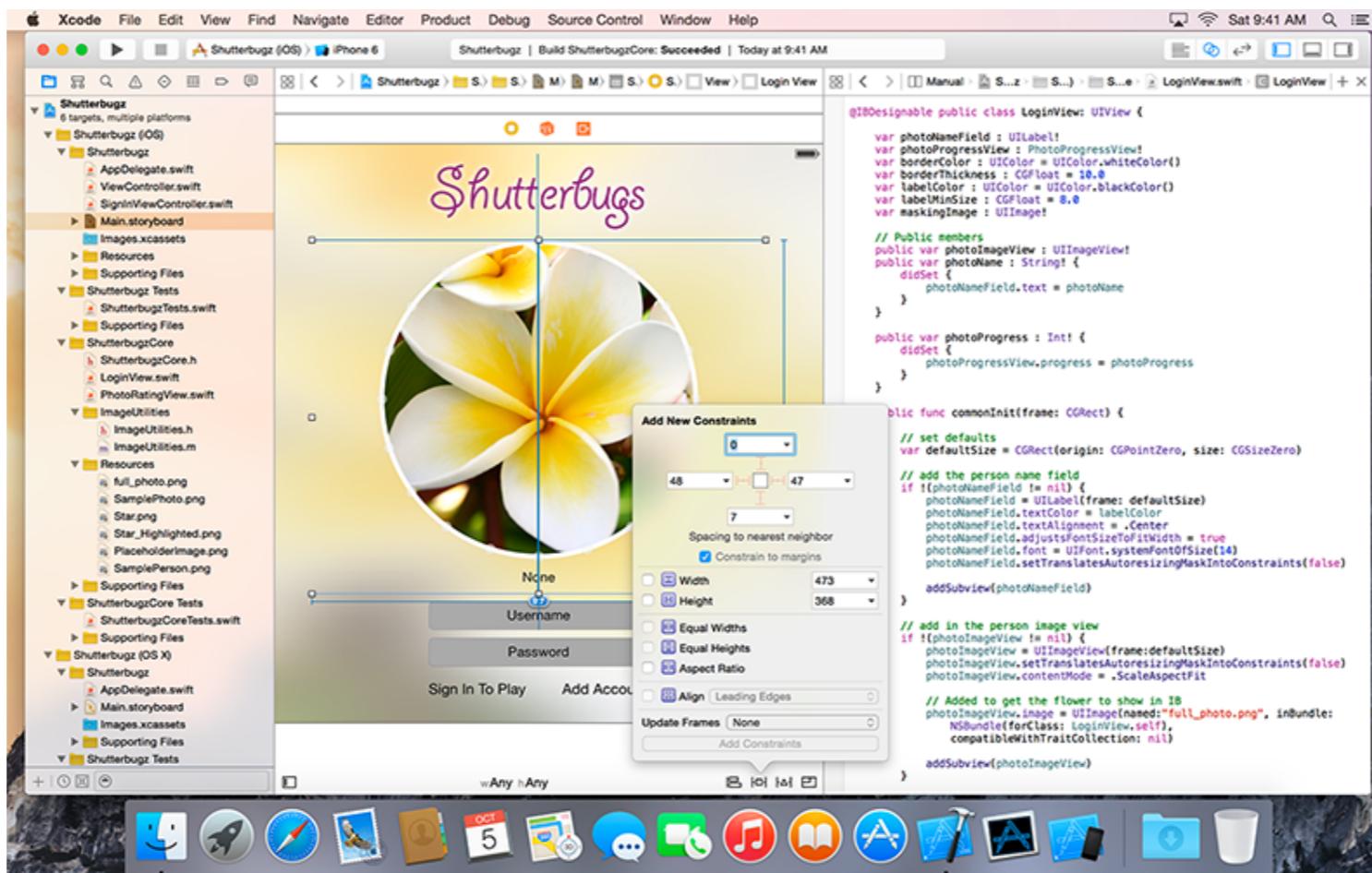
GETTING STARTED

DEV WORKFLOW OVERVIEW

- Launch Xcode
- Create new project
- Briefly discuss the different project templates
- Add user interface elements to project
- Change user interface element properties
- Build / run the app / test it
- Iterate
- Publish

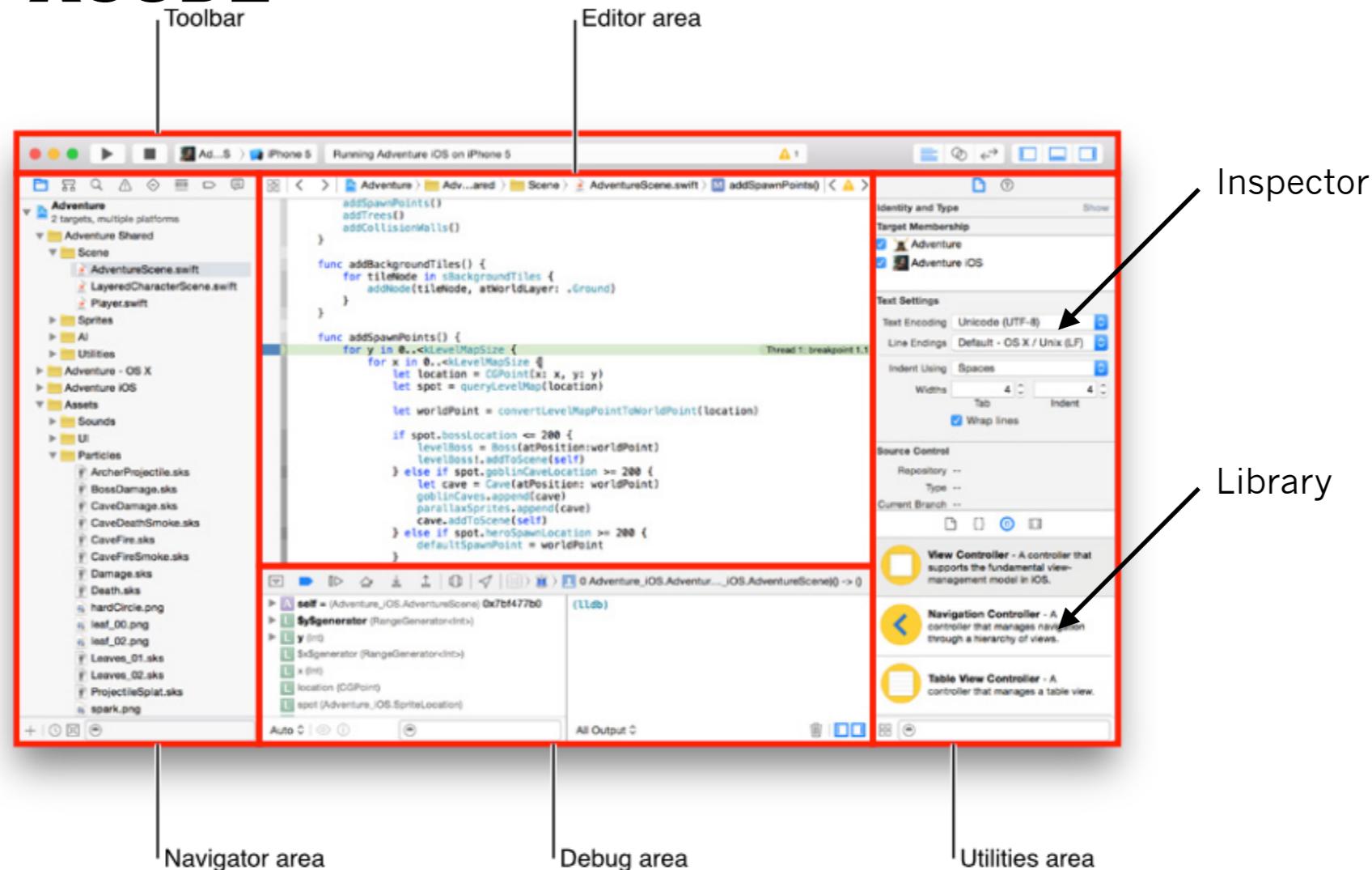
GETTING STARTED

WHAT IS XCODE?



