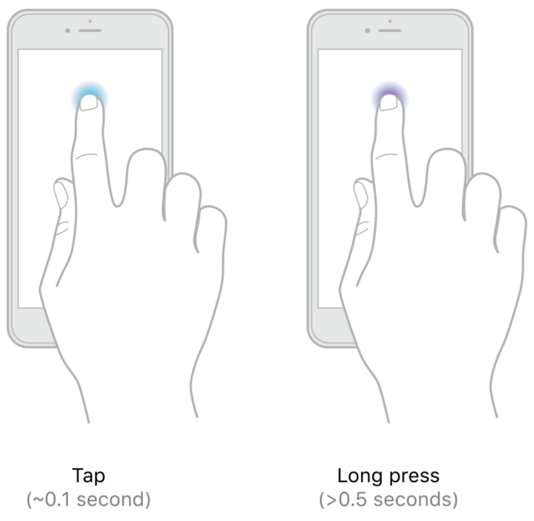
Handling Tap and Long-Press Gestures

On This Page

Tap and long-press (also known as press-and-hold) gestures detect one or more fingers touching the screen. The fingers involved in these gestures must not move significantly from the initial touch points, and you can configure the number of times the fingers must touch the screen. For example, you might configure tap or long-press gesture recognizers to detect single taps, double taps, or triple taps.

- A *tap gesture* occurs when the user's finger touches the screen only briefly. Use the `UITapGestureRecognizer` class to detect tap gestures.
- A *long-press gesture* occurs when the user's finger touches the screen for an extended period of time. You configure the minimum duration of the press gesture. (The gesture recognizer is triggered only by the duration of the touches and not by the force associated with them.) Use the `UILongPressGestureRecognizer` class to detect long-press gestures.

Figure 2-1 Tap and long-press gestures



You can attach a gesture recognizer in one of these ways:

- Programmatically. Call the `addGestureRecognizer:` method of your view.
- In Interface Builder. Drag the appropriate object from the library and drop it onto your view

Handling a Tap Gesture

A `UITapGestureRecognizer` object provides event handling capabilities similar to those of a button—it detects a tap in its view and reports that tap to your action method. Tap gestures are discrete, so your action method is called only when the tap gesture is recognized successfully. You can configure a tap gesture recognizer to require any number of taps—for example, single taps or double taps—before your action method is called.

Listing 2-1 shows an action method that responds to a successful tap in a view by animating that view to a new location. Always check the gesture recognizer's `state` property before taking any actions, even for a discrete gesture recognizer. Gestures can be cancelled for many reasons, so checking the `state` property ensures that your code responds correctly to success or failure.

Listing 2-1 Handling a tap gesture

```
@IBAction func tapPiece(_ gestureRecognizer : UIPanGestureRecognizer) {
    let piece = gestureRecognizer.view
    self.adjustAnchorPoint(gestureRecognizer: gestureRecognizer)
    let center = piece?.center

    if gestureRecognizer.state == .ended {
        UIView.animate(withDuration: 0.2, animations: {
            piece?.center = CGPoint(x: ((center?.x)! + 100), y: ((center?.y)! +
100))
```

```
        })
    }
}
```

If the code for your tap gesture recognizer is not called, check to see if the following conditions are true, and make corrections as needed:

- The `userInteractionEnabled` property of the view is set to `YES`. Image views and labels set this property to `NO` by default.
- The number of taps was equal to the number specified in the `numberOfTapsRequired` property.
- The number of fingers was equal to the number specified in the `numberOfTouchesRequired` property.

Handling a Long-Press Gesture

A `UILongPressGestureRecognizer` object detects a tap where a finger or stylus remains in contact with the screen for an extended period of time. Long-press gesture recognizers use the duration—not the force—of the touches to determine the success or failure of the gesture. You might use a long-press gesture to initiate an action on the object being pressed. For example, you might use it to display a context-sensitive menu. Long-press gestures are continuous gestures, meaning that your action method may be called multiple times as the state changes.

A long-press gesture recognizer enters the `UIGestureRecognizerStateBegan` state after the gesture conditions have been met but while the user's finger is still on the screen. Any updates to the touch events cause the gesture recognizer to enter the `UIGestureRecognizerStateChanged` state. When the user's fingers lift from the screen, the gesture recognizer enters the `UIGestureRecognizerStateEnded` state.

Listing 2-2 shows an action method that displays a context menu on top of the view. It displays the context menu at the beginning of the gesture, while the user's finger is still on the screen. The view controller that implements this method also sets itself as the first responder so that it can respond to menu actions selected by the user.

Listing 2-2 Handling a long-press gesture

```swift
@IBAction func showResetMenu(_ gestureRecognizer: UILongPressGestureRecognizer) {
    if gestureRecognizer.state == .began {
        self.becomeFirstResponder()
        self.viewForReset = gestureRecognizer.view

        // Configure the menu item to display
        let menuItemTitle = NSLocalizedString("Reset", comment: "Reset menu item title")
        let action = #selector(ViewController.resetPiece(controller:))
        let resetMenuItem = UIMenuItem(title: menuItemTitle, action: action)

        // Configure the shared menu controller
        let menuController = UIMenuController.shared
        menuController.menuItems = [resetMenuItem]

        // Set the location of the menu in the view.
        let location = gestureRecognizer.location(in: gestureRecognizer.view)
        let menuLocation = CGRect(x: location.x, y: location.y, width: 0, height: 0)
        menuController.setTargetRect(menuLocation, in: gestureRecognizer.view!)

        // Show the menu.
        menuController.setMenuVisible(true, animated: true)
    }
}
```

If the code for your tap gesture recognizer is not called, check to see if the following conditions are true, and make corrections as needed:

- The `userInteractionEnabled` property of the view is set to `YES`. Image views and labels set this property to `NO` by default.
- The tap duration was greater than what is specified in the `minimumPressDuration` property.

- The number of taps was equal to the number in the `numberOfTapsRequired` property.
- The number of fingers was equal to the number specified in the `numberOfTouchesRequired` property.

On This Page