

Stanford CS193p

Developing Applications for iOS

Fall 2013-14



Multiple MVCS

- ⦿ Why?

When your application gets more features than can fit in one MVC.

- ⦿ How to add a new MVC to your storyboard

Drag “View Controller” from Object Palette.

Create a subclass of UIViewController using New File menu item.

Set that subclass as the class of your new Controller in the Attributes Inspector.

- ⦿ How to present this new MVC to the user

UINavigationController

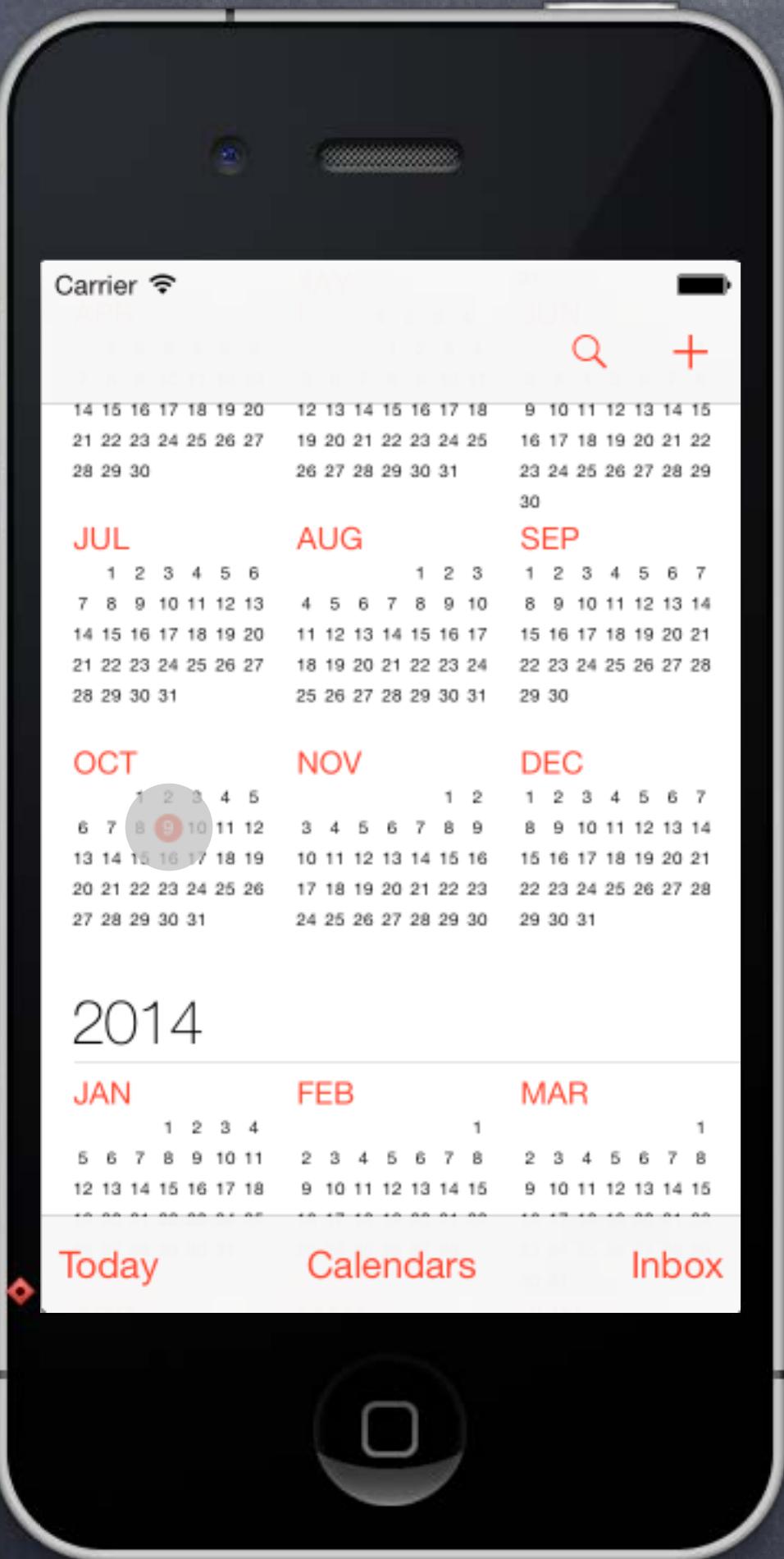
UITabBarController

Other mechanisms we'll talk about later in the course (popover, modal, etc.).

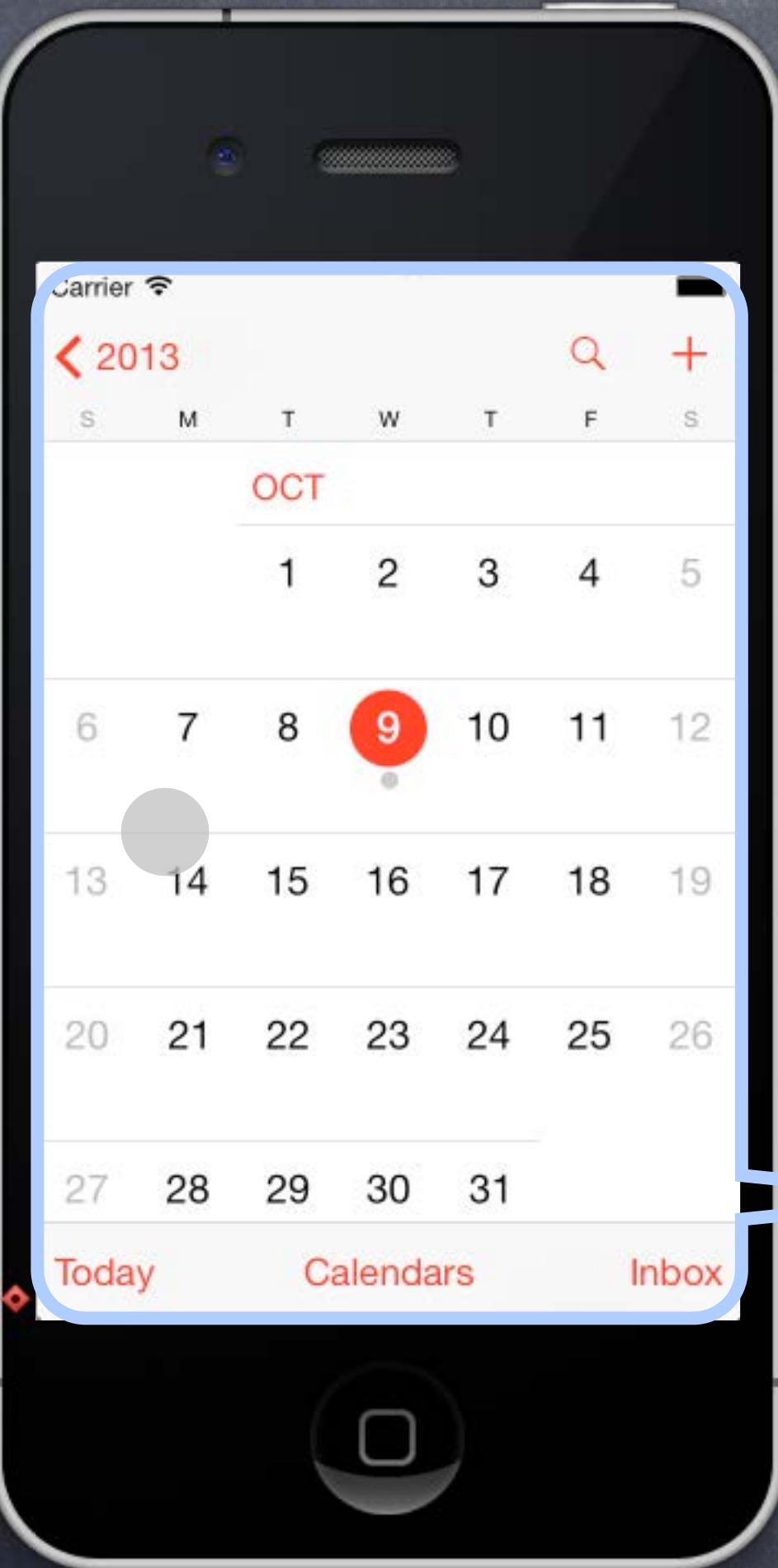
UINavigationController

When to use it?

When the user wants to “dive down” into more detail.



UINavigationController



When to use it?

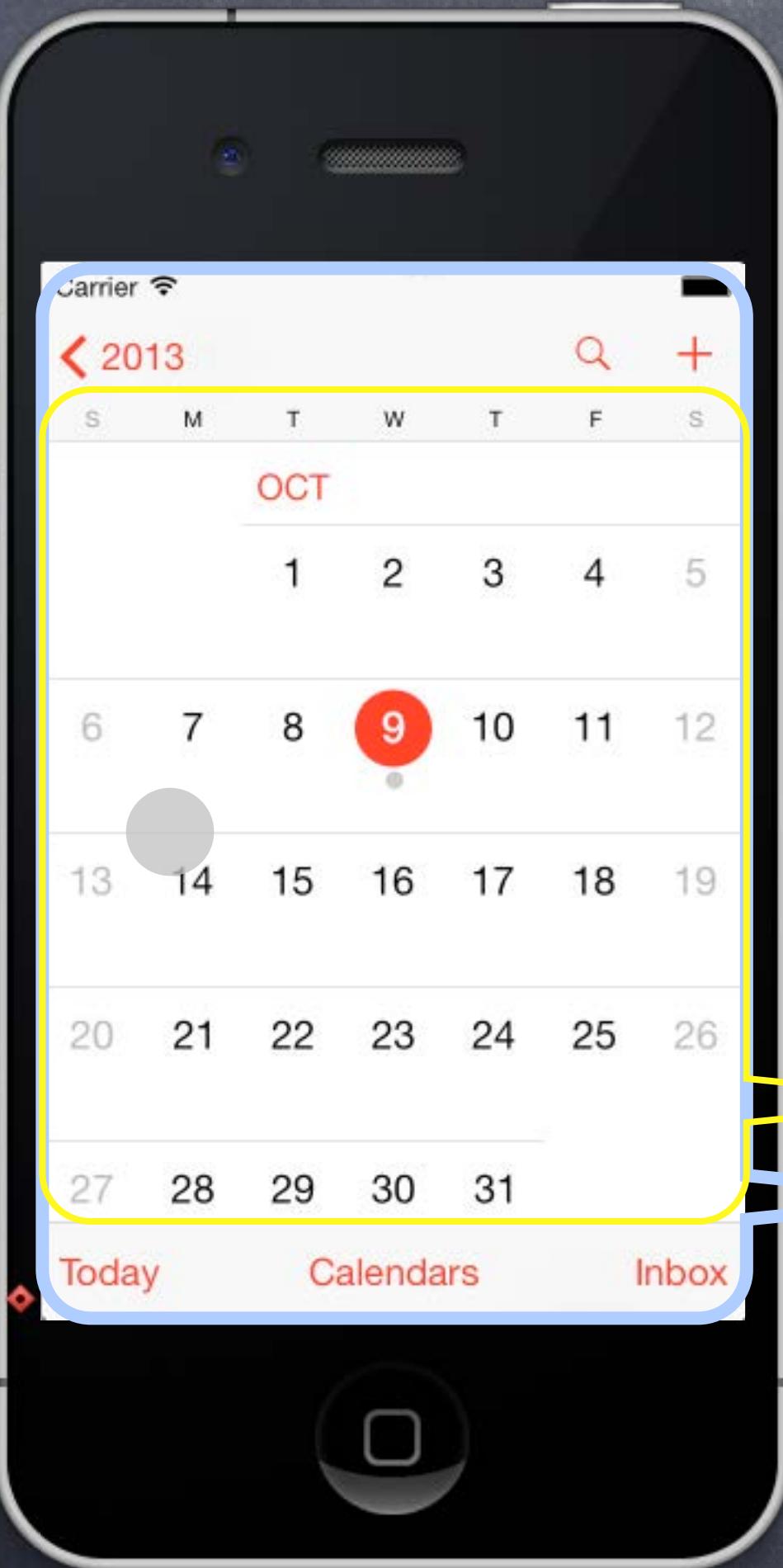
When the user wants to “dive down” into more detail.

How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC “segue” to the other MVCs.

This is the UINavigationController's View.

UINavigationController



When to use it?

When the user wants to "dive down" into more detail.

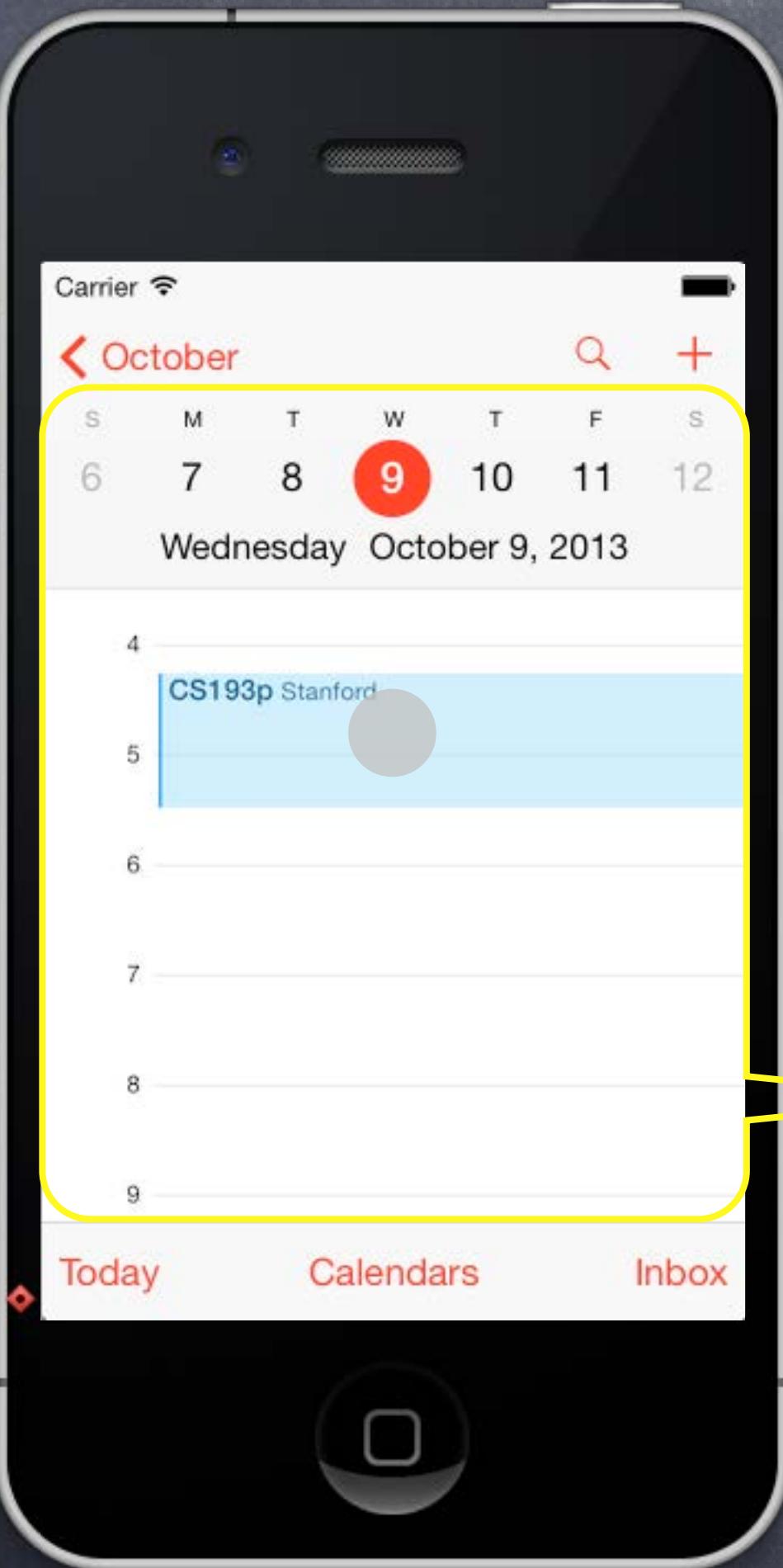
How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

This is a Month MVC's View.

This is the UINavigationController's View.

UINavigationController



When to use it?

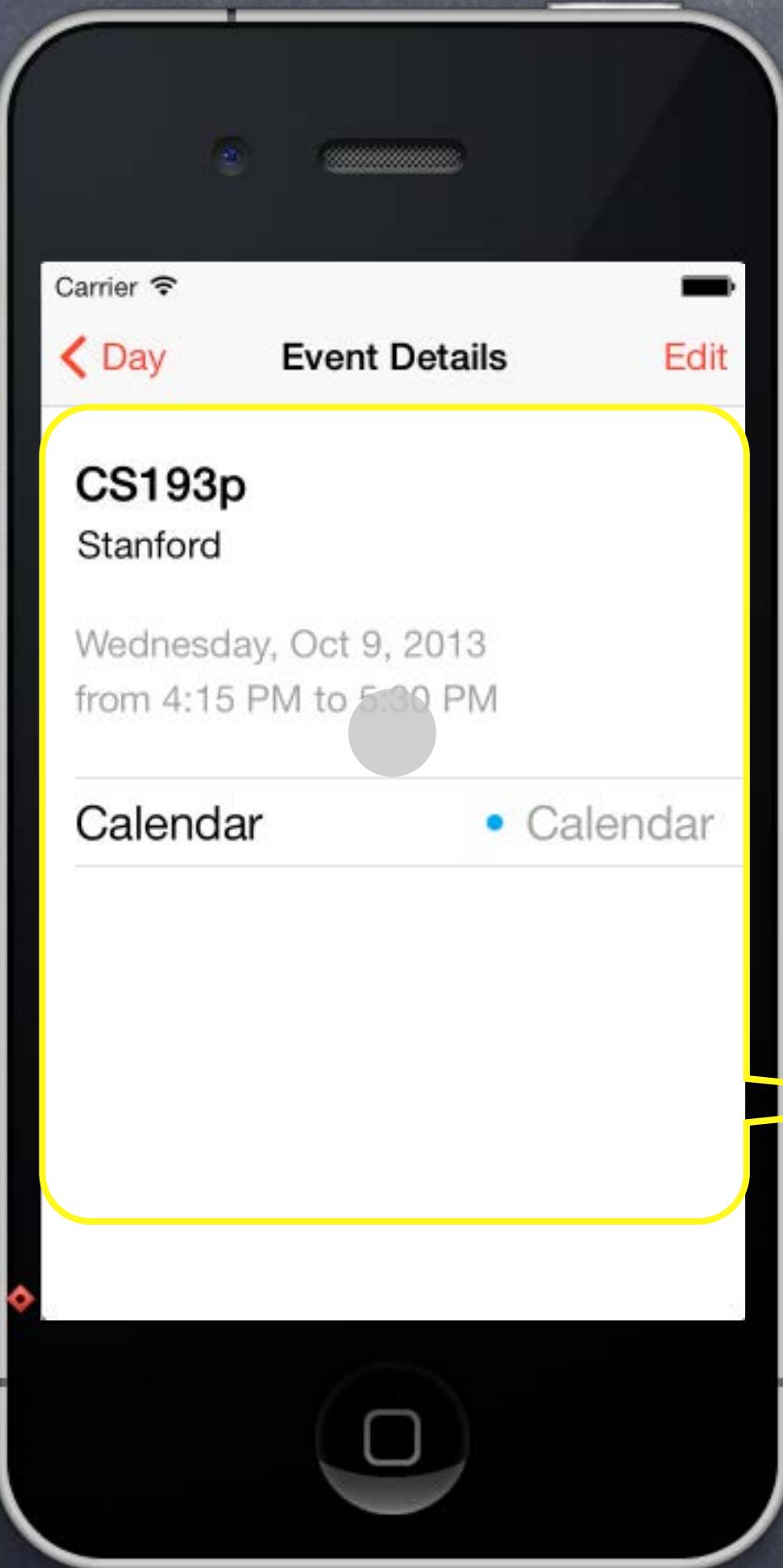
When the user wants to "dive down" into more detail.

How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

This is a Day MVC's View.

UINavigationController



When to use it?

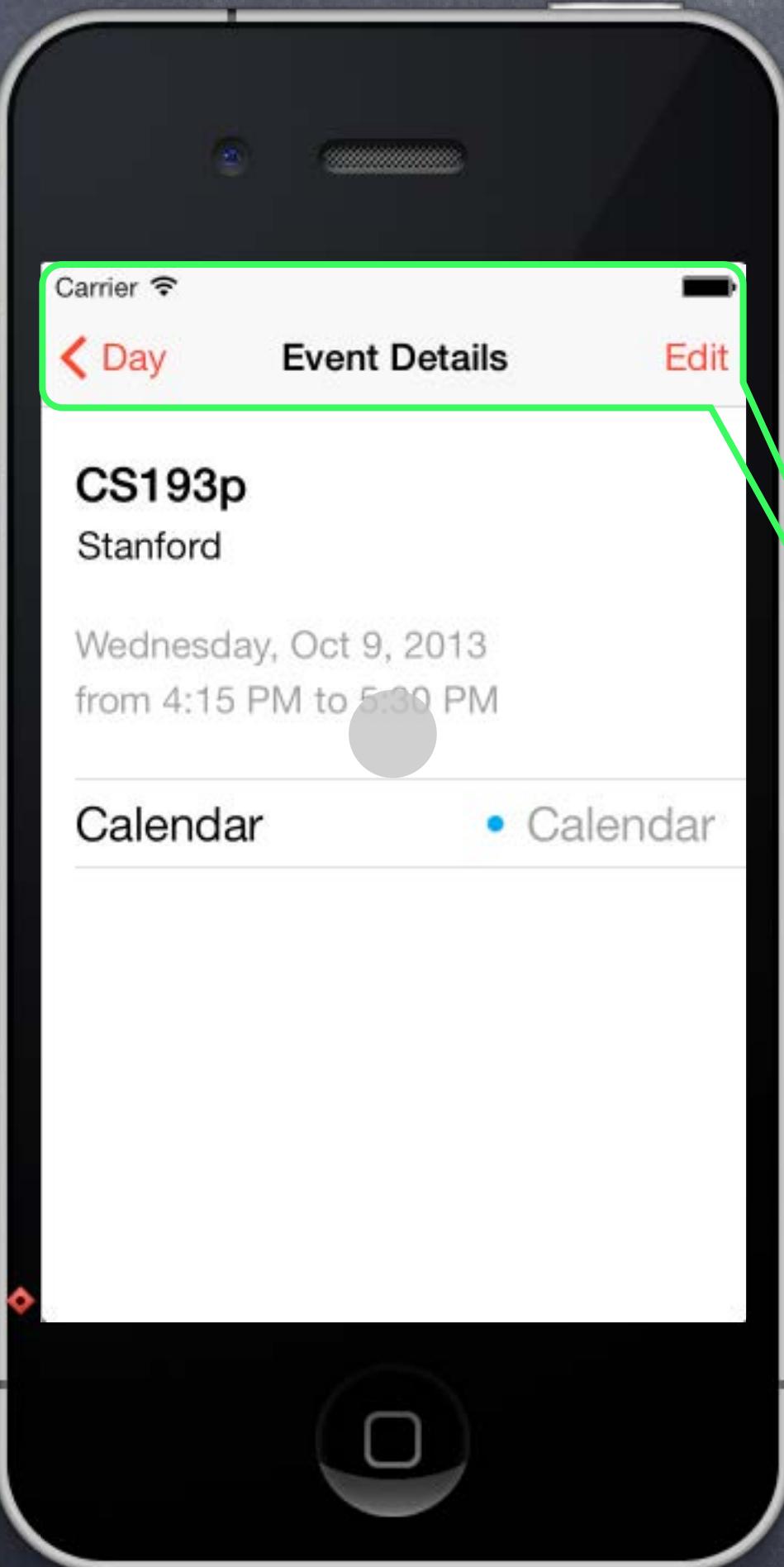
When the user wants to "dive down" into more detail.

How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

This is a Calendar Event MVC's View.

UINavigationController



⌚ When to use it?

When the user wants to "dive down" into more detail.

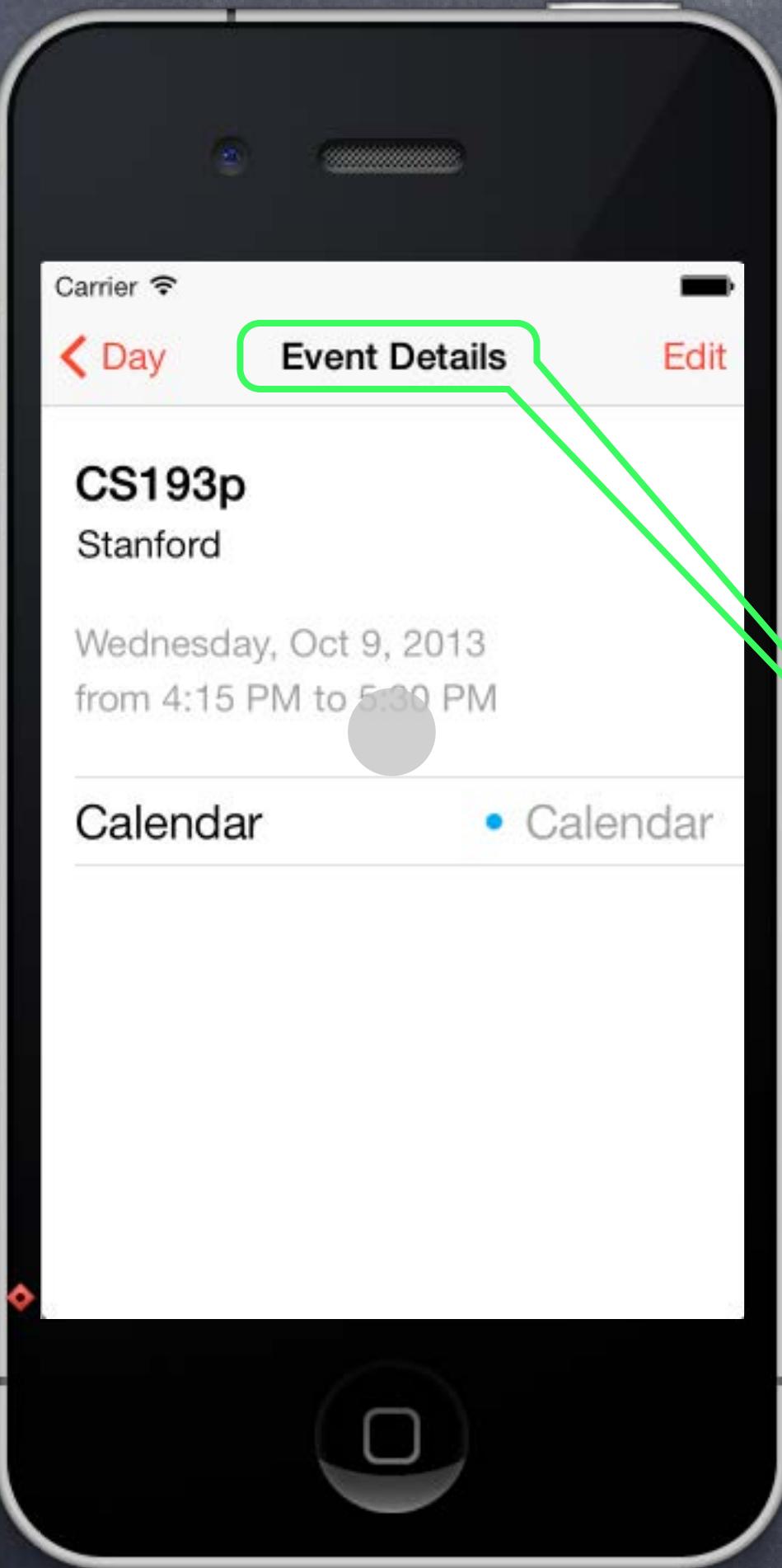
⌚ How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

⌚ Components of a UINavigationController

Navigation Bar (contents determined by embedded MVC's **navigationItem**).

UINavigationController



- ⌚ When to use it?

When the user wants to "dive down" into more detail.