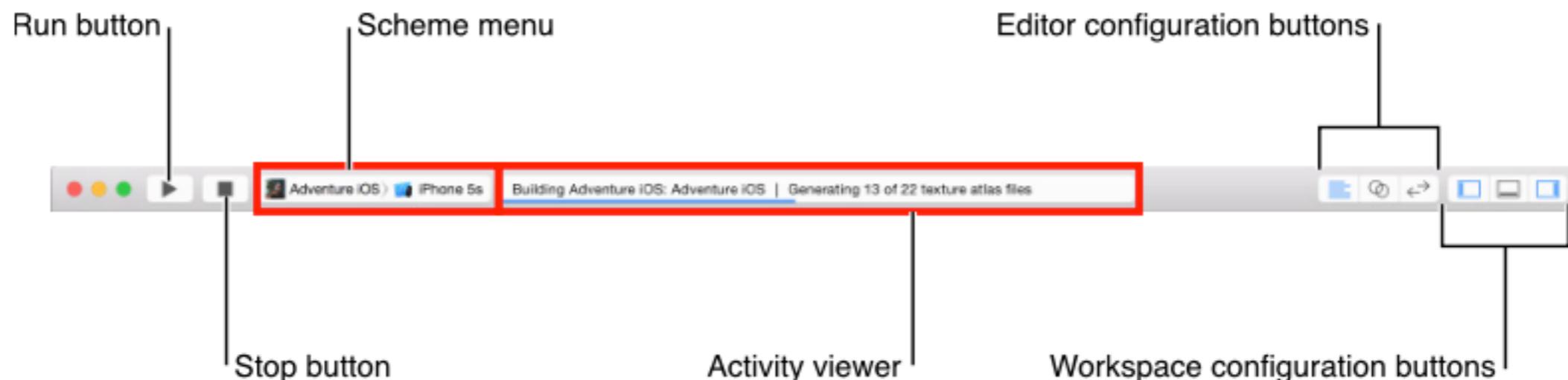
GETTING STARTED

NAVIGATING XCODE



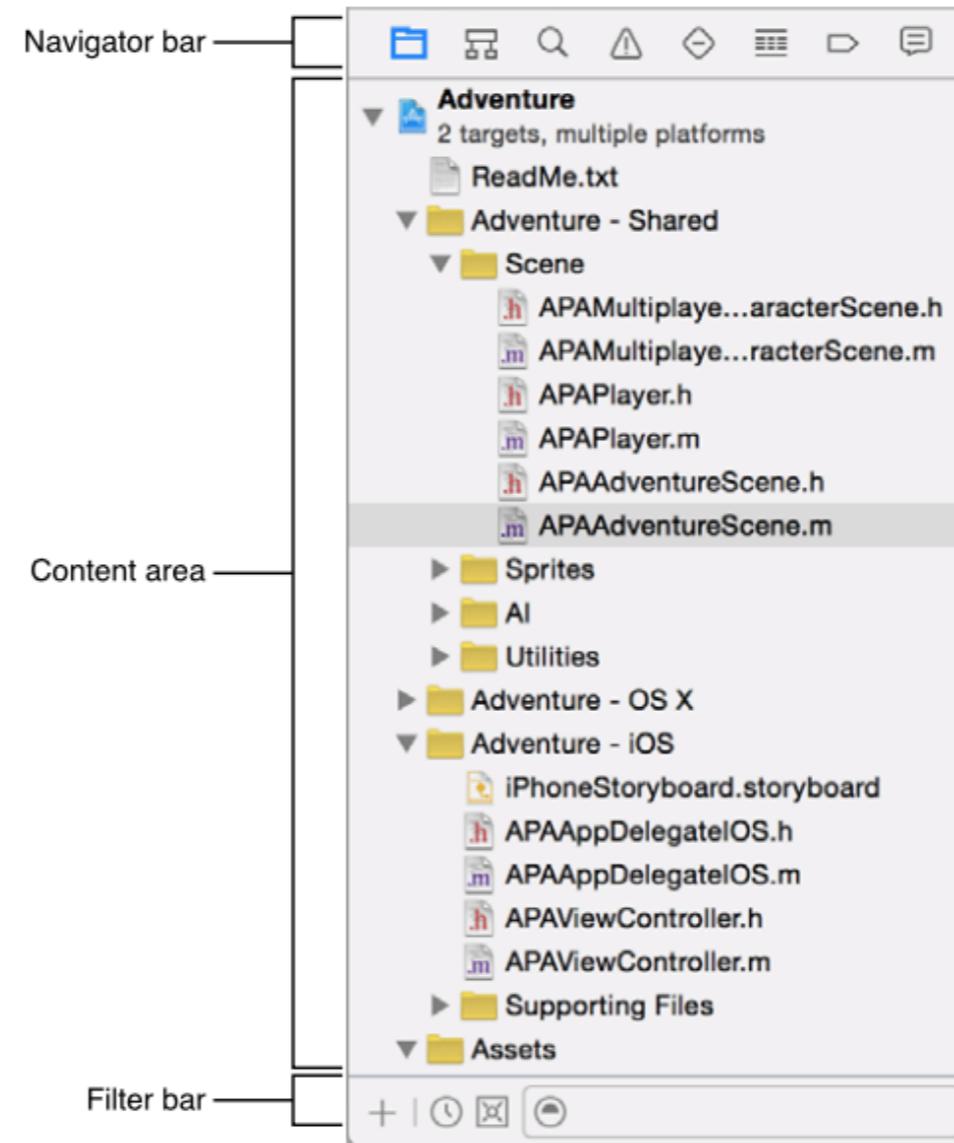
GETTING STARTED

WORKSPACE TOOLBAR



GETTING STARTED

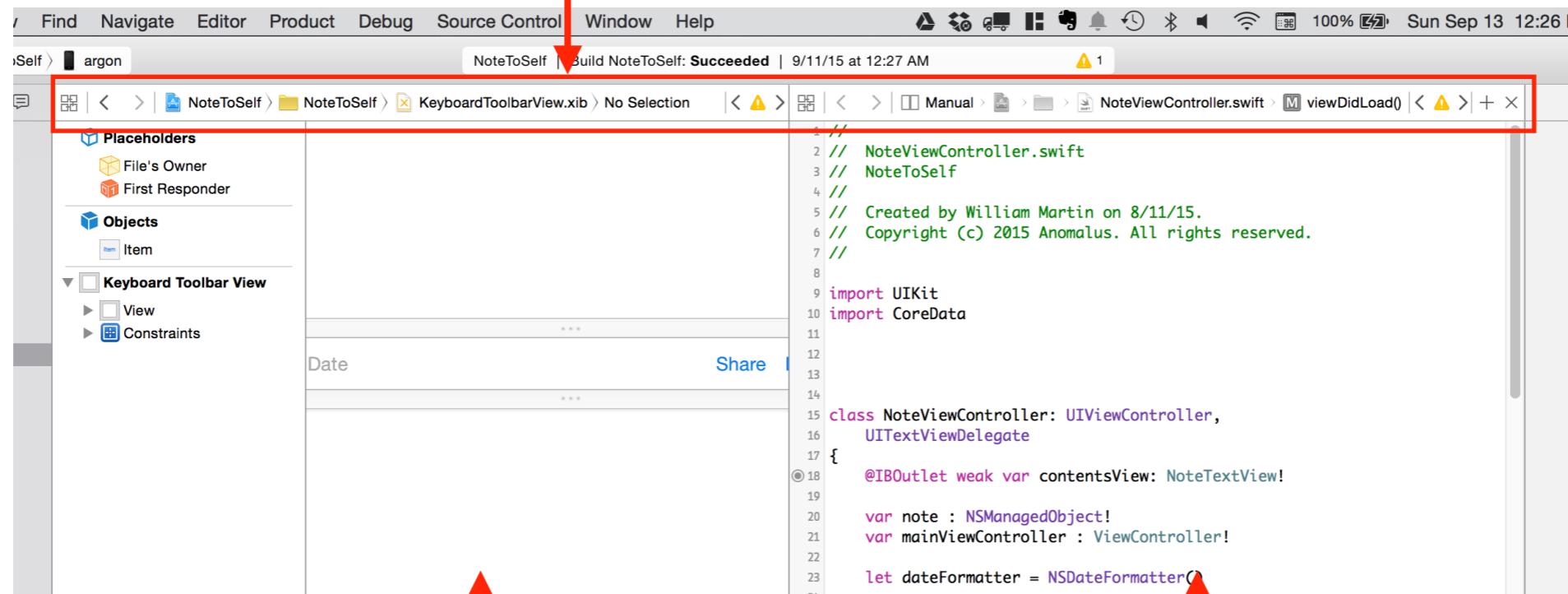
NAVIGATOR AREA



GETTING STARTED

JUMP BAR AND EDITOR PANES

Jump bars

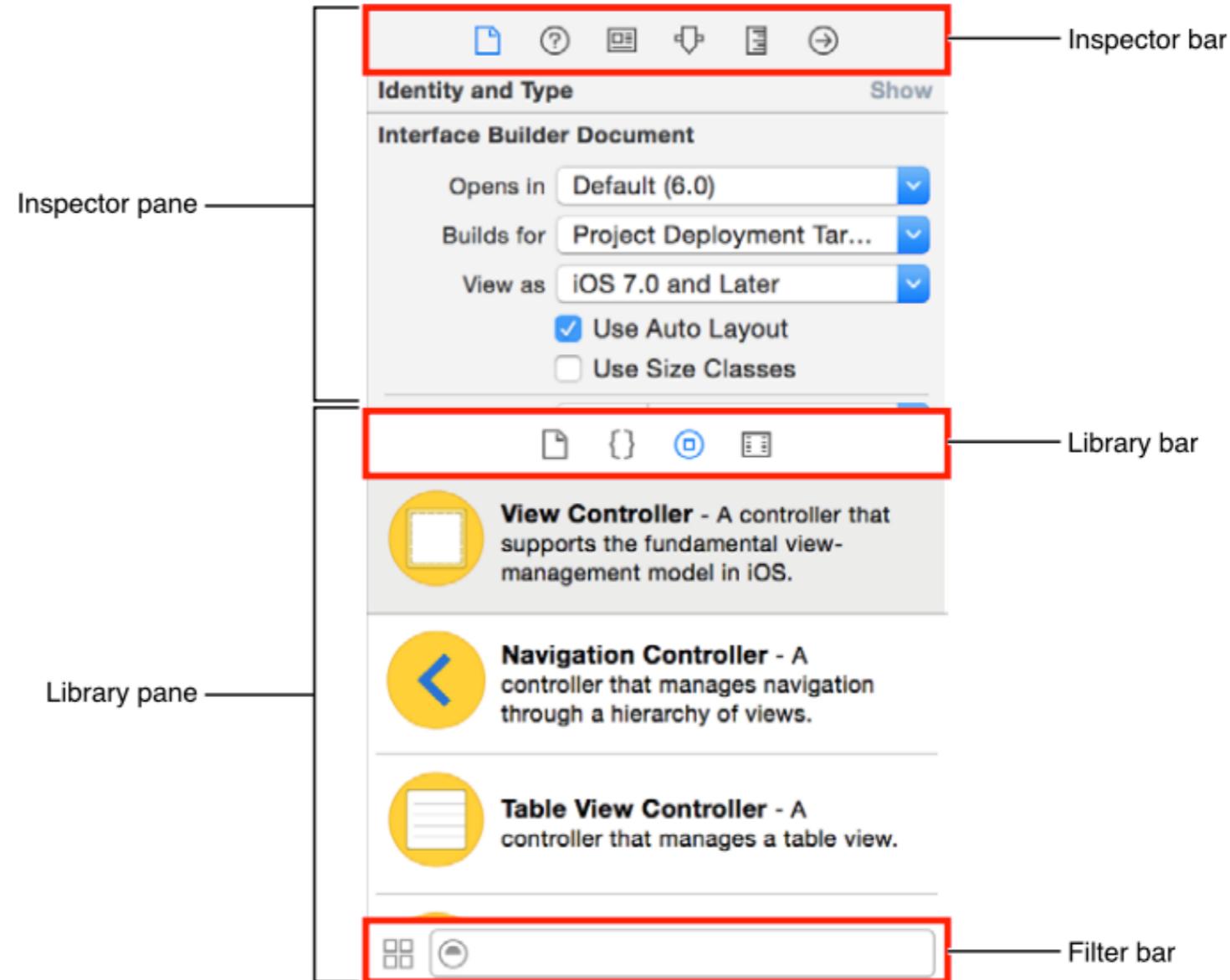


Editor pane

Assistant editor pane

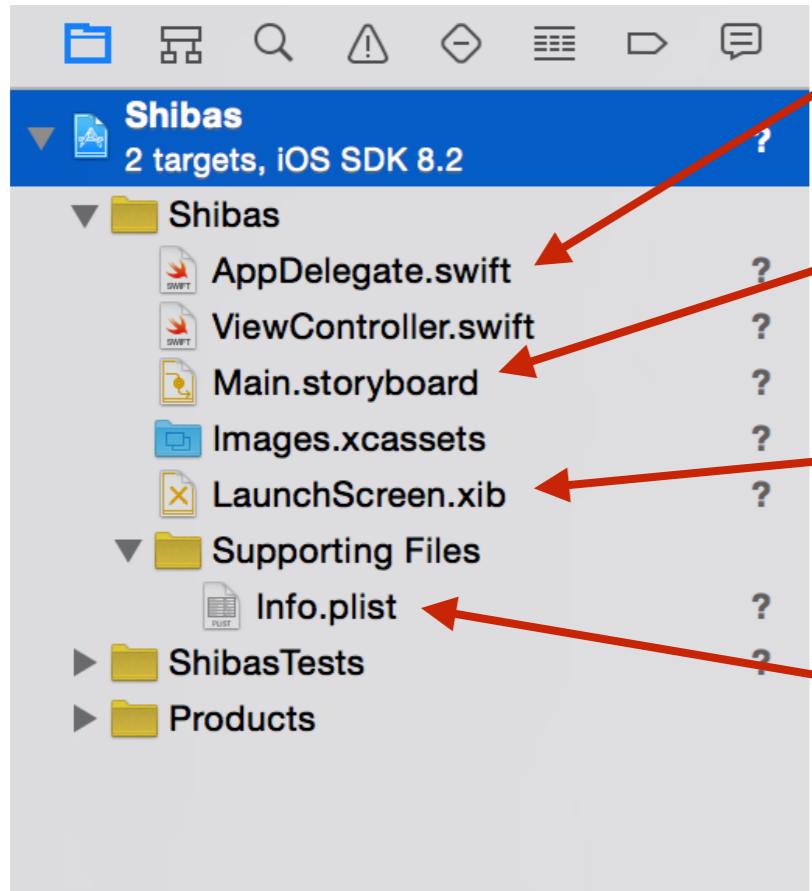
GETTING STARTED

UTILITIES AREA



GETTING STARTED

FILE TYPES



- .swift
Swift source code file
- .storyboard
an Interface Builder “Storyboard” file
- .xib
an Interface Builder “NIB” file
- .plist
a “property list”

GETTING STARTED

RUNNING YOUR APP

GETTING STARTED

RUNNING YOUR APP

- Run app on simulator (Cmd + R or click Play)
- BONUS: Outline the steps for deploying an app to device

GETTING STARTED

RUNNING AN APP IN THE IOS SIMULATOR

- Select iOS version in toolbar area
- Select “Build and then Run” in toolbar area (⌘R)

Note

- iPad apps only run on iPad simulator
- iPhone and universal apps run on both iPad and iPhone simulators

