

# Quick Interaction Techniques for watchOS

Session 211

Tom Witkin watchOS Engineer

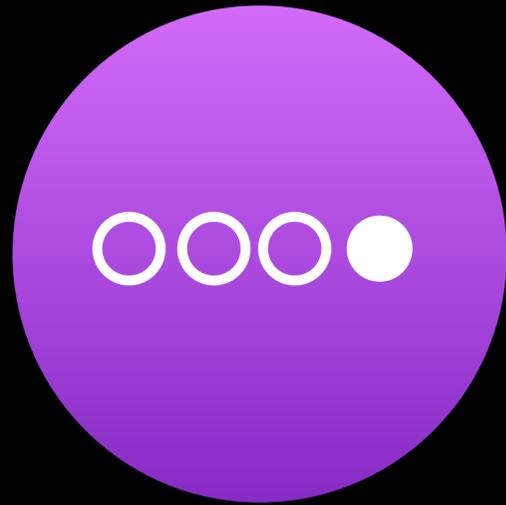
Miguel Sanchez watchOS Engineer

What is a quick interaction?

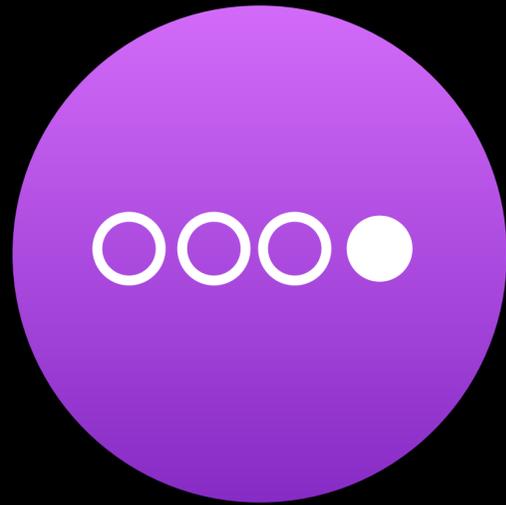








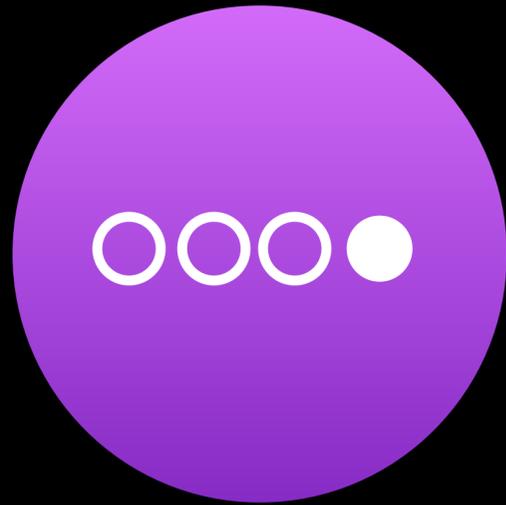
Glanceable



Glanceable



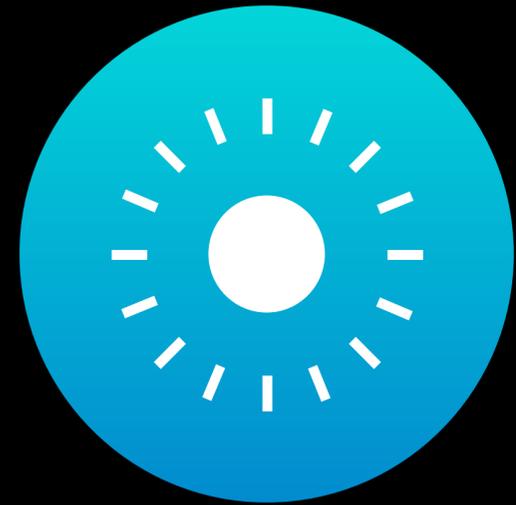
Actionable



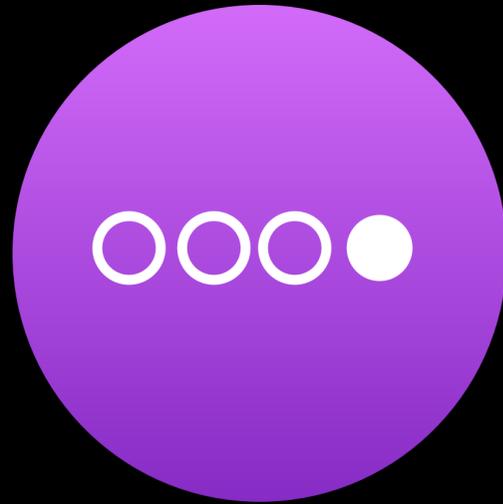
Glanceable



Actionable



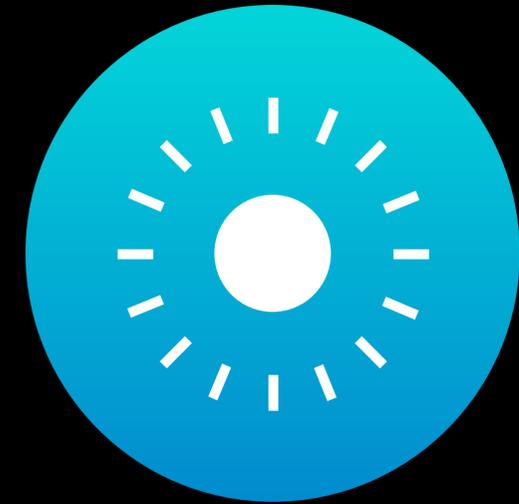
Responsive



Glanceable



Actionable



Responsive

---

Designing Great Apple Watch Experiences

Presidio

Wednesday 1:40PM

---

What's New in watchOS 3

WWDC 2016 Videos

---

How long is a  
quick interaction?

How long is a  
quick interaction?



2 seconds

watchOS

① New Features

- ① New Features
- ② Interaction Techniques

# New WatchKit Features

APIs for quick interactions

Miguel Sanchez watchOS Engineer

# watchOS 3 APIs

Glanceable, Actionable, Responsive

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

- Gesture recognizers

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

- Gesture recognizers
- Digital Crown rotation

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

- Gesture recognizers
- Digital Crown rotation

New ways of displaying and updating information

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

- Gesture recognizers
- Digital Crown rotation

New ways of displaying and updating information

- Improved table navigation

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

- Gesture recognizers
- Digital Crown rotation

New ways of displaying and updating information

- Improved table navigation
- Support for new Notifications Framework

# watchOS 3 APIs

Glanceable, Actionable, Responsive

Access new kinds of user input

- Gesture recognizers
- Digital Crown rotation

New ways of displaying and updating information

- Improved table navigation
- Support for new Notifications Framework
- SceneKit and SpriteKit integration

# Gesture Recognizers

Expanding your UI interactions

# Gesture Recognizers

NEW



# Gesture Recognizers

NEW

Discrete



# Gesture Recognizers

NEW

Discrete

`WKSwipeGestureRecognizer`



# Gesture Recognizers

NEW

Discrete

`WKSwipeGestureRecognizer`

`WKTapGestureRecognizer`



# Gesture Recognizers

NEW

Discrete

`WKSwipeGestureRecognizer`

`WKTapGestureRecognizer`

Continuous



# Gesture Recognizers

NEW

Discrete

`WKSwipeGestureRecognizer`

`WKTapGestureRecognizer`

Continuous

`WKPanGestureRecognizer`



# Gesture Recognizers

NEW

Discrete

`WKSwipeGestureRecognizer`

`WKTapGestureRecognizer`

Continuous

`WKPanGestureRecognizer`

`WKLongPressGestureRecognizer`



GestureTester: Ready | Today at 10:24 PM

GestureTester > Ge...pp > Int...ard > Int...se > Int...ene > Int...ller > Group > Swipe Gesture Recognizer

**GestureTester**

- GestureTester
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- GestureTester WatchKit App
  - Interface.storyboard M
  - Assets.xcassets
  - Info.plist
- GestureTester WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - Group
    - Tap Gesture Recognizer
    - Swipe Gesture Recognizer
    - Pan Gesture Recognizer
  - Main Entry Point

10:09

**Swipe Gesture Recognizer**

- Swipe: Right

**Gesture Recognizer**

- State:  Enabled
- Cancels touches in view
- Delays touches began
- Delays touches ended
- Installed
- Must Fail First:  Tap Gesture Recognizer  Pan Gesture Recognizer

