

Lecture 8

Gestures

Gestures

- People interact with iOS applications through gestures
 - Up to now, we have seen taps on buttons - there were many ways to press a button and we saw that there was an extensive list of IBActions to support these
- Gestures are great from a usability perspective. We can allow users to interact with content in an intuitive way (e.g. pinch to resize a photo)

Gestures

- iOS offers a range of gestures:
 - tap - often used for selection (including button press) (d)
 - pinch - used, for example, to resize an image (d)
 - rotate - used, for example, to rotate an image (c)
 - swipe - used to dismiss, or segue (d)
 - pan - essentially a drag gesture (c)
 - screen edge - used to reveal a menu (d)
 - long press - used to detect a press of some duration (d)
 - custom - we can define our own gestures
- All of these can be customised to create very specific actions
- Some of these are discrete (generate one call to our action)
- Others are continuous (generate many calls in rapid succession to our action)



Tap Gesture Recognizer - Recognizes tap gestures, including double-tap or multiple-touch.



Pinch Gesture Recognizer - Recognizes pinch gestures.



Rotation Gesture Recognizer - Recognizes rotation gestures.



Swipe Gesture Recognizer - Recognizes swipe gestures.



Pan Gesture Recognizer - Recognizes pan (dragging) gestures.



Screen Edge Pan Gesture Recognizer - Recognizes pan (dragging) gestures that start near a...



Long Press Gesture Recognizer - Recognizes long press gestures, based on the number and duration of...



Custom Gesture Recognizer - Recognizes custom gestures. Set a custom subclass in the Identity inspe...

Gestures

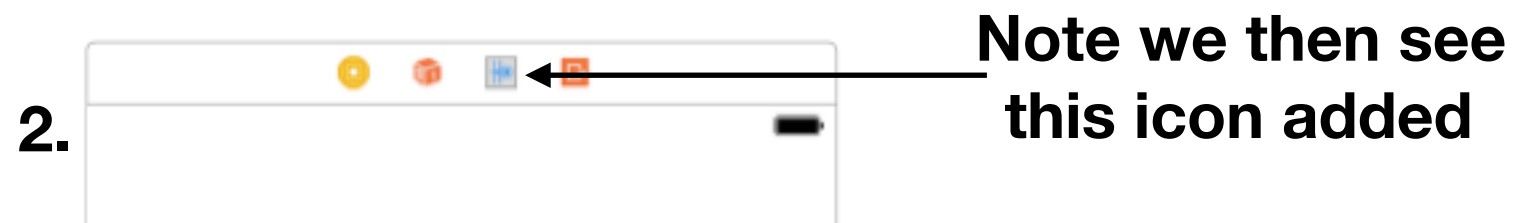
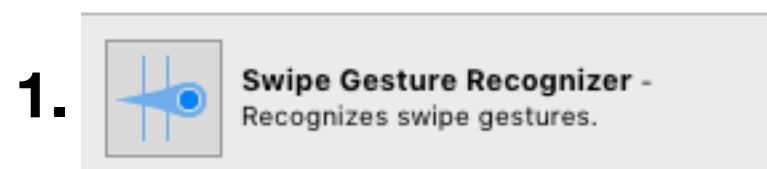
- We should observe Apple's suggested **usability standards** for gesture use
 - Use Standard Gestures - don't force the user to learn something new
 - Don't use Standard Gestures if the interaction is non-standard (e.g. pinch action to swipe content out of the view)
 - Don't block system gestures (top and bottom of screen in particular)
 - Don't rely on gestures only for app navigation - always provide button based alternatives
 - Consider the place of multi-finger gestures

Implementing Gestures

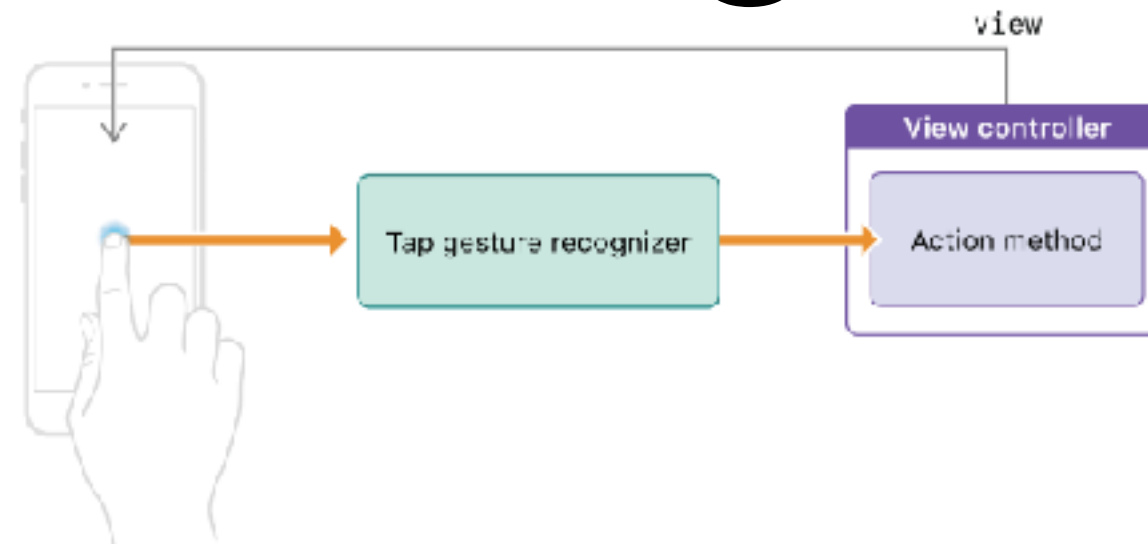
- Example 1 - using swipes to invoke segues
 - There are two approaches
 - Recognise the gesture and invoke a gesture handler to perform the segue
 - Relate the segue to the gesture in the storyboard
- To use the first approach, we must:
 - Add a gesture recogniser
 - Add an action handler to capture and handle the recogniser's events
 - Add a connection between these