GETTING STARTED

NAVIGATING THE IOS SIMULATOR

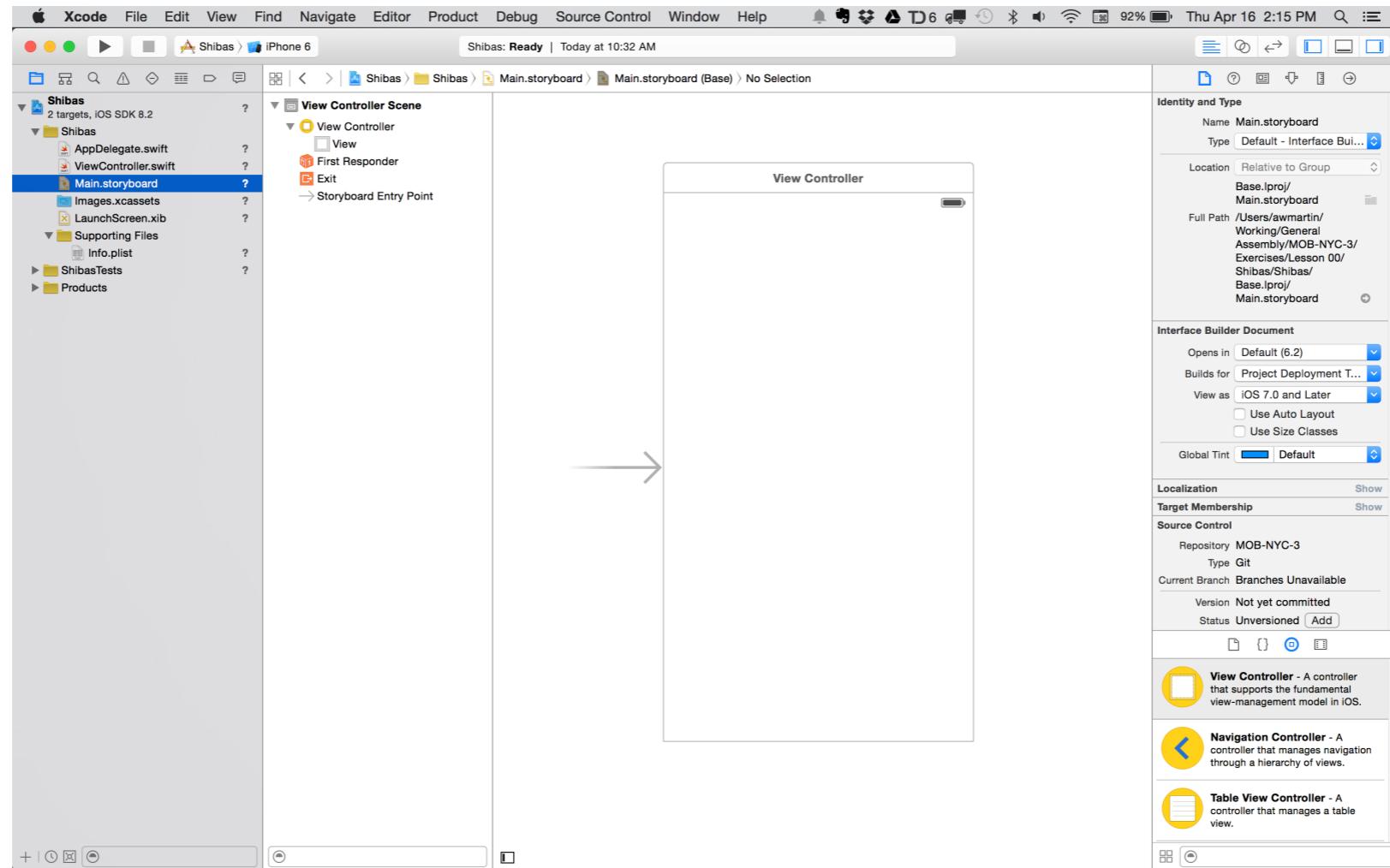
- › To run Simulator without running a project, select:
Xcode -> Open Developer Tool -> iOS Simulator
- › To select the “Home” button on simulator press $\text{⌘} + \text{↑} + \text{H}$.

GETTING STARTED

GETTING THINGS ON THE SCREEN

GETTING STARTED

INTERFACE BUILDER



GETTING STARTED

GETTING VIEWS ON SCREEN

- › To start understanding iOS apps, we'll first tackle 'views' on the screen
- › Almost everything we see on screen is a **view**.
- › There are lots of kinds of views:
 - › Buttons, labels, tables, images, etc
- › There are several ways to lay things out on screen, we'll cover these later in class
- › Until then, our views may look a little misaligned.

GETTING STARTED

VIEWS ON THE WHITEBOARD

USER INTERFACES

BUILDING USER INTERFACES

USER INTERFACES

WHAT IS COCOA TOUCH?



- Cocoa Touch contains key “frameworks” for building iOS apps. These frameworks are pre-built mechanisms for the appearance and behavior of iPhone apps.
- They also provide the basic app infrastructure and support for key technologies such as multitasking, touch-based input, push notifications, and many system services.
- When designing your apps, you should investigate the technologies here first to see if they meet your needs.
- [From the Apple docs here.](#)

USER INTERFACES

HOW DOES XCODE ENABLE YOU TO BUILD UI'S?

- A tool called "Interface Builder" that enables you to build "Storyboards."
- A Storyboard is a file that enables you to compose:
 - multiple Scenes,
 - the transitions between them, and
 - their respective Views
- all in a single place.

USER INTERFACES

MORE ON VIEWS

- A View is a UI element, typically rectangular, drawn at some location (or “position”) with a size.
- Views can “contain” other views (i.e. “subviews”).
- Most views have some level of interactivity (e.g. scrolling, detecting taps or other gestures, etc.).
- You can also develop your own Views.

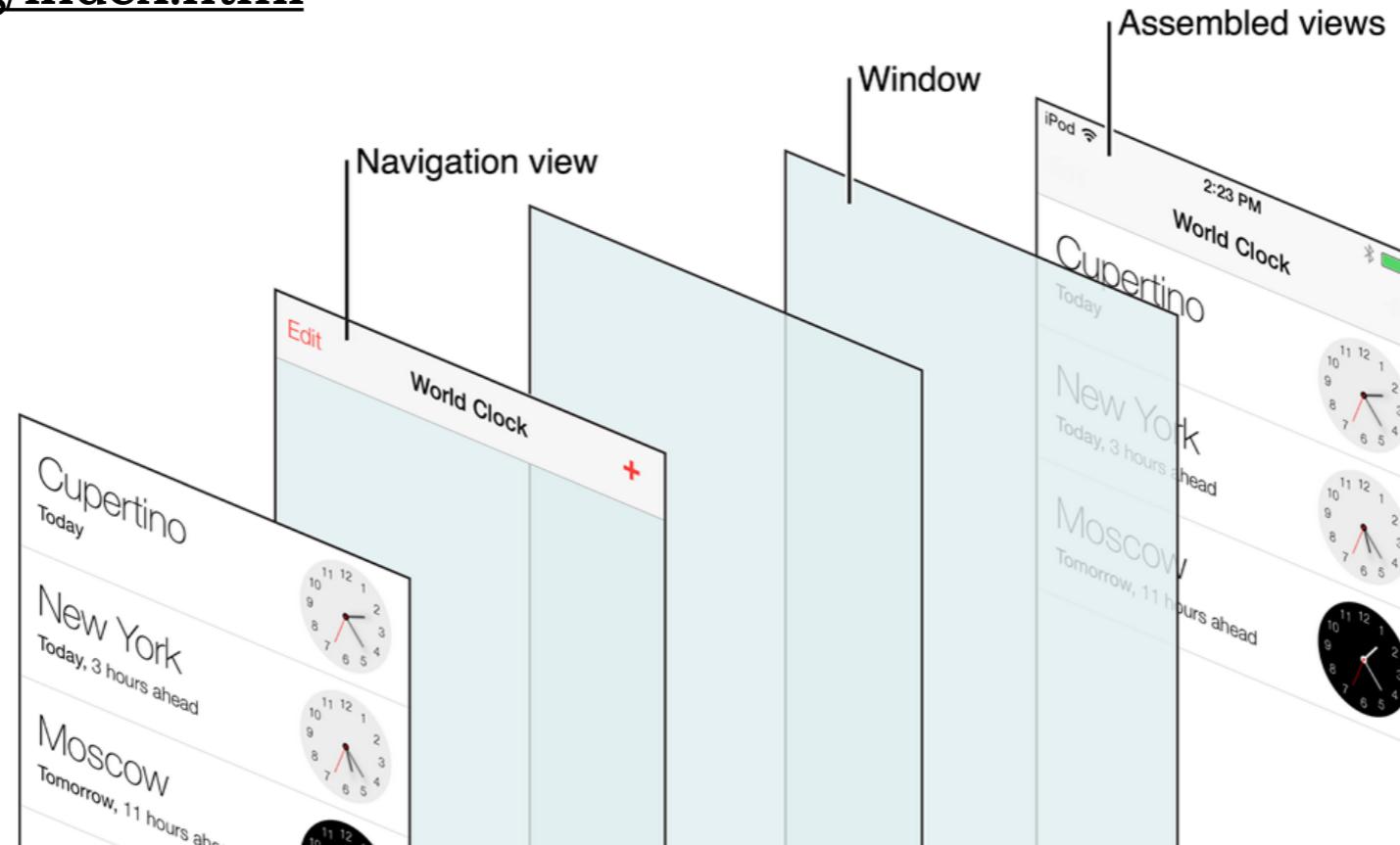
 View Controller - A controller that supports the fundamental view-management model in iOS.	 Text Field - Displays editable text and sends an action message to a target object when Return is tapped.	 Text View - Displays multiple lines of editable text and sends an action message to a target object when Return is tapped.	 Rotation Gesture Recognizer - Provides a recognizer for rotation gestures which are invoked on the view.	 Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.
 Navigation Controller - A controller that manages navigation through a hierarchy of views.	 Slider - Displays a continuous range of values and allows the selection of a single value.	 ScrollView - Provides a mechanism to display content that is larger than the size of the application's window.	 Swipe Gesture Recognizer - Provides a recognizer for swipe gestures which are invoked on the view.	 Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.
 Table View Controller - A controller that manages a table view.	 Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle the value.	 DatePicker - Displays multiple rotating wheels to allow users to select dates and times.	 Pan Gesture Recognizer - Provides a recognizer for panning (dragging) gestures which are invoked on the view.	 View - Represents a rectangular region in which it draws and receives events.
 Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.	 Activity Indicator View - Provides feedback on the progress of a task or process of unknown duration.	 Picker View - Displays a spinning-wheel or slot-machine motif of values.	 Screen Edge Pan Gesture Recognizer - Provides a recognizer for panning (dragging) gestures which are invoked on the view and sta...	 Container View - Defines a region of a view controller that can include a child view controller.
 Split View Controller - A composite view controller that manages left and right view controllers.	 Progress View - Depicts the progress of a task over time.	 Visual Effect View - Provides a blur effect	 Long Press Gesture Recognizer - Provides a recognizer for long press gestures which are invoked on the view.	
 Page View Controller - Presents a sequence of view controllers as pages.	 Page Control - Displays a dot for each open page in an application and supports sequential navigation through the pages.	 Visual Effect Views with Blur and Vibrancy - Provides a blur effect, plus vibrancy for nested views	 Navigation Bar - Provides a mechanism for displaying a navigation bar just below the status bar.	
 GLKit View Controller - A controller that manages a GLKit view.	 Stepper - Provides a user interface for incrementing or decrementing a value.	 MapKit View - Displays maps and provides an embeddable interface to navigate map content.	 Navigation Item - Represents a state of the navigation bar, including a title.	
 Object - Provides a template for objects and controllers not directly available in Interface Builder.	 Table View - Displays data in a list of plain, sectioned, or grouped rows.	 GLKit View - Provides a default implementation of an OpenGL ES-aware view.	 Toolbar - Provides a mechanism for displaying a toolbar at the bottom of the screen.	
 Collection View Controller - A controller that manages a collection view.	 Table View Cell - Defines the attributes and behavior of cells (rows) in a table view.	 iAd BannerView - The ADBannerView class provides a view that displays banner advertisements to the user.	 Item - Represents an item on a UIToolbar or UINavigationItem object.	
 AVKit Player View Controller - A view controller that manages a AVPlayer object.	 Image View - Displays a single image, or an animation described by an array of images.	 SceneKit View - A view for displaying a 3D scene.	 Tab Bar - Provides a mechanism for displaying tabs at the bottom of the screen.	
 Label Label - A variably sized amount of static text.	 Collection View - Displays data in a collection of cells.	 Web View - Displays embedded web content and enables content navigation.	 Tab Bar Item - Represents an item on a UITabBar object.	
 Button - Intercepts touch events and sends an action message to a target object when it's tapped.	 Collection View Cell - Defines the attributes and behavior of cells in a collection view.	 Tap Gesture Recognizer - Provides a recognizer for tap gestures which land on the view.	 Search Bar - Displays an editable search bar, containing the search icon, that sends an action message to a target object when Return is tapp...	
 Segmented Control - Displays multiple segments, each of which functions as a discrete button.	 Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a section header or foo...	 Pinch Gesture Recognizer - Provides a recognizer for pinch gestures which are invoked on the view.	 Search Bar and Search Display Controller - Displays an editable search bar connected to a search display controller for managing searching.	