**Tap Gesture Recognizer** - A recognizer for tap gestures.

**Swipe Gesture Recognizer** - A recognizer for swipe gestures.

**Long Press Gesture Recognizer** - A recognizer for long press gestures.

**Pan Gesture Recognizer** - A recognizer for pan gestures.

View as: Apple Watch 38mm | 100%

gesture

iPhone 6s + Apple Watch - 42mm | GestureTester: Ready | Today at 10:24 PM

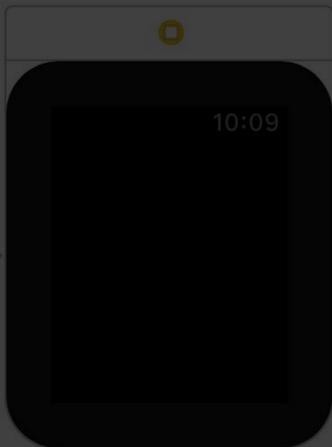
GestureTester > Ge...pp > Int...ard > Int...se > Int...ene > Int...ller > Group > Swipe Gesture Recognizer

**GestureTester**

- GestureTester
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- GestureTester WatchKit App
  - Interface.storyboard M
  - Assets.xcassets
  - Info.plist
- GestureTester WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - Group
    - Tap Gesture Recognizer
    - Swipe Gesture Recognizer
    - Pan Gesture Recognizer
  - Main Entry Point



**Swipe Gesture Recognizer**

+ Swipe Right

**Gesture Recognizer**

+ State  Enabled

+  Cancels touches in view

+  Delays touches began

+  Delays touches ended

+  Installed

Must Fail First  Tap Gesture Recognizer  Pan Gesture Recognizer

**Tap Gesture Recognizer** - A recognizer for tap gestures.

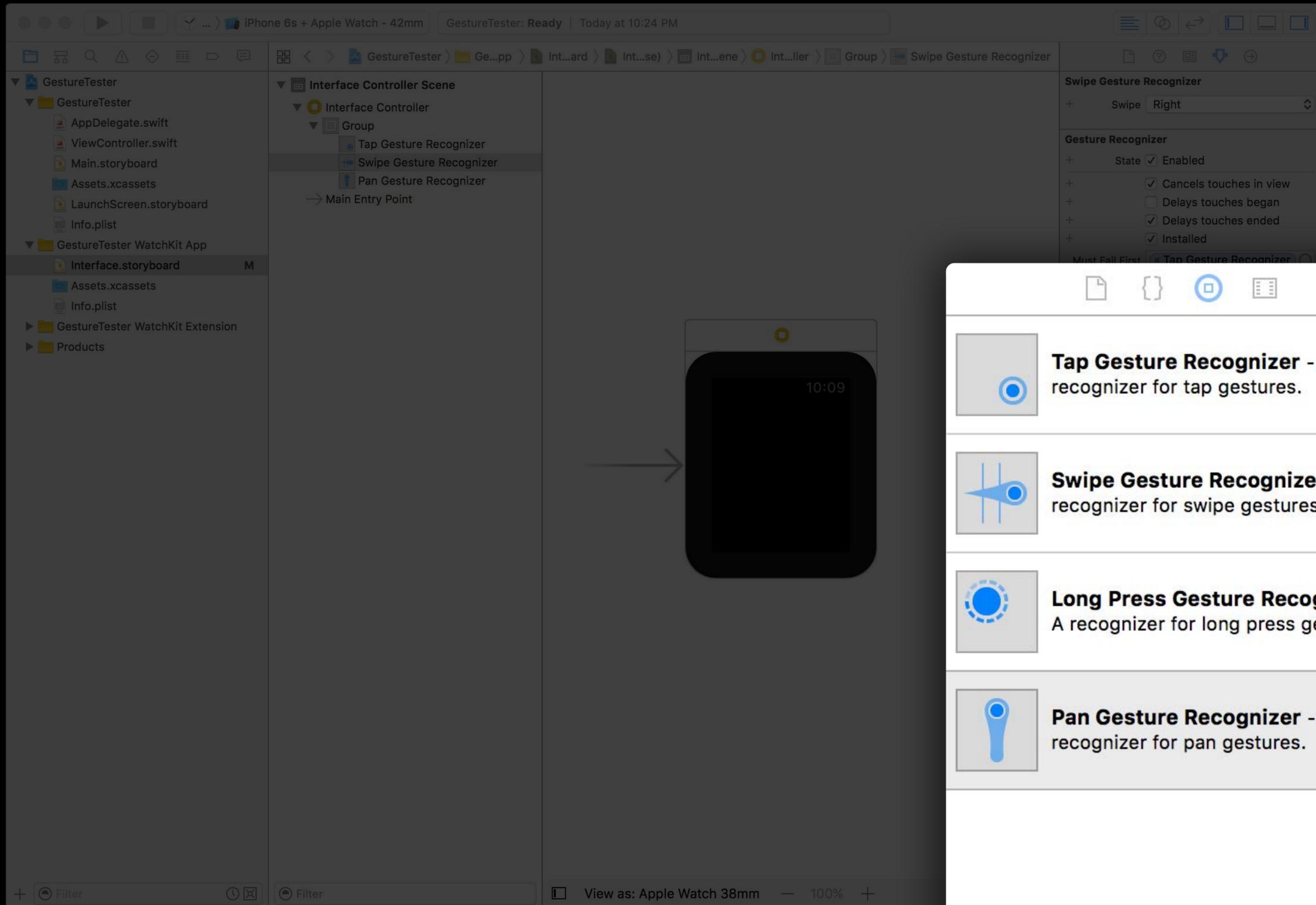
**Swipe Gesture Recognizer** - A recognizer for swipe gestures.

**Long Press Gesture Recognizer** - A recognizer for long press gestures.

**Pan Gesture Recognizer** - A recognizer for pan gestures.

View as: Apple Watch 38mm 100%

gesture



gesture

-  **Tap Gesture Recognizer** - A recognizer for tap gestures.
-  **Swipe Gesture Recognizer** - A recognizer for swipe gestures.
-  **Long Press Gesture Recognizer** - A recognizer for long press gestures.
-  **Pan Gesture Recognizer** - A recognizer for pan gestures.

gesture

iPhone 6s + Apple Watch - 42mm | GestureTester: Ready | Today at 10:24 PM

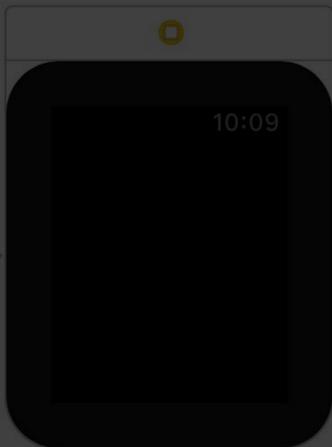
GestureTester > Ge...pp > Int...ard > Int...se > Int...ene > Int...ller > Group > Swipe Gesture Recognizer

**GestureTester**

- GestureTester
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- GestureTester WatchKit App
  - Interface.storyboard M
  - Assets.xcassets
  - Info.plist
- GestureTester WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - Group
    - Tap Gesture Recognizer
    - Swipe Gesture Recognizer
    - Pan Gesture Recognizer
  - Main Entry Point