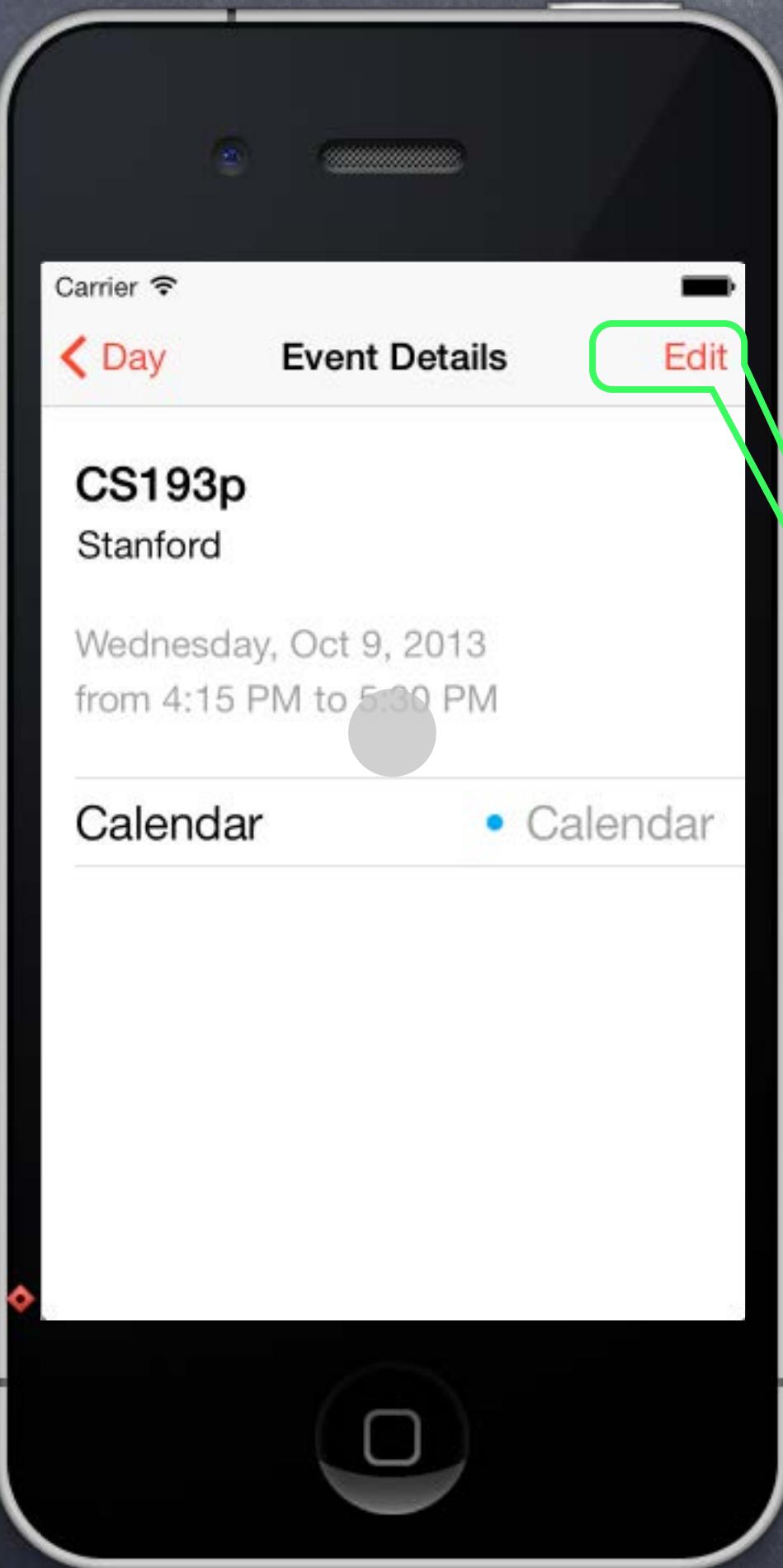
- ⌚ How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

- ⌚ Components of a UINavigationController

Navigation Bar (contents determined by embedded MVC's `navigationItem`).
Title (by default is `title` property of the embedded MVC)

UINavigationController



When to use it?

When the user wants to "dive down" into more detail.

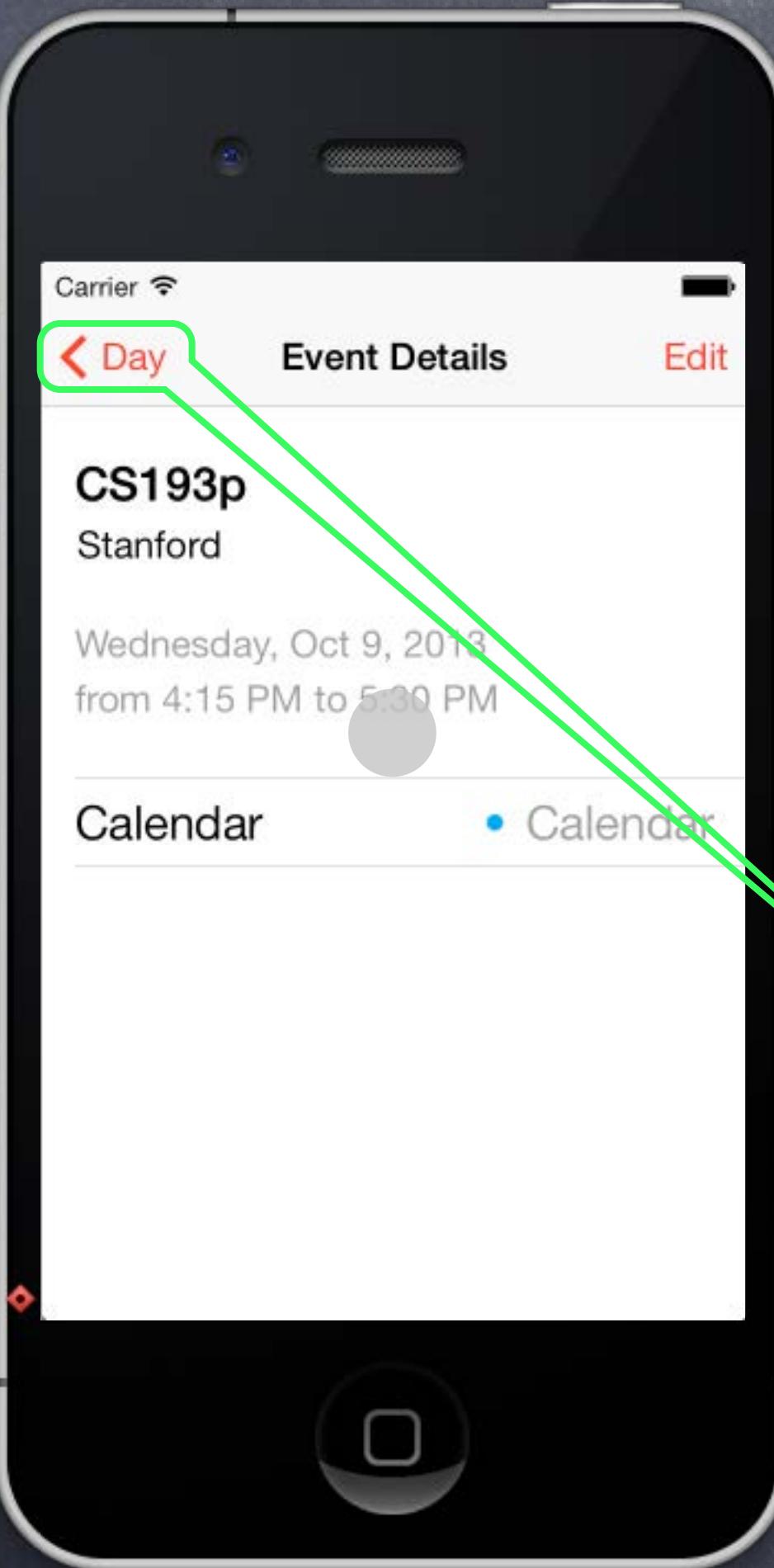
How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

Components of a UINavigationController

Navigation Bar (contents determined by embedded MVC's `navigationItem`).
Title (by default is `title` property of the embedded MVC)
Embedded MVC's `navigationItem.rightBarButtonItemItems`
(an NSArray of UIBarButtonItems)

UINavigationController



When to use it?

When the user wants to "dive down" into more detail.

How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

Components of a UINavigationController

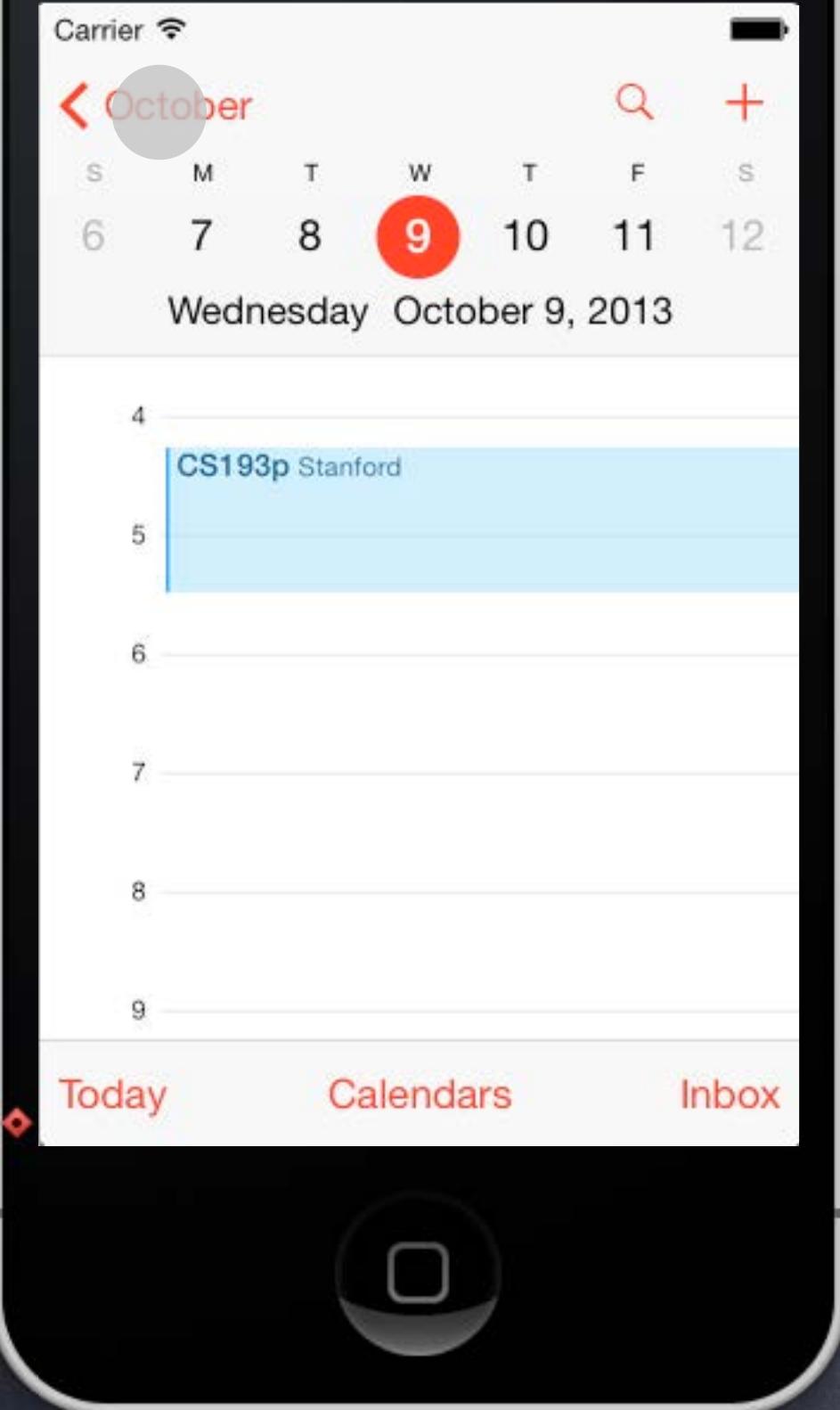
Navigation Bar (contents determined by embedded MVC's `navigationItem`).

Title (by default is `title` property of the embedded MVC)

Embedded MVC's `navigationItem.rightBarButtonItems`
(an NSArray of UIBarButtonItems)

Back Button (automatic)

UINavigationController



When to use it?

When the user wants to “dive down” into more detail.

How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).

Touches in one MVC “segue” to the other MVCs.

Components of a UINavigationController

Navigation Bar (contents determined by embedded MVC's `navigationItem`).

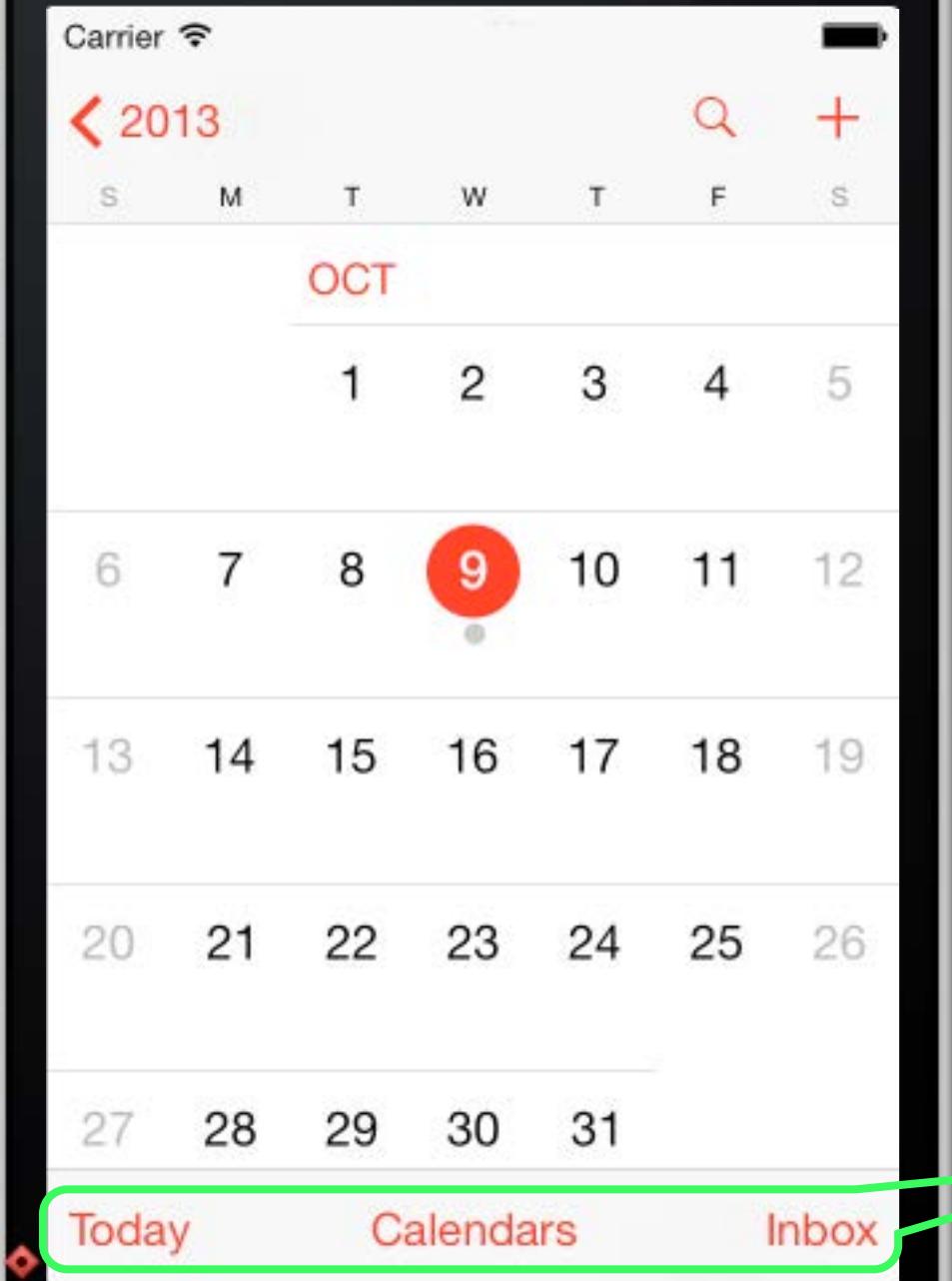
Title (by default is `title` property of the embedded MVC)

Embedded MVC's `navigationItem.rightBarButtonItems`

(an NSArray of UIBarButtonItems)

Back Button (automatic)

UINavigationController



When to use it?

When the user wants to "dive down" into more detail.

How does it work?

Encloses other MVCs (like the Year MVC and the Month MVC).
Touches in one MVC "segue" to the other MVCs.

Components of a UINavigationController

Navigation Bar (contents determined by embedded MVC's `navigationItem`).

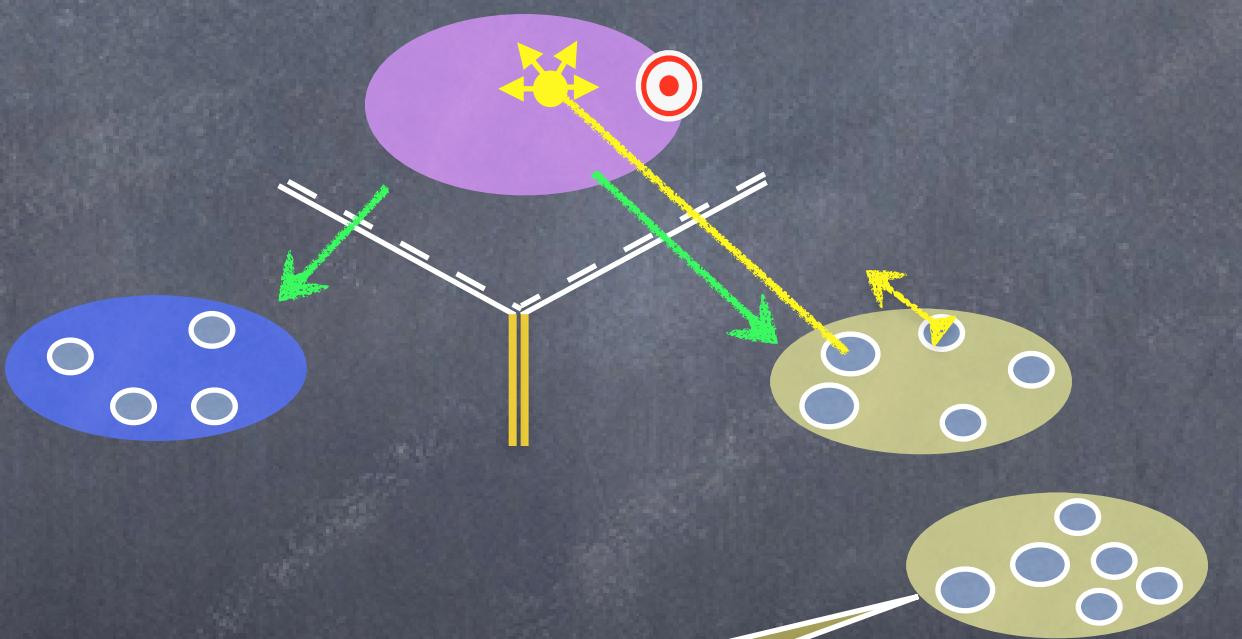
Title (by default is `title` property of the embedded MVC)

Embedded MVC's `navigationItem.rightBarButtonItem`
(an NSArray of UIBarButtonItems)

Back Button (automatic)

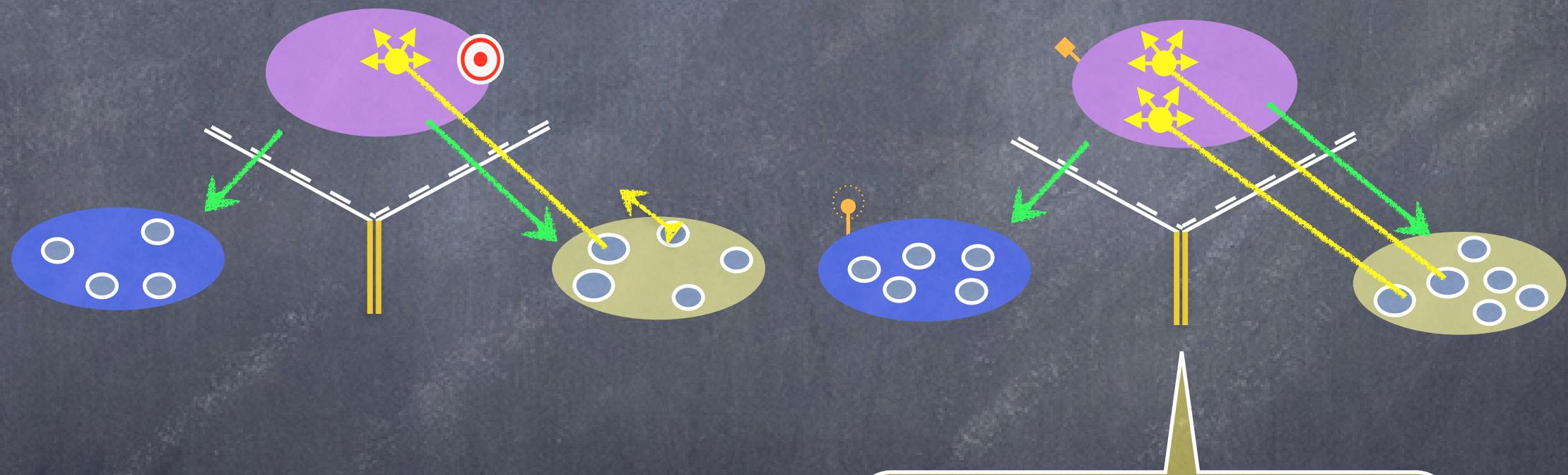
Embedded MVC's `toolbarItems` property
(also an NSArray of UIBarButtonItems)

MVCs working together



I want more features, but it doesn't make sense to put them all in one MVC!

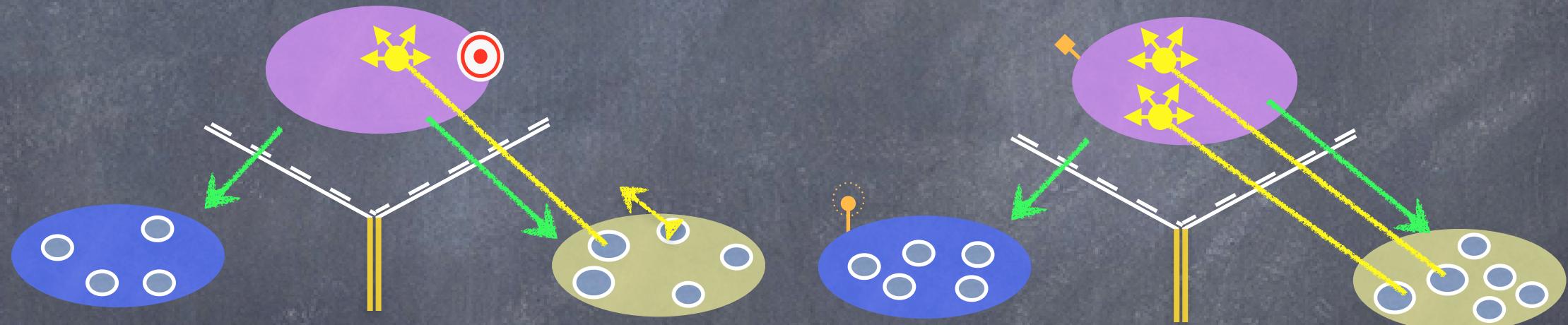
MVCs working together



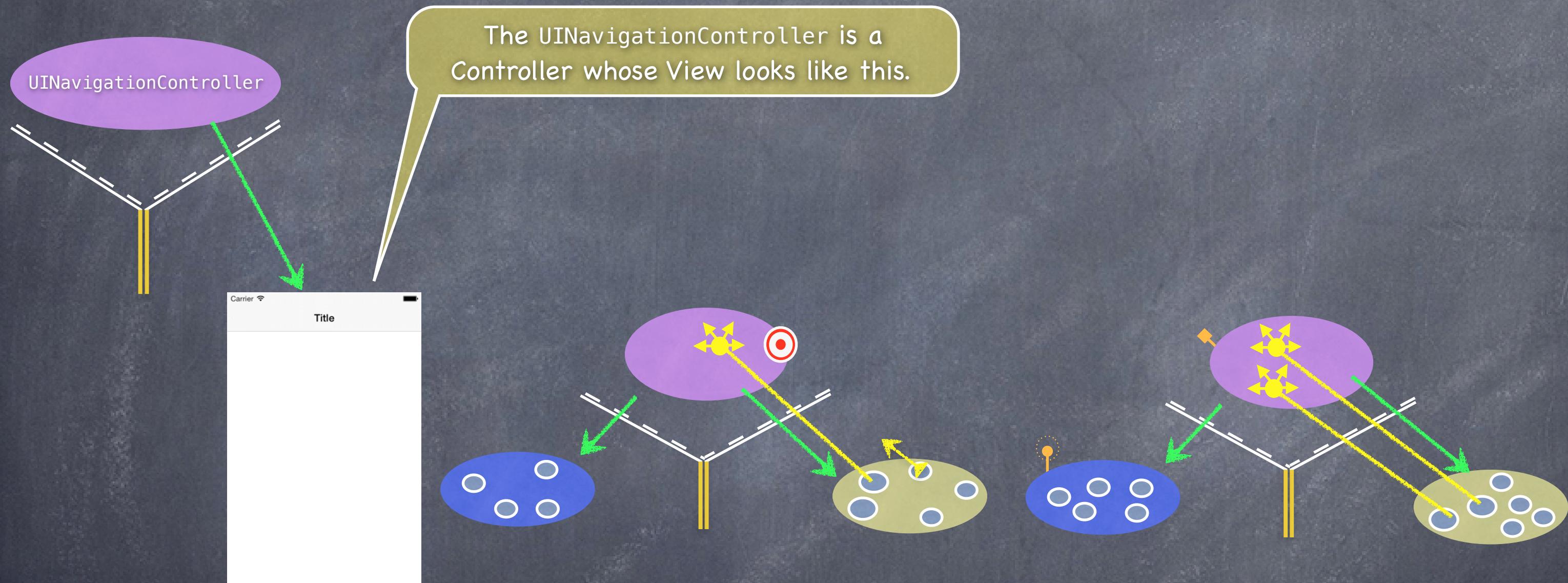
So I create a new MVC to encapsulate that functionality.

MVCs working together

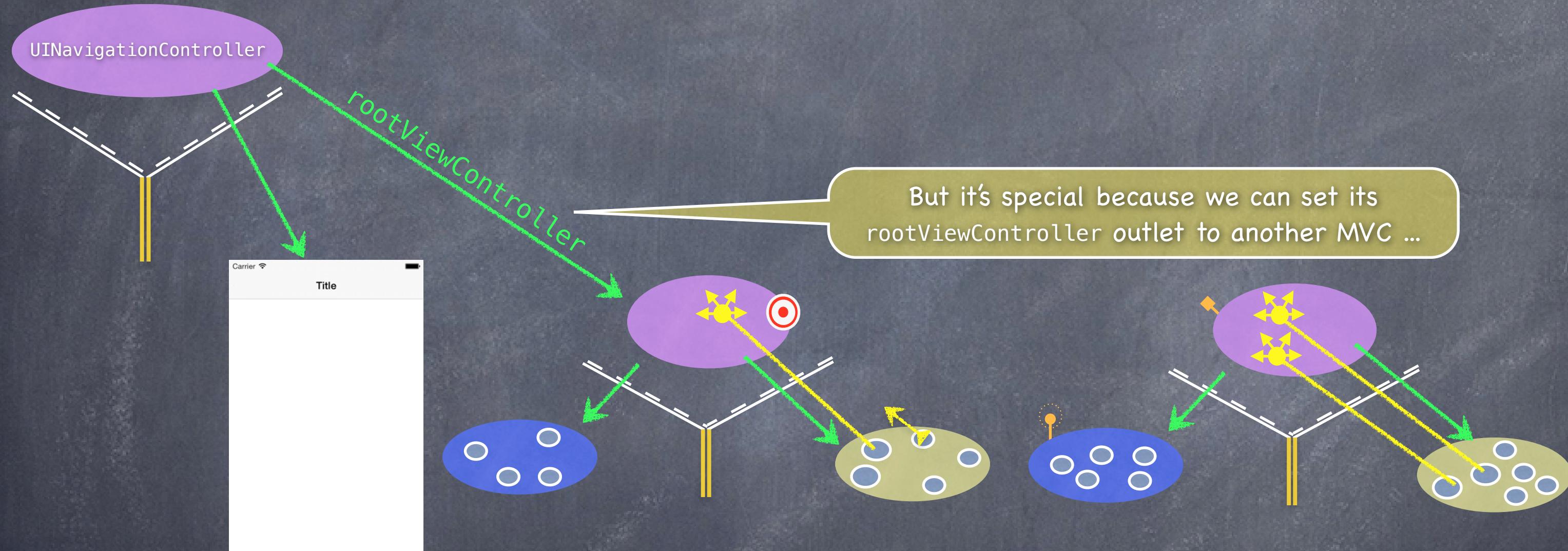
If the relationship between these two MVCs is “more detail,” we use a `UINavigationController` to let them share the screen.