USER INTERFACES

MORE ON VIEWS

UIKit User Interface Catalog

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/index.html>

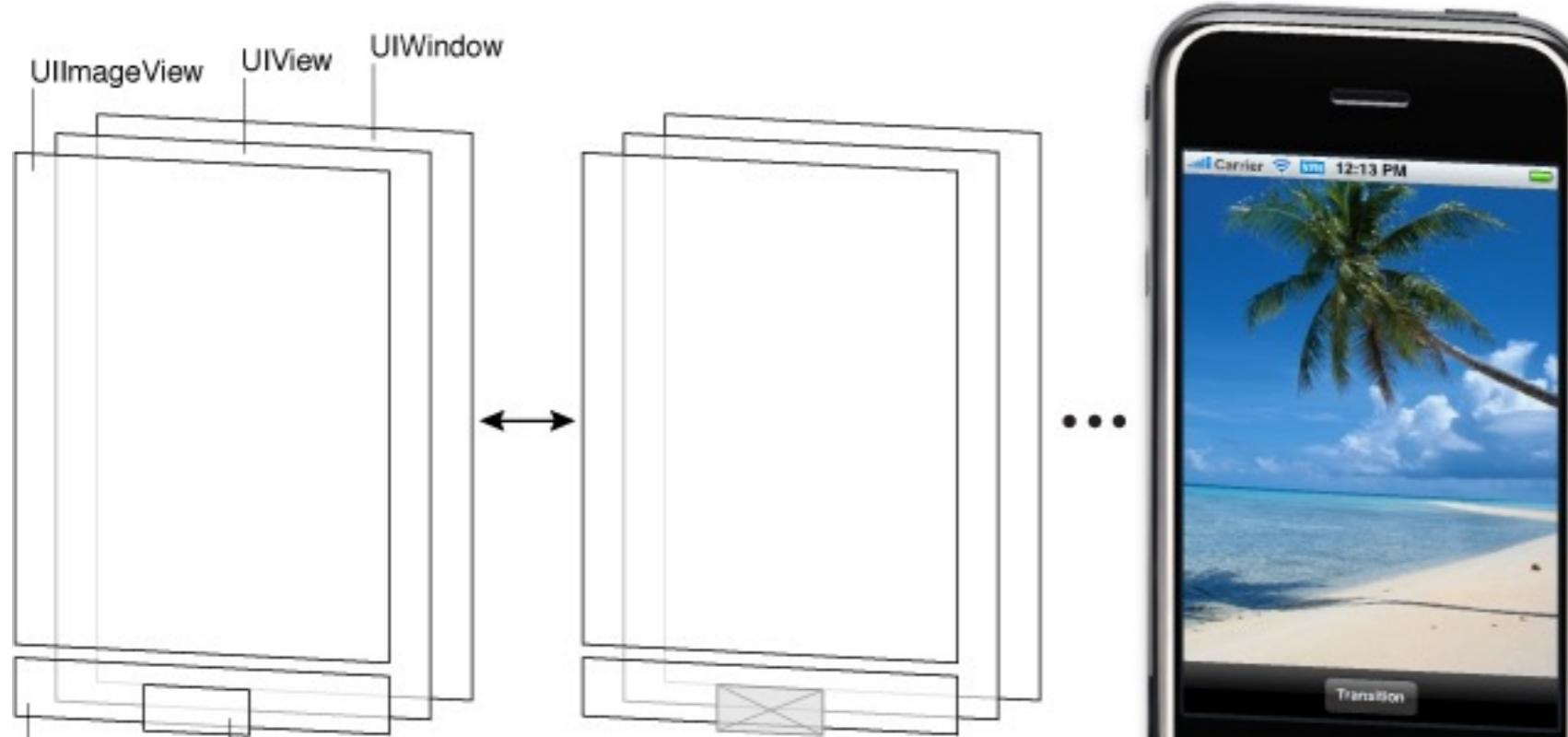


USER INTERFACES

MORE ON VIEWS

View Programming Guide for iOS

https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/Introduction/Introduction.html



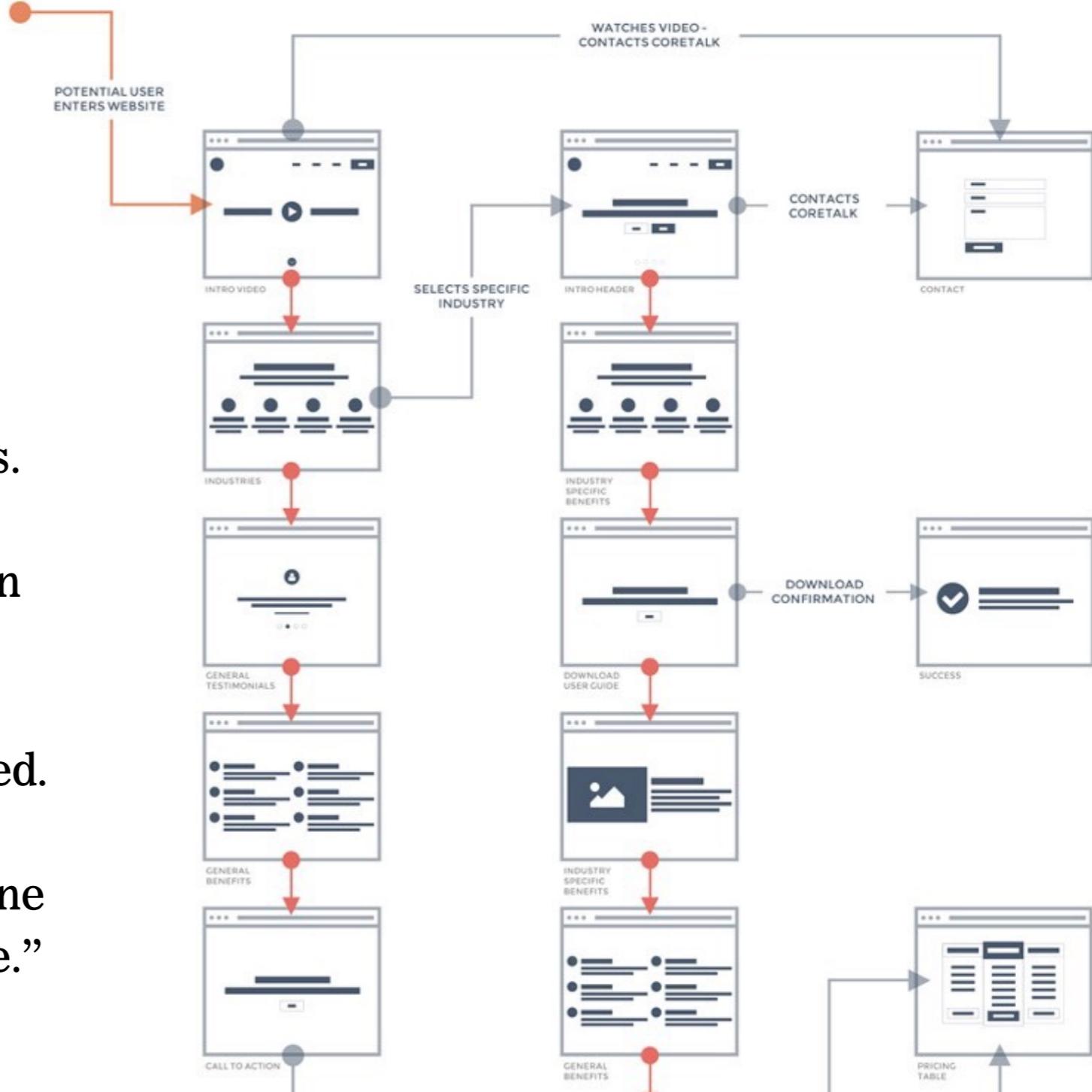
USER INTERFACES

WEBSITE SITEMAPS

Web and UX designers draw sitemap flows that show the structure of a site and describe the “flow” between pages.

Storyboards are like this. They show an app’s screens (here called “Scenes”, each of which correlates to a View Controller), and how they are connected.

Each line of flow (or transition) from one Scene to the another is called a “Segue.”



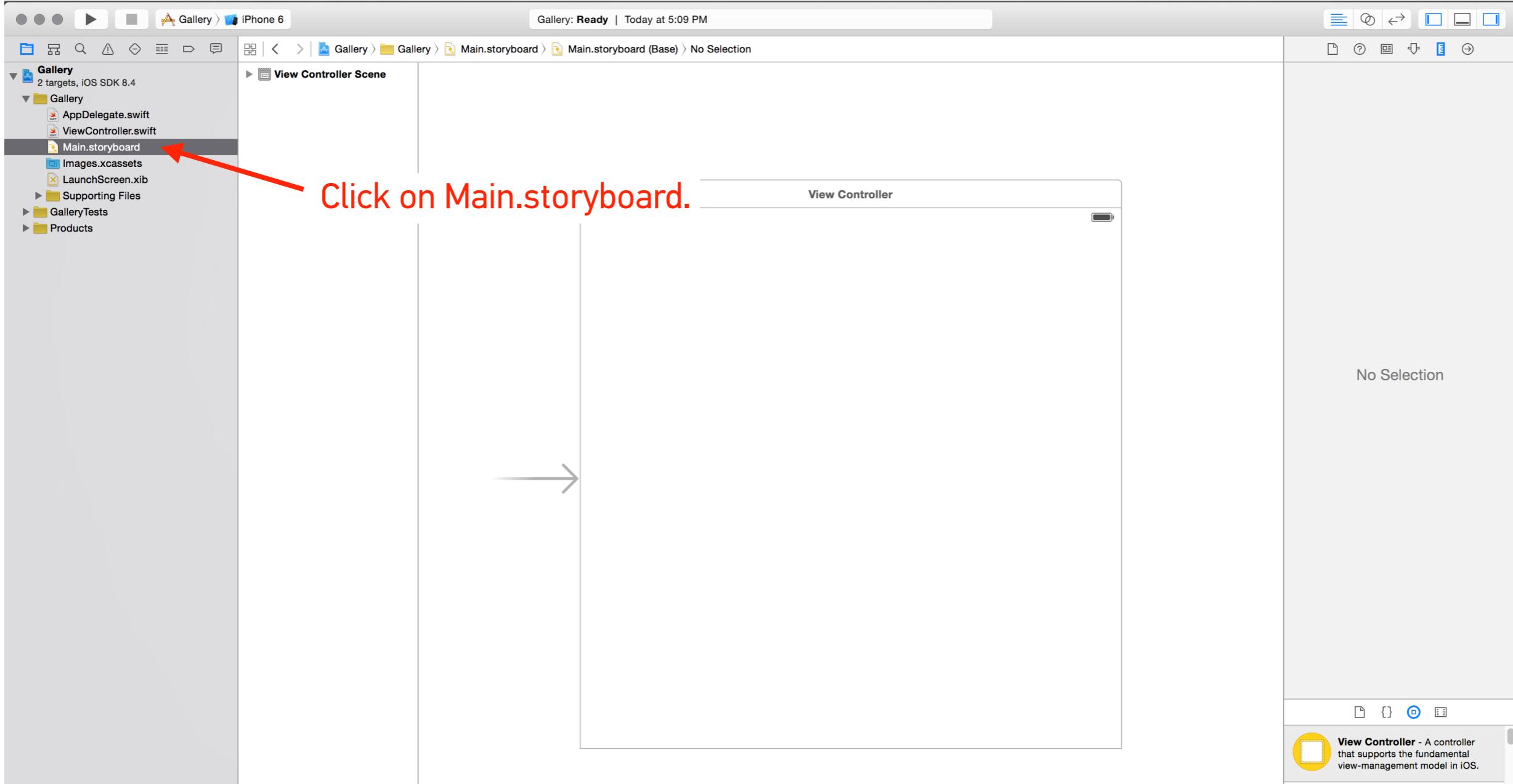
USER INTERFACES

SO, WHAT ARE STORYBOARDS?

- Storyboards can serve as the scaffolding for the various screens of an app.
- By themselves, they can be deployed as a way to prototype apps without code.
- Unlike sitemaps, however, Storyboards aren't passive descriptions, they are active documents that we can use to author a significant part of our apps.

HOW-TO

DISABLING AUTO LAYOUT



Click on Main.storyboard.

For the first half of the course, we won't be using Auto Layout.
So before the Midterm Project, do this for every app you create.

Gallery > iPhone 6 Gallery: Ready | Today at 5:09 PM

View Controller Scene

View Controller

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

Identity and Type

Name Main.storyboard

Type Default - Interface Build...

Location Relative to Group

Base.lproj/Main.storyboard

Full Path /Users/awmartin/Working/General Assembly/Working/Gallery/Gallery/Base.lproj/Main.storyboard

Interface Builder Document

Opens in Default (6.2)

Builds for Project Deployment Tar...

