# Introduction to Segues

Segues allow us to have multiple scenes (pages/screens) in one application. In this workbook we will look at linking scenes together and passing data between scenes.

It includes explanations, code descriptions and code examples, which you should read. It also includes activities, for example to create or modify some code. The activities reinforce what you have read, but also give you vital coding practice. The workbook is design to be read in sequence, if you skip parts it may make it harder to understand later parts. It also assumes you have completed the previous workbooks.

Whilst the workbook includes everything you need to cover, you may find it useful to refer to other sources of information. There is a lot of information and documentation available at <https://swift.org/documentation/> including a guide <https://docs.swift.org/swift-book/LanguageGuide/TheBasics.html>

The codeacademy website has a course on Swift, but it is new and doesn’t contain many lessons at the moment. Our library also has books on Swift.

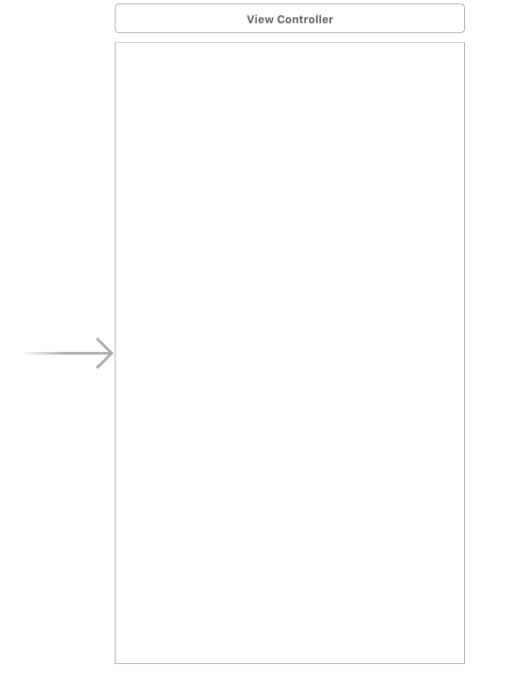
There are different versions of Xcode and Swift, so watch out for incompatibilities!

## Segues

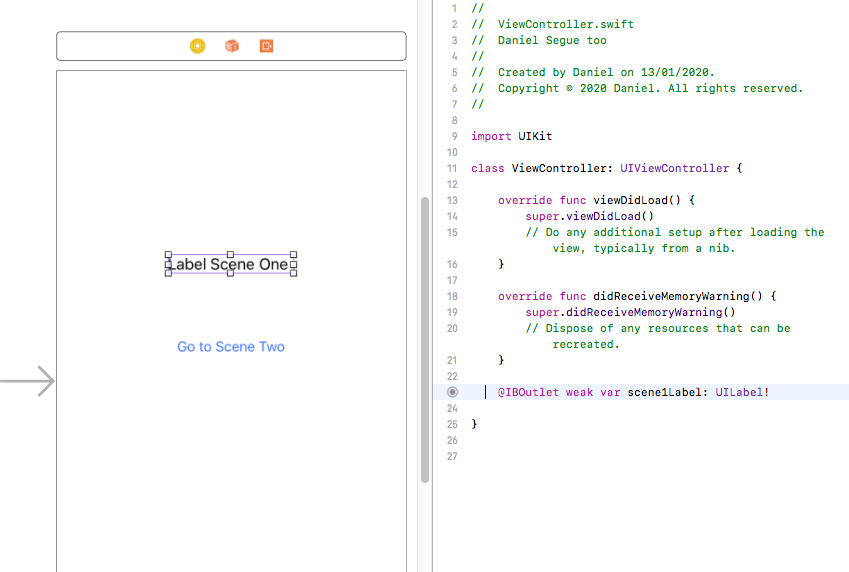
Segues allow us to have multiple scenes (pages/screens) in one application. In this workbook we will look at linking scenes together and passing data between scenes.

1. Create a new Single View Application

If we look at the Storyboard, we will see an arrow pointing to the screen. This indicates the first scene that will be shown when the app runs. When we have multiple scenes we can move this arrow to point to any of them.