**Swipe Gesture Recognizer**

+ Swipe Right

**Gesture Recognizer**

+ State  Enabled

+  Cancels touches in view

+  Delays touches began

+  Delays touches ended

+  Installed

Must Fail First  Tap Gesture Recognizer  Pan Gesture Recognizer

**Tap Gesture Recognizer** - A recognizer for tap gestures.

**Swipe Gesture Recognizer** - A recognizer for swipe gestures.

**Long Press Gesture Recognizer** - A recognizer for long press gestures.

**Pan Gesture Recognizer** - A recognizer for pan gestures.

View as: Apple Watch 38mm 100%

gesture

▼  **Interface Controller Scene**

▼  **Interface Controller**

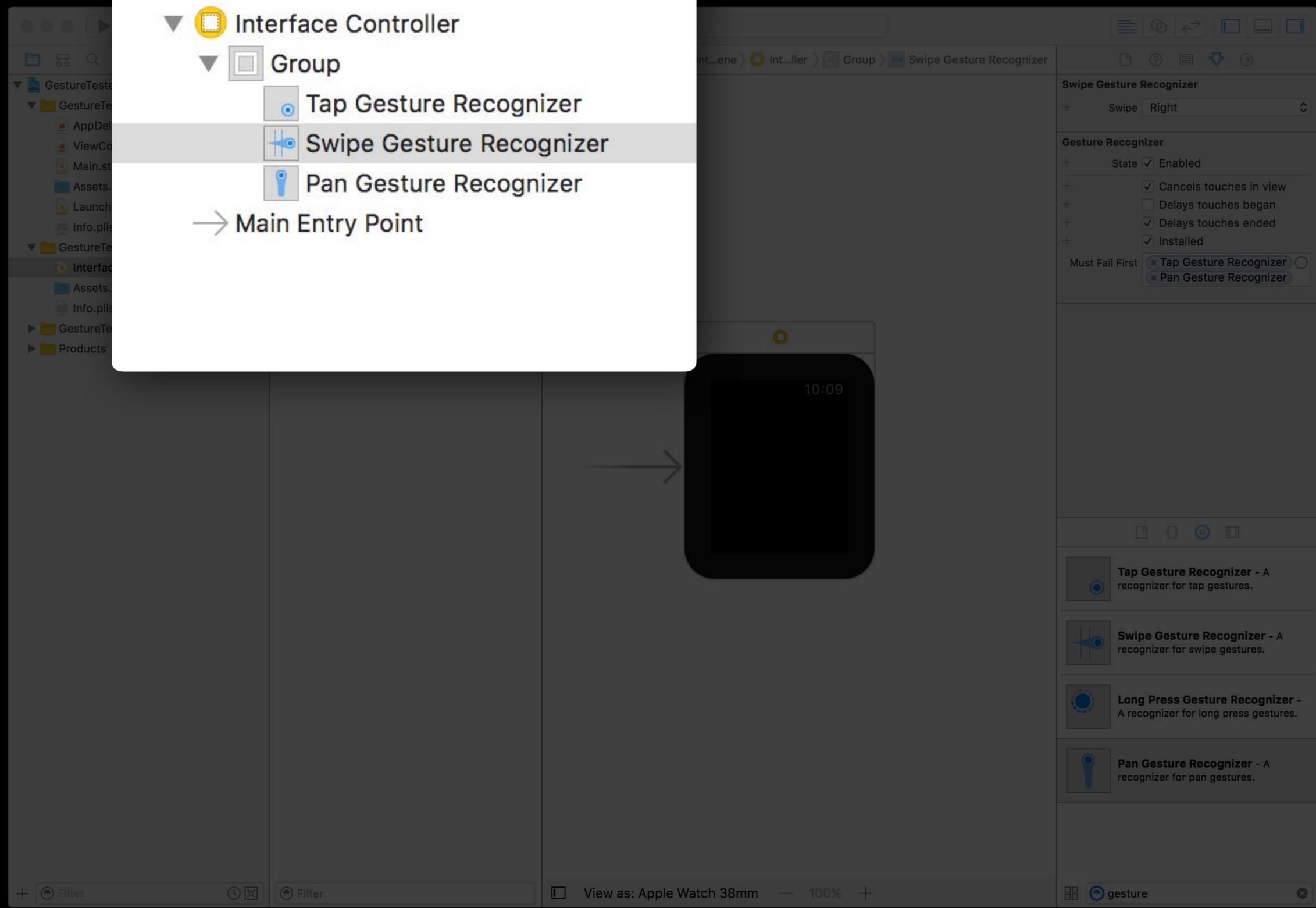
▼  **Group**

 Tap Gesture Recognizer

 **Swipe Gesture Recognizer**

 Pan Gesture Recognizer

→ Main Entry Point



iPhone 6s + Apple Watch - 42mm | GestureTester: Ready | Today at 10:24 PM

GestureTester > Ge...pp > Int...ard > Int...se > Int...ene > Int...ller > Group > Swipe Gesture Recognizer

**GestureTester**

- GestureTester
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- GestureTester WatchKit App
  - Interface.storyboard M
  - Assets.xcassets
  - Info.plist
- GestureTester WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - Group
    - Tap Gesture Recognizer
    - Swipe Gesture Recognizer
    - Pan Gesture Recognizer
  - Main Entry Point

10:09

**Swipe Gesture Recognizer**

- Swipe: Right

**Gesture Recognizer**

- State:  Enabled
- Cancels touches in view
- Delays touches began
- Delays touches ended
- Installed
- Must Fail First:
  - Tap Gesture Recognizer
  - Pan Gesture Recognizer

Tap Gesture Recognizer - A recognizer for tap gestures.

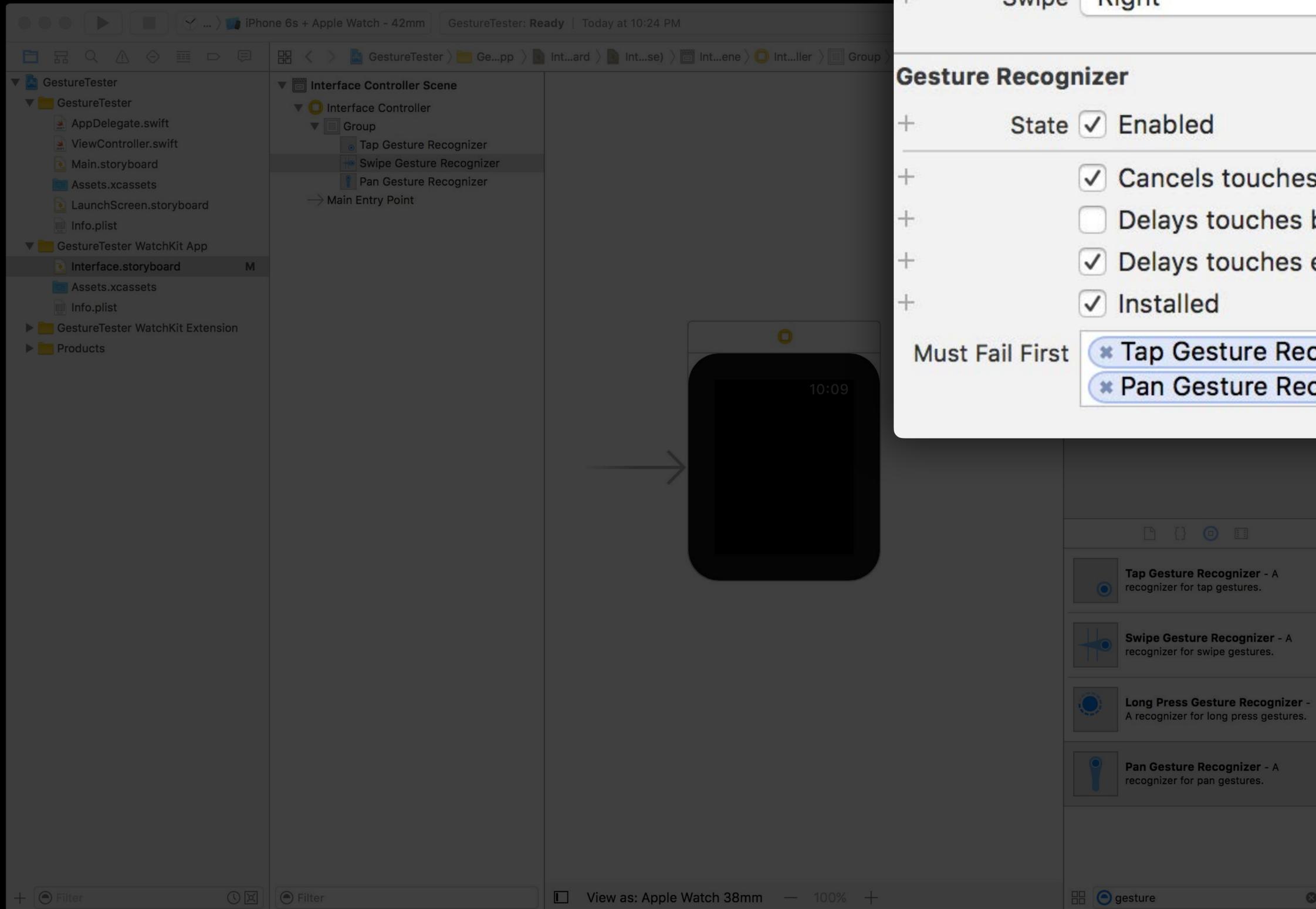
Swipe Gesture Recognizer - A recognizer for swipe gestures.

Long Press Gesture Recognizer - A recognizer for long press gestures.

Pan Gesture Recognizer - A recognizer for pan gestures.