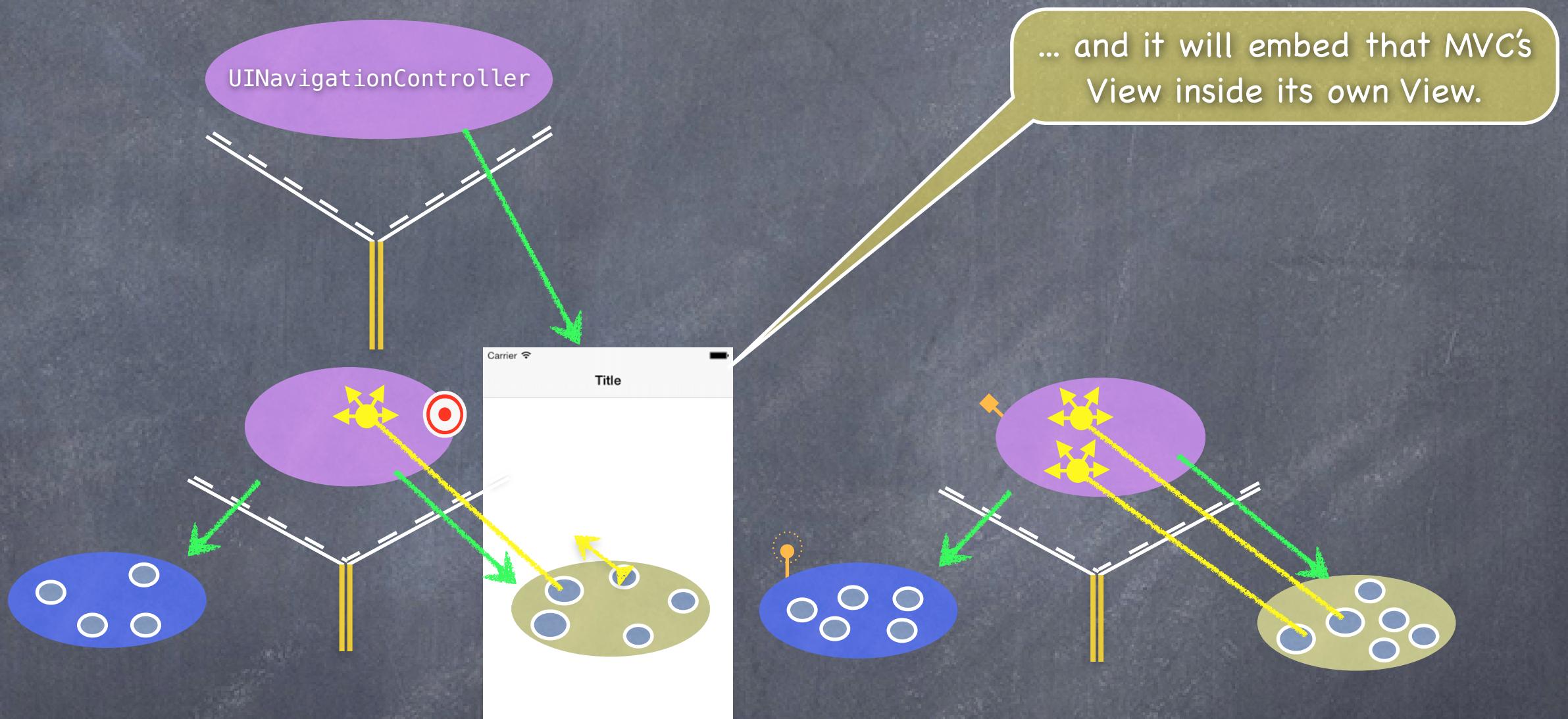
MVCs working together



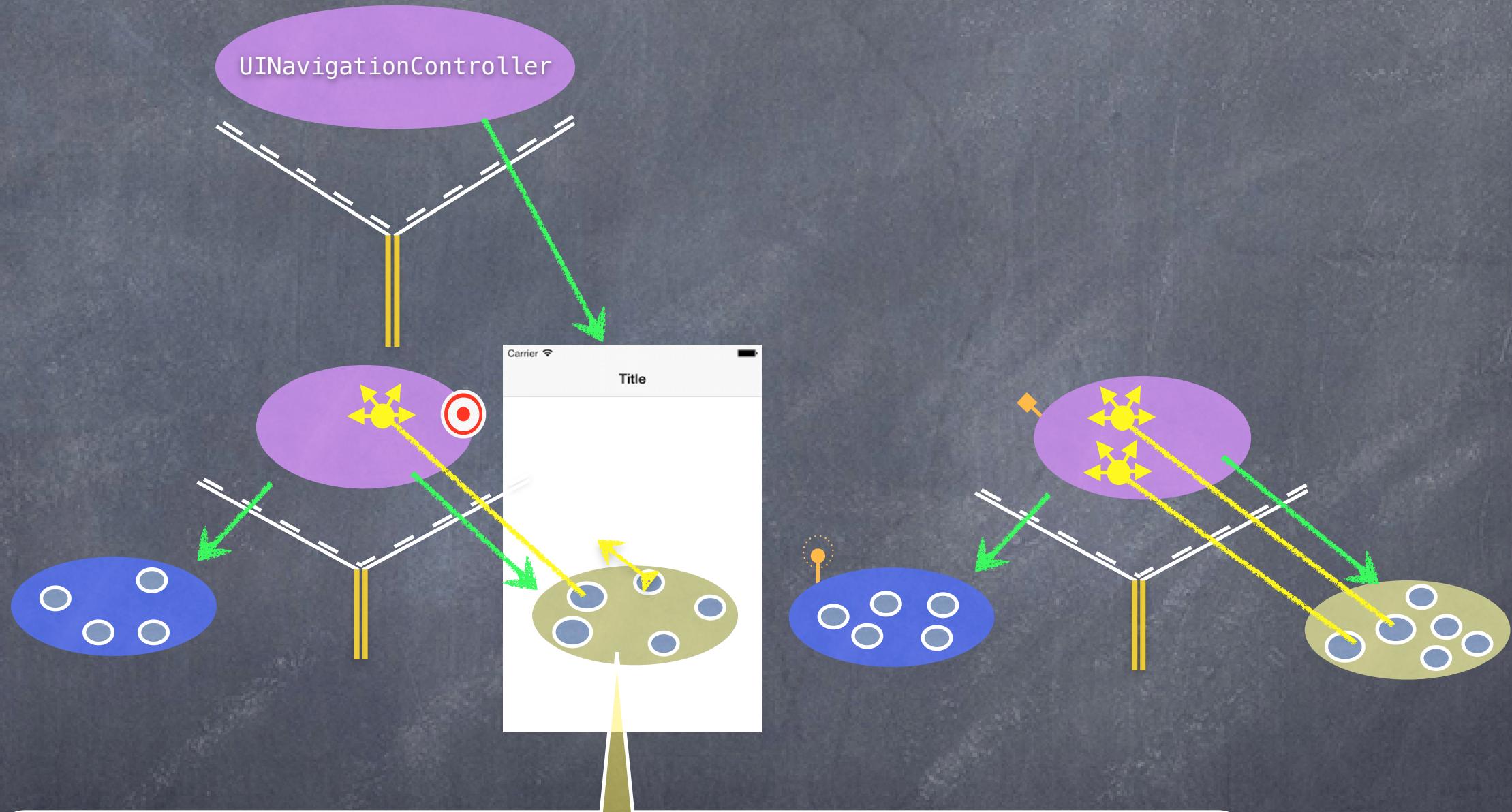
MVCs working together



MVCs working together

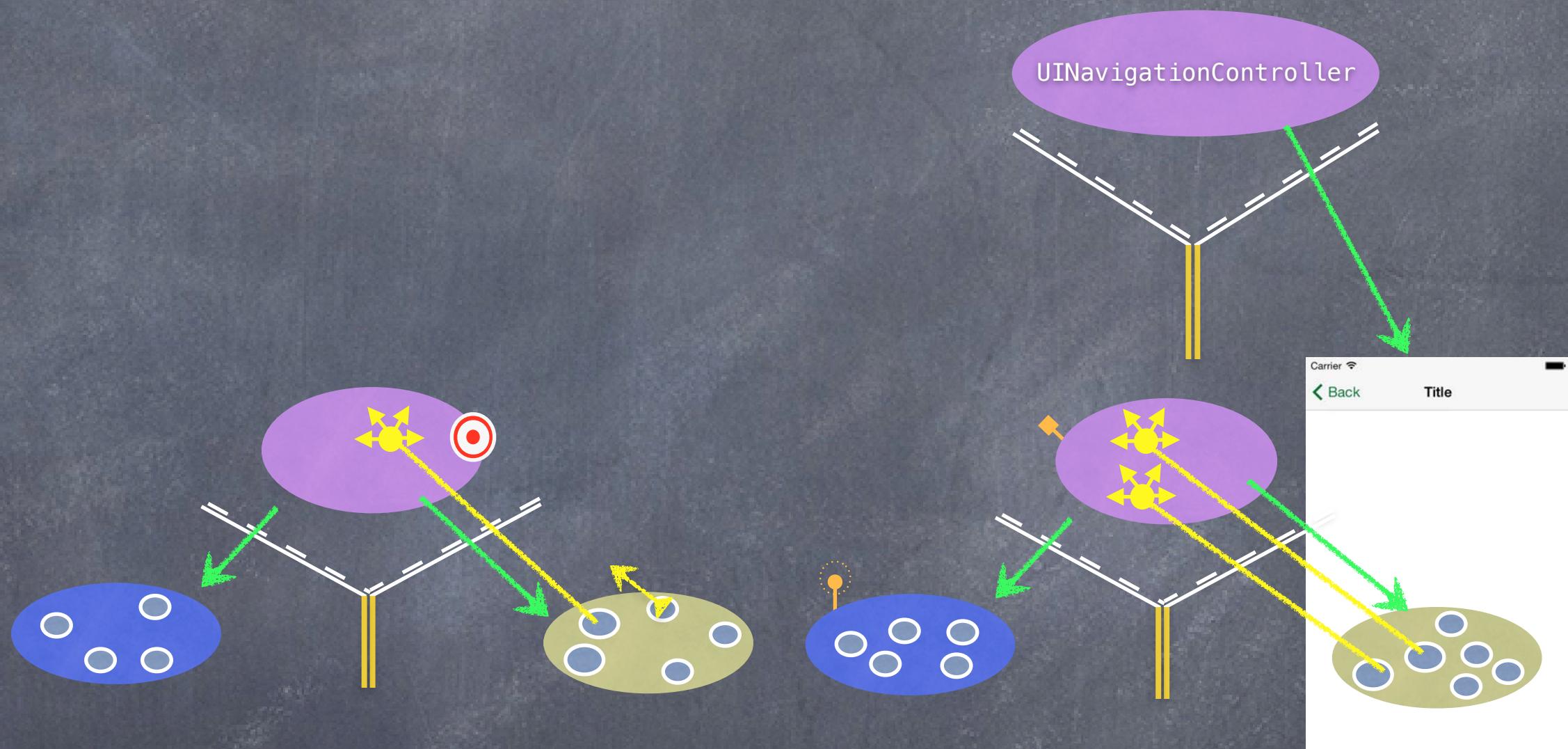


MVCs working together



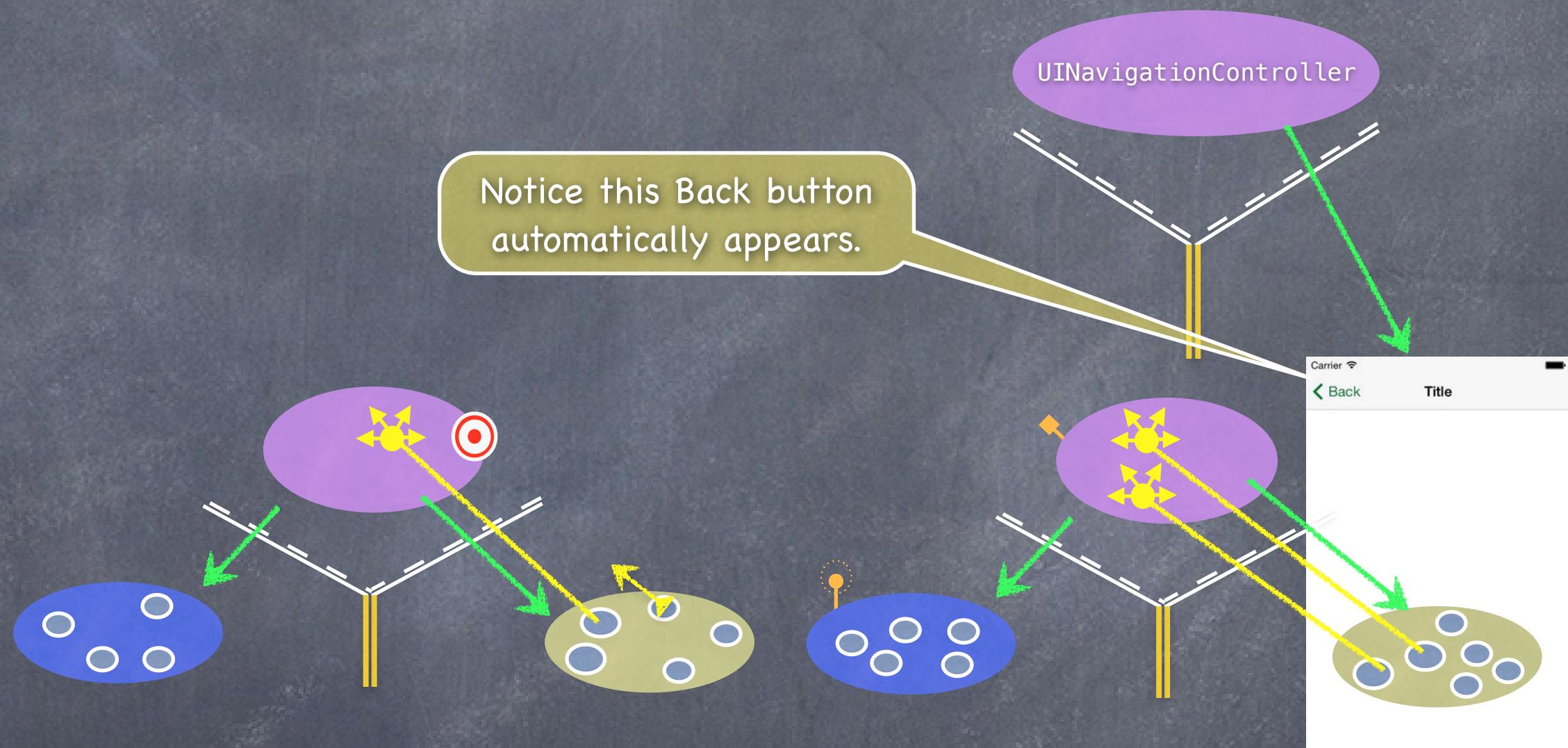
Then a UI element in this View (e.g. a `UIButton`) can segue to the other MVC and its View will now appear in the `UINavigationController` instead.

MVCs working together

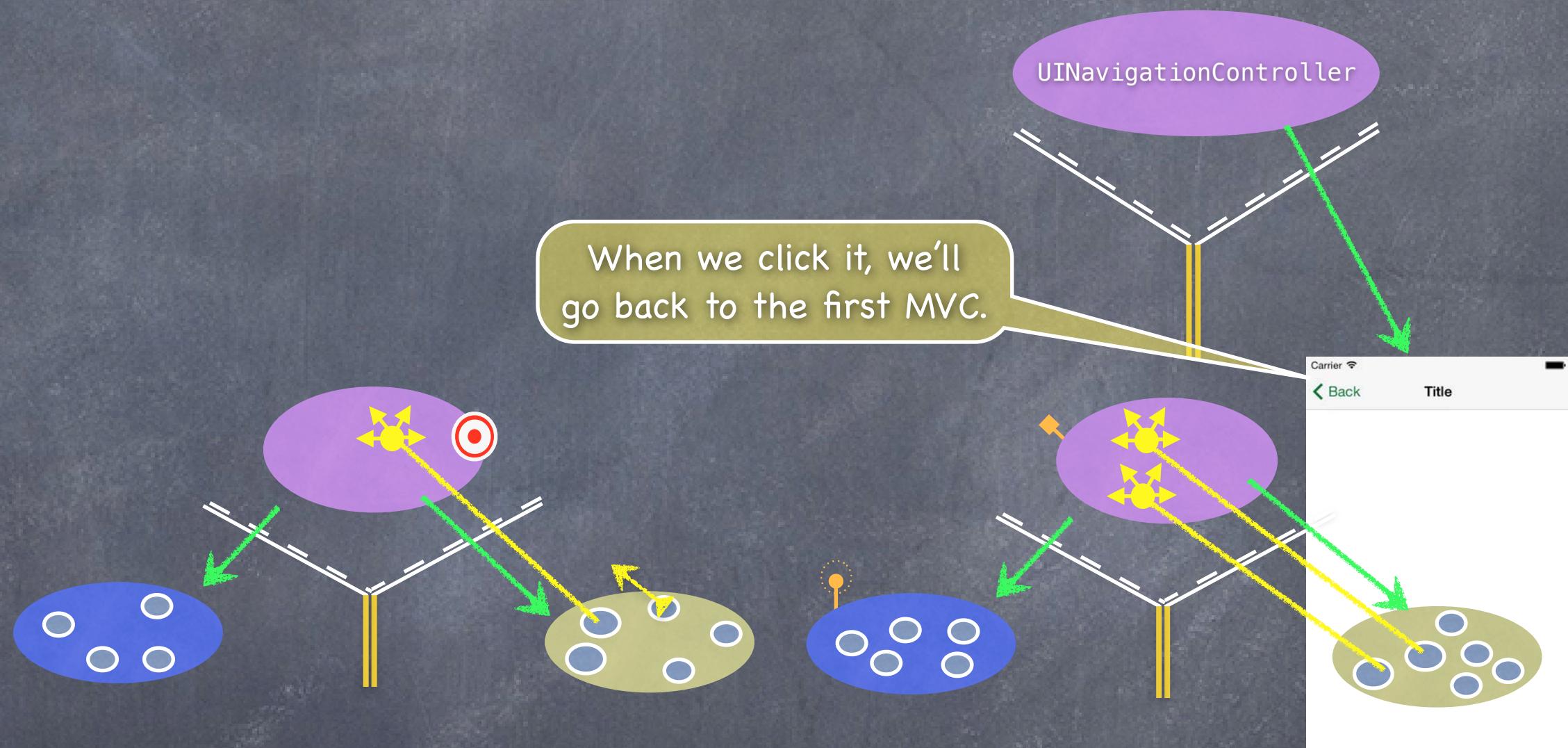


We call this kind of segue
a “push segue”.

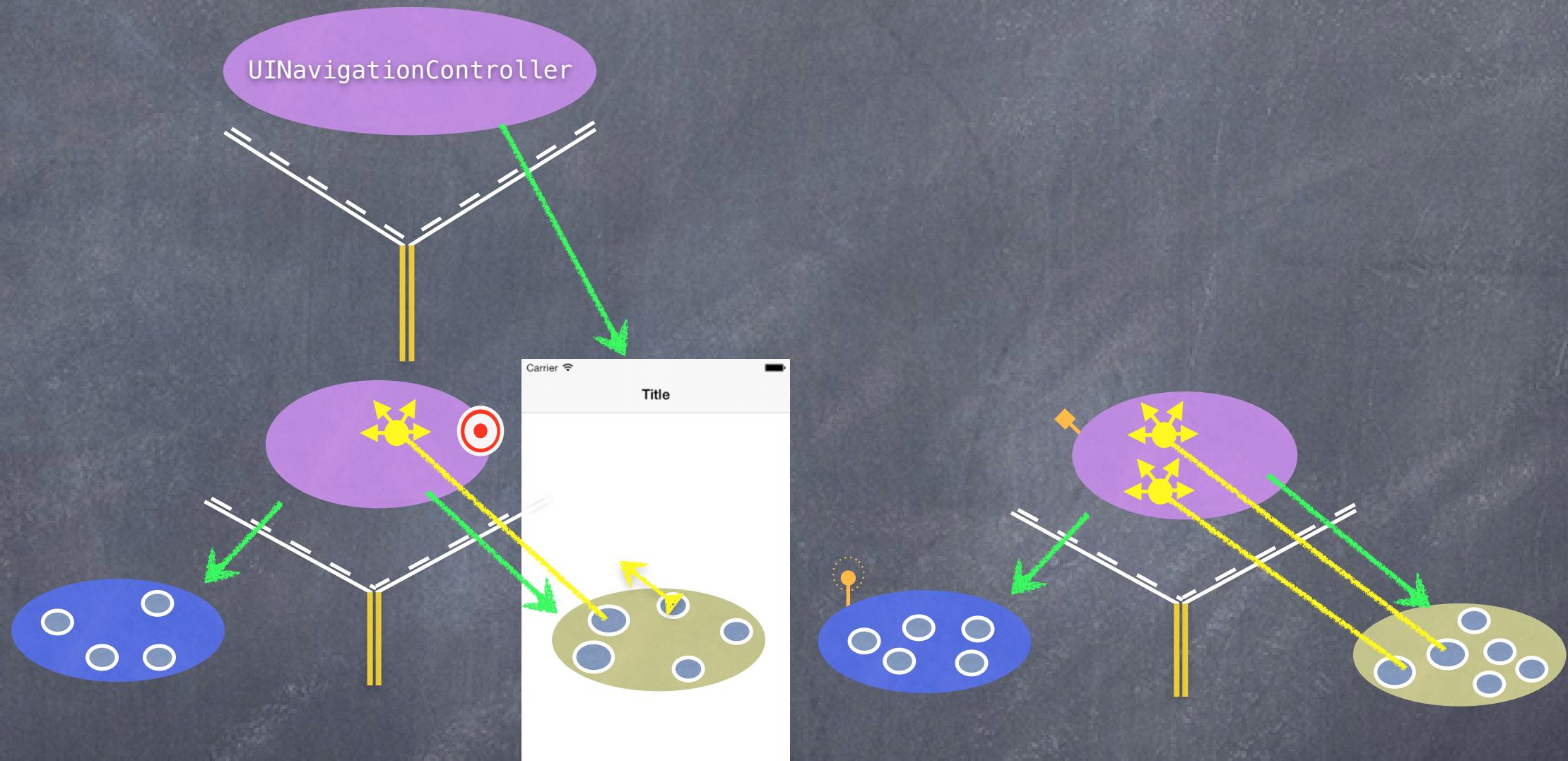
MVCs working together



MVCs working together



MVCs working together



Segues

- Let's talk about how the segue gets set up first
Then we'll look at how we create a UINavigationController in our storyboard.

So far, you've only had a single MVC in your application.

So how do you create a second one?

It's a two-step process.
First, drag a View Controller
into your Storyboard ...



Example View Controller

... second, set its class.
This is almost always a class
that you create using
File > New > File ...
Don't forget that it has to be
a subclass of
UIViewController.

It is a VERY common mistake
to forget this step!
If you do, you'll wonder why you
can't hook up any outlets or
actions inside this MVC!

Custom Class		
Identity	Type	Value
Storyboard ID	UIStoryboard	ExampleViewController
Restoration ID	UINavigationController	GLKViewController
Use Storyboard ID	UICollectionViewController	UIActivityViewController
	UIImagePickerController	



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

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

Collection View Controller - A
controller that manages a
collection view.

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

Tab Bar Controller - A
controller that manages a set of

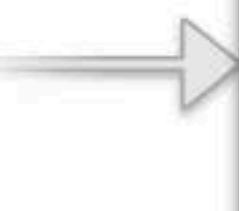
Stanford CS193p
Fall 2013



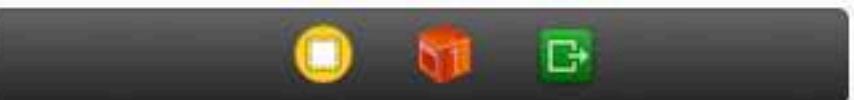
We call a particular layout of a View for a Controller in Xcode a "scene".

This is a scene.

This is a scene.



View Controller



View

Mode **Scale To Fill**
Tag **0**
Interaction User Interaction Enabled
 Multiple Touch
Alpha **1**
Background **White Color**
Tint **Default**
Drawing Opaque Hidden
 Clears Graphics Context
 Clip Subviews
 Autoresizes Subviews
Stretching **0 0**
X **0** Y **0**
Width **1** Height **1**

Label Label - A variably sized amount of static text.