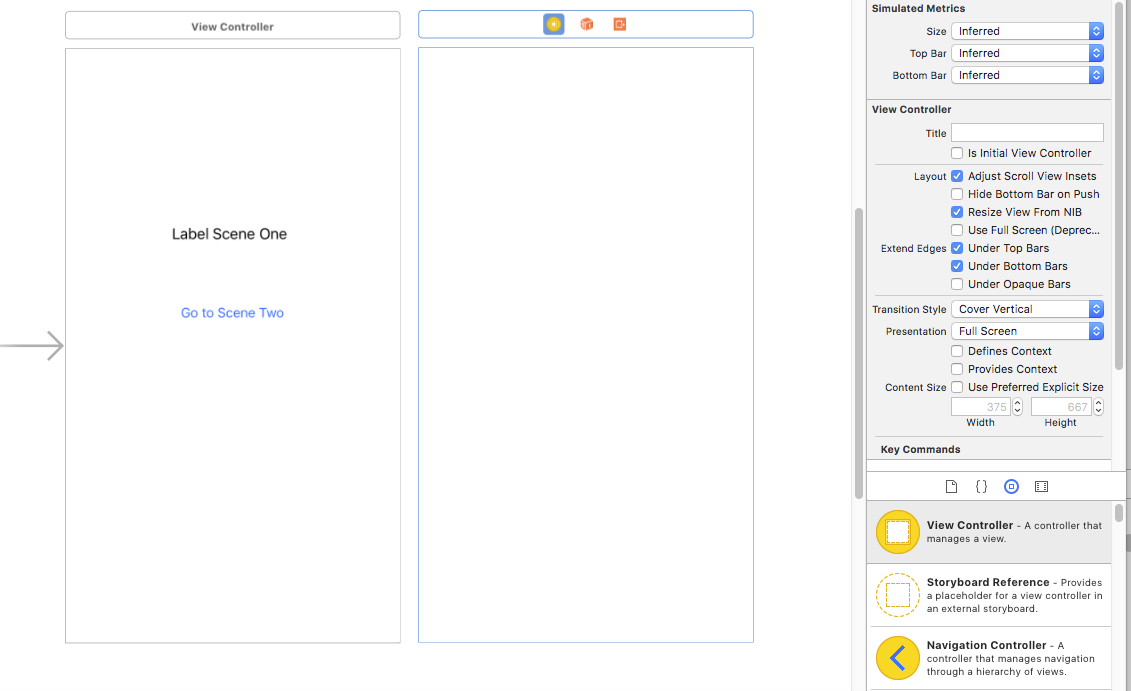


1. Add a label
2. Change the text of the label to "Label Scene One"
3. Add a new Referencing Outlet for the label named "scene1Label"
4. Add a button
5. Change the text of the button to "Go to Scene Two"



We will now add a second scene

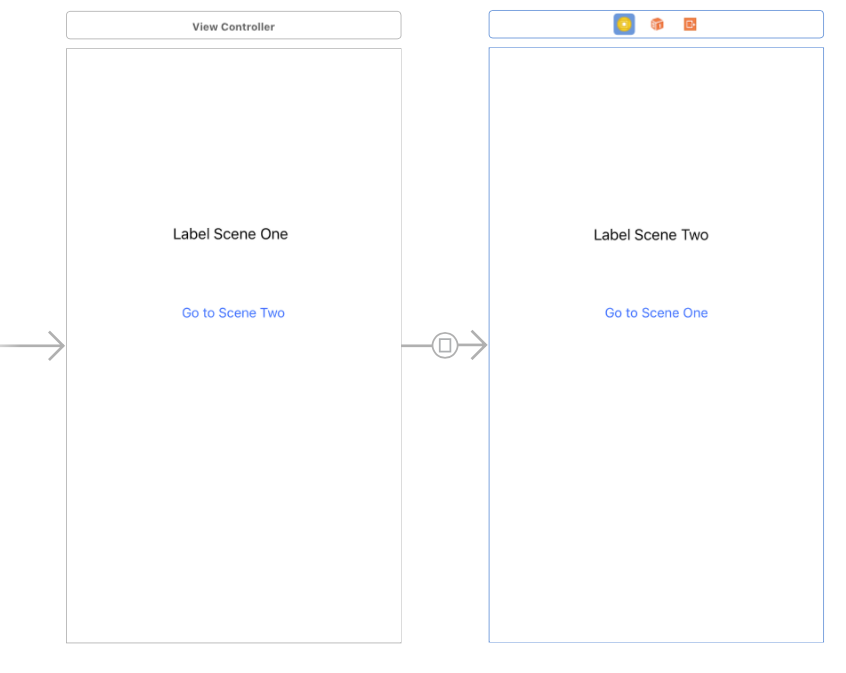
1. Add a View Controller – this will add a new screen to our Storyboard.



