

Gesturizer

Lesson 1



Description

Add a text label and a Tap Gesture Recognizer to the interface, and update the label text when the screen is tapped.

Learning Outcomes

- Practice adding objects to the user interface, modifying interface object attributes, and establishing controller outlet and action connections.
- Discover how iOS encapsulates gesture recognition in `UIGestureRecognizer` classes.
- Apply a Tap Gesture Recognizer to update the interface when the screen is tapped.
- Practice using the connections inspector to observe bindings between objects.



Vocabulary

gesture	<code>UIGestureRecognizer</code>	Attributes Inspector
outlet	property	Tap Gesture Recognizer
Document Outline	Connections Inspector	controller action

Materials

- **Gesturizer Lesson 1** Xcode project

Opening

How is interacting with an iOS touch screen different than using a conventional computer interface?

Agenda

- Discuss the kinds of touch-based interaction students have used in the past.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIGestureRecognizer` class reference, drawing attention to the concrete subclass names.
- Discuss how iOS provides abstractions for common gestures, such as taps, making it easy to detect the gestures in an app.
- Using Interface Builder and the Object Library (⇧⌘L), add a Text Label near the middle of the view.
- Change the label text to **Gesturizer**, and, using the Attributes Inspector (⇧⌘4), customize the label font, size, alignment and color.
- Add layout constraints to the label for its horizontal and vertical center, and use the menu item *Editor > Resolve Auto Layout Issues > Update Frames* (⇧⌘=) to update its size.
- Using the Attributes Inspector (⇧⌘4), check the text label's *Drawing > Hidden* attribute.
- Open the Assistant Editor (⇧⌘⇥), and Control-drag from the label to the `ViewController` to create an outlet property.

```
@IBOutlet weak var gestureName: UILabel!
```

- Using Interface Builder and the Object Library (⇧⌘L), drag a Tap Gesture Recognizer to the bottom of the Document Outline (⇧⌘0).
- Within the Document Outline (⇧⌘0), click and rename the Tap Gesture Recognizer to **Single Tap**.
- With the Single Tap gesture recognizer still selected, view the Attributes Inspector (⇧⌘4).
- Discuss the attributes of the gesture recognizer, such as the number of taps and touches it is configured to recognize.
- With the Single Tap gesture recognizer still selected, view the Connections Inspector (⇧⌘6).
- Discuss how the gesture recognizer has only one Sent Actions connection called `selector`, representing a method that will be called when the gesture recognizer recognizes a gesture.

- Using the Assistant Editor (⌘⇧↵), Control-drag a connection from the Single Tap gesture recognizer to a new controller method called `singleTap:`.

```
@IBAction func singleTap(sender: UITapGestureRecognizer) {  
}
```

- Implement the `singleTap:` method, such that it changes the visibility of the `gestureName` text label and sets its content to "Tap".

```
@IBAction func singleTap(sender: UITapGestureRecognizer) {  
    gestureName.text = "Tap"  
    gestureName.hidden = false  
}
```

- Run the app (⌘R), tap the screen (single-click) and observe how nothing happens.
- Explain how, although one taps on the view, the view is not bound to the Single Tap gesture recognizer.
- Using Interface Builder and the Document Outline (⇧⌘O), add the Single Tap gesture recognizer to the View's `gestureRecognizers` outlet connection by Control-dragging from the View to the Single Tap gesture recognizer.
- With the View still selected, open the Connections Inspector (⌘⇧6) and observe the Outlet Collections.
- In the Document Outline (⇧⌘O), select the Single Tap recognizer, open the Connections Inspector (⌘⇧6), and observe the Referencing Outlet Collections.
- Run the app (⌘R), tap (click) the view, and observe the **Tap** label appear.

Closing

How might you get the recognizer to only respond to taps on a label or other subview component?

Modifications and Extensions

- Delete the Single Tap gesture recognizer from the Document Outline, create a `UITapGestureRecognizer` within the `ViewController` class definition, bind it to the view, and retain the same functionality.

Resources

Event Handling Guide for iOS <http://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/Introduction/Introduction.html>

UIGestureRecognizer Class Reference https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIGestureRecognizer_Class/index.html

UITapGestureRecognizer Class Reference http://developer.apple.com/library/ios/documentation/uikit/reference/UITapGestureRecognizer_Class/Reference/Reference.html

Creating an Outlet Connection https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingOutlet.html

Adding an Object to Your Interface https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/AddingObject.html

Configuring Object Attributes https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/ObjectAttributes.html

Creating an Action Connection https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingAction.html

Cocoa Core Competencies: Target-Action <http://developer.apple.com/library/ios/documentation/General/Conceptual/Devpedia-CocoaApp/TargetAction.html>