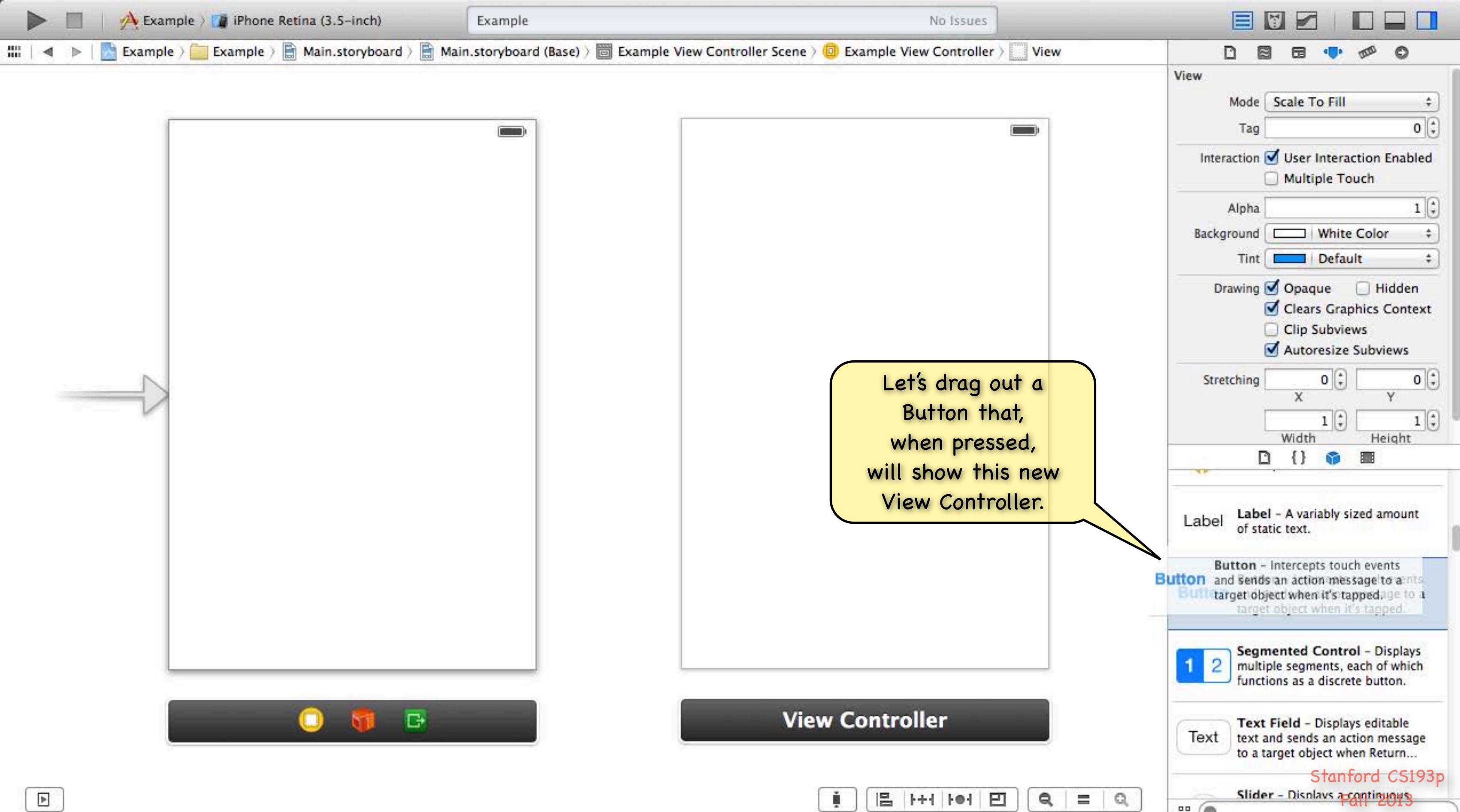
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

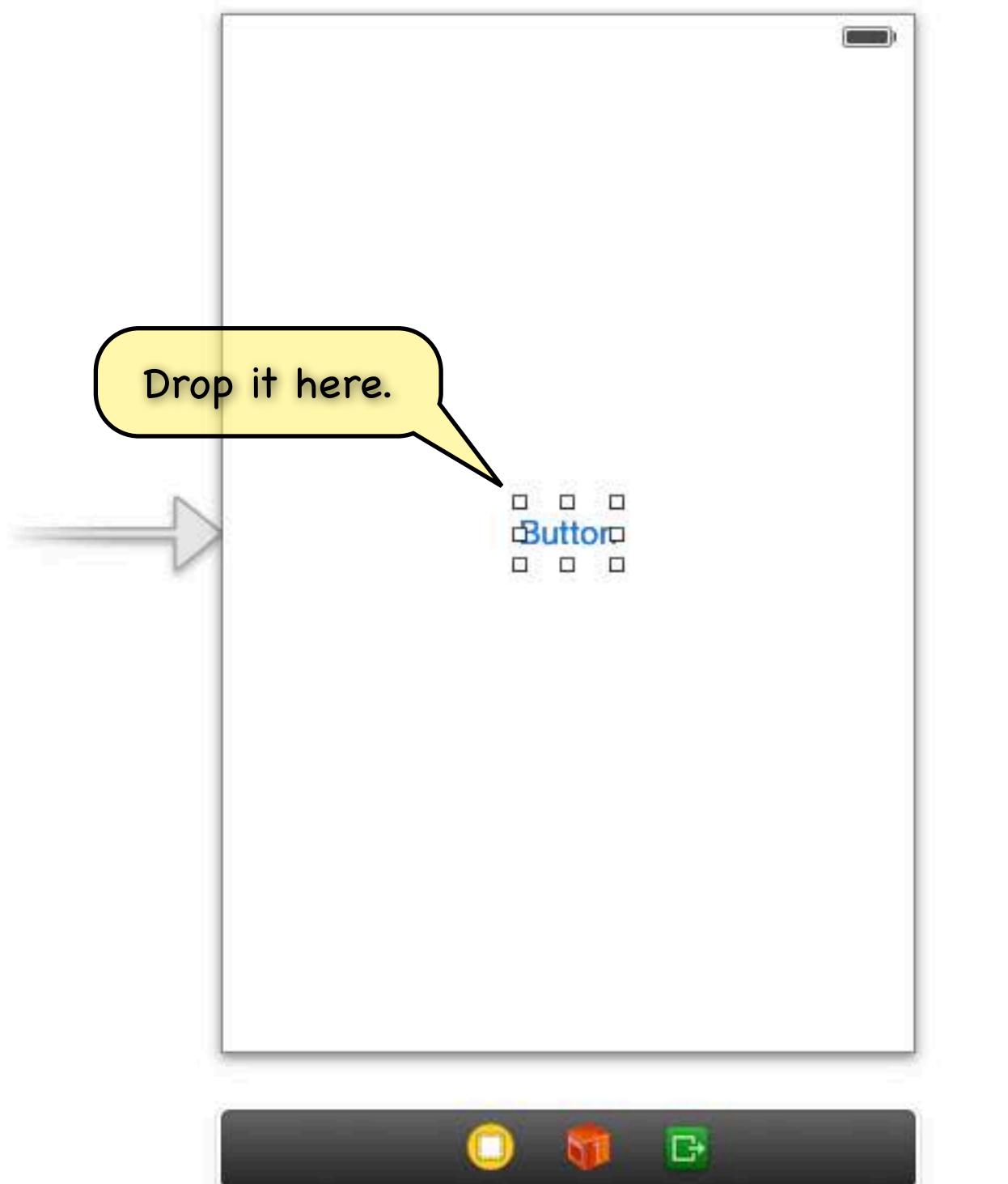
Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text Field - Displays editable text and sends an action message to a target object when Return...

Stanford CS193p

Fall 2013





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 0.0

Width: Height

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Label Label - A variably sized amount of static text.

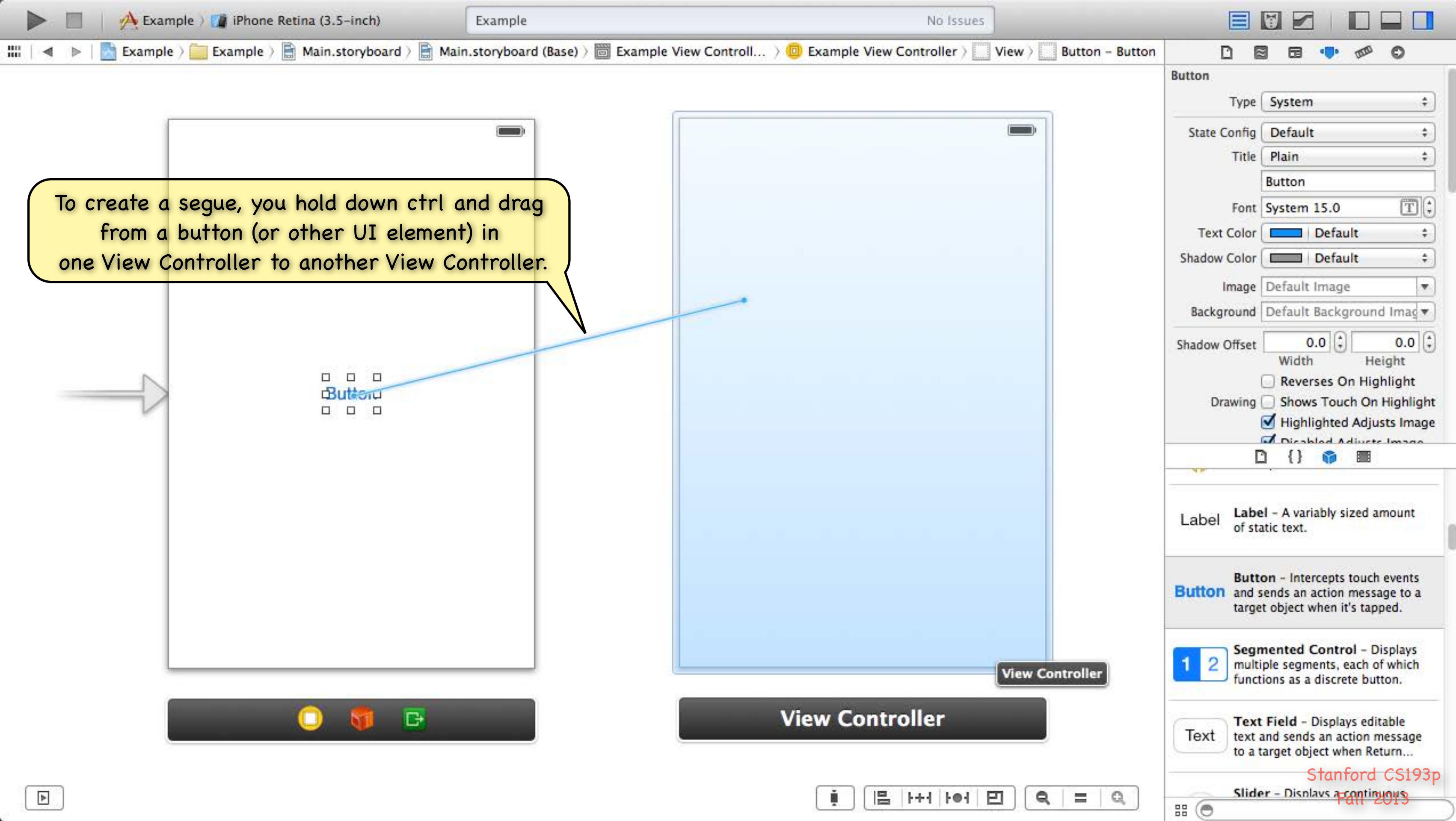
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text Field - Displays editable text and sends an action message to a target object when Return...

Slider Slider - Displays a continuous

Stanford CS193p Fall 2013





When you let go of the mouse, Xcode will ask what sort of segue you want to occur when Button is pressed.

"Push" is the kind of segue you use when the two Controllers are inside a UINavigationController.

Action Segue
push
modal
custom



□ □ □
Button
□ □ □

View Controller



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 0.0
Width Height
<input type="checkbox"/> Reverses On Highlight
<input type="checkbox"/> Shows Touch On Highlight
<input checked="" type="checkbox"/> Highlighted Adjusts Image
<input checked="" type="checkbox"/> Disabled Adjusts Image
□ { } ⚙
Label Label - A variably sized amount of static text.
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.
Text Text Field - Displays editable text and sends an action message to a target object when Return...
Slider Slider - Displays a continuous

Example > iPhone Retina (3.5-inch) Example No Issues

Example > Example > Main.storyboard > Main.storyboard (Base) > Example View Controller Scene > Push segue from Button to View Controller

Storyboard Segue

Identifier **Do Something**

Identifier
The identifier for the segue object.
(read-only)

Related Methods
-[UIStoryboardSegue identifier]

This segue will be created.

Button

View Controller

Label Label - A variably sized amount of static text.

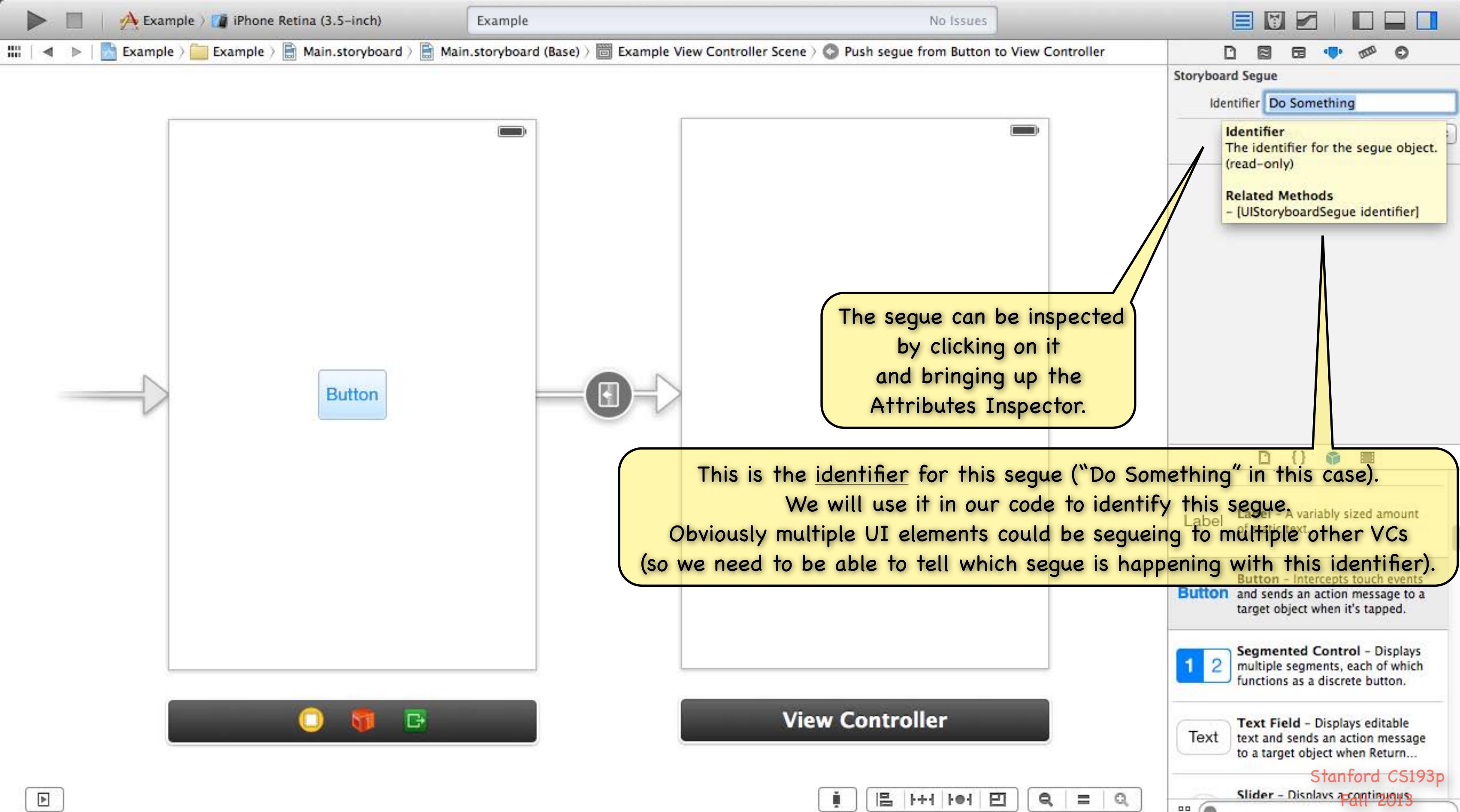
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

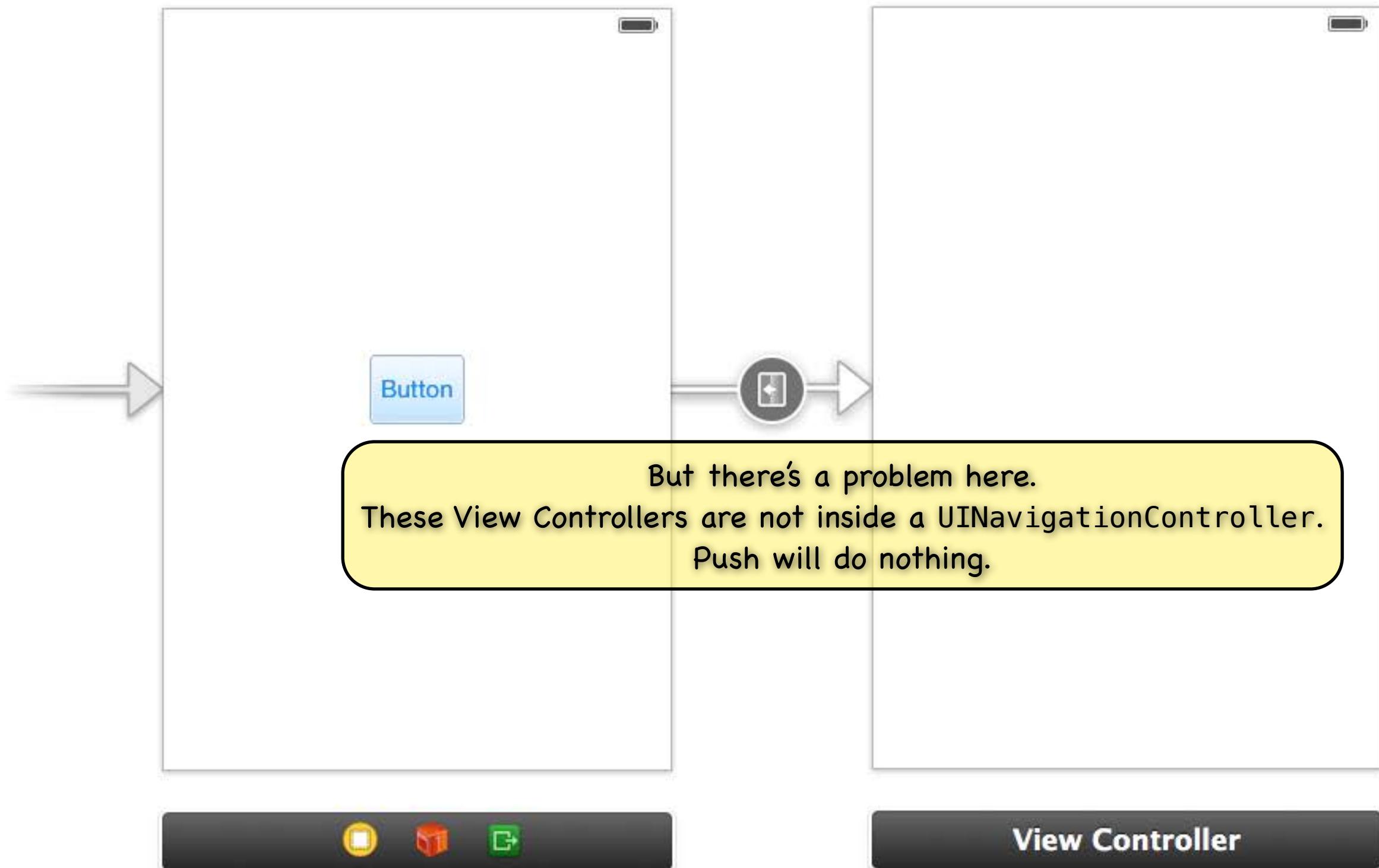
Segmented Control 1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text Field - Displays editable text and sends an action message to a target object when Return...
Slider Slider - Displays a continuous

Stanford CS193p Fall 2013

The screenshot shows the Xcode interface with the storyboard open. On the left, there is a blue rectangular view containing a white button labeled "Button". A segue arrow originates from the bottom right corner of the button and points to the right, leading into a second view. This second view has a dark grey header bar with the text "View Controller" in white. A yellow callout bubble with a black border and a black arrow points from the text "This segue will be created." towards the segue arrow. The top navigation bar shows the project path: "Example > Example > Main.storyboard > Main.storyboard (Base) > Example View Controller Scene > Push segue from Button to View Controller". The top right corner of the screen shows the Xcode toolbar with various icons. The bottom of the screen features the standard Mac OS X dock with several application icons.



**Storyboard Segue****Identifier** Do Something**Identifier**The identifier for the segue object.
(read-only)**Related Methods**

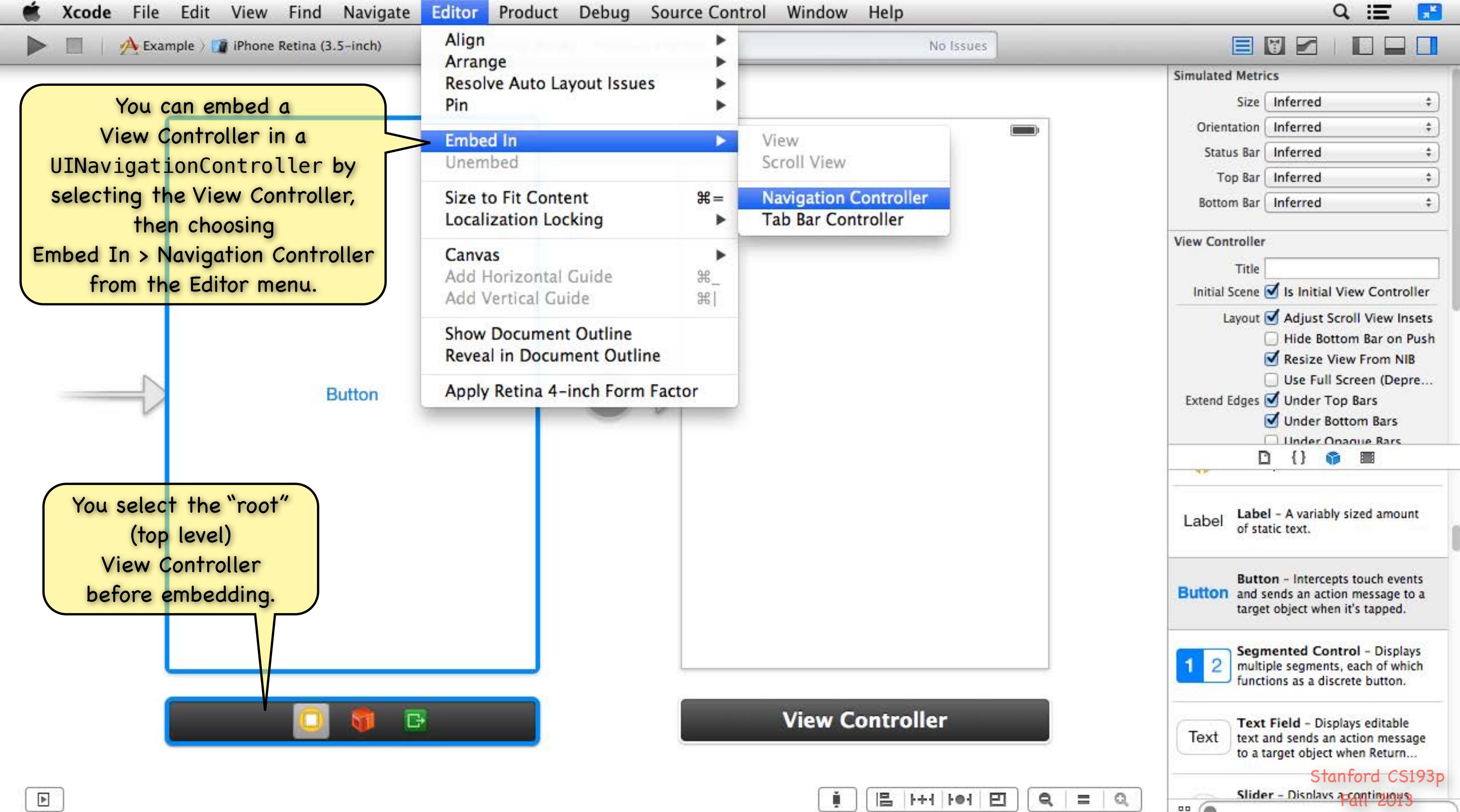
-[UIStoryboardSegue identifier]

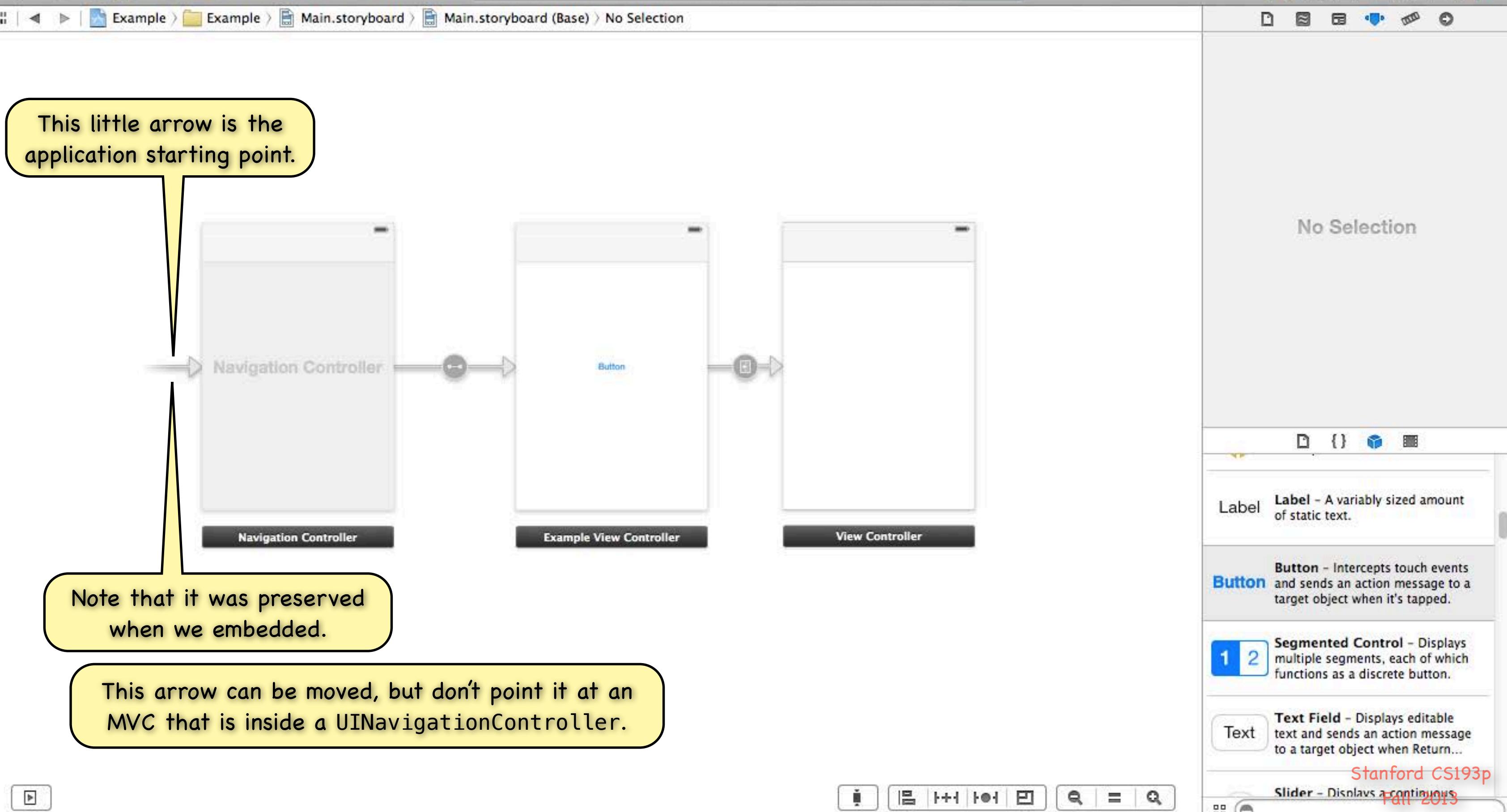
**Label** Label - A variably sized amount of static text.**Button** Button - Intercepts touch events and sends an action message to a target object when it's tapped.**Segmented Control** Segmented Control - Displays multiple segments, each of which functions as a discrete button.**Text** Text Field - Displays editable text and sends an action message to a target object when Return...

