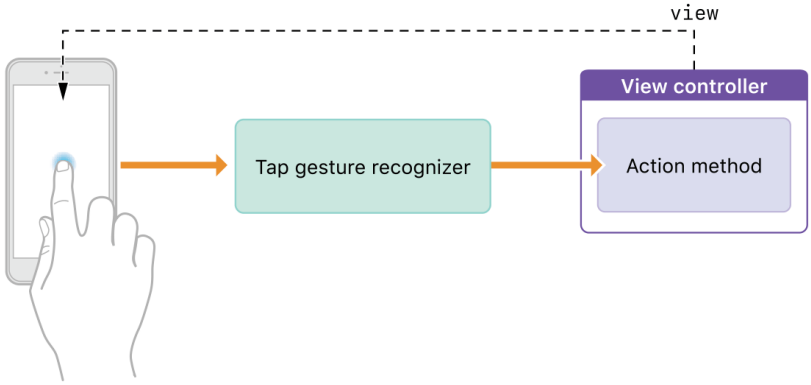
Gesture Recognizer Basics

Gesture recognizers are the simplest way to handle touch or press events in your views. You can attach one or more gesture recognizers to any view. Gesture recognizers encapsulate all of the logic needed to process and interpret incoming events for that view and match them to a known pattern. When a match is detected, the gesture recognizer notifies its assigned target object, which can be a view controller, the view itself, or any other object in your app. Notifications are sent using the target-action design pattern, an example of which is shown in Figure 1-1. When the `UITapGestureRecognizer` object detects a single-finger tap in the view, it calls an action method of the view's view controller, which you use to provide a response.

Figure 1-1 A gesture recognizer notifying its target



Gesture recognizers come in two types: discrete and continuous. A *discrete gesture recognizer* calls your action method once after the gesture is recognized. A *continuous gesture recognizer* may call your action method many times, including at the beginning and end of the gesture and each time the details of a tracked event change. For example, a `UIPanGestureRecognizer` object calls your action method whenever the position of a touch event changes.

Interface Builder includes objects for each of the standard UIKit gesture recognizers. It also includes a custom gesture recognizer object that you can use to represent your custom `UIGestureRecognizer` subclasses.

To configure a gesture recognizer

1. In your storyboard, drag the gesture recognizer onto your view.
2. Implement an action method to be called when the gesture is recognized; see Listing 1-1.
3. Connect your action method to the gesture recognizer.

   You can create this connection in Interface Builder by right-clicking the gesture recognizer and connecting its Sent Action selector to the appropriate object in your interface. You can also configure the action method programmatically using the `addTarget:action:` method of the gesture recognizer.

Listing 1-1 shows the generic format for the action method of a gesture recognizer. If you prefer, you can change the parameter type to match a specific gesture recognizer subclass.

Listing 1-1 Gesture recognizer action methods

OBJECTIVE-C

```
- (IBAction)myActionMethod:(UIGestureRecognizer*)sender
```

SWIFT

```
@IBAction func myActionMethod(_ sender: UIGestureRecognizer)
```

Responding to Gestures

The action method associated with a gesture recognizer provides your app's response to that gesture. For discrete gestures, your action method is similar to the action method for a button. Once the action method is called, you perform whatever task is appropriate for that gesture. For continuous gestures, your action method can respond to the recognition of the gesture, but it can also track events before the gesture is recognized. Tracking events lets you create a more interactive experience. For example, you might use the updates from a `UIPanGestureRecognizer` object to reposition content in your app.

The `state` property of a gesture recognizer communicates the object's current state of recognition. For continuous gestures, the gesture recognizer updates the value of this property from `UIGestureRecognizerStateBegan` to `UIGestureRecognizerStateChanged` to `UIGestureRecognizerStateEnded`. Your action metl
use the began a
make those changes permanent. Always check the value of the `state` property of a gesture recognizer before taking actions.

For examples of how to handle specific types of gestures, see the following information:

- Handling Tap and Long-Press Gestures
- Handling Pan and Swipe Gestures
- Handling Pinch and Rotation Gestures

For more information about gesture recognizer states and how they affect your code, see Implementing a Custom Gesture Recognizer.