1. Add a label to the new screen
2. Change the text of the label to "Label Scene Two"
3. Add a button
4. Change the text of the button to "Go to Scene One"

We now need to add a segue between the two scenes

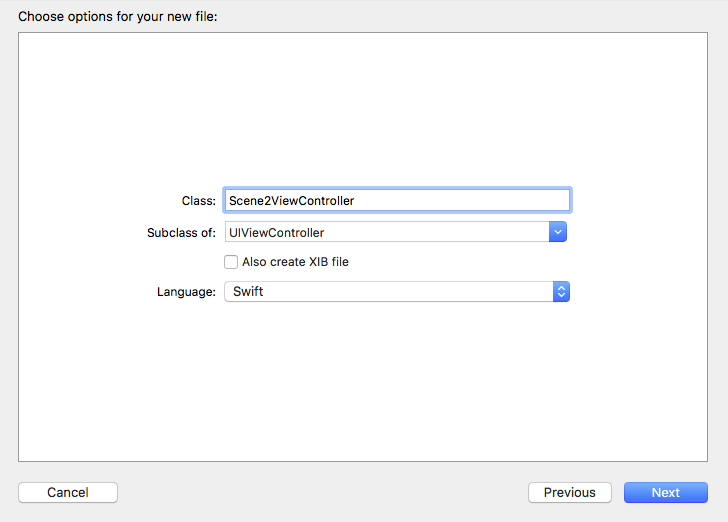
1. Hold down the ctrl key and click and hold the mouse over the scene one button
2. Drag over to the second screen and release
3. Select the "Present Modally" option



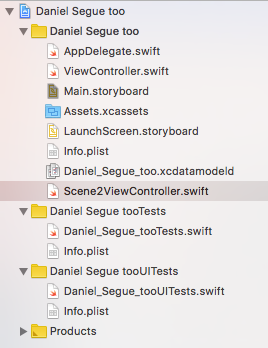
1. We can now see a segue link between our two scenes. Click on the Segue Link
2. In the Attributes panel, select the Cross Dissolve transition.
3. Run the simulator – click the scene one button to go to scene two.

At the moment we have no no code associated with it. Next we will associate the second scene with a new View Controller Swift file

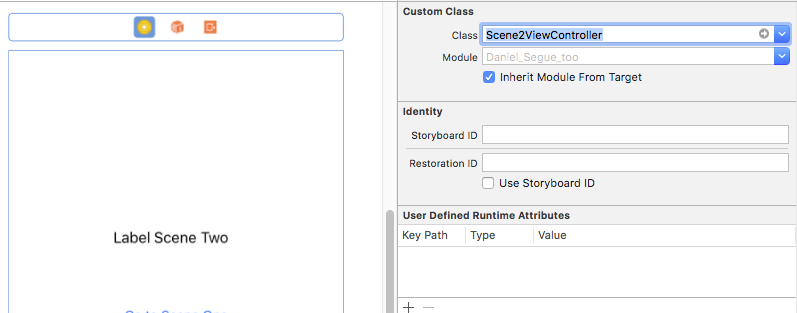
1. Ctrl click on the top folder in the Navigation pane
2. Select "New File…"
3. Select "Cocoa Touch Class"
4. Name the new class "Scene2ViewController"



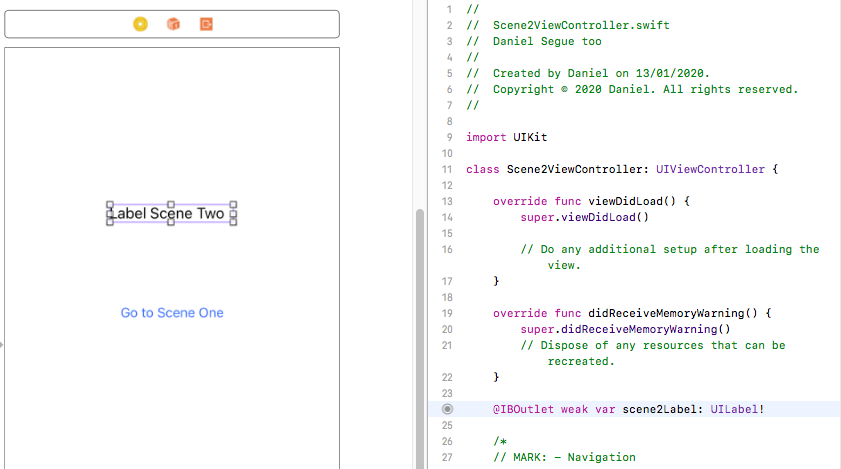
You should see the new View Controller file listed in the Navigation pane



1. On the Storyboard, select the View Controller icon on the second scene
2. Then on the Identity Inspector pane, select the class "Scene2ViewController"