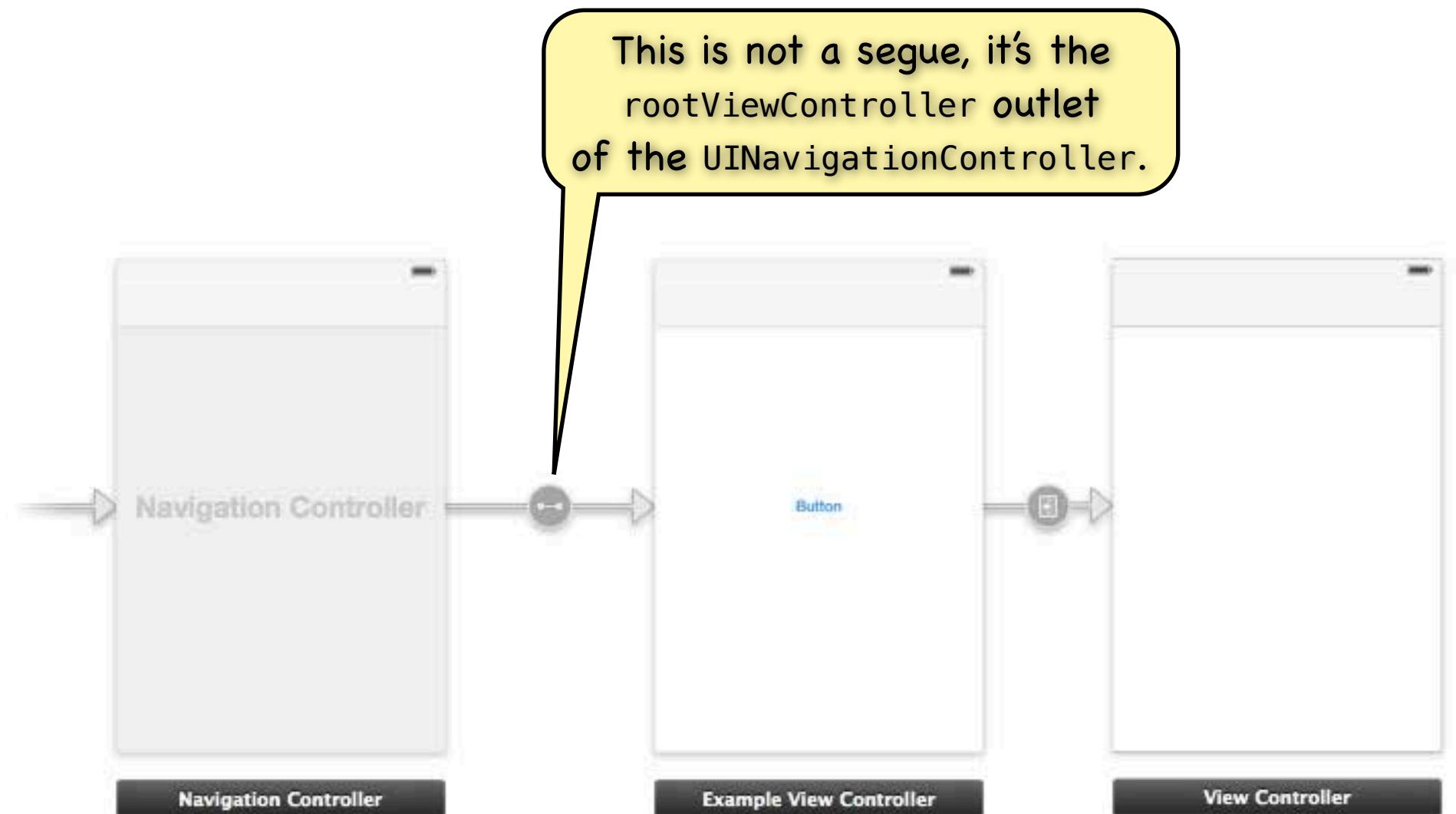
Stanford CS193p

Slider Slider - Displays a continuous

Fall 2013

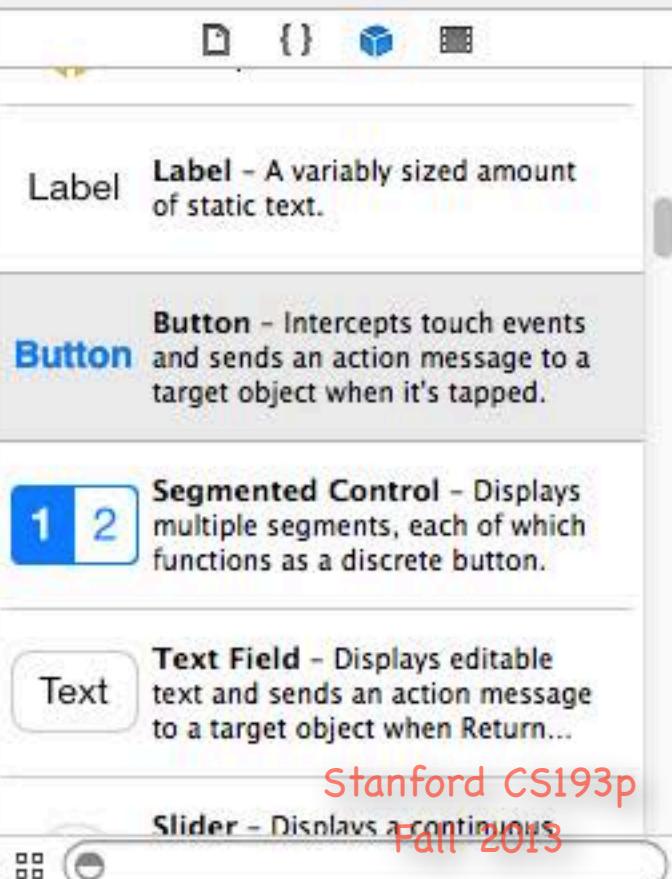


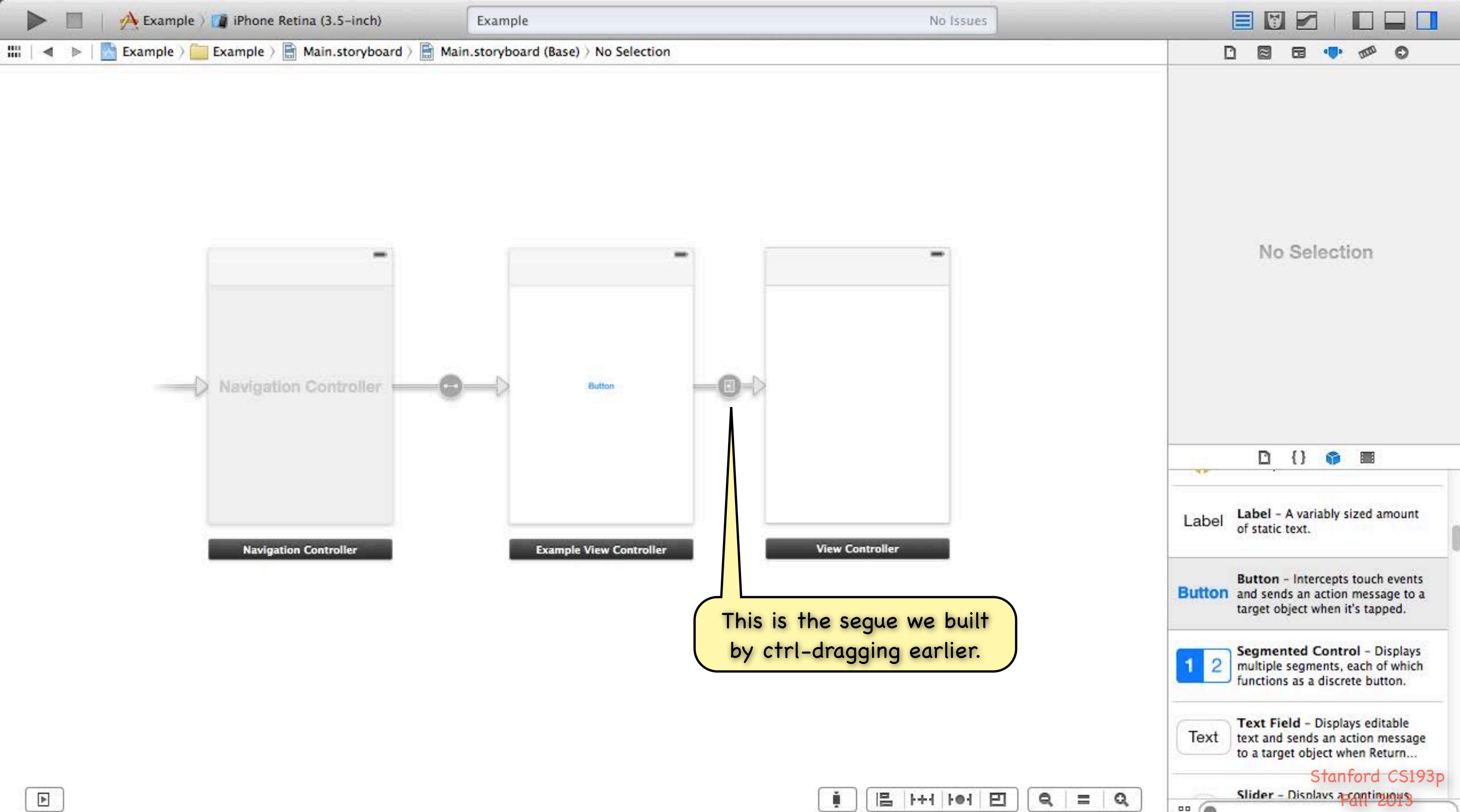




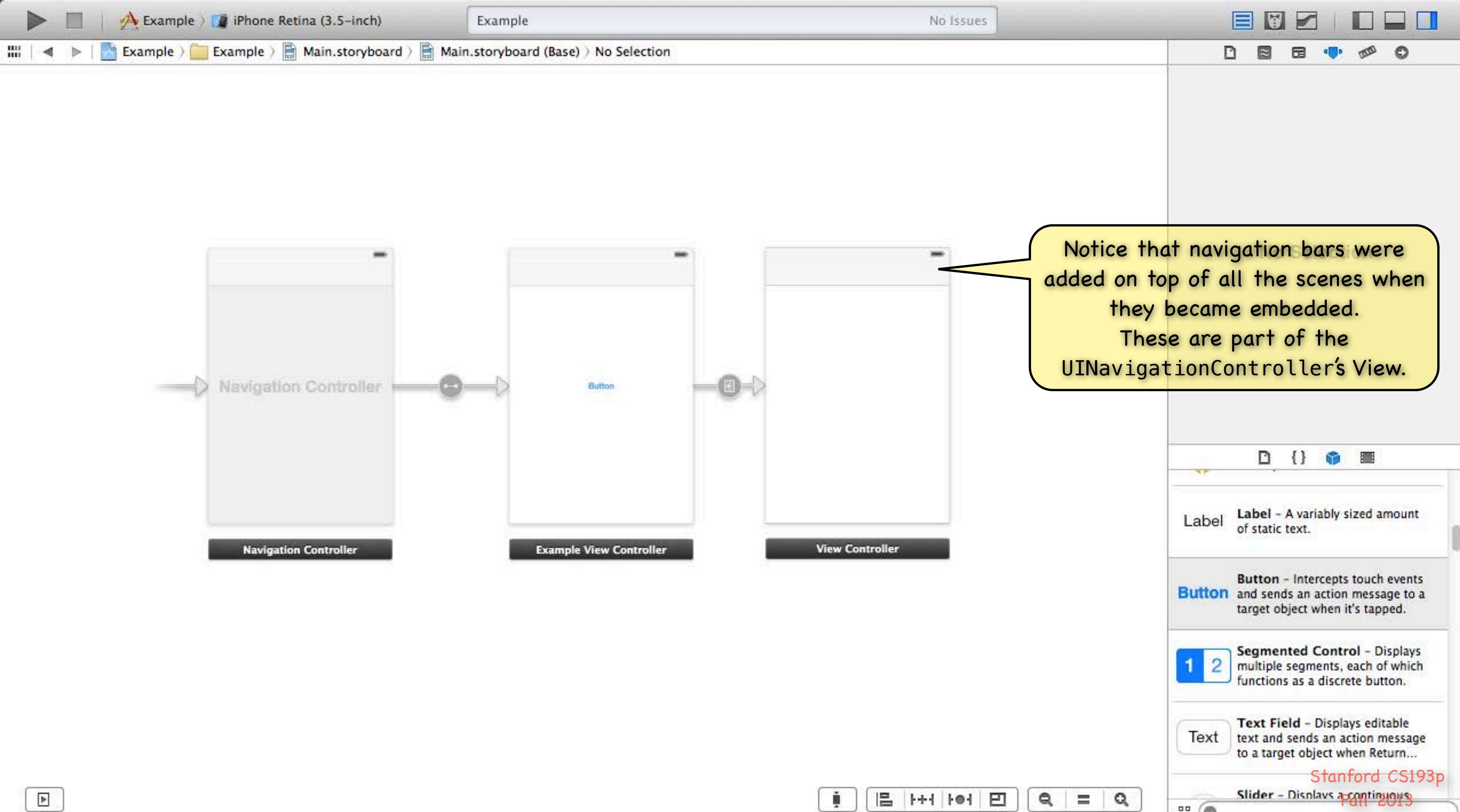
This is not a segue, it's the
rootViewController outlet
of the UINavigationController.

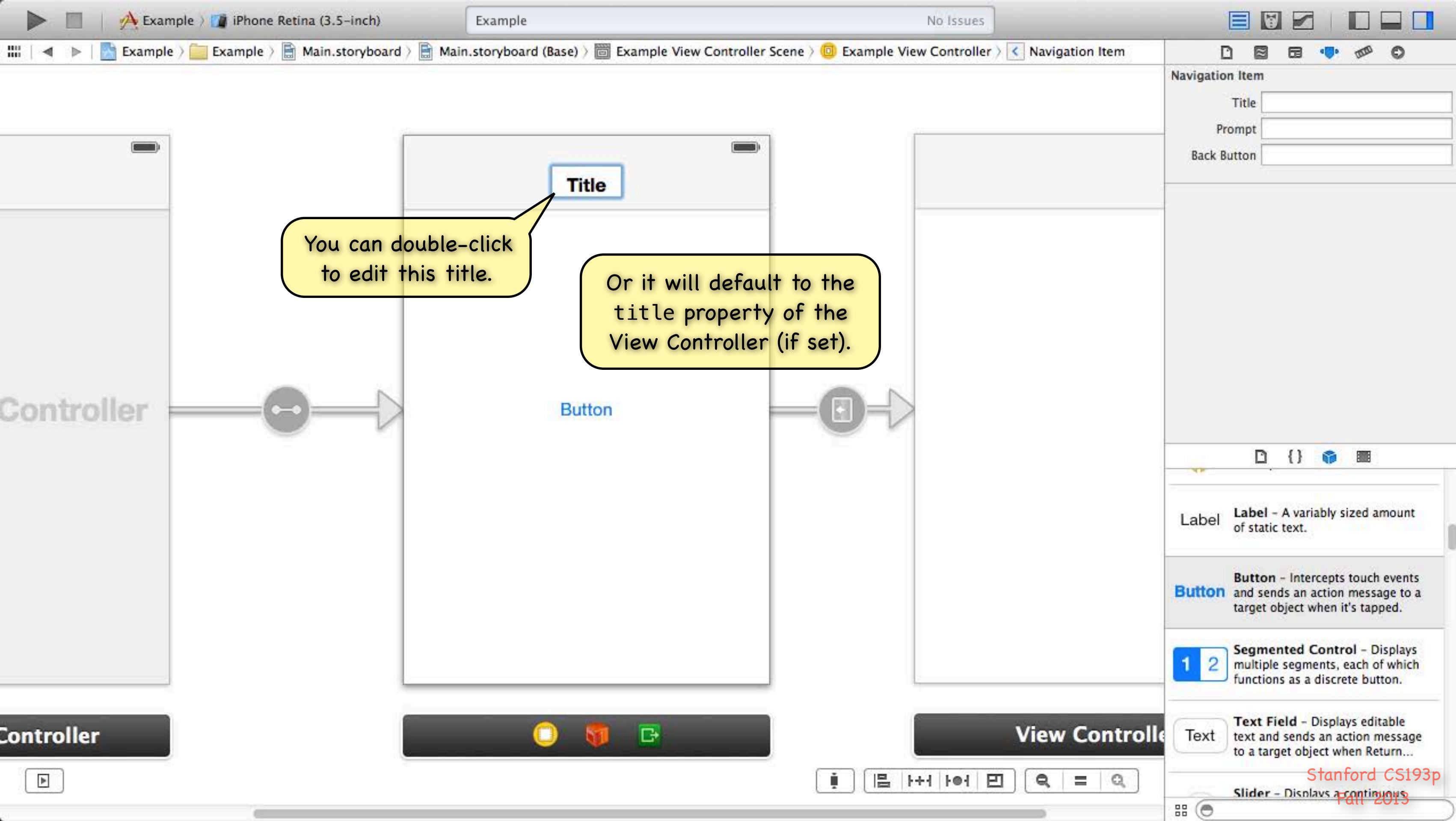
No Selection





This is the segue we built
by ctrl-dragging earlier.





Navigation Item

Title

Prompt

Back Button



Label Label - A variably sized amount of static text.

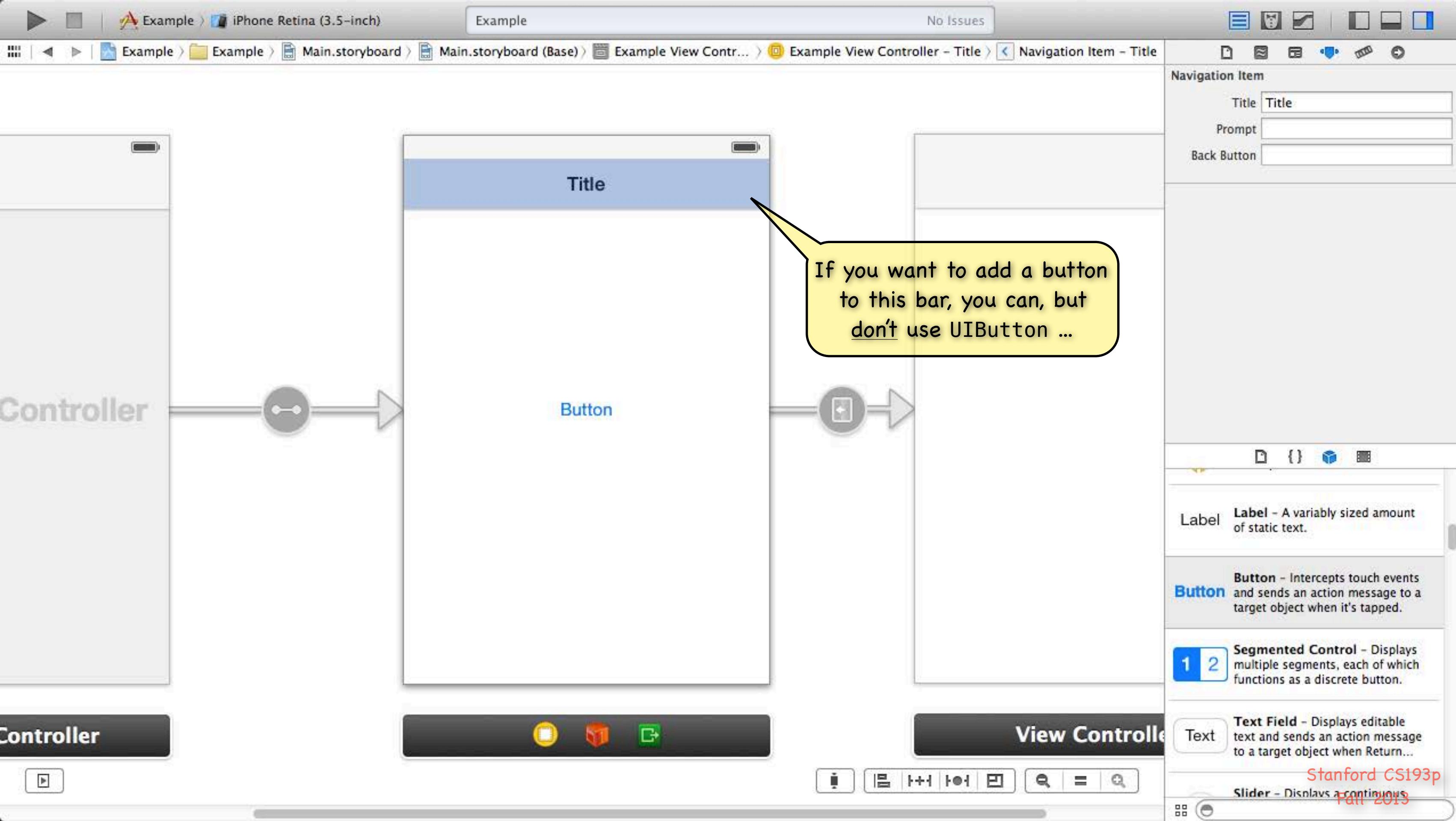
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

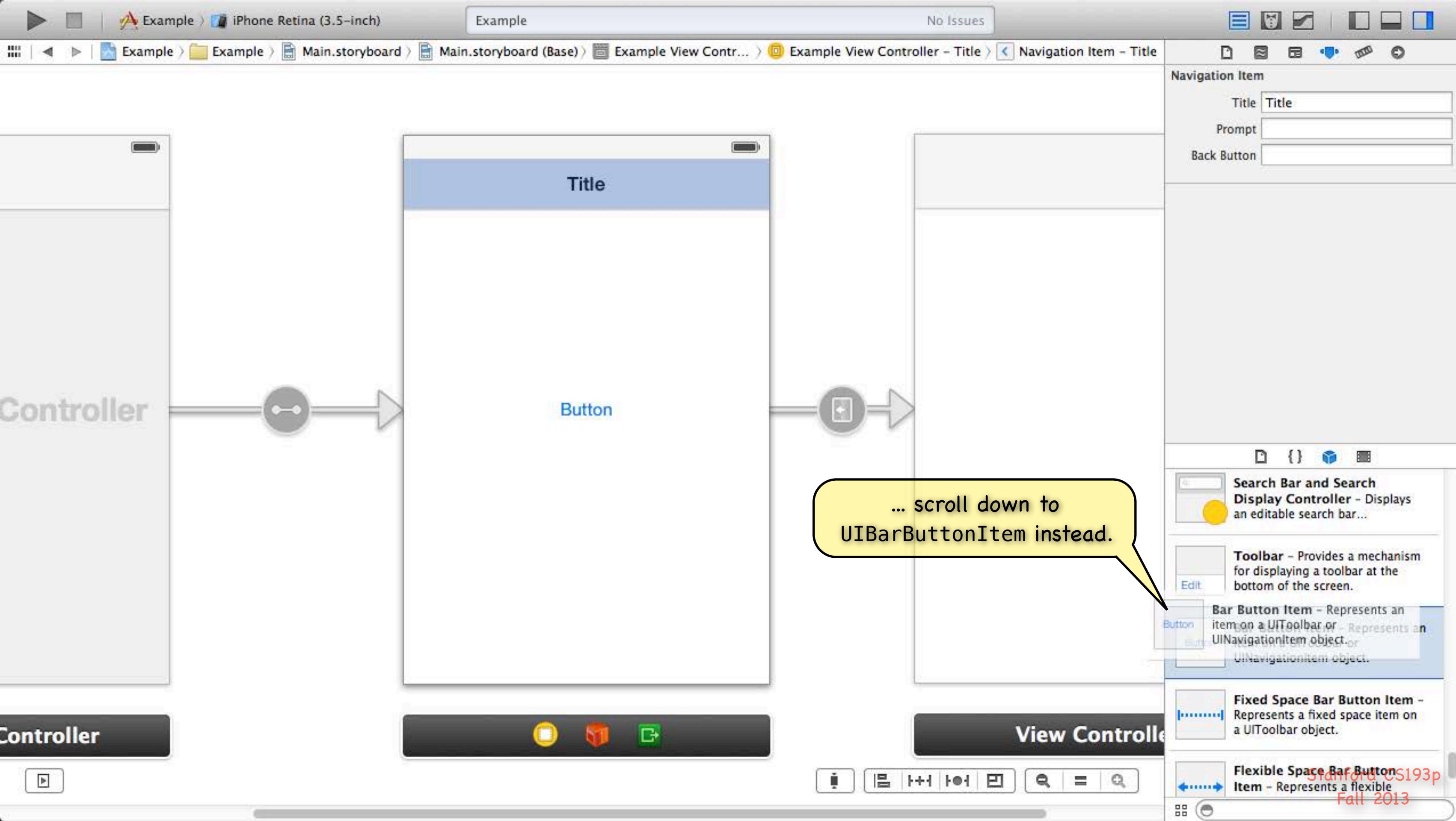
Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

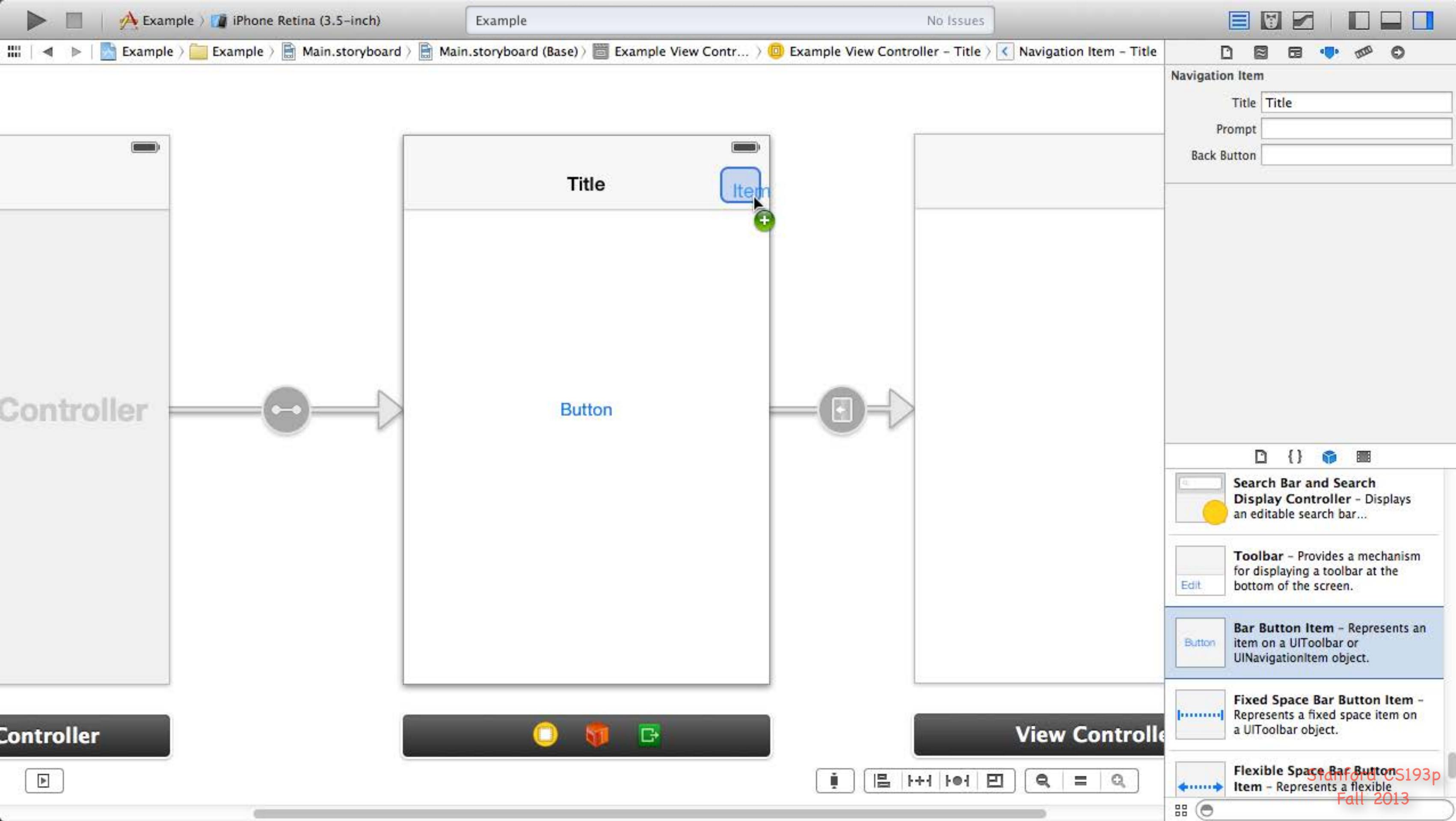
Text Text Field - Displays editable text and sends an action message to a target object when Return...

Stanford CS193p

Fall 2013







Example > iPhone Retina (3.5-inch)

Example

No Issues

Example > Example > Main.storyboard > Main.storybo... > Example View... > Example View... > Navigation Item - Title > Bar Button Item - Item

Bar Button Item

Style Bordered

Identifier Custom

Tint Default

Bar Item

Title Item

Image

Tag 0

Enabled

Controller

Controller

View Controller

Button

This button is now associated with this View Controller in this scene and will be displayed when this View Controller is the currently-showing scene in the UINavigationController.

The screenshot shows the Xcode interface with a storyboard open. A navigation item titled 'Item' is selected. A bar button item is also present. A callout bubble points from the bar button item to a text annotation: 'This button is now associated with this View Controller in this scene and will be displayed when this View Controller is the currently-showing scene in the UINavigationController.' Below the storyboard, a toolbar contains three icons: a yellow square, an orange cube, and a green square. The bottom right corner of the screen has a red watermark reading 'Stanford CS193p Fall 2013'.

UINavigationController

When does a pushed MVC pop off?

Usually because the user presses the “back” button (shown on the previous slide).

But it can happen programmatically as well with this UINavigationController instance method

- (void)popViewControllerAnimated:(BOOL)animated;

This does the same thing as clicking the back button.

Somewhat rare to call this method. Usually we want the user in control of navigating the stack.

But you might do it if some action the user takes in a view makes it irrelevant to be on screen.

Example

Let's say we push an MVC which displays a database record and has a delete button w/this action:

```
- (IBAction)deleteCurrentRecord:(UIButton *)sender
{
    // delete the record we are displaying
    // we just deleted the record we are displaying!
    // so it does not make sense to be on screen anymore, so pop
    [self.navigationController popViewControllerAnimated:YES];
}
```

Notice that all UIViewControllers know the UINavigationController they are in.
This is nil if they are not in one.