View as iOS 7.0 and Later

Use Auto Layout

Use Size Classes

Global Tint Default

Localization Show

Target Membership Show

Source Control

Repository --

Type --

Current Branch --

Version --

Status No changes

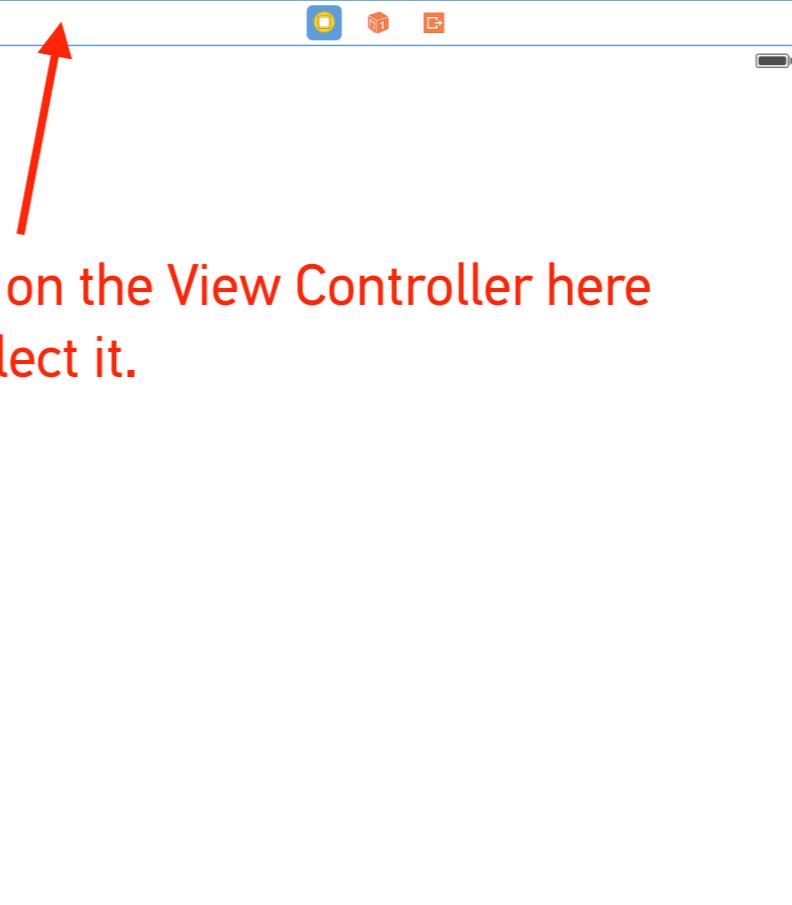
Location

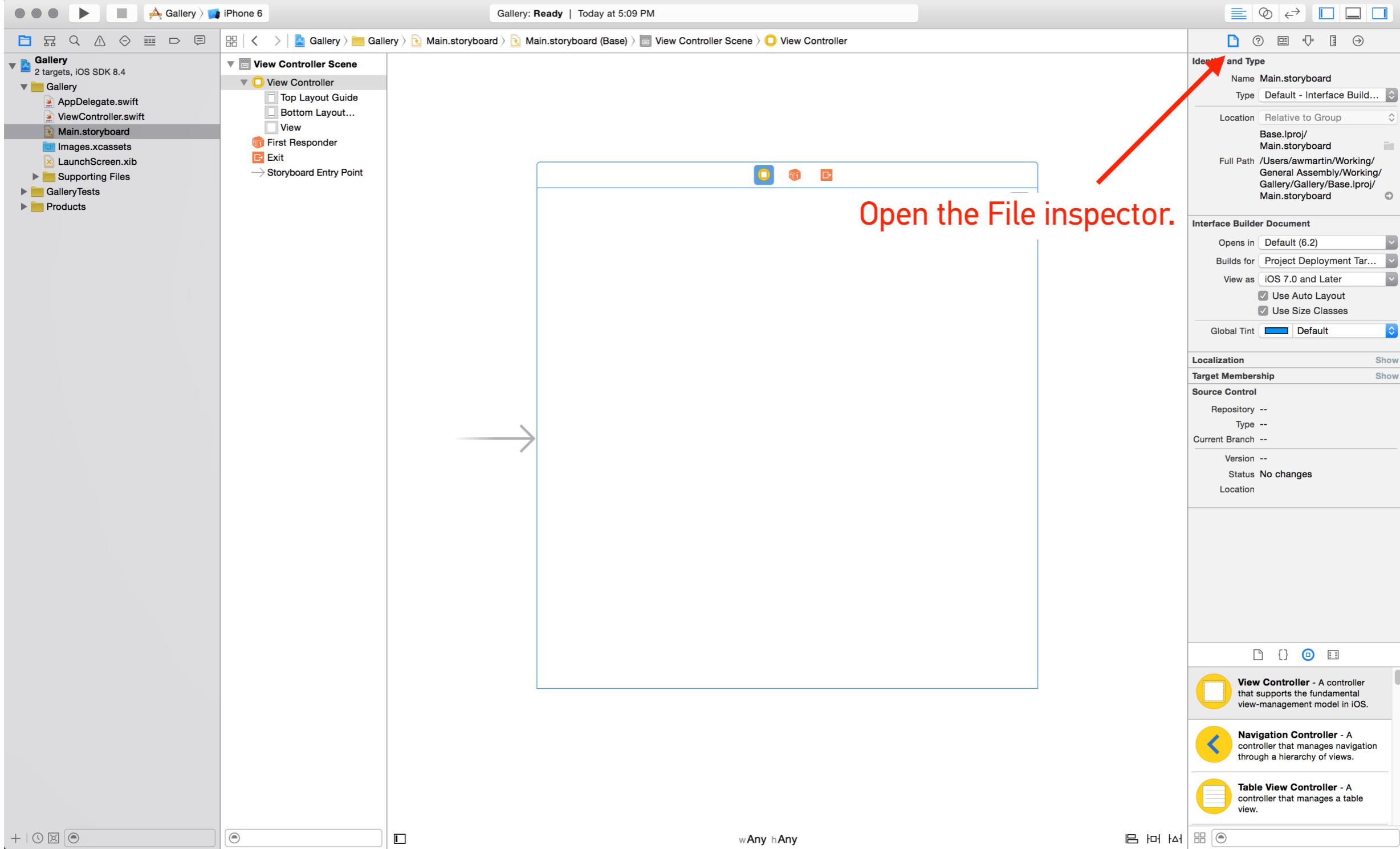
View Controller - A controller that supports the fundamental view-management model in iOS.

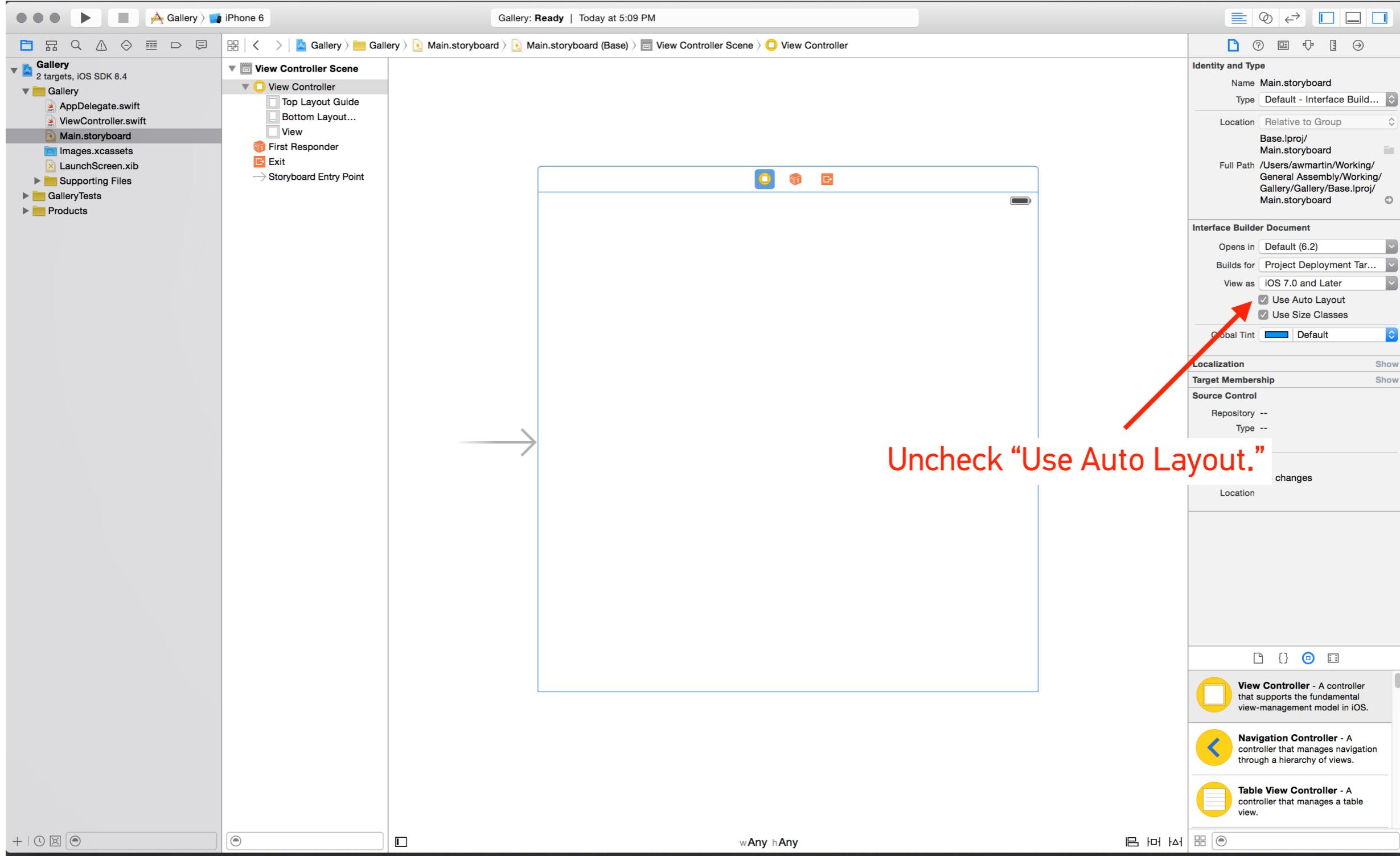
Navigation Controller - A controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table view.

Click on the View Controller here to select it.







Gallery > iPhone 6 Gallery: Ready | Today at 5:09 PM

View Controller Scene

Using Size Classes Requires Auto Layout

Disabling size classes will limit this document to storing data for a single device family. The data for the size class best representing the targeted device will be retained, and all other data will be removed. In addition, segues will be converted to their non-adaptive equivalents.

Keep size class data for: iPhone

Cancel Disable Size Classes

Identity and Type

Name Main.storyboard
Type Default - Interface Build...
Location Relative to Group
Base.lproj/
Main.storyboard
Full Path /Users/awmartin/Working/
General Assembly/Working/
Gallery/Gallery/Base.lproj/
Main.storyboard

Interface Builder Document

Opens in Default (6.2)
Builds for Project Deployment Tar...
View as iOS 7.0 and Later
 Use Auto Layout
 Use Size Classes
Global Tint Default

Localization Show
Target Membership Show
Source Control

Repository --
Type --
Current Branch --
Version --
Status No changes
Location

View Controller - A controller that supports the fundamental view-management model in iOS.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.

Click "Disable Size Classes."
Don't worry about what this means yet.

Gallery > iPhone 6 Gallery: Ready | Today at 5:09 PM

View Controller Scene

- View Controller
- View
- First Responder
- Exit
- Storyboard Entry Point

Identity and Type

- Name Main.storyboard
- Type Default - Interface Build...
- Location Relative to Group
- Base.lproj/Main.storyboard
- Full Path /Users/awmartin/Working/General Assembly/Working/Gallery/Gallery/Base.lproj/Main.storyboard

Interface Builder Document

- Opens in Default (6.2)
- Builds for Project Deployment Tar...
- View as iOS 7.0 and Later
- Use Auto Layout
- Use Size Classes
- Global Tint Default

Localization Show

Target Membership Show

Source Control

- Repository --
- Type --
- Current Branch --
- Version --
- Status No changes
- Location

View Controller - A controller that supports the fundamental view-management model in iOS.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

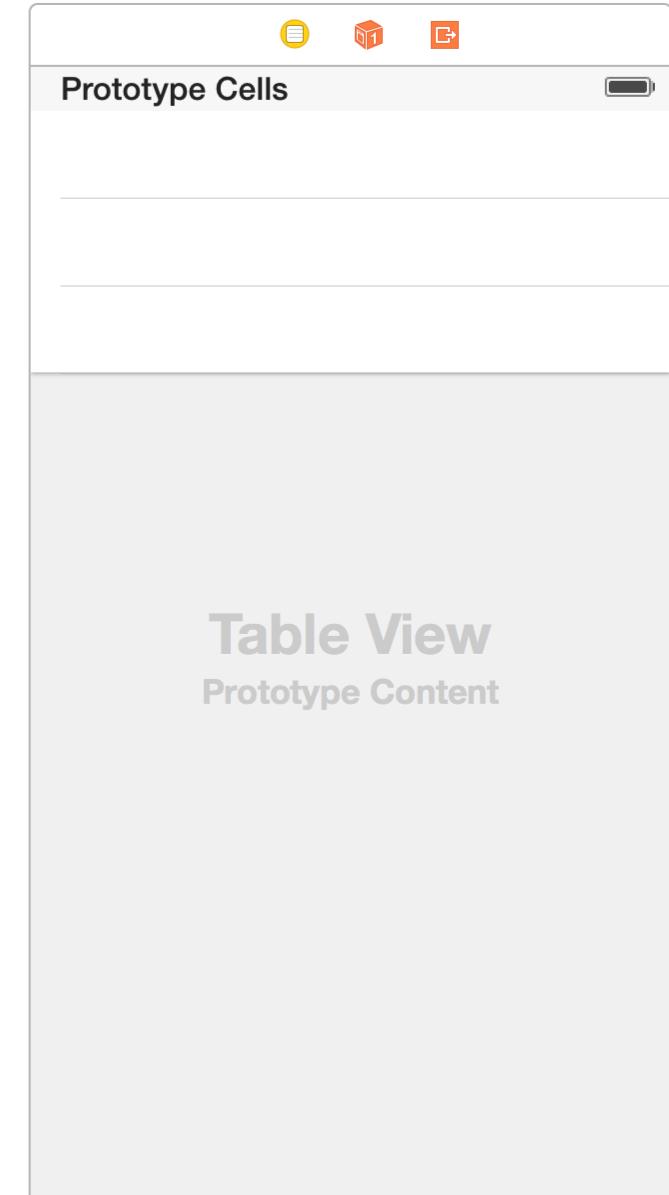
Table View Controller - A controller that manages a table view.