View as: Apple Watch 38mm | 100%

gesture



### Swipe Gesture Recognizer

+ Swipe Right

---

### Gesture Recognizer

+ State  Enabled

+  Cancels touches in view

+  Delays touches began

+  Delays touches ended

+  Installed

Must Fail First  Tap Gesture Recognizer  Pan Gesture Recognizer

# WKGestureRecognizer

Base Class

# WKGestureRecognizer

Base Class

Associate recognizer with action method

```
IBAction func handleGesture(recognizer : WKGestureRecognizer?)
```

# WKGestureRecognizer

## Base Class

Associate recognizer with action method

```
IBAction func handleGesture(recognizer : WKGestureRecognizer?)
```

Recognizer properties

```
func locationInObject() -> CGPoint  
func objectBounds() -> CGRect  
var state : WKGestureRecognizerState
```

# WKGestureRecognizerState

Discrete gestures

Action method called when recognized

# WKGestureRecognizerState

Discrete gestures

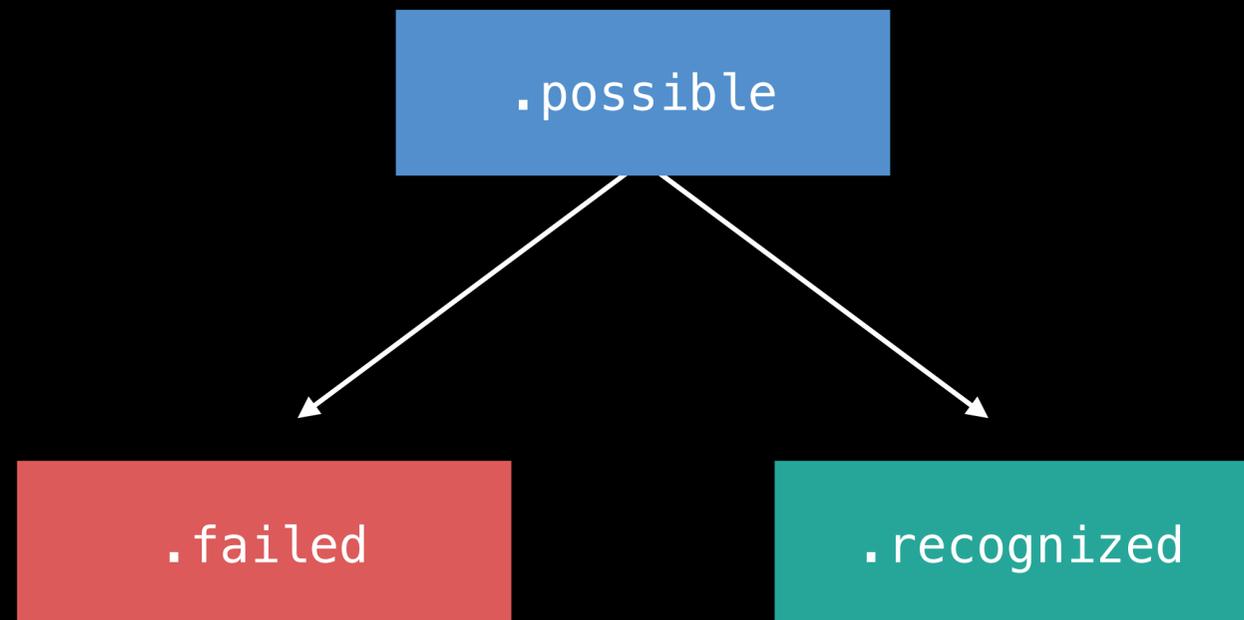
Action method called when recognized

```
.possible
```

# WKGestureRecognizerState

Discrete gestures

Action method called when recognized



# WKGestureRecognizerState

Continuous gestures

Action method called many times

# WKGestureRecognizerState

Continuous gestures

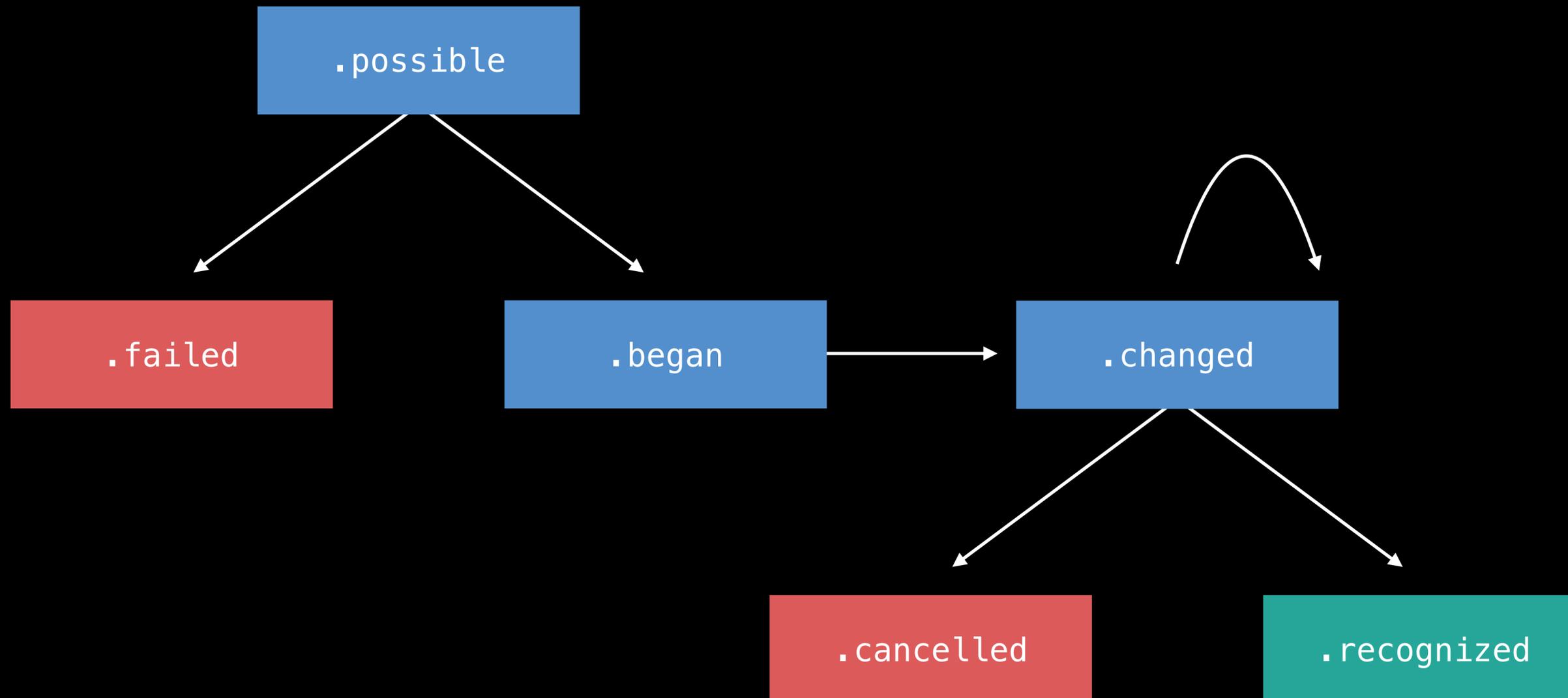
Action method called many times

```
.possible
```