View Controller

- ⦿ Other kinds of segues besides Push

Replace - Replaces the right-hand side of a UISplitViewController (iPad only)

Popover - Puts the view controller on the screen in a popover (iPad only)

Modal - Puts the view controller up in a way that blocks the app until it is dismissed

Custom - You can create your own subclasses of UIStoryboardSegue

- ⦿ We'll talk about iPad-related segues in future lectures

Replace & Popover

- ⦿ We'll talk about Modal segues later in the quarter too

People often use Modal UIs as a crutch, so we don't want to go to that too early.

View Controller

⌚ Firing off a segue from code

Sometimes it makes sense to segue directly when a button is touched, but not always.

For example, what if you want to conditionally segue?

You can programmatically invoke segues using this method in UIViewController:

– `(void)performSegueWithIdentifier:(NSString *)segueId sender:(id)sender;`

The segueId is set in the attributes inspector in Xcode (seen on previous slide).

The sender is the initiator of the segue (a UIButton or yourself (UIViewController) usually).

– `(IBAction)rentEquipment`

`{`

`if (self.snowTraversingTalent == Skiing) {`

`[self performSegueWithIdentifier:@"AskAboutSkis" sender:self];`

`} else {`

`[self performSegueWithIdentifier:@"AskAboutSnowboard" sender:self];`

`}`

`}`

Segues

When a segue happens, what goes on in my code?

The segue offers the source VC the opportunity to “prepare” the new VC to come on screen. This method is sent to the VC that contains the button that initiated the segue:

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"DoSomething"]) {
        if ([segue.destinationViewController isKindOfClass:[DoSomethingVC class]]) {
            DoSomethingVC *doVC = (DoSomethingVC *)segue.destinationViewController;
            doVC.neededInfo = ...;
        }
    }
}
```

You should pass data the new VC needs here and “let it run.”

Think of the new VC as part of the View of the Controller that initiates the segue.

It must play by the same rules as a View.

For example, it should not talk back to you (except through blind communication like delegation).

Segues

- ⦿ You can prevent a segue from happening

Your Controller usually just always segues.

But if you respond **NO** to this method, it would prevent the identified segue from happening.

```
- (BOOL)shouldPerformSegueWithIdentifier:(NSString *)identifier sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"DoAParticularThing"]) {
        return [self canDoAParticularThing] ? YES : NO;
    }
}
```

Do not create “dead UI” with this (e.g. buttons that do nothing).

This is a very rare method to ever implement.

Unwinding

- There are also ways to unwind from a series of segues

Some people think of this as “reverse segueing”.

Used if you want to dismiss the VC you are in and go back to a previous VC that segued to you.

For example, what if you wanted to pop back multiple levels in a navigation controller?

(if you were only going back one level, you could just use `popViewControllerAnimated:`).

The little green button in the black bar at the bottom of a scene can be used to wire that up.

We will probably cover this when we talk about the Modal segue type (i.e. later).

You need to master segueing forward before you start thinking about going backward!



This is the “little green button.”

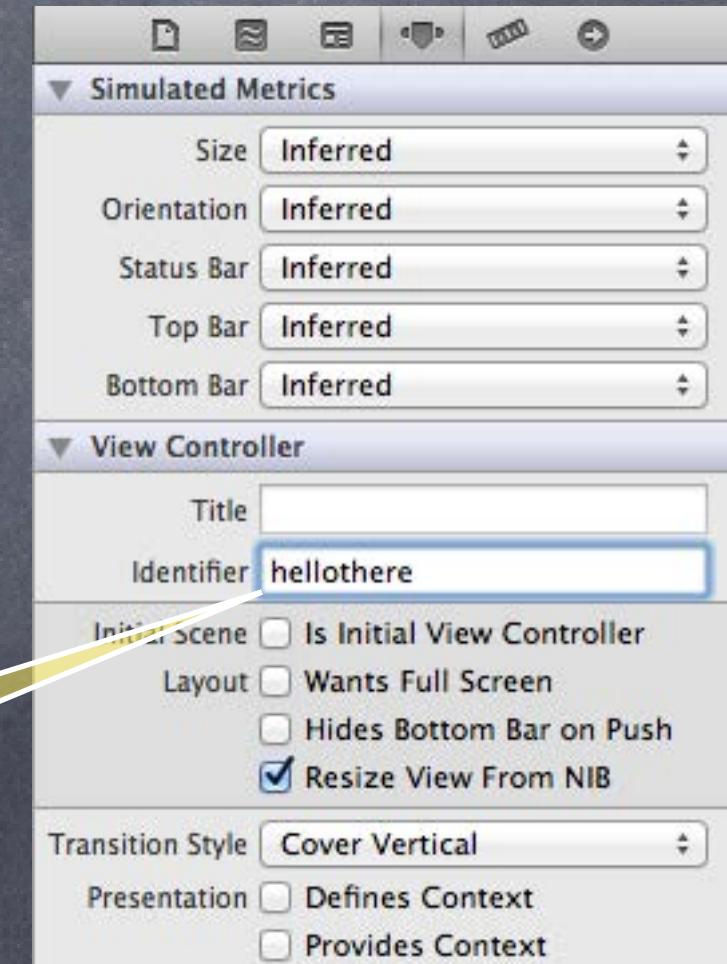
View Controller

⌚ Instantiating a UIViewController by name from a storyboard

Sometimes (very rarely) you might want to put a VC on screen yourself (i.e., not use a segue).

```
NSString *vcid = @“something”;  
UIViewController *controller = [storyboard instantiateViewControllerWithIdentifier:vcid];
```

Usually you get the storyboard above from `self.storyboard` in an existing UIViewController.
The identifier `vcid` must match a string you set in Xcode to identify a UIViewController there.



This UIViewController in the storyboard can be instantiated using the identifier “hellothere”.

View Controller

⌚ Instantiating a UIViewController by name from a storyboard

Sometimes (very rarely) you might want to put a VC on screen yourself (i.e., not use a segue).

```
NSString *vcid = @“something”;  
UIViewController *controller = [storyboard instantiateViewControllerWithIdentifier:vcid];
```

Usually you get the storyboard above from self.storyboard in an existing UIViewController.
The identifier vcid must match a string you set in Xcode to identify a UIViewController there.

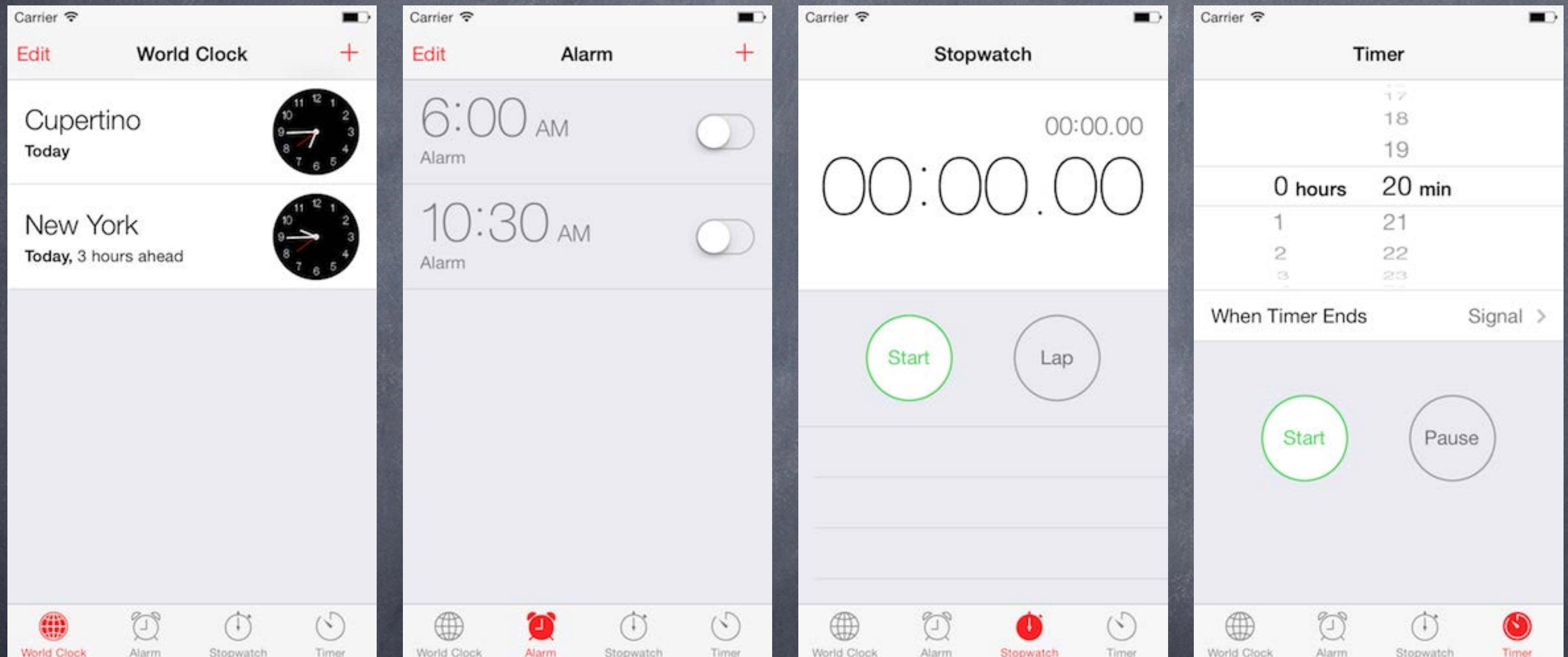
⌚ Example: creating a UIViewController in a target/action method

Lay out the View for a DoitViewController in your storyboard and name it “doit1”.

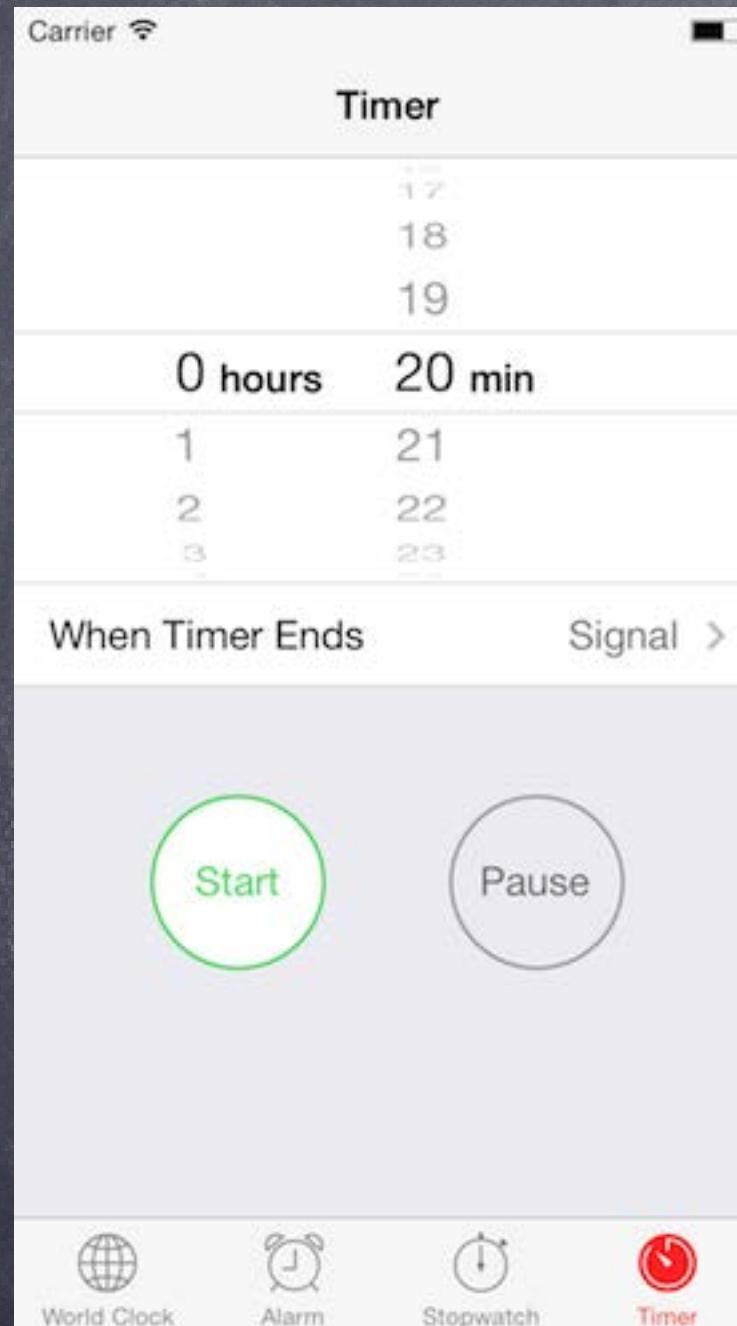
```
- (IBAction)doit  
{  
  
