

# Gesturizer

## Lesson 9



### Description

Enable the controller to detect a shake motion event.

### Learning Outcomes

- Distinguish a shake motion event from a UI gesture.
- Describe how iOS uses accelerometer data to send event notifications to responders.
- Apply `UIResponder` methods to enable a controller to respond to motion events.



### Vocabulary

motion events	accelerometer	UIEvent
responder	UIResponder	UIEventSubtype
enumeration		

### Materials

- **Gesturizer Lesson 9** Xcode project

### Opening

How can we detect when the user shakes the device?

## Agenda

- Using the Xcode Documentation and API Reference (⇧⌘0), explore the Motion Events documentation and draw attention to the section Detecting Shake-Motion Events with UIEvent.
- Discuss how shaking the device causes iOS to examine the accelerometer data and creates a UIEvent, which iOS sends to the active app.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the UIEvent class reference and draw attention to the UIEventSubtype enumeration.
- In the ViewController class, override the UIResponder canBecomeFirstResponder and motionEnded:withEvent: methods.

```
override func canBecomeFirstResponder() -> Bool {  
    return true  
}  
  
override func motionEnded(motion: UIEventSubtype,  
    withEvent event: UIEvent) {  
    if motion == .MotionShake {  
        showGestureName("Shake")  
    }  
}
```

- Explain how, when the accelerometer motion ends, iOS calls the motionEnded:withEvent: method, passing it the type of the motion and the UIEvent object associated with the motion.
- Run the app (⌘R), use the Simulator menu item *Hardware > Shake Gesture* (⌘Z) to simulate shaking the device, and observe the **Shake** text appear.

## Closing

We can implement our own UIGestureRecognizer classes, and detect touches and movement on our own. What are the benefits and drawbacks of using the built-in gesture recognizers? Do you have to swipe differently in one app than you do in another? Why do you think that is? In what situations might you implement your own UIGestureRecognizer?

## Modifications and Extensions

- Implement a UIGestureRecognizer that recognizes triangle or circle shaped gestures.

## Resources

Motion Events [https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/motion\\_event\\_basics/motion\\_event\\_basics.html](https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/motion_event_basics/motion_event_basics.html)

UIResponder Class Reference [https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIResponder\\_Class/index.html](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIResponder_Class/index.html)

UIEvent Class Reference [https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIEvent\\_Class/index.html](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIEvent_Class/index.html)