# WKGestureRecognizerState

Continuous gestures

Action method called many times



# Tips for Using Gesture Recognizers

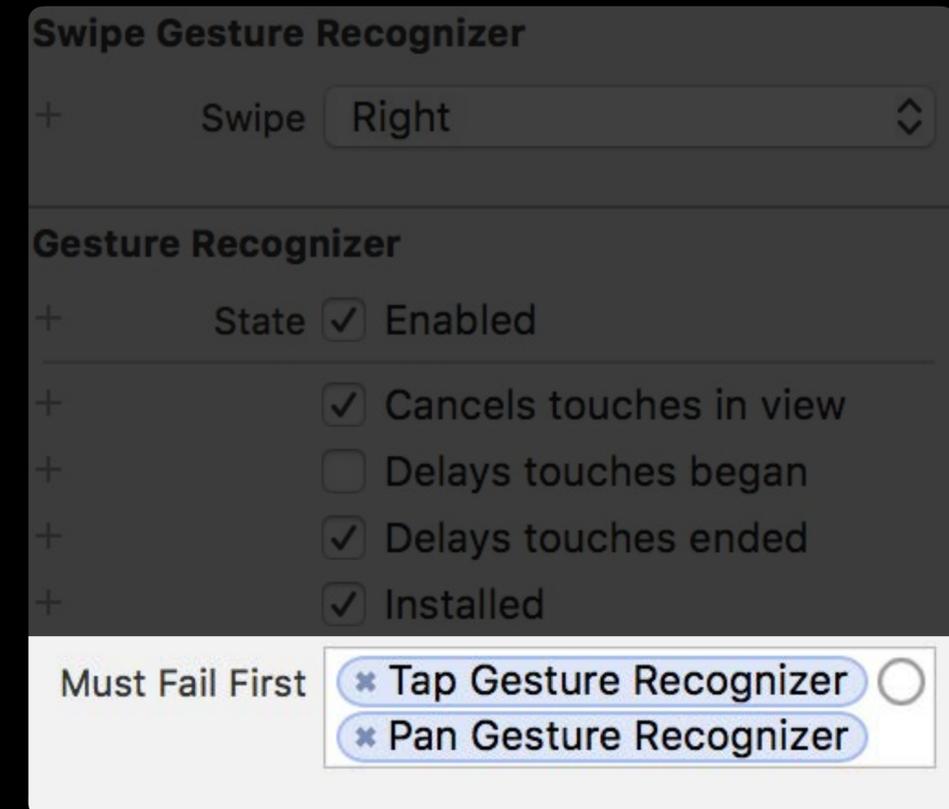
# Tips for Using Gesture Recognizers

Add gestures to groups instead of controls

# Tips for Using Gesture Recognizers

Add gestures to groups instead of controls

Set dependency relationships in storyboard

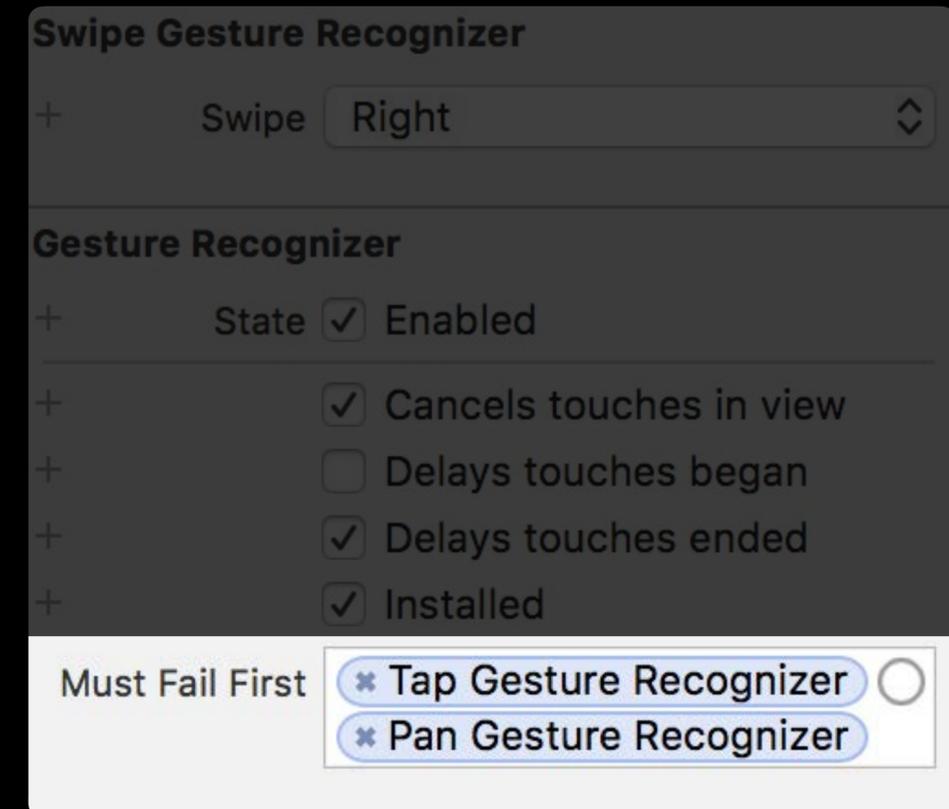


# Tips for Using Gesture Recognizers

Add gestures to groups instead of controls

Set dependency relationships in storyboard

Some system gestures will have precedence



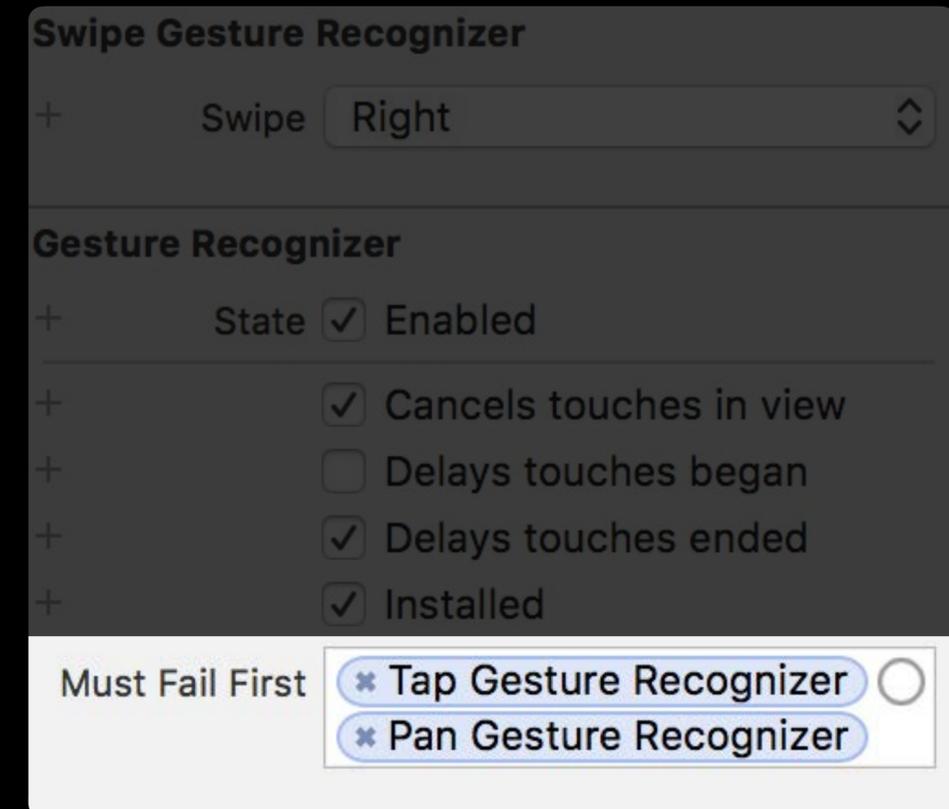
# Tips for Using Gesture Recognizers

Add gestures to groups instead of controls

Set dependency relationships in storyboard

Some system gestures will have precedence

- Scrolling
- Force Touch



# Digital Crown Rotation

Increasing speed and precision of interactions

# WKInterfacePicker Refresher

Introduced in watchOS 2

# WKInterfacePicker Refresher

Introduced in watchOS 2

You provide

- A list of **WKPickerItems**
- Picker style

# WKInterfacePicker Refresher

Introduced in watchOS 2

You provide

- A list of **WKPickerItems**
- Picker style



# WKInterfacePicker Refresher

Introduced in watchOS 2

You provide

- A list of **WKPickerItems**
- Picker style



# WKInterfacePicker Refresher

Introduced in watchOS 2

You provide

- A list of **WKPickerItems**
- Picker style



# WKInterfacePicker Refresher

Introduced in watchOS 2

You provide

- A list of **WKPickerItems**
- Picker style

WatchKit handles

- Look and feel of interface element
- Processing of Digital Crown events
- Calling action on item selection



# Digital Crown Support

Direct access to rotation events

NEW

# Digital Crown Support

Direct access to rotation events

Control your own UI elements with rotation values

NEW

# Digital Crown Support

Direct access to rotation events

Control your own UI elements with rotation values

## `WKCrownSequencer`

- Gives rotations per second

NEW

# Digital Crown Support

NEW

Direct access to rotation events

Control your own UI elements with rotation values

## **WKCrownSequencer**

- Gives rotations per second

## **WKCrownDelegate** Protocol

- Callbacks with rotational deltas

# Rotations Per Second

Useful for physics based animations

# Rotations Per Second

Useful for physics based animations

Access sequencer from interface controller

```
var crownSequencer: WKCrownSequencer { get }
```

# Rotations Per Second

Useful for physics based animations

Access sequencer from interface controller

```
var crownSequencer: WKCrownSequencer { get }
```

Get RPS from sequencer

```
var rotationsPerSecond: Double { get }
```

# Rotational Deltas

Counting number of rotations

# Rotational Deltas

Counting number of rotations

Argument in WKCrownDelegate callback

```
func crownDidRotate(sequencer : WKCrownSequencer?, rotationalDelta: Double)
```