    DoitViewController *doit =  
        [self.storyboard instantiateViewControllerWithIdentifier:@“doit1”];  
    doit.infoDoitNeeds = self.info;  
    [self.navigationController pushViewController:doit animated:YES];  
}
```

Note use of self.navigationController again.

UITabBarController



UITabBarController



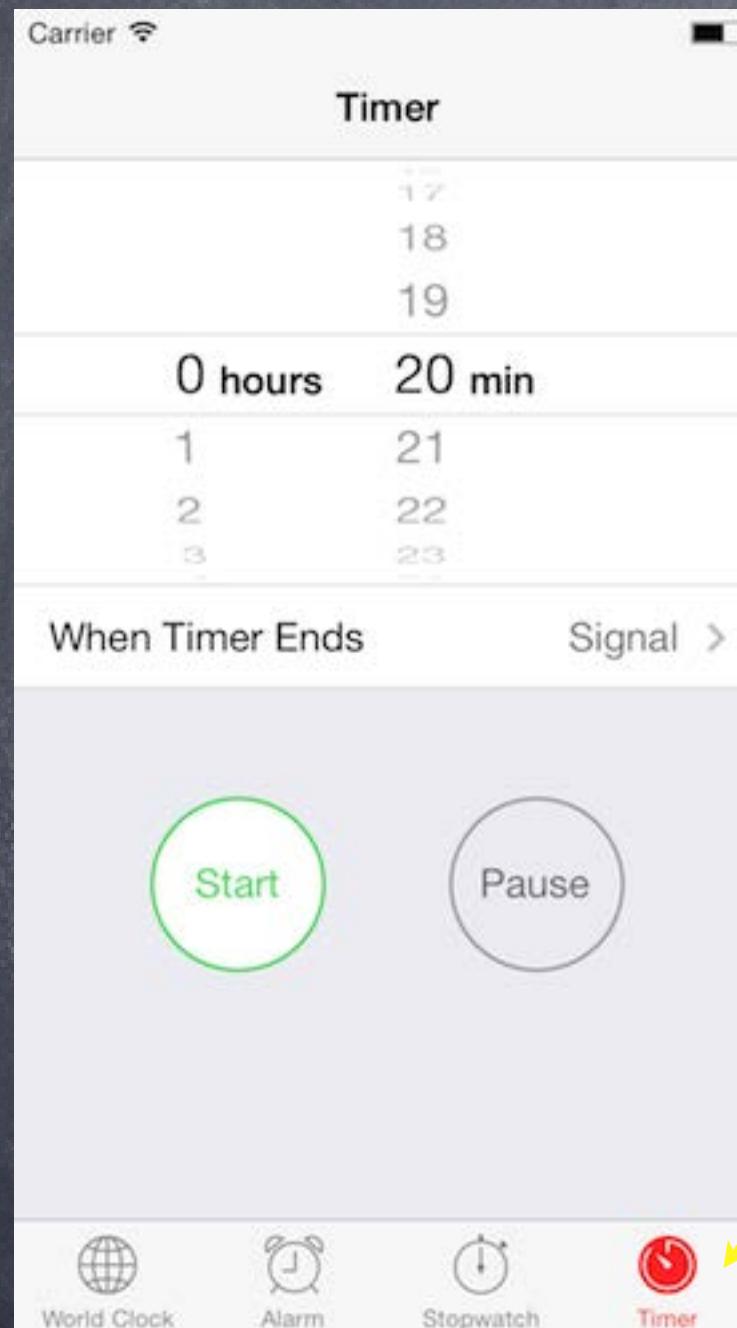
Tab Bar
Controller



You control drag to
create these
connections in Xcode.

Doing so is setting
`@property (nonatomic, strong) NSArray *viewControllers;`
inside your UITabBarController.

UITabBarController



Tab Bar
Controller

View Controller

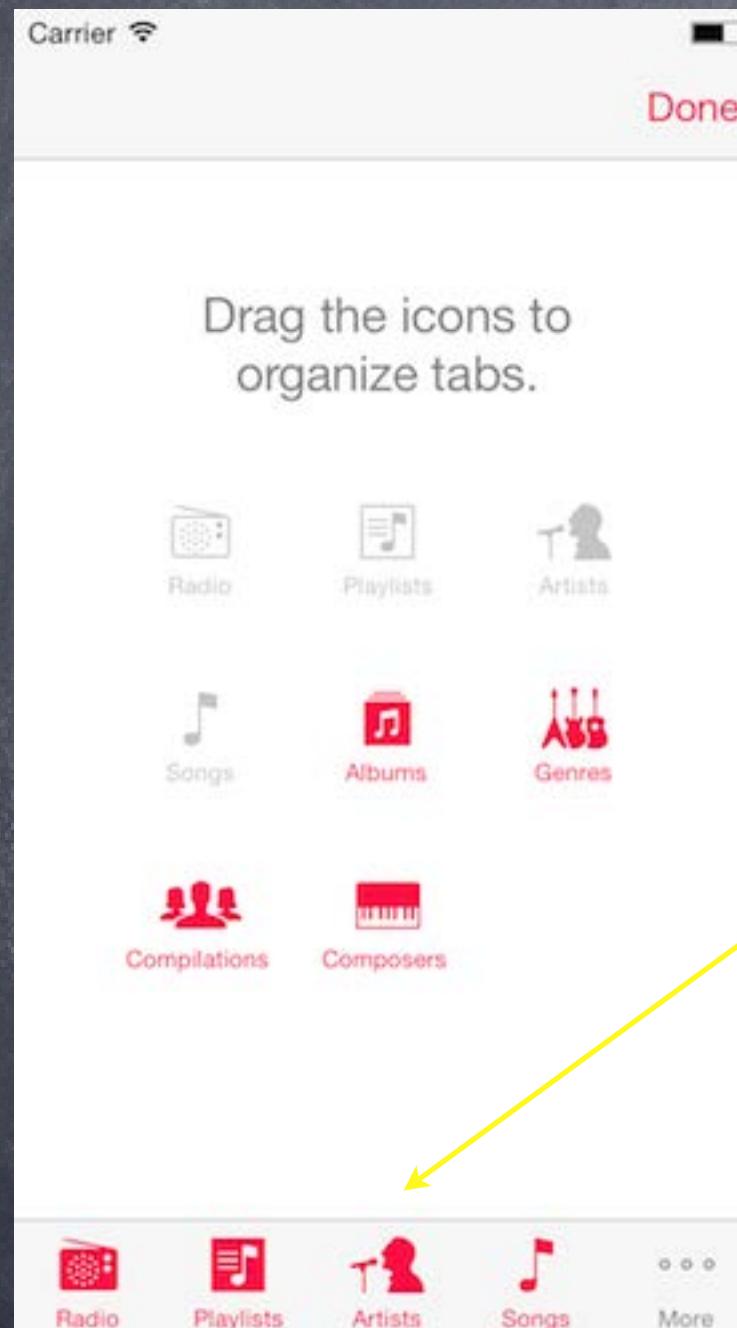
View Controller

View Controller

By default this is
the **UIViewController's**
title property
(and no image)

But usually you set
both of these in your
storyboard in Xcode.

UITabBarController

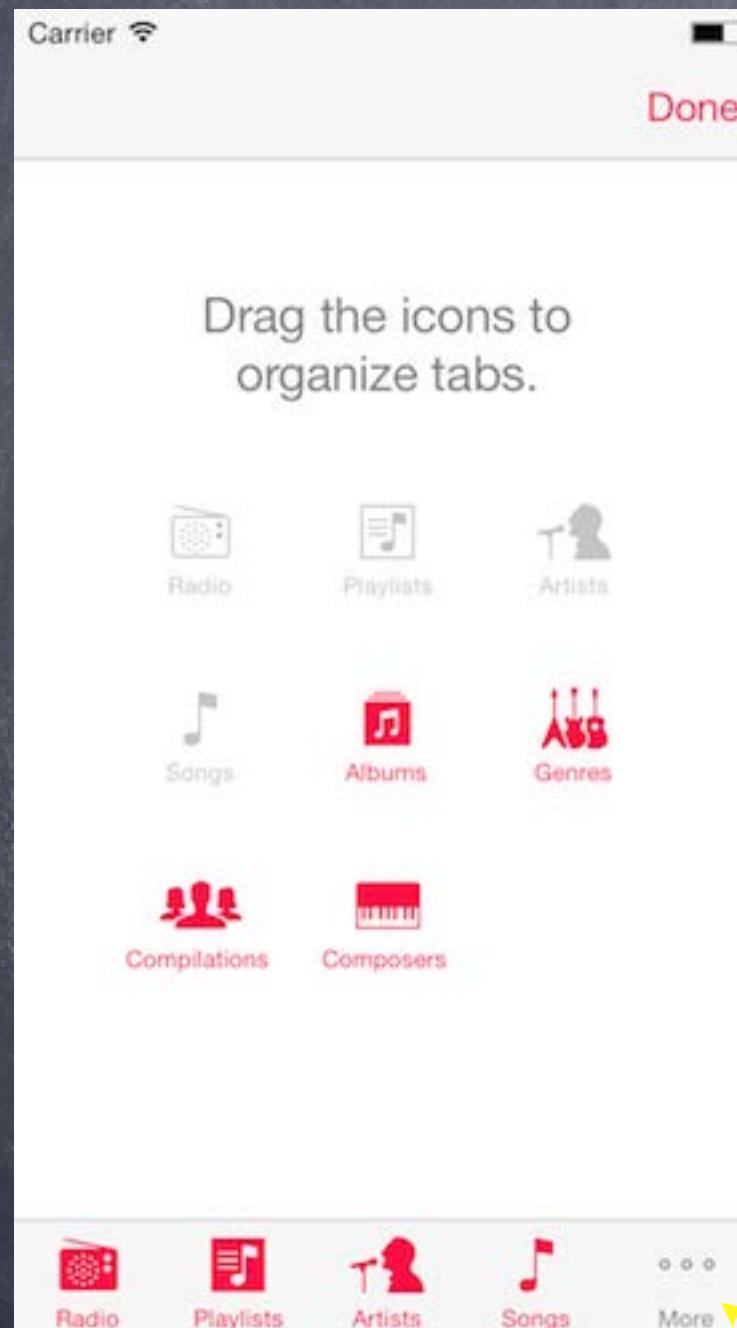


Tab Bar
Controller

What if there are
more than 4 View
Controllers?

View Controller

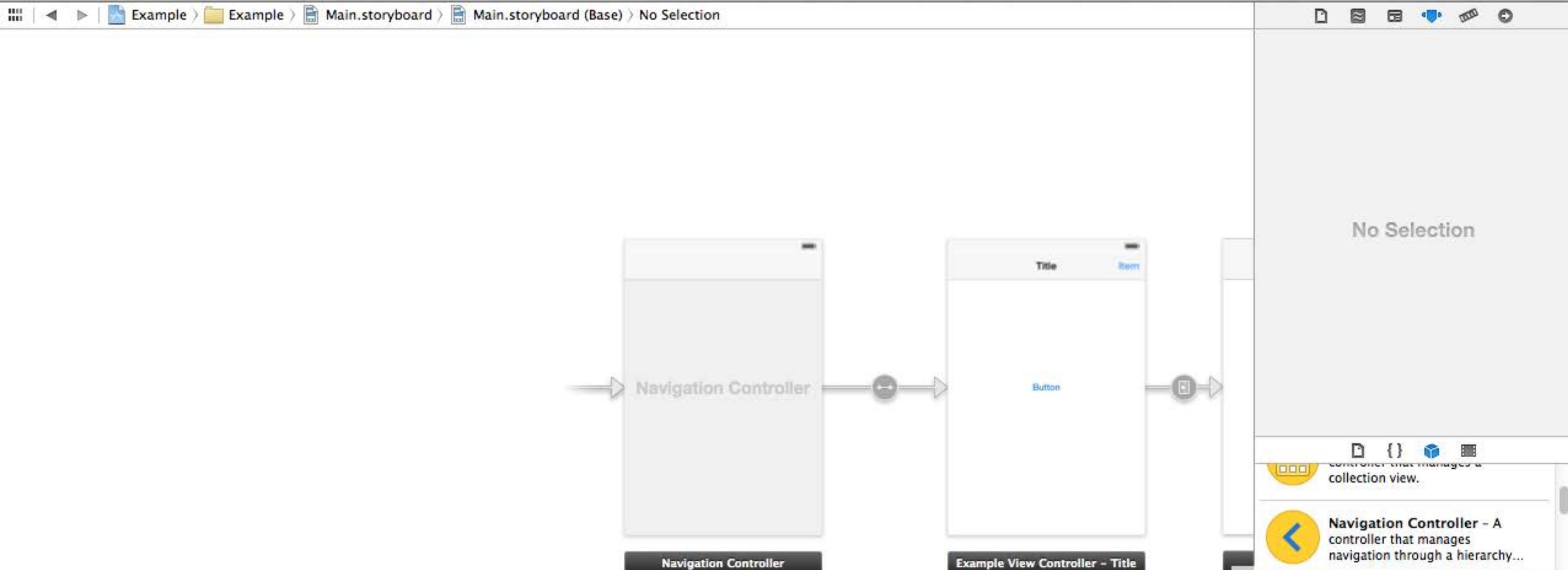
UITabBarController



Tab Bar
Controller



A More button appears.



You create a Tab Bar Controller by dragging it from the object palette.

No Selection

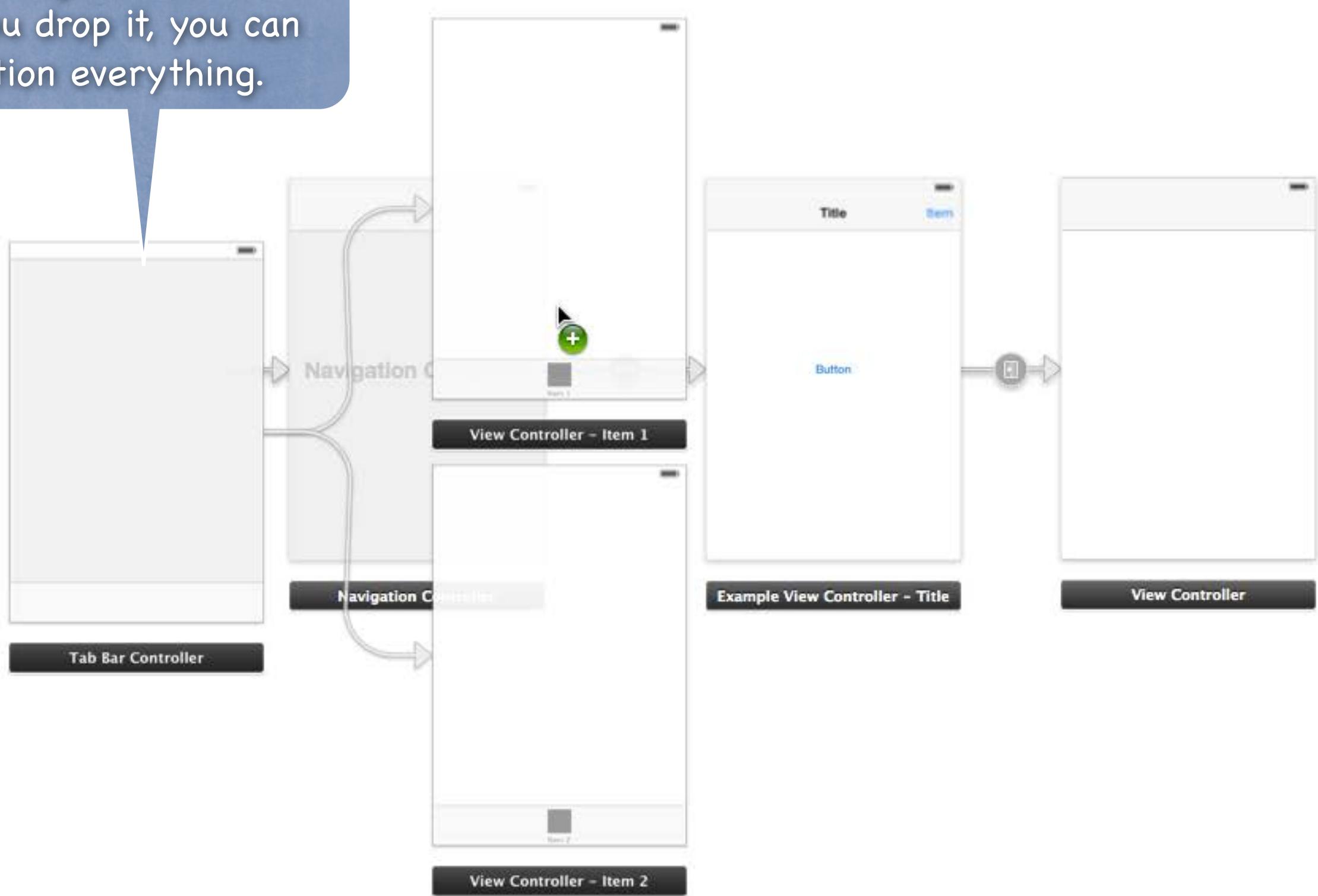
collection view.

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

Tab Bar Controller - A controller that manages a set of view controllers that represent a tab bar.

Page View Controller - Presents a sequence of view controllers as pages.

GLKit View Controller - A controller that manages a GLKit view.



You can drag it anywhere.
After you drop it, you can
reposition everything.

No Selection

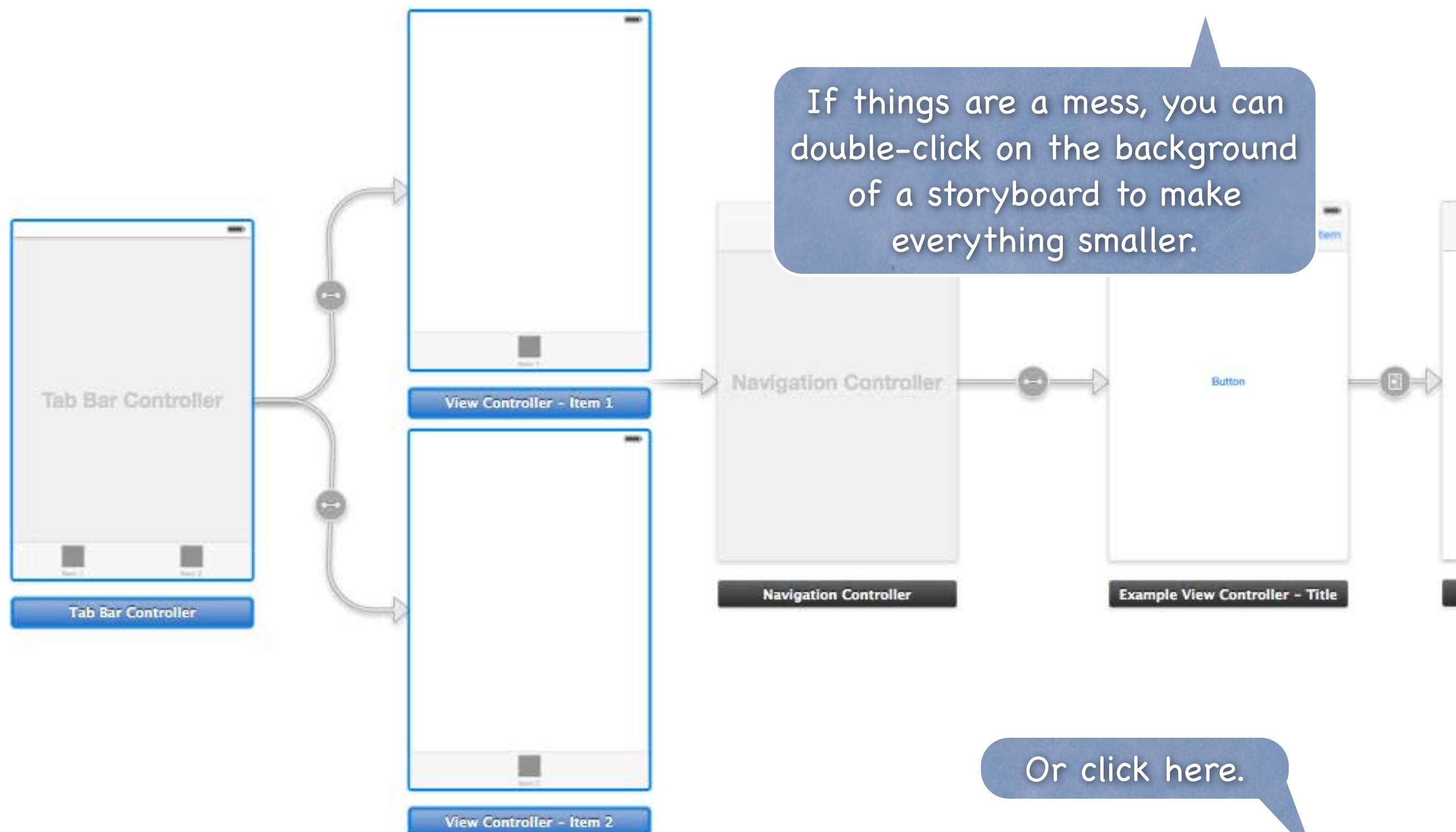
collection view.

 **Navigation Controller** – A controller that manages navigation through a hierarchy...

Tab Bar Controller - A controller that manages a set of view controllers that represent...

 **Page View Controller** -
Presents a sequence of view
controllers as pages.

GLKit View Controller - A controller that manages a GLKit view.

**Simulated Metrics**

Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Multiple Values

View Controller

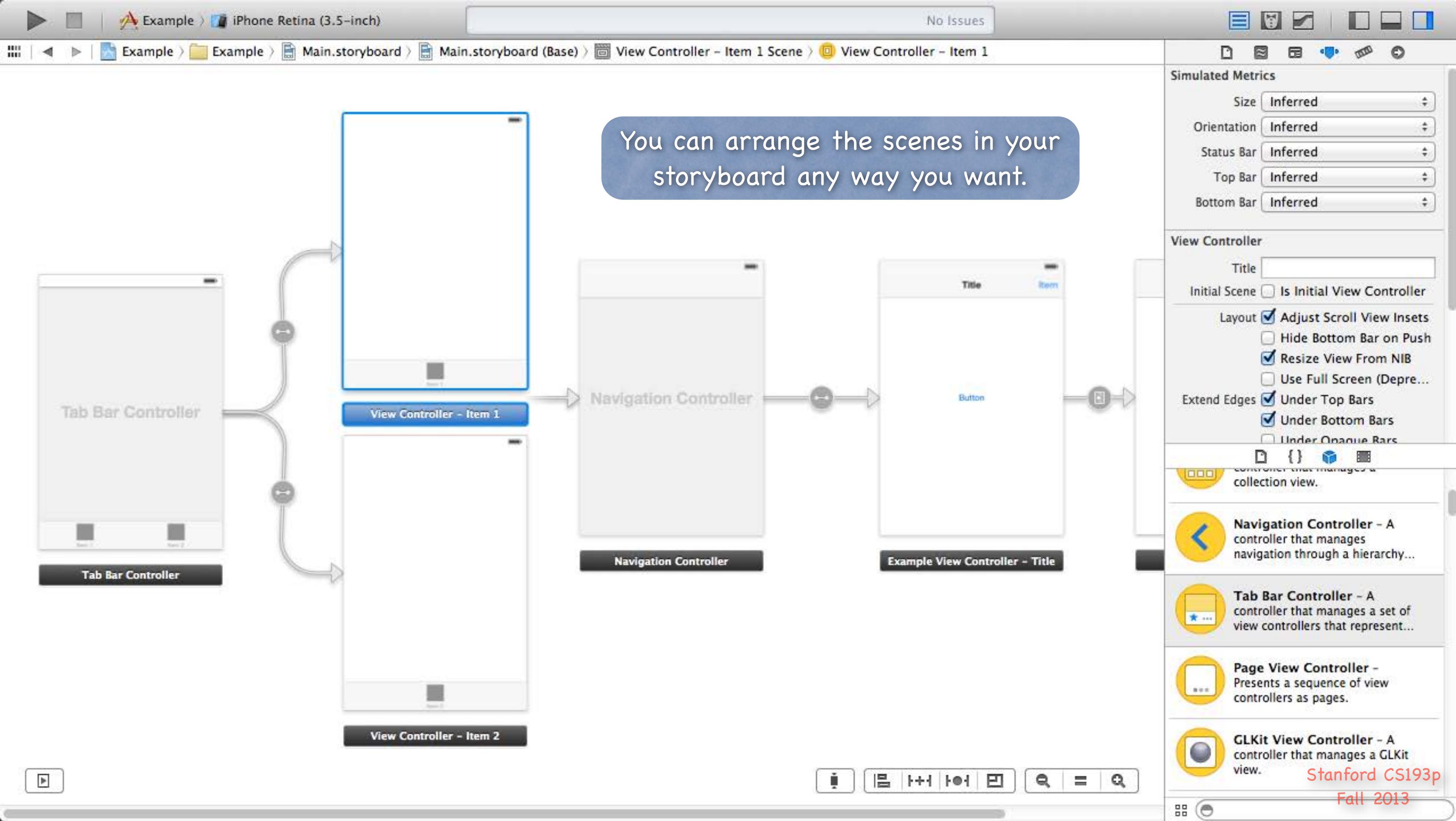
Title	<input type="text"/>
Initial Scene	<input type="checkbox"/> Is Initial View Controller
Layout	<input checked="" type="checkbox"/> Adjust Scroll View Insets <input type="checkbox"/> Hide Bottom Bar on Push <input checked="" type="checkbox"/> Resize View From NIB <input type="checkbox"/> Use Full Screen (Depre...)
Extend Edges	<input checked="" type="checkbox"/> Under Top Bars <input checked="" type="checkbox"/> Under Bottom Bars <input type="checkbox"/> Under Opaque Bars
	collection view.

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

Tab Bar Controller - A controller that manages a set of view controllers that represent...
--

Page View Controller - Presents a sequence of view controllers as pages.
--

GLKit View Controller - A controller that manages a GLKit view.



You can arrange the scenes in your storyboard any way you want.

Simulated Metrics

Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Inferred

View Controller

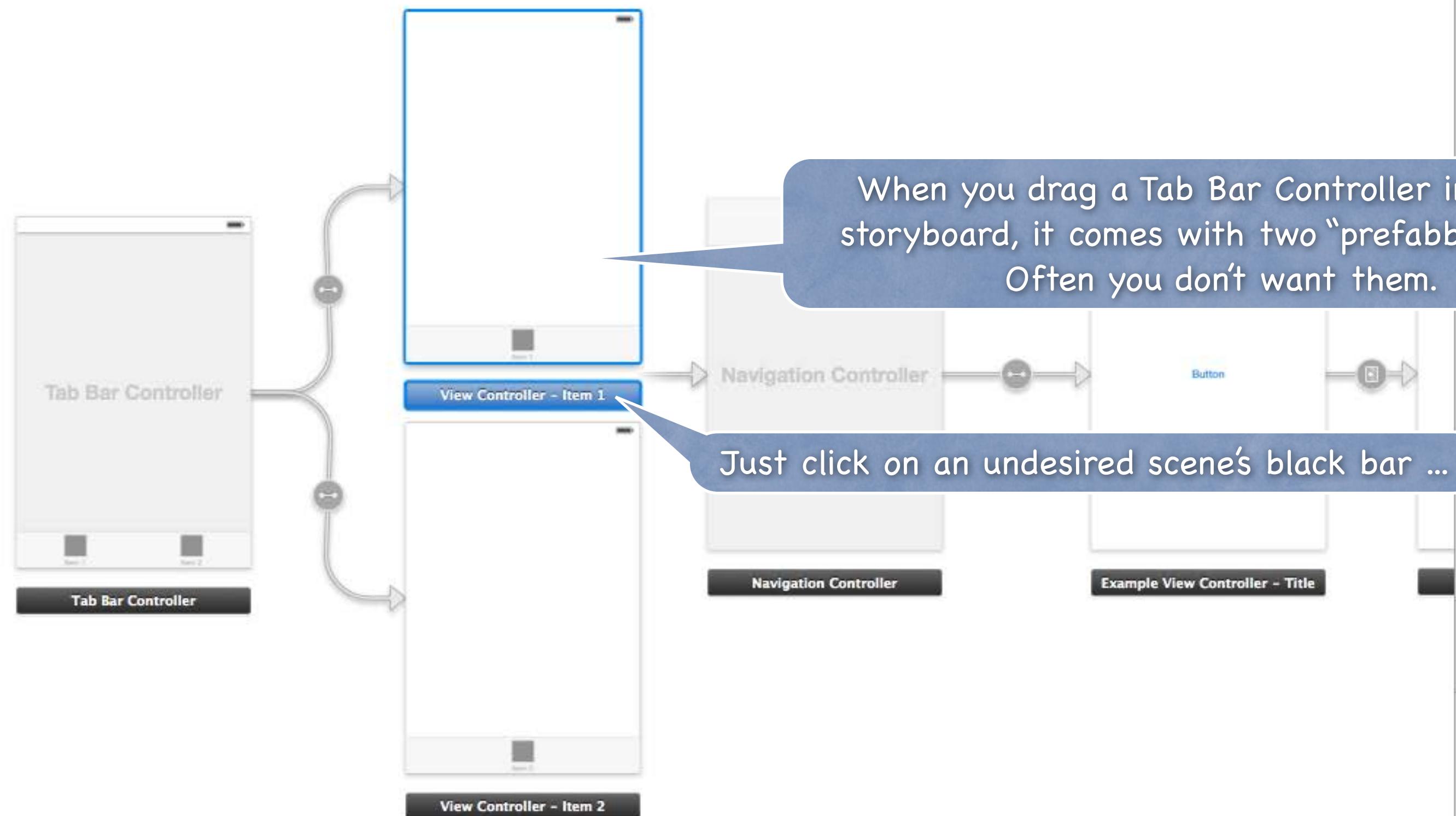
Title	<input type="text"/>
Initial Scene	<input type="checkbox"/> Is Initial View Controller
Layout	<input checked="" type="checkbox"/> Adjust Scroll View Insets <input type="checkbox"/> Hide Bottom Bar on Push <input checked="" type="checkbox"/> Resize View From NIB <input type="checkbox"/> Use Full Screen (Depre...)
Extend Edges	<input checked="" type="checkbox"/> Under Top Bars <input checked="" type="checkbox"/> Under Bottom Bars <input type="checkbox"/> Under Opaque Bars
	   

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

Tab Bar Controller - A controller that manages a set of view controllers that represent

 **Page View Controller** -
Presents a sequence of view
controllers as pages.

GLKit View Controller - A controller that manages a GLKit view.



When you drag a Tab Bar Controller into your storyboard, it comes with two “prefabbed” tabs. Often you don’t want them.

Just click on an undesired scene's black bar ...

Simulated Metrics

Size	Inferred	▲
Orientation	Inferred	▲
Status Bar	Inferred	▲
Top Bar	Inferred	▲
Bottom Bar	Inferred	▲

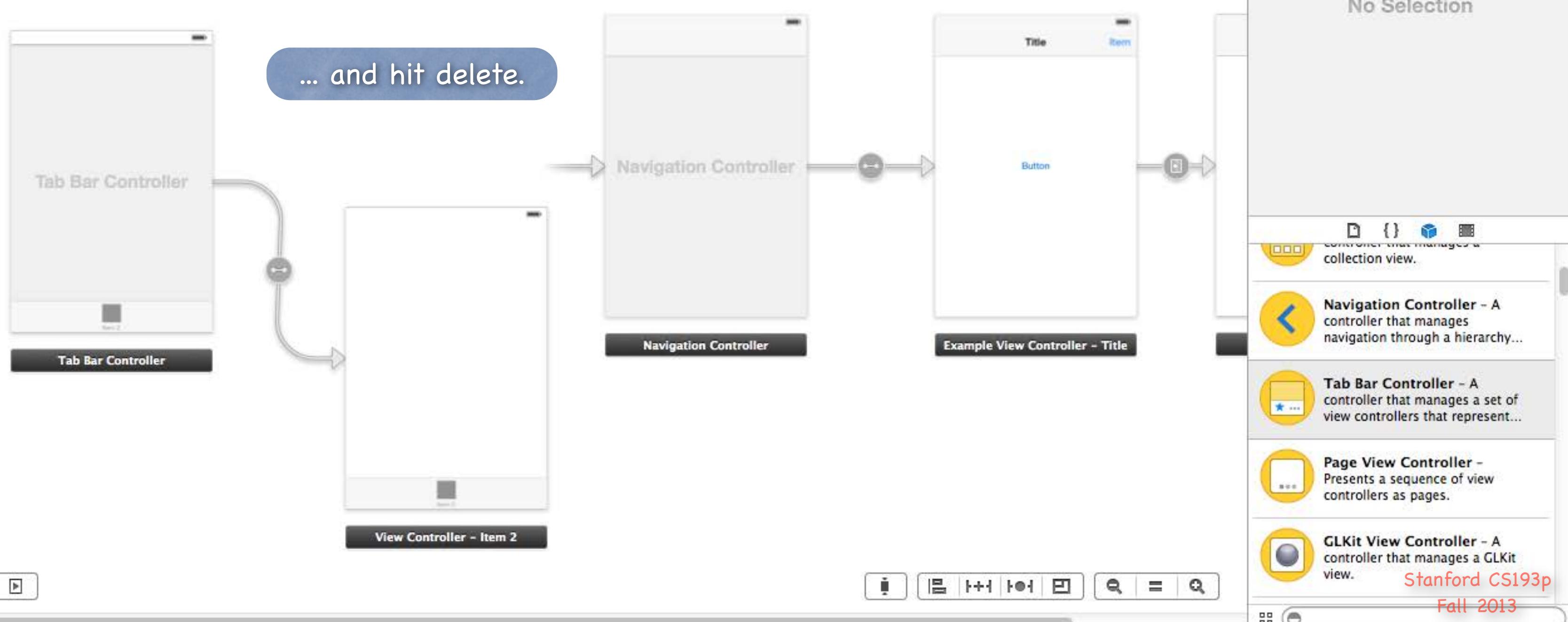
- Resize View From NIB
- Use Full Screen (Depre...
Extend Edges Under Top Bars
 Under Bottom Bars
 Under Opaque Bars

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

 **Tab Bar Controller - A**
controller that manages a set of
view controllers that represent...

 **Page View Controller** –
Presents a sequence of view
controllers as pages.

 **GLKit View Controller** - A controller that manages a GLKit view.



... and hit delete.

No Selection

Tab Bar Controller

Tab Bar Controller

Navigation Controller

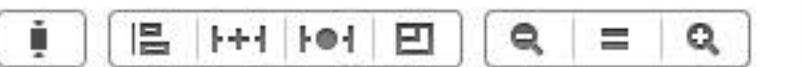
Navigation Controller

Example View Controller - Title

Button

1000

View Controller - Item 2



GLKit View Controller - A
controller that manages a GLKit
view.

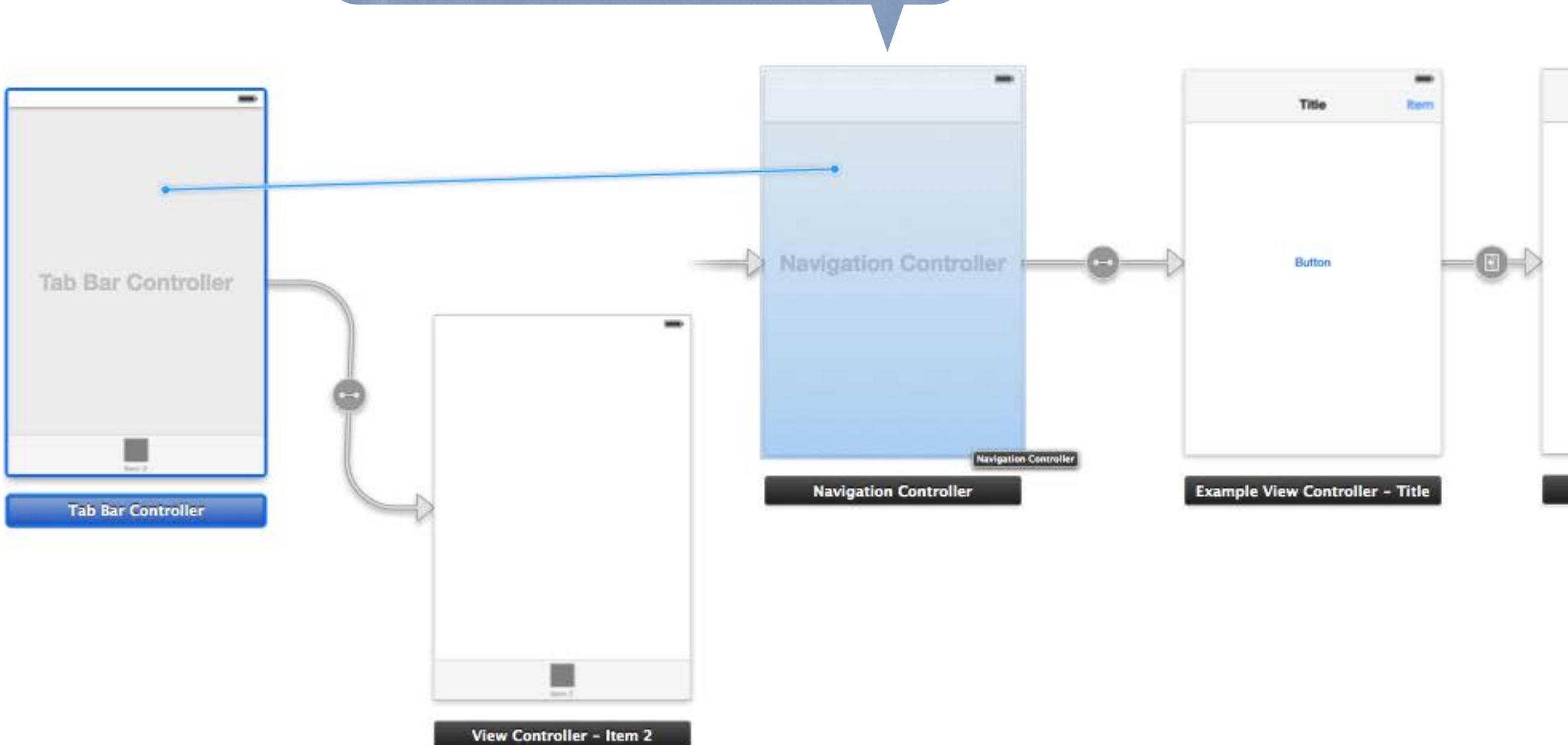
Stanford CS193p

Fall 2013



In the same way as a UINavigationController, a UITabBarController is itself the Controller of an MVC.

It's View consists of other MVCs.

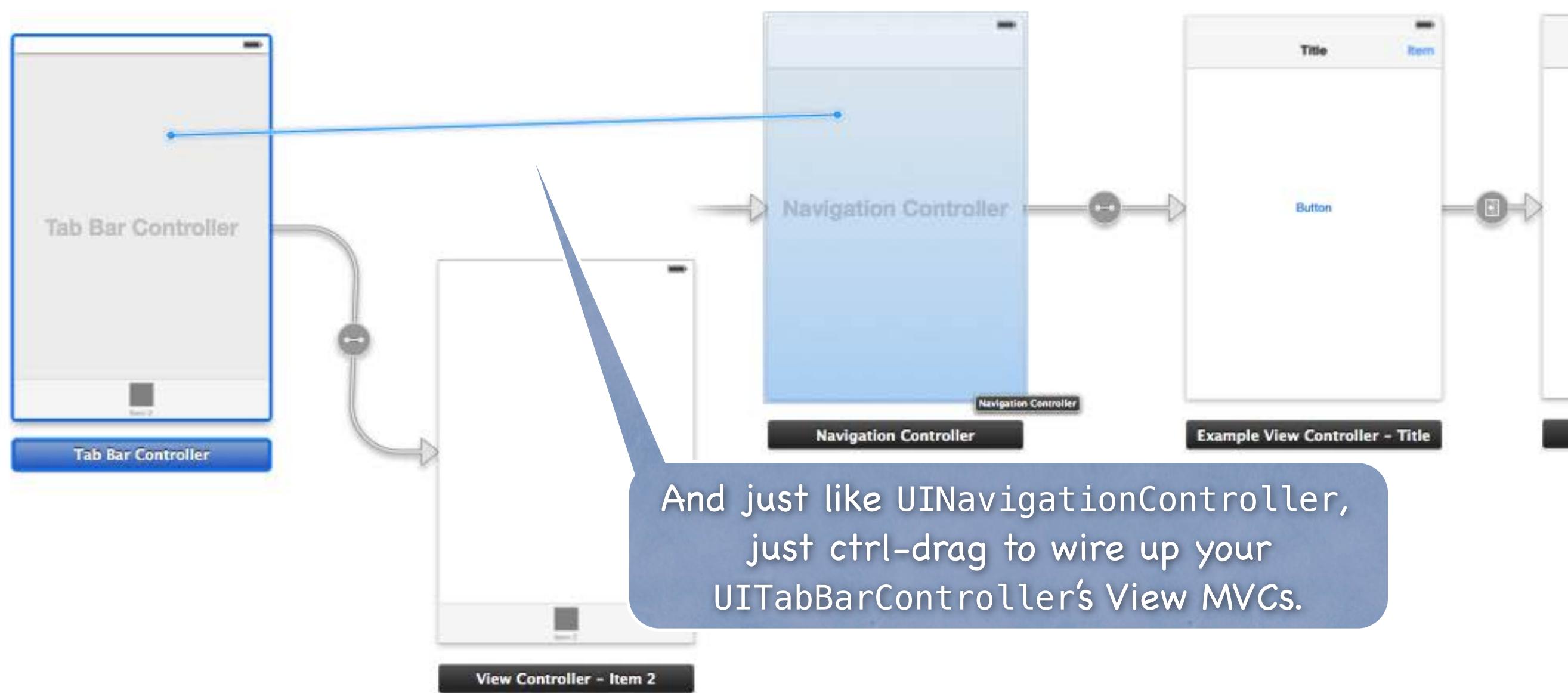


Simulated Metrics

Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Translucent Tab Bar

View Controller

- Title
- Initial Scene Is Initial View Controller
- Layout Adjust Scroll View Insets
 Hide Bottom Bar on Push
 Resize View From NIB
 Use Full Screen (Depre...)
- Extend Edges Under Top Bars
 Under Bottom Bars
 Under Opaque Bars
- collection view.
- Navigation Controller - A controller that manages navigation through a hierarchy...
- Tab Bar Controller - A controller that manages a set of view controllers that represent...
- Page View Controller - Presents a sequence of view controllers as pages.
- GLKit View Controller - A controller that manages a GLKit view.



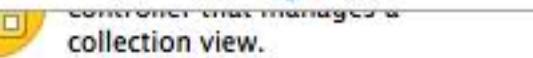
And just like UINavigationController,
just ctrl-drag to wire up your
UITabBarController's View MVCs.

ulated Metrics

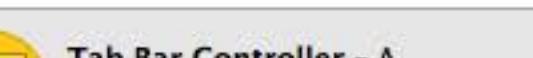
Size	Inferred	▲
Orientation	Inferred	▲
Status Bar	Inferred	▲
Top Bar	Inferred	▲
Bottom Bar	Translucent Tab Bar	▲

New Controller

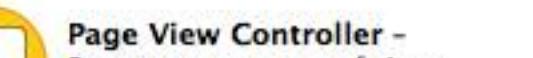
Title	<input type="text"/>
Initial Scene	<input checked="" type="checkbox"/> Is Initial View Controller
Layout	<input checked="" type="checkbox"/> Adjust Scroll View Insets <input type="checkbox"/> Hide Bottom Bar on Push <input checked="" type="checkbox"/> Resize View From NIB <input type="checkbox"/> Use Full Screen (Depre...)
and Edges	<input checked="" type="checkbox"/> Under Top Bars <input checked="" type="checkbox"/> Under Bottom Bars <input type="checkbox"/> Under Opaque Bars



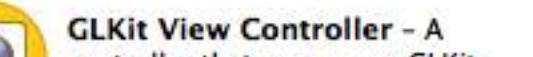
Navigation Controller - A controller that manages navigation through a hierarchy...



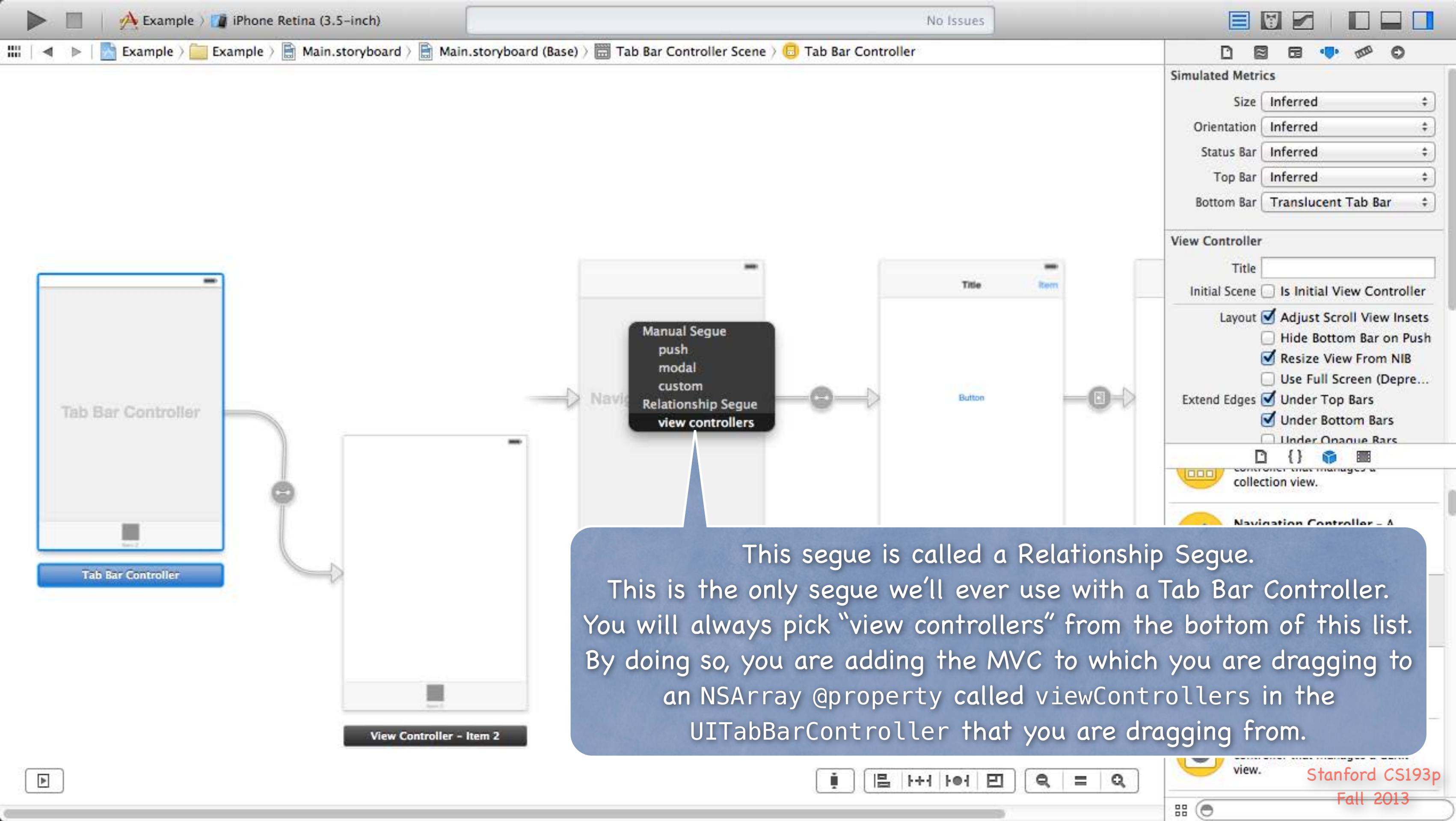
Tab Bar Controller - A
Controller that manages a set of
view controllers that represent...



Page View Controller -
represents a sequence of view
controllers as pages.



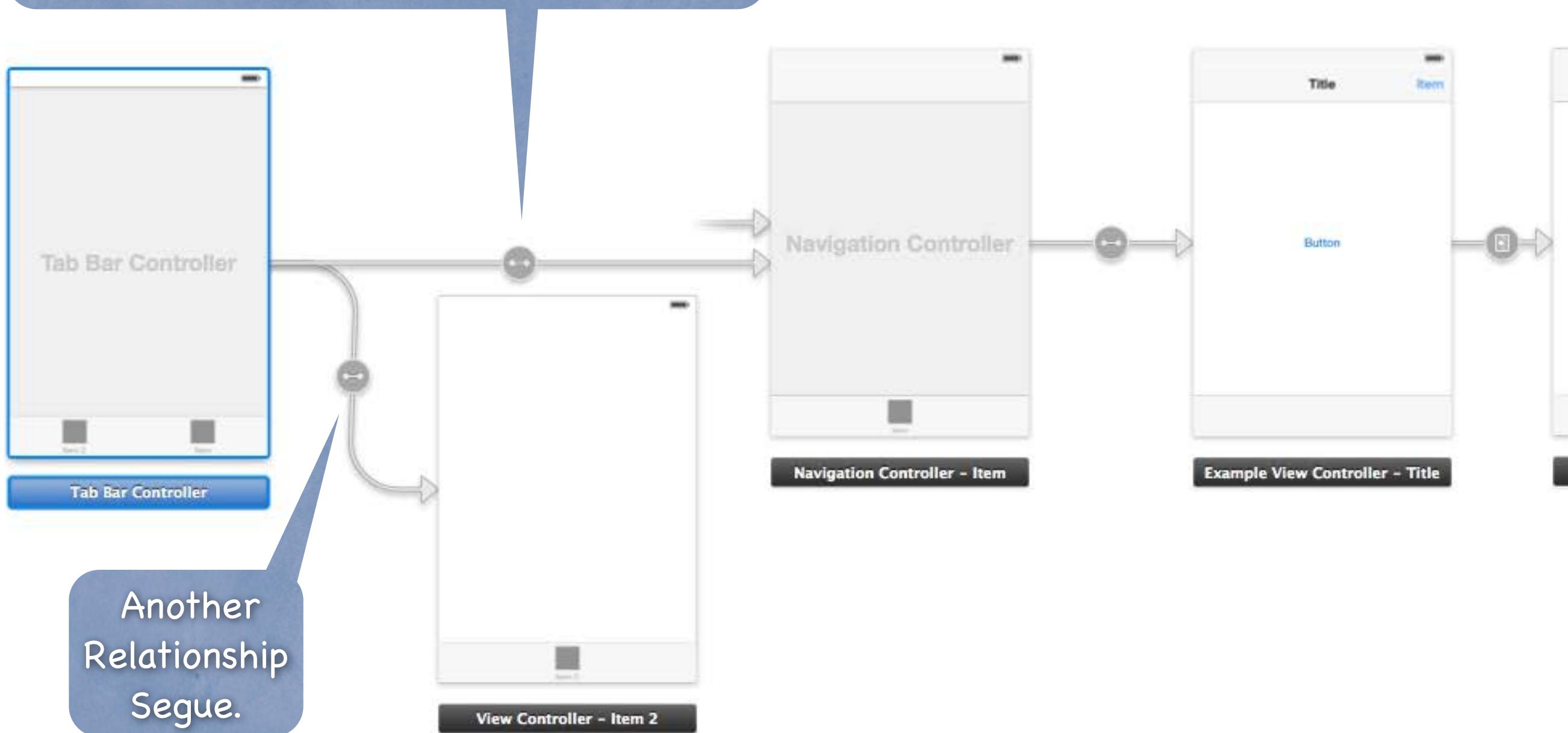
GLKit View Controller - A controller that manages a GLKit view.



This segue is called a Relationship Segue. This is the only segue we'll ever use with a Tab Bar Controller. You will always pick "view controllers" from the bottom of this list. By doing so, you are adding the MVC to which you are dragging to an NSArray @property called viewControllers in the UITabBarController that you are dragging from.



Here is the Relationship Segue.
You don't need to set an identifier on it.



Another
Relationship
Segue.

Simulated Metrics

Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Translucent Tab Bar

View Controller

- Title
- Initial Scene Is Initial View Controller
- Layout Adjust Scroll View Insets
 Hide Bottom Bar on Push
 Resize View From NIB
 Use Full Screen (Depre...)
- Extend Edges Under Top Bars
 Under Bottom Bars
 Under Opaque Bars
- collection view.
- Navigation Controller - A controller that manages navigation through a hierarchy...
- Tab Bar Controller - A controller that manages a set of view controllers that represent...
- Page View Controller - Presents a sequence of view controllers as pages.
- GLKit View Controller - A controller that manages a GLKit view.

Example > iPhone Retina (3.5-inch) No Issues

Example > Example > Main.storyboard > Main.storyboard (Base) > Tab Bar Controller Scene > Tab Bar Controller

Simulated Metrics

- Size Inferred
- Orientation Inferred
- Status Bar Inferred
- Top Bar Inferred

Note that room has been made at the bottom of each scene for the tab bar. This might cover up some of your UI and require some repositioning.

The screenshot shows the Xcode Storyboard Editor. A Tab Bar Controller is on the left, connected to two View Controllers via Navigation Controllers. The top navigation controller is labeled "Navigation Controller - Item" and the bottom one is "View Controller - Item 2". Each navigation controller is connected to an "Example View Controller - Title" view controller. A blue callout bubble points to the bottom of the storyboard area, stating: "Note that room has been made at the bottom of each scene for the tab bar. This might cover up some of your UI and require some repositioning." On the right side, there is a sidebar with various controller icons and their descriptions:

- Extend Edges Under Top Bars Under Bottom Bars Under Opaque Bars
- Navigation Controller - A controller that manages navigation through a hierarchy...
- Tab Bar Controller - A controller that manages a set of view controllers that represent...
- Page View Controller - Presents a sequence of view controllers as pages.
- GLKit View Controller - A controller that manages a GLKit view.

Stanford CS193p Fall 2013

Example > iPhone Retina (3.5-inch) No Issues

Example > Example > Main.storyboard > Main.storyboard (Base) > Tab Bar Controller Scene > Tab Bar Controller

Simulated Metrics

- Size Inferred
- Orientation Inferred
- Status Bar Inferred
- Top Bar Inferred
- Bottom Bar Translucent Tab Bar

View Controller

Title

Initial Scene Is Initial View Controller

Layout Adjust Scroll View Insets

Hide Bottom Bar on Push

Resize View From NIB

Use Full Screen (Depre...)

Extend Edges Under Top Bars

Under Bottom Bars

Under Opaque Bars

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

Tab Bar Controller - A controller that manages a set of controllers that represent...

Controller - sequence of view is pages.

View Controller - A flat manager a GLKit

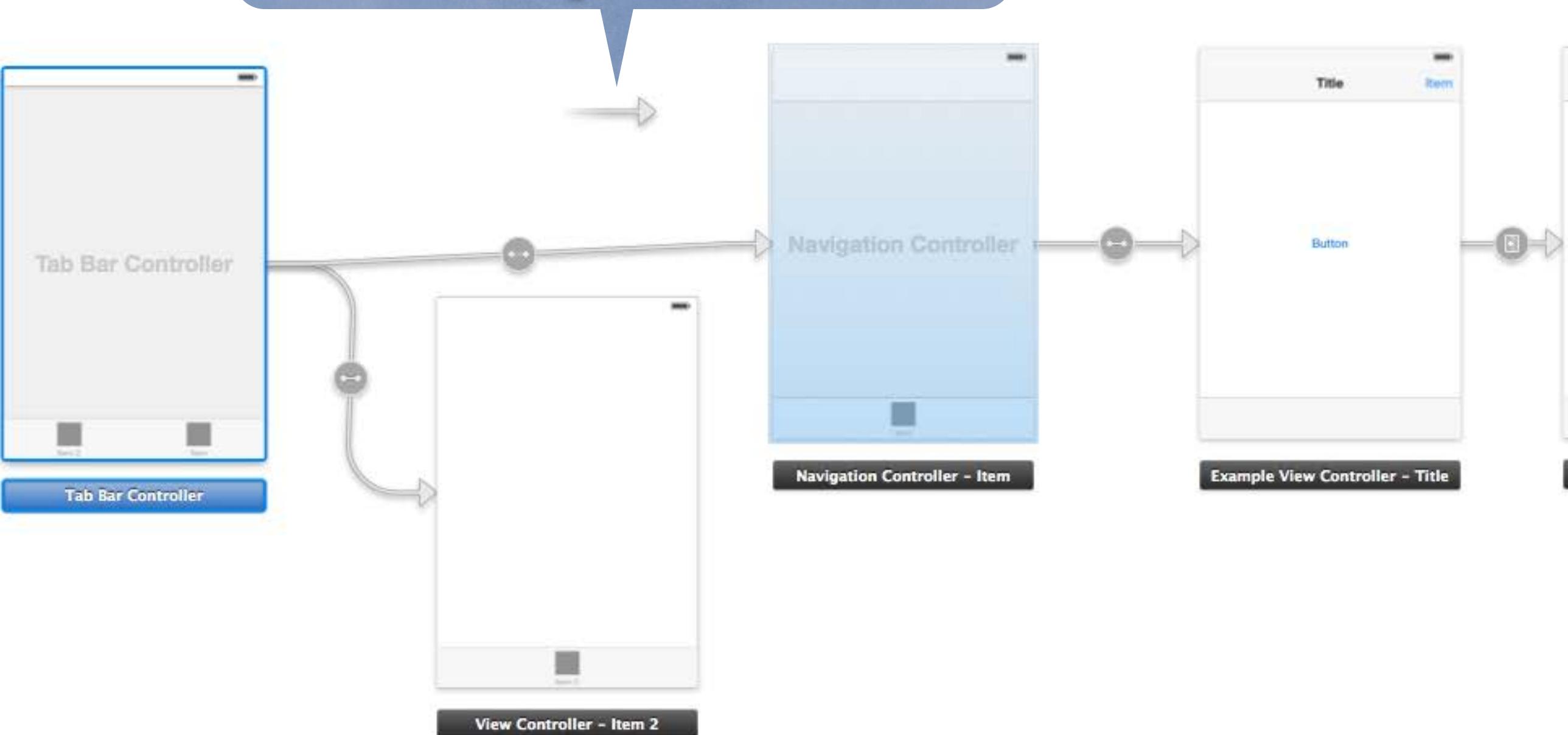
Stanford CS193p Fall 2013