Implementing Gestures

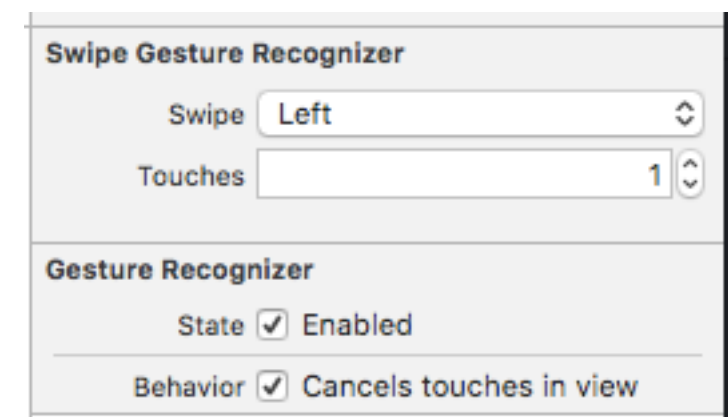
- For the first case, we initialise our application by:
 - adding two view controllers to the storyboard (and adding labels so that we can distinguish between them)
 - adding a manual segue between the view controllers on the storyboard and we give these segues identifiers
- We can then add a swipe gesture recogniser to the first view controller by dragging an instance from the object library and dropping it on the view controller



Implementing Gestures



- We can configure the gesture in the ‘Attribute Inspector’
 - In the case of a swipe gesture, we can add a swipe direction, and a number of touches required (number of fingers making the swipe)
- Each gesture has its own attributes



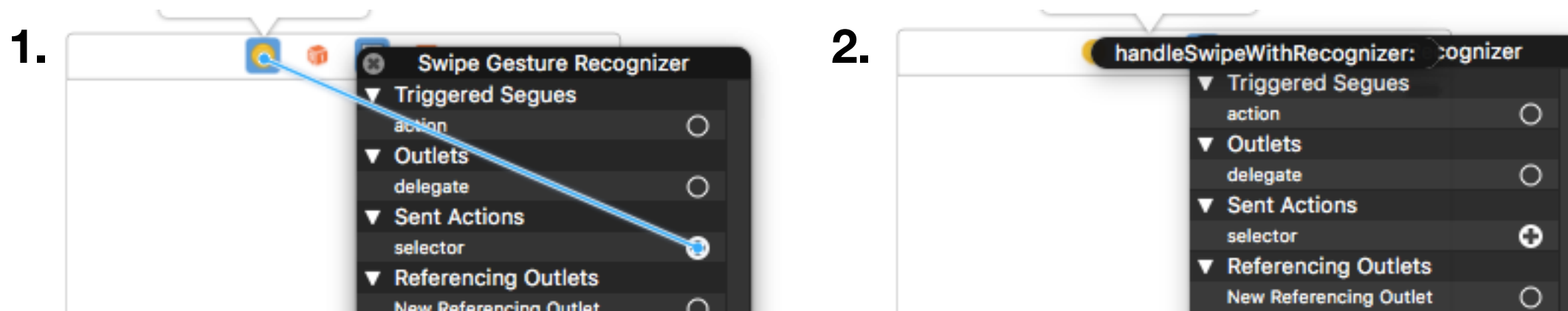
- We can then add an action method to handle the gesture recogniser's events

Implementing Gestures

- In the ViewController's swift file, we add a method to handle the event generated by the handler:

```
@IBAction func handleSwipe(recognizer:UISwipeGestureRecognizer) {  
    self.performSegue(withIdentifier:"SwipeToSecond", sender: self)  
}
```

- This invokes a segue programatically for us
- We associate this IBAction with our gesture recogniser by right clicking on the storyboard's gesture and dragging to the view controller. We will be given an option to click on the IBAction to create a connection



Implementing Gestures

- We could also have taken a second approach in the particular case of a segue.
- The action list available for a gesture includes adding a segue
- Add a second swipe gesture, but this time to the second view controller in the storyboard

1.



2.