# Rotational Deltas

Counting number of rotations

Argument in WKCrownDelegate callback

```
func crownDidRotate(sequencer : WKCrownSequencer?, rotationalDelta: Double)
```

Maintain accumulator to determine number of rotations

# Rotational Deltas

Counting number of rotations

Argument in WKCrownDelegate callback

```
func crownDidRotate(sequencer : WKCrownSequencer?, rotationalDelta: Double)
```

Maintain accumulator to determine number of rotations

- 1 full rotation = accumulated deltas of 1

# Rotational Deltas

Counting number of rotations

Argument in WKCrownDelegate callback

```
func crownDidRotate(sequencer : WKCrownSequencer?, rotationalDelta: Double)
```

Maintain accumulator to determine number of rotations

- 1 full rotation = accumulated deltas of 1
- Half rotation = accumulated deltas of 0.5

# Rotational Deltas

Counting number of rotations

Argument in WKCrownDelegate callback

```
func crownDidRotate(sequencer : WKCrownSequencer?, rotationalDelta: Double)
```

Maintain accumulator to determine number of rotations

- 1 full rotation = accumulated deltas of 1
- Half rotation = accumulated deltas of 0.5

Determine how count corresponds to sensitivity of changes in your content

# Reported Values for RPS and Deltas

Sign indicates direction

# Reported Values for RPS and Deltas

Sign indicates direction



# Reported Values for RPS and Deltas

Sign indicates direction

Positive up rotation



# Reported Values for RPS and Deltas

Sign indicates direction

Positive up rotation

Negative down rotation



# Reported Values for RPS and Deltas

Sign indicates direction

Positive up rotation

Negative down rotation

Values are orientation agnostic



# Digital Crown Focus

Sequencer only reports values when focused

# Digital Crown Focus

Sequencer only reports values when focused

Other elements can take over Digital Crown focus

- Picker
- Slider
- Scrolling controller

# Digital Crown Focus

Sequencer only reports values when focused

Other elements can take over Digital Crown focus

- Picker
- Slider
- Scrolling controller

Can use a gesture to gain focus on your UI element

# Digital Crown Focus

Sequencer only reports values when focused

Other elements can take over Digital Crown focus

- Picker
- Slider
- Scrolling controller

Can use a gesture to gain focus on your UI element

Focus WKCrownSequencer to receive values

```
func focus()
```

# Vertical Paging

Faster table row detail navigation

# Hierarchical Navigation



# Hierarchical Navigation



# Hierarchical Navigation



# Hierarchical Navigation



# Vertical Paging

NEW



# Vertical Paging

NEW



# Make Your App More Responsive

Doesn't require navigating back to table

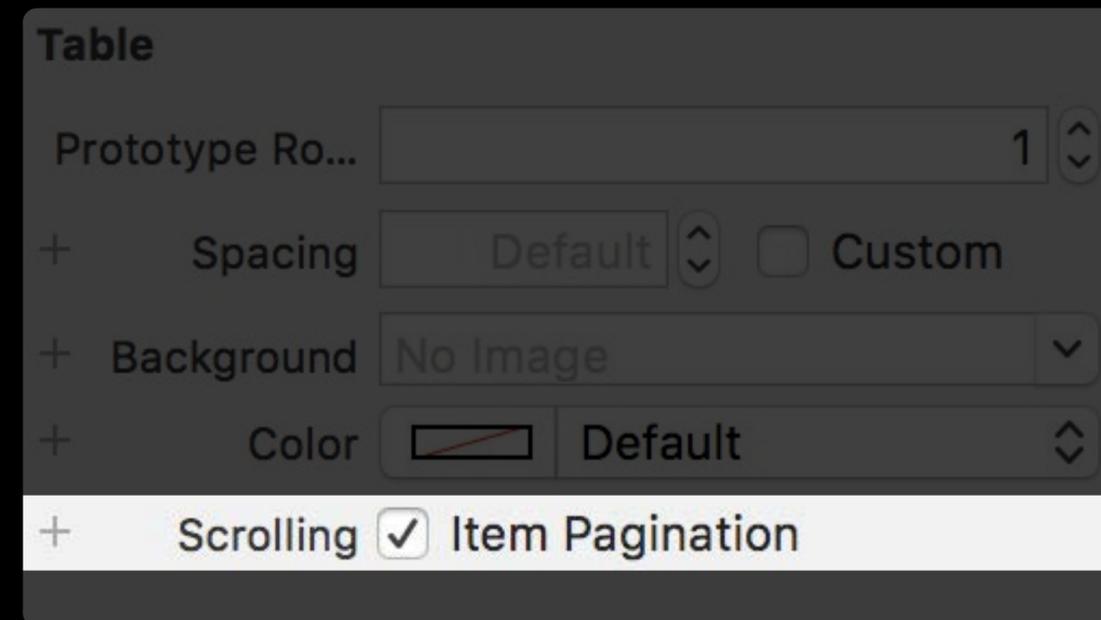
NEW

# Make Your App More Responsive

NEW

Doesn't require navigating back to table

One checkbox in Storyboard



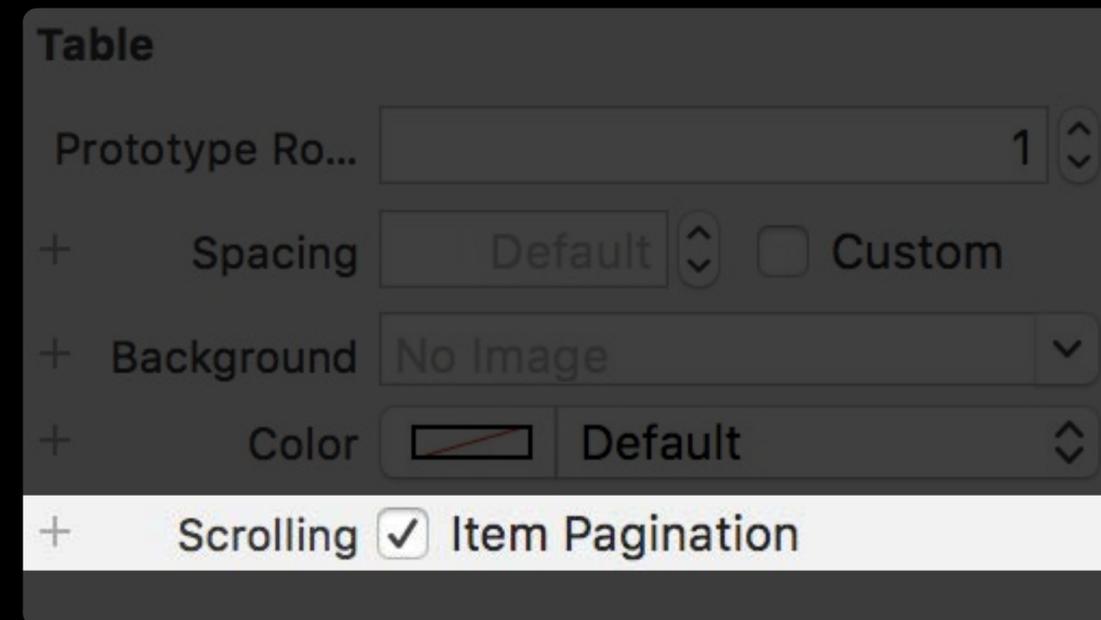
# Make Your App More Responsive

NEW

Doesn't require navigating back to table

One checkbox in Storyboard

Table must be using segues



# watchOS 3 Table API

Mostly the same

# watchOS 3 Table API

Mostly the same

New method to segue into vertical paging

```
func performSegue(row : Int)
```

# watchOS 3 Table API

Mostly the same

New method to segue into vertical paging

```
func performSegue(row : Int)
```

WatchKit handles preheating of controllers

- Some controller lifecycle methods called when not yet visible

# watchOS 3 Table API

Mostly the same

New method to segue into vertical paging

```
func performSegue(row : Int)
```

WatchKit handles preheating of controllers

- Some controller lifecycle methods called when not yet visible

# Notifications

The original quick interactions

# Notifications Refresher

Apple Watch integral part of notification flow

# Notifications Refresher

Apple Watch integral part of notification flow

Quick interaction

# Notifications Refresher

Apple Watch integral part of notification flow

Quick interaction

- User feels a haptic

# Notifications Refresher

Apple Watch integral part of notification flow

Quick interaction

- User feels a haptic
- Sees a Short Look

# Notifications Refresher

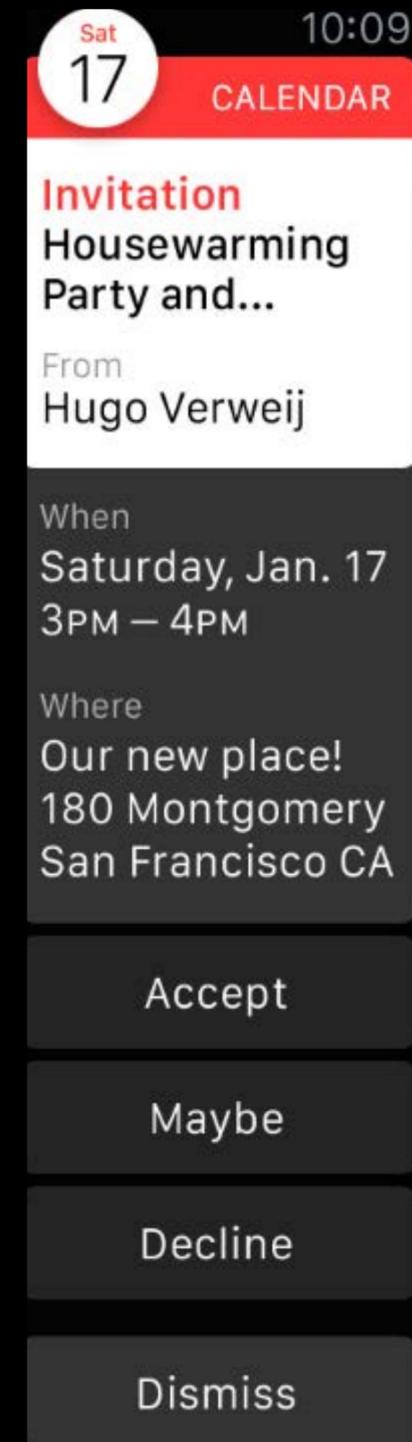
Apple Watch integral part of notification flow