```
graph LR; TabBar[Tab Bar Controller] --> NavCtrl[Navigation Controller]; NavCtrl --> ViewController[Example View Controller - Title];
```

Here we have
UINavigationController INSIDE a
UITabBarController.
Perfectly legal (the opposite is not).



The MVC at launch is still set to the
UINavigationController.
It needs to be the UITabBarController.
Just drag this arrow ...

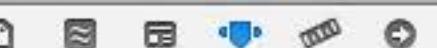


Simulated Metrics

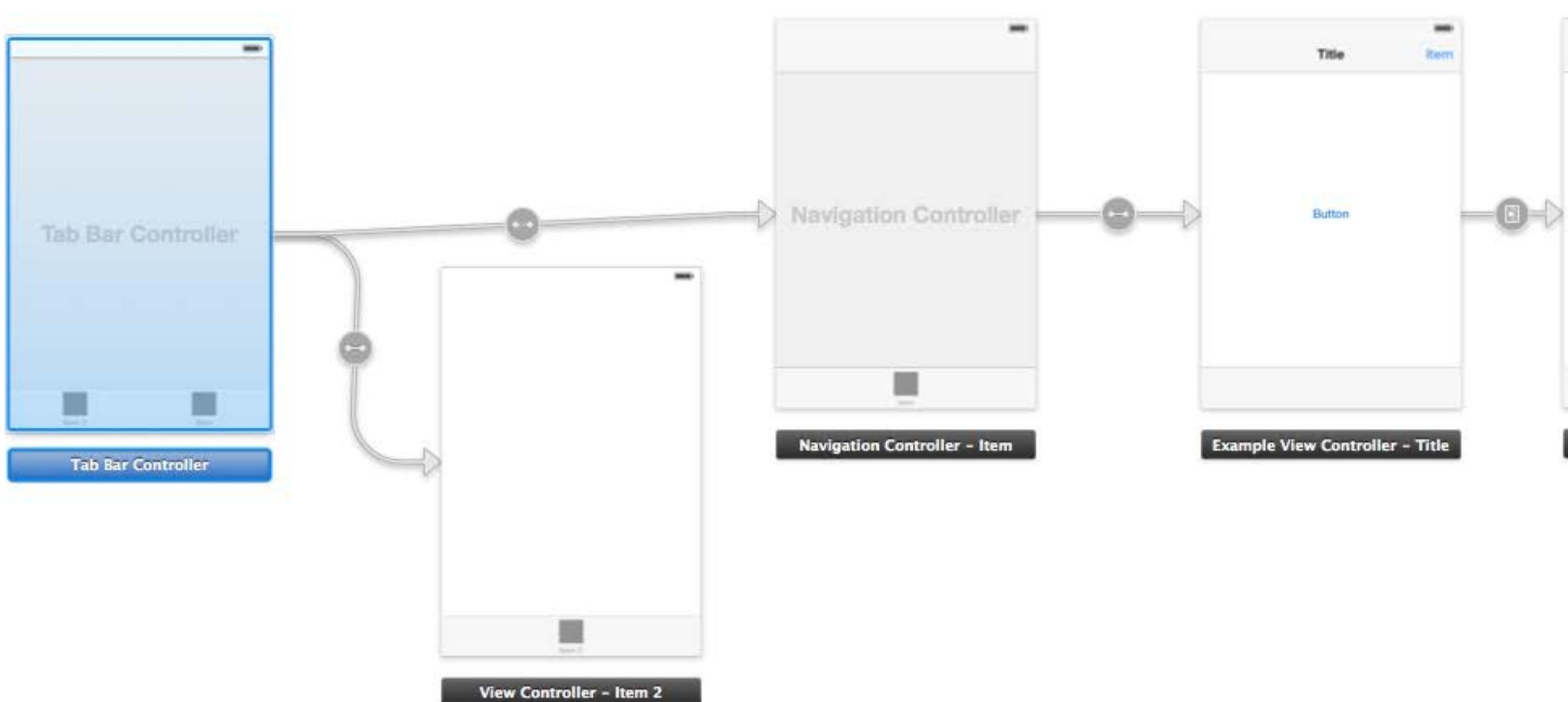
Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Translucent Tab Bar

View Controller

- Title
- Initial Scene Is Initial View Controller
- Layout Adjust Scroll View Insets
 Hide Bottom Bar on Push
 Resize View From NIB
 Use Full Screen (Depre...)
- Extend Edges Under Top Bars
 Under Bottom Bars
 Under Opaque Bars
- collection view.
- Navigation Controller - A controller that manages navigation through a hierarchy...
- Tab Bar Controller - A controller that manages a set of view controllers that represent...
- Page View Controller - Presents a sequence of view controllers as pages.
- GLKit View Controller - A controller that manages a GLKit view.



... over near the
UITabBarController MVC ...



Simulated Metrics

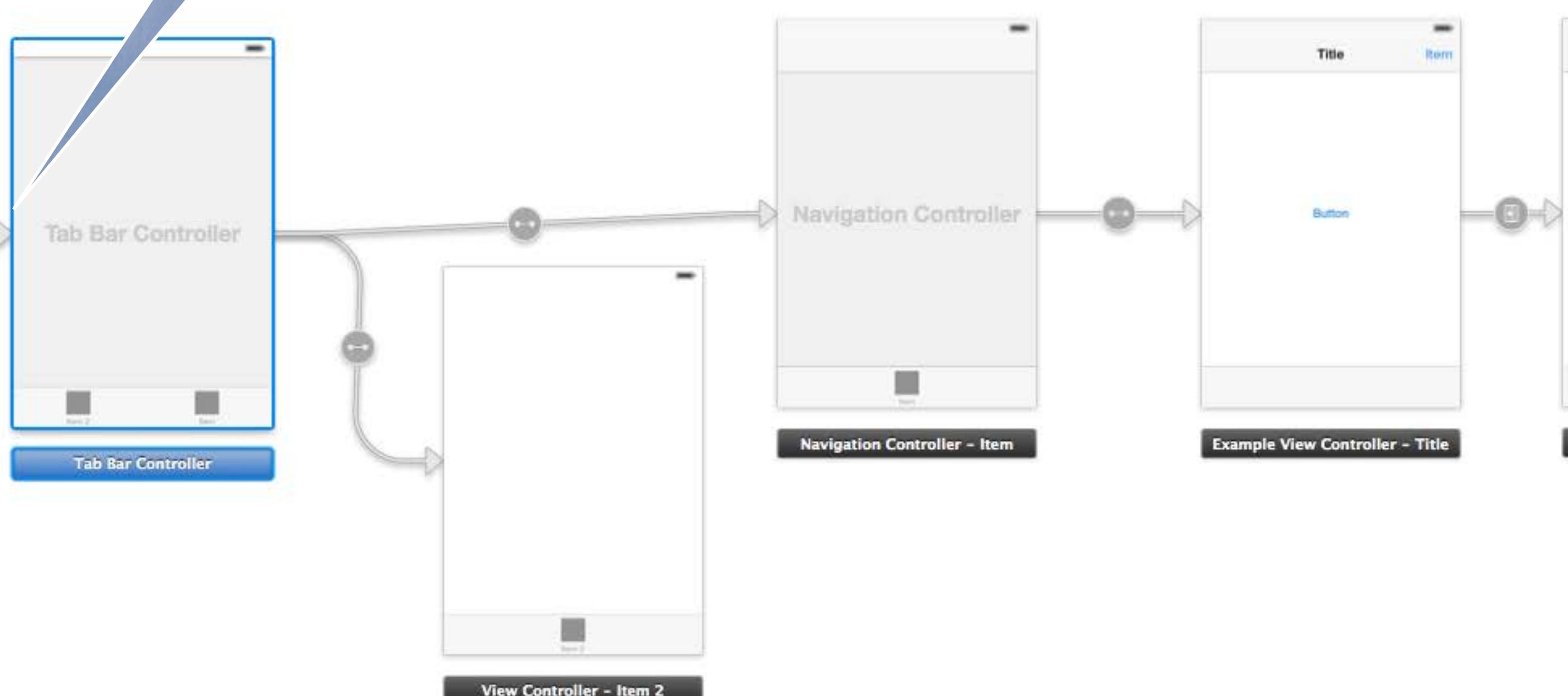
Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Translucent Tab Bar

View Controller

- Title
- Initial Scene Is Initial View Controller
- Layout Adjust Scroll View Insets
 Hide Bottom Bar on Push
 Resize View From NIB
 Use Full Screen (Depre...)
- Extend Edges Under Top Bars
 Under Bottom Bars
 Under Opaque Bars
- collection view.
- Navigation Controller - A controller that manages navigation through a hierarchy...
- Tab Bar Controller - A controller that manages a set of view controllers that represent...
- Page View Controller - Presents a sequence of view controllers as pages.
- GLKit View Controller - A controller that manages a GLKit view.



... and drop it
(it will snap onto the
UITabBarController).



Simulated Metrics

Size	Inferred
Orientation	Inferred
Status Bar	Inferred
Top Bar	Inferred
Bottom Bar	Translucent Tab Bar

View Controller

- Title
- Initial Scene Is Initial View Controller
- Layout Adjust Scroll View Insets
 Hide Bottom Bar on Push
 Resize View From NIB
 Use Full Screen (Depre...)
- Extend Edges Under Top Bars
 Under Bottom Bars
 Under Opaque Bars
- collection view.
- Navigation Controller - A controller that manages navigation through a hierarchy...
- Tab Bar Controller - A controller that manages a set of view controllers that represent...
- Page View Controller - Presents a sequence of view controllers as pages.
- GLKit View Controller - A controller that manages a GLKit view.

Example > iPhone Retina (3.5-inch)

No Issues

Navigation Controller Scene > Navigation Controller - Item > Tab Bar Item - Item

Navigation Controller

Item

View Controller - Item 2

The name of each tab can be edited directly in Xcode.

Navigation Controller - Item

Tab Bar Item

- Badge
- Identifier: Custom
- Title: Default Position

Bar Item

- Title: Item
- Image
- Tag: 0
- Enabled

View Controller - Item 2

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

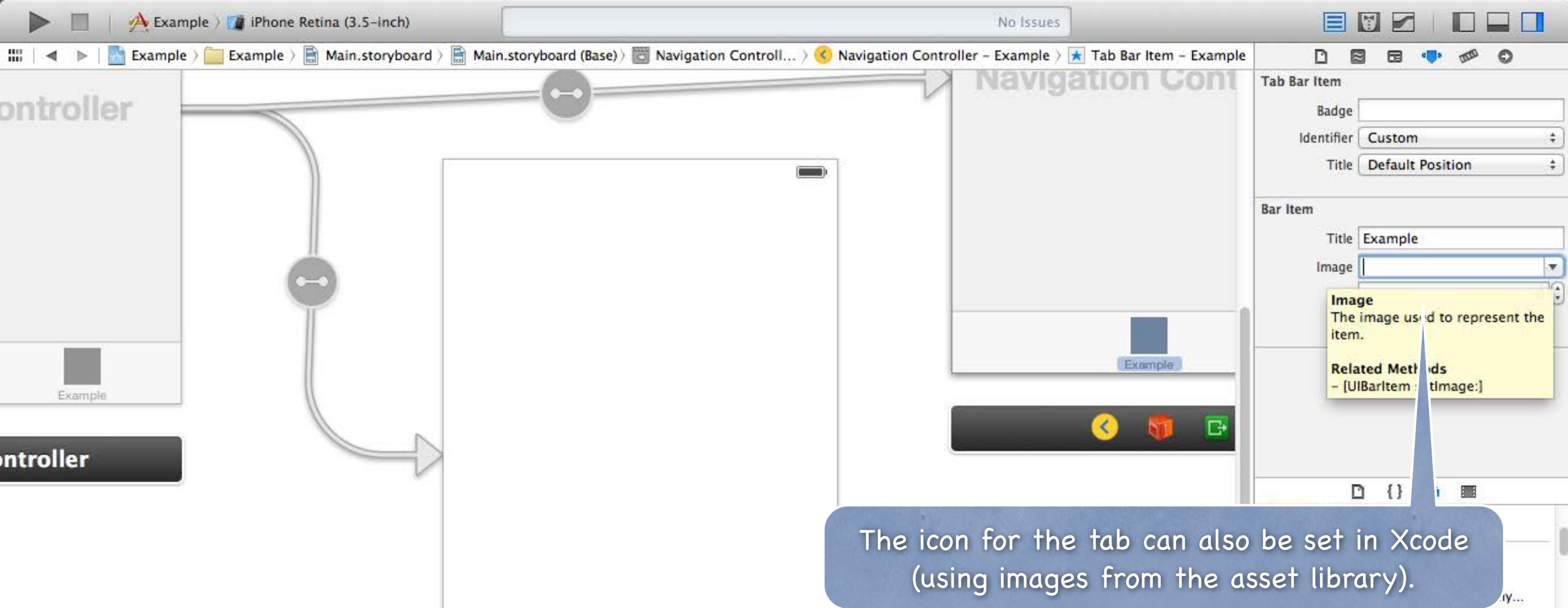
Tab Bar Controller - A controller that manages a set of view controllers that represent...

Page View Controller - Presents a sequence of view controllers as pages.

GLKit View Controller - A controller that manages a GLKit view.

Stanford CS193p
Fall 2013

This screenshot shows the Xcode interface with a storyboard open. The storyboard contains two tabs: 'Item' and 'Item 2'. A callout bubble with the text 'The name of each tab can be edited directly in Xcode.' points to the 'Example' tab bar item. The right side of the screen displays the Utilities panel, specifically the Attributes Inspector, which shows settings for the selected tab bar item, including badge, identifier, and title options. Below the storyboard, a navigation bar is visible with icons for back, forward, and search.



Tab Bar icons are 30x30, alpha channel only.

 Page View Controller -
Presents a sequence of view
controllers as pages.

 GLKit View Controller - A
controller that manages a GLKit
view.