Your View Controller should take on a more iPhone-like proportion.

STORYBOARDS

VIEW CONTROLLERS

- Each screen of our app is made up of at least one View Controller and at least one View.
- A View Controller typically supports one “Scene,” or a single UI screen.
- A View Controller “manages” a set of Views.



 View Controller - A controller that supports the fundamental view-management model in iOS.	 Text Field - Displays editable text and sends an action message to a target object when Return is tapped.	 Text View - Displays multiple lines of editable text and sends an action message to a target object when Return is tapped.	 Rotation Gesture Recognizer - Provides a recognizer for rotation gestures which are invoked on the view.	 Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.
 Navigation Controller - A controller that manages navigation through a hierarchy of views.	 Slider - Displays a continuous range of values and allows the selection of a single value.	 ScrollView - Provides a mechanism to display content that is larger than the size of the application's window.	 Swipe Gesture Recognizer - Provides a recognizer for swipe gestures which are invoked on the view.	 Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.
 Table View Controller - A controller that manages a table view.	 Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle the value.	 DatePicker - Displays multiple rotating wheels to allow users to select dates and times.	 Pan Gesture Recognizer - Provides a recognizer for panning (dragging) gestures which are invoked on the view.	 View - Represents a rectangular region in which it draws and receives events.
 Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.	 Activity Indicator View - Provides feedback on the progress of a task or process of unknown duration.	 Picker View - Displays a spinning-wheel or slot-machine motif of values.	 Screen Edge Pan Gesture Recognizer - Provides a recognizer for panning (dragging) gestures which are invoked on the view and sta...	 Container View - Defines a region of a view controller that can include a child view controller.
 Split View Controller - A composite view controller that manages left and right view controllers.	 Progress View - Depicts the progress of a task over time.	 Visual Effect View	 Long Press Gesture Recognizer - Provides a recognizer for long press gestures which are invoked on the view.	
 Page View Controller - Presents a sequence of view controllers as pages.	 Page Control - Displays a dot for each open page in an application and supports sequential navigation through the pages.	 Visual Effect Views	 Navigation Bar - Provides a mechanism for displaying a navigation bar just below the status bar.	
 GLKit View Controller - A controller that manages a GLKit view.	 Stepper - Provides a user interface for incrementing or decrementing a value.	 MapKit View - Displays maps and provides an embeddable interface to navigate map content.	 Navigation Item - Represents a state of the navigation bar, including a title.	
 Object - Provides a template for objects and controllers not directly available in Interface Builder.	 Table View - Displays data in a list of plain, sectioned, or grouped rows.	 GLKit View - Provides a default implementation of an OpenGL ES-aware view.	 Toolbar - Provides a mechanism for displaying a toolbar at the bottom of the screen.	
 Collection View Controller - A controller that manages a collection view.	 Table View Cell - Defines the attributes and behavior of cells (rows) in a table view.	 iAd BannerView - The ADBannerView class provides a view that displays banner advertisements to the user.	 Item	
 AVKit Player View Controller - A view controller that manages a AVPlayer object.	 Image View - Displays a single image, or an animation described by an array of images.	 SceneKit View - A view for displaying a 3D scene.	 Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.	
 Label Label - A variably sized amount of static text.	 Collection View - Displays data in a collection of cells.	 Web View - Displays embedded web content and enables content navigation.	 Tab Bar Item - Represents an item on a UITabBar object.	
 Button - Intercepts touch events and sends an action message to a target object when it's tapped.	 Collection View Cell - Defines the attributes and behavior of cells in a collection view.	 Tap Gesture Recognizer - Provides a recognizer for tap gestures which land on the view.	 Search Bar - Displays an editable search bar, containing the search icon, that sends an action message to a target object when Return is tapp...	
 Segmented Control - Displays multiple segments, each of which functions as a discrete button.	 Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a section header or foo...	 Pinch Gesture Recognizer - Provides a recognizer for pinch gestures which are invoked on the view.	 Search Bar and Search Display Controller - Displays an editable search bar connected to a search display controller for managing searching.	

HOW-TO

PRACTICE: ADDING VIEWS

STORYBOARDS

WHAT ARE STORYBOARDS?

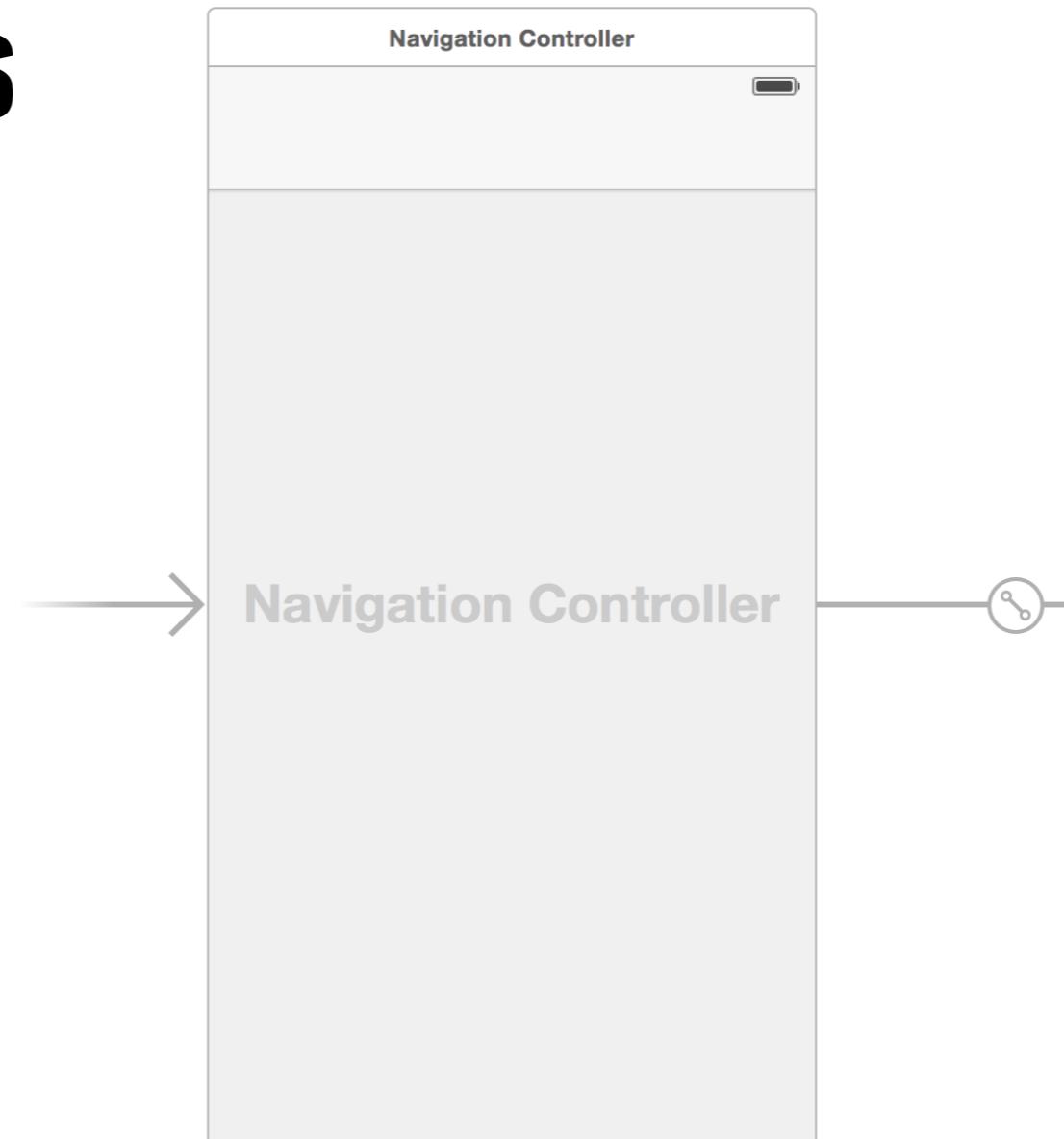
STORYBOARDS

NAVIGATION CONTROLLERS

- Navigation Controllers manage several View Controllers.
- One View Controller is “presented” to the user at a time.
- They organize the View Controllers in a browser-history-like fashion (think forward and back buttons).
- By default, they place Navigation Bars on all the View Controllers they manage.

STORYBOARDS

NAVIGATION CONTROLLERS



 View Controller - A controller that supports the fundamental view-management model in iOS.	 Text Field - Displays editable text and sends an action message to a target object when Return is tapped.	 Text View - Displays multiple lines of editable text and sends an action message to a target object when Return is tapped.	 Rotation Gesture Recognizer - Provides a recognizer for rotation gestures which are invoked on the view.	 Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.
 Navigation Controller - A controller that manages navigation through a hierarchy of views.	 Slider - Displays a continuous range of values and allows the selection of a single value.	 ScrollView - Provides a mechanism to display content that is larger than the size of the application's window.	 Swipe Gesture Recognizer - Provides a recognizer for swipe gestures which are invoked on the view.	 Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.
 Table View Controller - A controller that manages a table view.	 Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle the value.	 DatePicker - Displays multiple rotating wheels to allow users to select dates and times.	 Pan Gesture Recognizer - Provides a recognizer for panning (dragging) gestures which are invoked on the view.	 View - Represents a rectangular region in which it draws and receives events.
 Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.	 Activity Indicator View - Provides feedback on the progress of a task or process of unknown duration.	 Picker View - Displays a spinning-wheel or slot-machine motif of values.	 Screen Edge Pan Gesture Recognizer - Provides a recognizer for panning (dragging) gestures which are invoked on the view and sta...	 Container View - Defines a region of a view controller that can include a child view controller.
 Split View Controller - A composite view controller that manages left and right view controllers.	 Progress View - Depicts the progress of a task over time.	 Visual Effect View	 Long Press Gesture Recognizer - Provides a recognizer for long press gestures which are invoked on the view.	
 Page View Controller - Presents a sequence of view controllers as pages.	 Page Control - Displays a dot for each open page in an application and supports sequential navigation through the pages.	 Visual Effect Views	 Title	 Navigation Bar - Provides a mechanism for displaying a navigation bar just below the status bar.
 GLKit View Controller - A controller that manages a GLKit view.	 Stepper - Provides a user interface for incrementing or decrementing a value.	 MapKit View - Displays maps and provides an embeddable interface to navigate map content.	 Navigation Item - Represents a state of the navigation bar, including a title.	
 Object - Provides a template for objects and controllers not directly available in Interface Builder.	 Table View - Displays data in a list of plain, sectioned, or grouped rows.	 GLKit View - Provides a default implementation of an OpenGL ES-aware view.	 Toolbar	
 Collection View Controller - A controller that manages a collection view.	 Table View Cell - Defines the attributes and behavior of cells (rows) in a table view.	 iAd BannerView - The ADBannerView class provides a view that displays banner advertisements to the user.	 Item	
 AVKit Player View Controller - A view controller that manages a AVPlayer object.	 Image View - Displays a single image, or an animation described by an array of images.	 SceneKit View - A view for displaying a 3D scene.	 Tab Bar - Provides a mechanism for displaying tabs at the bottom of the screen.	
 Label Label - A variably sized amount of static text.	 Collection View - Displays data in a collection of cells.	 Web View - Displays embedded web content and enables content navigation.	 Tab Bar Item	
 Button - Intercepts touch events and sends an action message to a target object when it's tapped.	 Collection View Cell - Defines the attributes and behavior of cells in a collection view.	 Tap Gesture Recognizer - Provides a recognizer for tap gestures which land on the view.	 Search Bar - Displays an editable search bar, containing the search icon, that sends an action message to a target object when Return is tapp...	
 Segmented Control - Displays multiple segments, each of which functions as a discrete button.	 Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a section header or foo...	Pinch Gesture Recognizer - Provides a recognizer for pinch gestures which are invoked on the view.	Search Bar and Search Display Controller - Displays an editable search bar connected to a search display controller for managing searching.	