Quick interaction

- User feels a haptic
- Sees a Short Look
- Sees Long Look after short interval

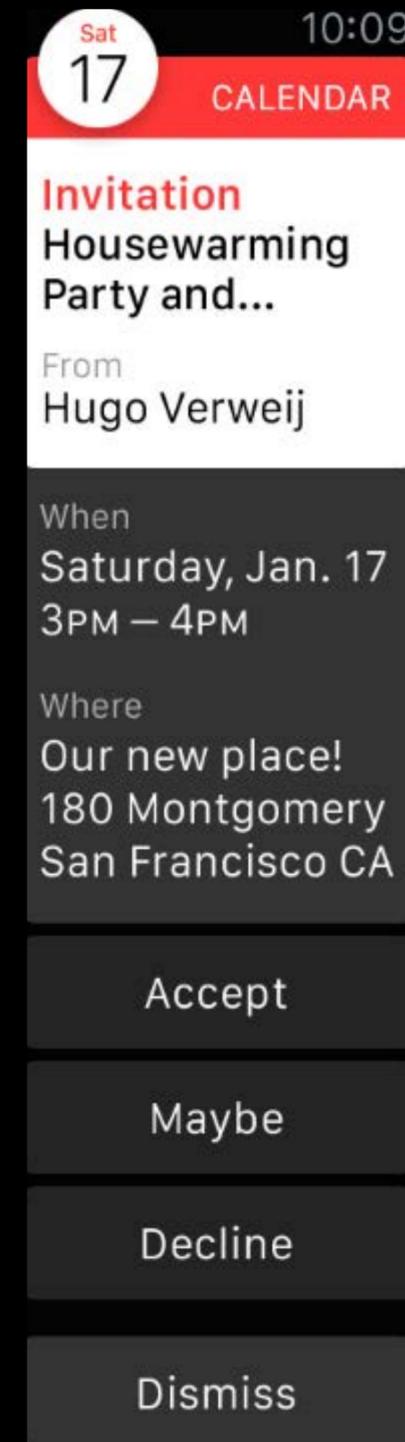
# User Response to Notification

# User Response to Notification



# User Response to Notification

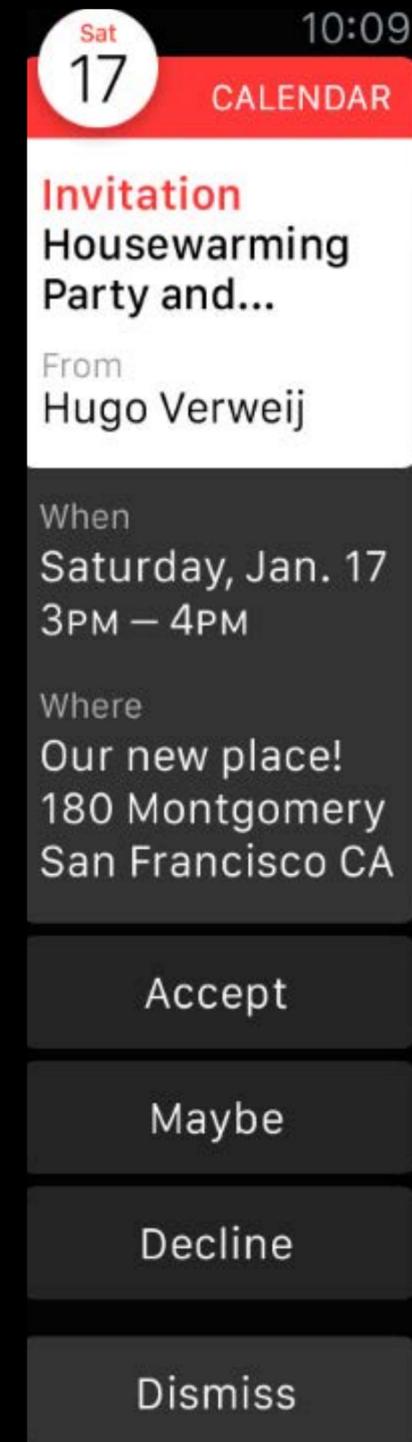
Dismiss notification



# User Response to Notification

Dismiss notification

Launch watch app

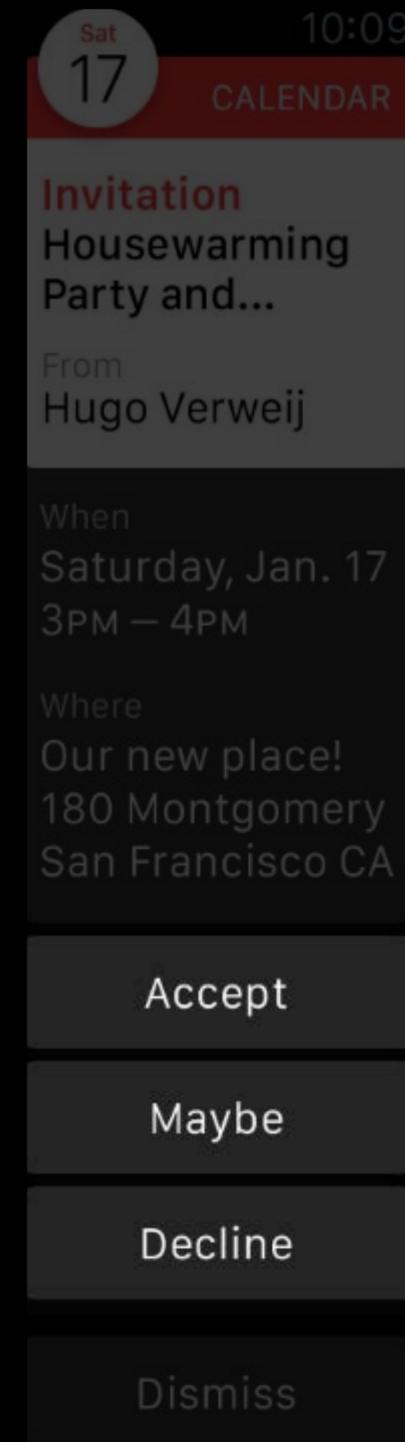


# User Response to Notification

Dismiss notification

Launch watch app

Notifications support custom actions



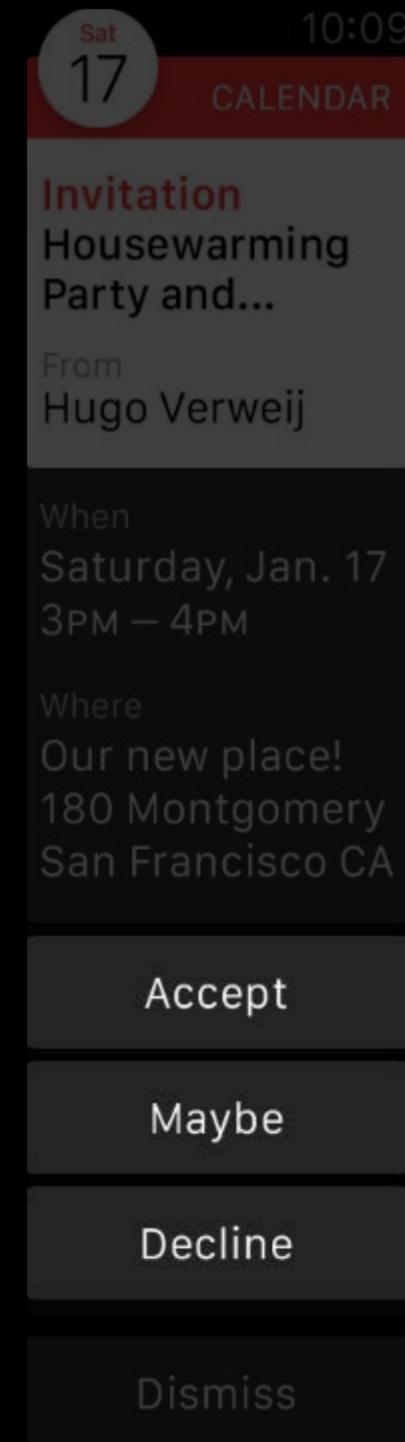
# User Response to Notification

Dismiss notification

Launch watch app

Notifications support custom actions

- Foreground actions make app active
- Background actions routed to iPhone app in watchOS 2



# watchOS 3 Notification Support

# watchOS 3 Notification Support

New User Notification Framework

---

Introduction to Notifications

---

WWDC 2016 Videos

# watchOS 3 Notification Support

New User Notification Framework

---

Introduction to Notifications

---

WWDC 2016 Videos

WatchKit

# watchOS 3 Notification Support

New User Notification Framework

---

Introduction to Notifications

---

WWDC 2016 Videos

WatchKit

- Notification handling APIs similar across platforms

# watchOS 3 Notification Support

New User Notification Framework

---

Introduction to Notifications

---

WWDC 2016 Videos

WatchKit

- Notification handling APIs similar across platforms
- Local notifications can now be scheduled on Apple Watch

# watchOS 3 Notification Support

New User Notification Framework

---

Introduction to Notifications

---

WWDC 2016 Videos

WatchKit

- Notification handling APIs similar across platforms
- Local notifications can now be scheduled on Apple Watch
- Background actions now delivered to extension if scheduled on Apple Watch

# UNNotificationCenter

NEW

Register to be delegate

```
class ExtensionDelegate: NSObject, WKExtensionDelegate, UNNotificationCenterDelegate {  
    func applicationDidFinishLaunching() {  
        UNNotificationCenter.current().delegate = self;  
    }  
}
```

# UNUserNotificationCenter

NEW

Register to be delegate

```
class ExtensionDelegate: NSObject, WKExtensionDelegate, UNUserNotificationCenterDelegate {  
    func applicationDidFinishLaunching() {  
        UNUserNotificationCenter.current().delegate = self;  
    }  
}
```

# UNNotificationCenter

NEW

Register to be delegate

```
class ExtensionDelegate: NSObject, WKExtensionDelegate, UNNotificationCenterDelegate {  
    func applicationDidFinishLaunching() {  
        UNNotificationCenter.current().delegate = self;  
    }  
}
```

# UNNotificationCenterDelegate

willPresent notification callback

NEW

# UNUserNotificationCenterDelegate

NEW

willPresent notification callback

Called only if application is active

```
func userNotificationCenter(center: UNUserNotificationCenter,  
                           willPresent: UNNotification,  
                           completionHandler: (UNNotificationPresentationOptions))
```

# UNUserNotificationCenterDelegate

NEW

willPresent notification callback

Called only if application is active

```
func userNotificationCenter(center: UNUserNotificationCenter,  
                           willPresent: UNNotification,  
                           completionHandler: (UNNotificationPresentationOptions))
```

Completion handler argument determines presentation

# UNUserNotificationCenterDelegate

NEW

willPresent notification callback

Called only if application is active

```
func userNotificationCenter(center: UNUserNotificationCenter,  
                           willPresent: UNNotification,  
                           completionHandler: (UNNotificationPresentationOptions))
```

Completion handler argument determines presentation

- `.alert`      Presents Long Look
- `.none`      App handles notification

# UNNotificationCenterDelegate

didReceive response callback

NEW

# UNNotificationCenterDelegate

NEW