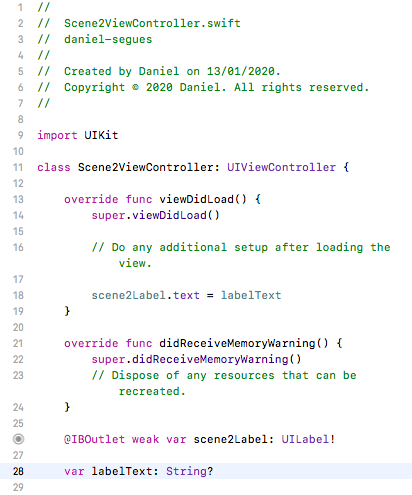


1. In your new Scene2ViewController Swift file (check the comments at the top to make sure you have the right ViewController file) create a new Referencing Outlet for the label named "scene2Label"

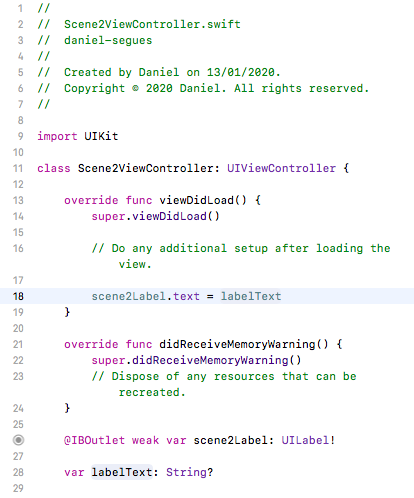


We are now going to pass a variable string from scene one to scene two when we move from scene one to scene 2. We will write the value of this variable to the label on scene two.

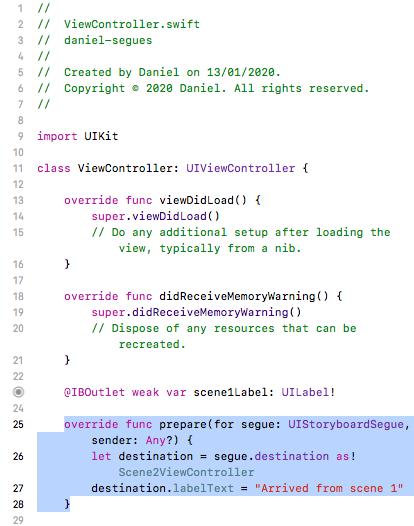
1. In scene two, add a variable called labelText with type optional String, as shown below.



1. Within the ViewDidLoad function set the label text for scene two equal to the value of this variable. This will run when the scene is loaded.

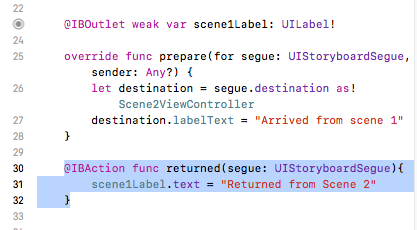


1. In scene one, add the code shown below. This code will change the variable in scene two when the segue is executed (i.e. when we click on the button to go to scene two)



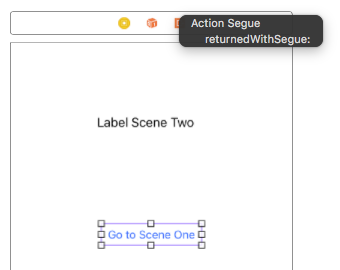
We will return from scene two to scene one using an unwind segue. When scene two exits (we will set this up in a moment) we will execute a function.

1. In scene one, add the code shown below. This code will be executed when we return from scene two.



We will now set the exit point for scene two.

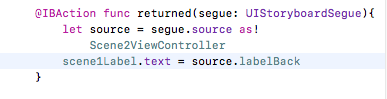
1. On the scene two storyboard, ctrl-click the button and drag to the Exit icon at the top of the storyboard (the right-hand icon) and release. Select the "returnedWithSegue" action.



1. Run the simulator. You can now step from scene one to scene two by clicking the button, and then unwind back to scene one by clicking the return button.

We will now pass back a variable from scene two to scene one.

1. In scene two, add a new constant called labelBack with the value "Hello from Scene Two"
2. In scene one, in the returned function change the assignment of "scene1Label.text" to be "source.labelBack" instead of "Returned from Scene Two"



1. Run the simulator

### Exercises:

**Add text entry to scene two**

Include the ability to type in text on Scene Two and return that text back to Scene One

Hint: Insert a Text box like we did last week

Hint: Remember, this text box needs to have an action and an outlet associated with it

Hint: Use labelBack to return the text

**Add a timer/counter (like last week) and insert it on scene one only**

See what happens as you switch between scenes