STORYBOARDS

BACK TO STORYBOARDS...

- Navigation Controllers enable us to link multiple Scenes (View Controllers) together with a very typical mobile navigation paradigm.
- They represent transitions between many Scenes. These transitions are encoded by “Segues”.

STORYBOARDS

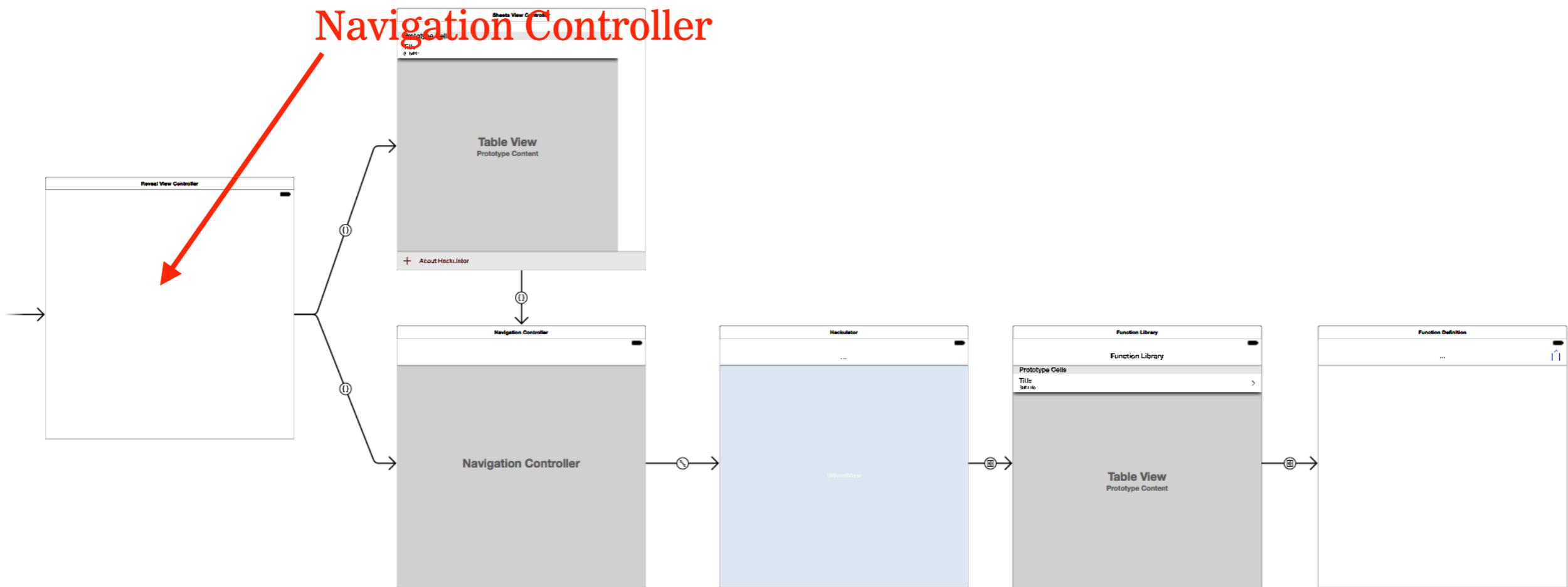
NAVIGATION CONTROLLERS

Some terms:

- “Initial View Controller” – The View Controller the app will load when an app is launched.
- “Root View Controller” – The View Controller that a Navigation Controller will first load when it is created.

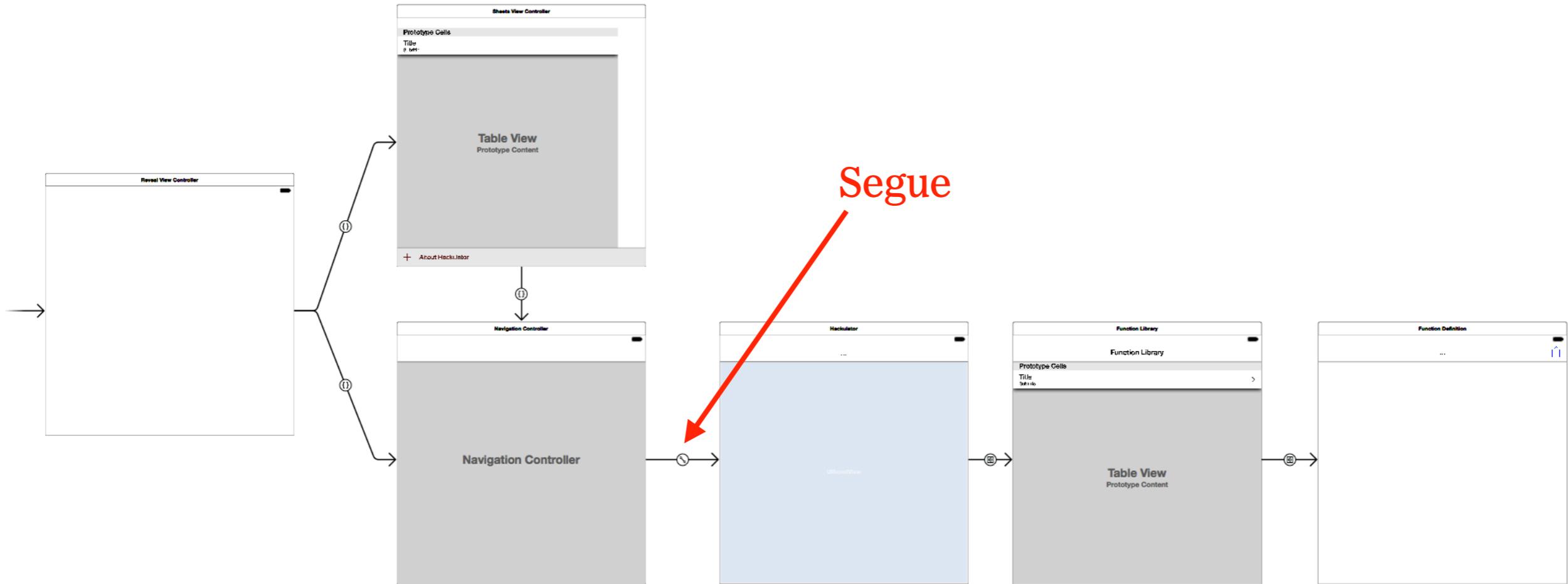
STORYBOARDS

SIMPLE STORYBOARD EXAMPLE



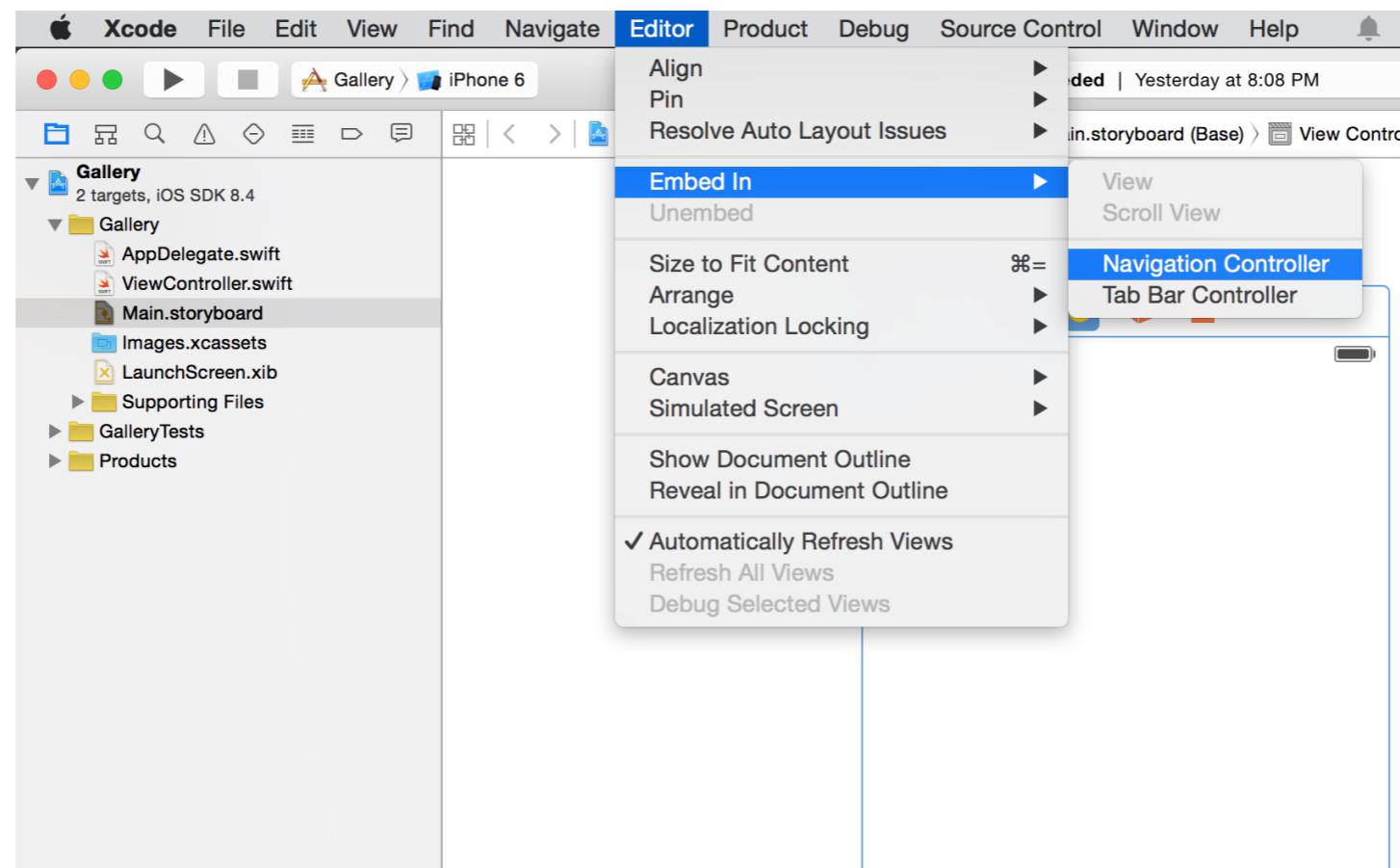
STORYBOARDS

SIMPLE STORYBOARD EXAMPLE



STORYBOARDS

NAVIGATION CONTROLLERS



STORYBOARDS

SEGUES

- A “Segue” is essentially a transition between two View Controllers.
- In code, we can use them to transfer information from one View Controller to another, like going from a list of emails to a single view of just one email.
- There are several types...

Name	Interface Builder Symbol	Description
Show		Present the content in the detail or master area depending on the content of the screen. If the app is displaying a master and detail view, the content is pushed onto the detail area. If the app is only displaying the master or the detail, the content is pushed on top of the current view controller stack.
Show Detail		Present the content in the detail area. If the app is displaying a master and detail view, the new content replaces the current detail. If the app is only displaying the master or the detail, the content replaces the top of the current view controller stack.
Present Modally		Present the content modally. There are options to choose a presentation style (<code>UIModalPresentationStyle</code>) and a transition style (<code>UIModalTransitionStyle</code>).
Present as Popover		Present the content as a popover anchored to an existing view. There is an option to specify the possible directions of the arrow shown on one edge of the popover view (<code>UIPopoverArrowDirection</code>). There is also an option to specify the anchor view.
Custom		A custom segue enabling you to write your own behaviors.
Push (Deprecated)		Present the content by pushing it onto the current stack of view controllers.
Modal (Deprecated)		Present the content modally on top of the existing screen. The options are the same as Present Modally.
Popover (Deprecated)		Present the content as a popover. The options are the same as Present as Popover.

STORYBOARDS

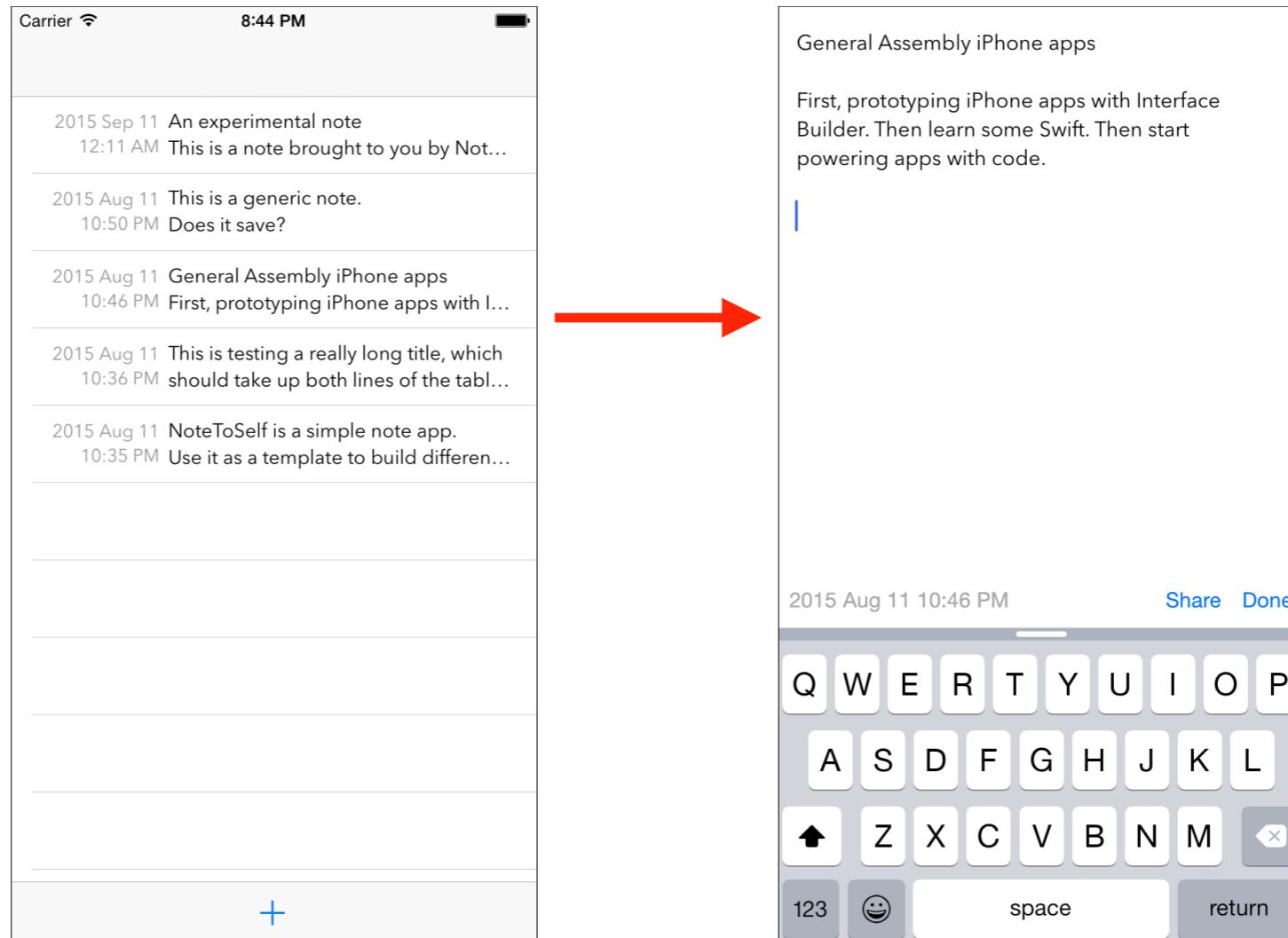
USER FLOW

- Storyboards can reflect the structure of an app's user interface.
- Consider different ways of structuring a notes app.

STORYBOARDS

USER FLOW

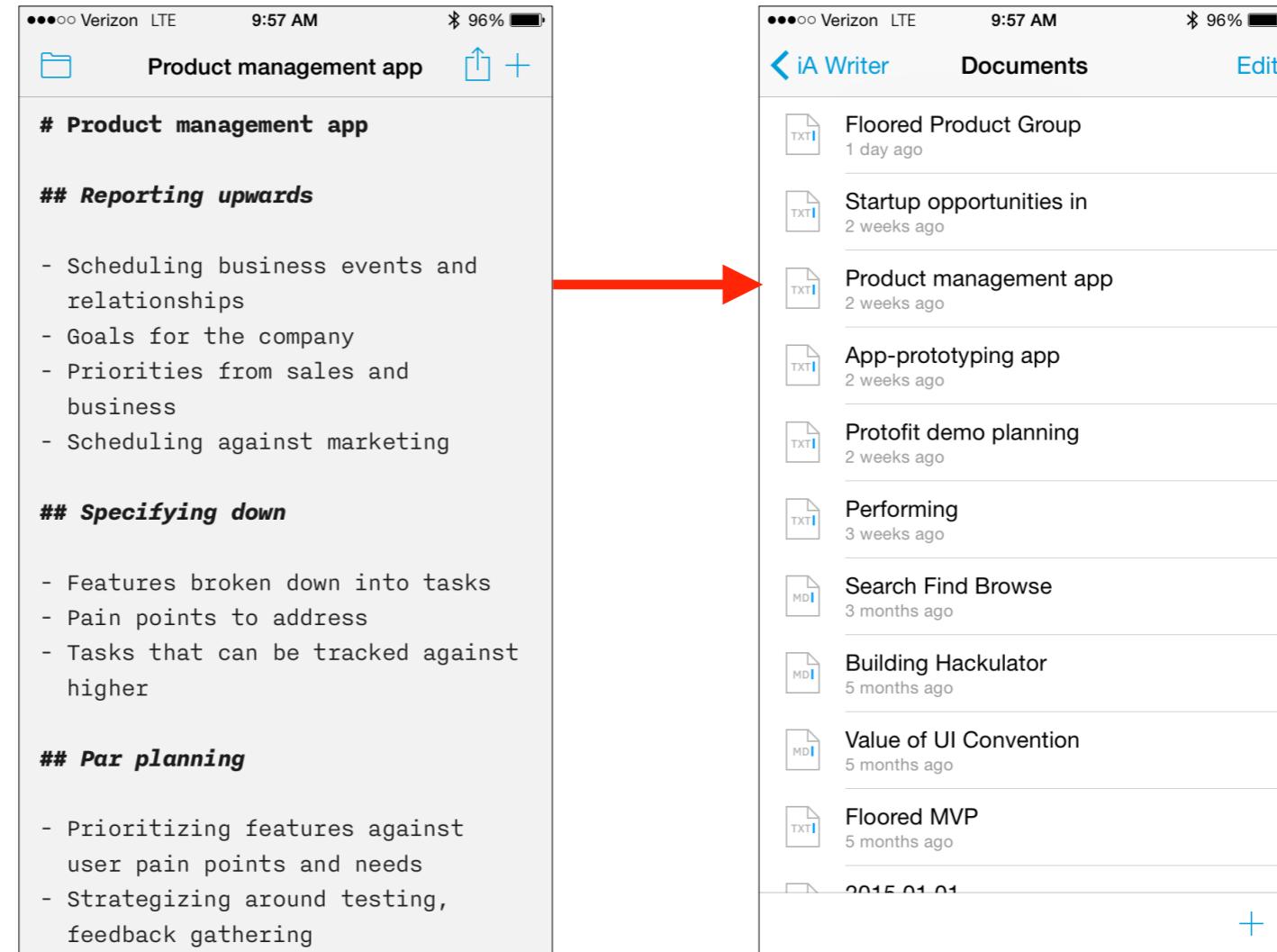
► Master / Detail



STORYBOARDS

USER FLOW

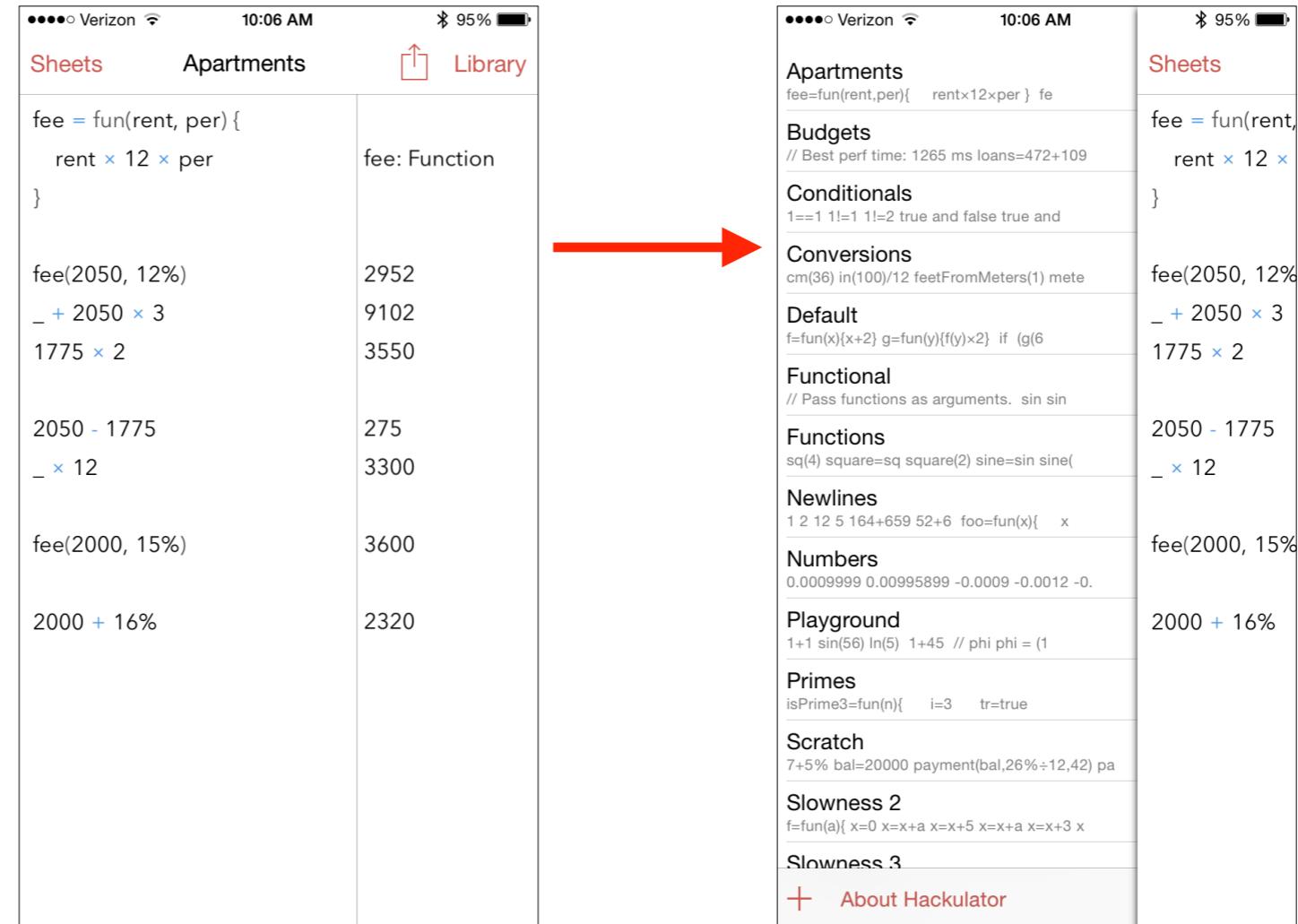
► Focus / List



STORYBOARDS

USER FLOW

► Focus / Reveal



Apartment	fee=fun(rent,per){ rent×12×per } fe
Budgets	// Best perf time: 1265 ms loans=472+109
Conditionals	1==1 1!=1 1!=2 true and false true and
Conversions	cm(36) in(100)/12 feetFromMeters(1) mete
Default	f=fun(x)(x+2) g=fun(y){f(y)×2} if (g(6
Functional	// Pass functions as arguments. sin sin
Functions	sq(4) square=sq square(2) sine=sin sine(
Newlines	1 2 12 5 164+659 52+6 foo=fun(x){ x
Numbers	0.0009999 0.00995899 -0.0009 -0.0012 -0.
Playground	1+1 sin(56) ln(5) 1+45 // phi phi = (1
Primes	isPrime3=fun(n){ i=3 tr=true
Scratch	7+5% bal=20000 payment(bal,26%÷12,42) pa
Slowness 2	f=fun(a){ x=0 x=x+a x=x+5 x=x+a x=x+3 x
Slowness 3	
About Hackulator	+ About Hackulator

STORYBOARDS

PRACTICE MAKING SEGUES

GETTING STARTED

Q&A