didReceive response callback

```
func userNotificationCenter(center : UNNotificationCenter,  
                           didReceive: UNNotificationResponse,  
                           completionHandler: (UNNotificationPresentationOptions))
```

# UNNotificationCenterDelegate

NEW

didReceive response callback

```
func userNotificationCenter(center : UNNotificationCenter,  
                           didReceive: UNNotificationResponse,  
                           completionHandler: (UNNotificationPresentationOptions))
```

Called when a foreground action makes app active

# UNNotificationCenterDelegate

NEW

didReceive response callback

```
func userNotificationCenter(center : UNNotificationCenter,  
                           didReceive: UNNotificationResponse,  
                           completionHandler: (UNNotificationPresentationOptions))
```

Called when a foreground action makes app active

Called for background actions on notifications

# UNNotificationCenterDelegate

NEW

didReceive response callback

```
func userNotificationCenter(center : UNNotificationCenter,  
                           didReceive: UNNotificationResponse,  
                           completionHandler: (UNNotificationPresentationOptions))
```

Called when a foreground action makes app active

Called for background actions on notifications

- If scheduled on Apple Watch

# Notification Scheduling and Delivery

Notifications on iPhone forward to Apple Watch when

- iPhone screen off
- Apple Watch on-wrist and unlocked

# Notification Scheduling and Delivery

# Notification Scheduling and Delivery

watchOS 3 allows scheduling of local notifications on Apple Watch

# Notification Scheduling and Delivery

watchOS 3 allows scheduling of local notifications on Apple Watch

- Delivered only on Apple Watch

# Notification Scheduling and Delivery

watchOS 3 allows scheduling of local notifications on Apple Watch

- Delivered only on Apple Watch
- Schedule iPhone notification if required

# Notification Scheduling and Delivery

watchOS 3 allows scheduling of local notifications on Apple Watch

- Delivered only on Apple Watch
- Schedule iPhone notification if required

Notification identifiers used to avoid duplicate alerts

# Notification Scheduling and Delivery

watchOS 3 allows scheduling of local notifications on Apple Watch

- Delivered only on Apple Watch
- Schedule iPhone notification if required

Notification identifiers used to avoid duplicate alerts

- Apple Watch notifications take precedence over forwarded with same id

# SpriteKit and SceneKit

Enhanced UI Interactions

# Classes to Display Scenes

NEW

# Classes to Display Scenes

NEW



`WKInterfaceSKScene`  
for SpriteKit

# Classes to Display Scenes

NEW



`WKInterfaceSKScene`  
for SpriteKit



`WKInterfaceSCNScene`  
for SceneKit

iPhone 6s + Apple Watch - 42mm | Scene Tests: Ready | Today at 10:48 PM

Scene Tests > Sc...App > Int...oard > Int...ase > Int...ene > Interface Controller > SceneKit Scene

**Scene Tests**

- Scene Tests
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Scene Tests WatchKit App
  - Interface.storyboard
  - Assets.xcassets
  - Info.plist
- Scene Tests WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - SceneKit Scene
  - Main Entry Point
- Interface Controller Scene
  - Interface Controller
    - SpriteKit Scene

SceneKit Scene

10:09

SceneKit Scene

Interface Controller

10:09

SpriteKit Scene

**SceneKit Scene**

- Scene: Scene name
- Antialiasing: None
- Frame Rate: Automatic Per Second
- Rendering:  Default Lighting
- Jittering
- Animations:  Play
- Loop

**View**

- Alpha: 1
- Hidden
- Installed
- Semantic: Unspecified

**Alignment**

- Horizontal: Left
- Vertical: Top

**Size**

- Width: Size To Fit Content
- Height: Size To Fit Content

SceneKit Scene - Displays SceneKit content.

SpriteKit Scene - Displays SpriteKit content.

View as: Apple Watch 42mm | 100%

scene

iPhone 6s + Apple Watch - 42mm | Scene Tests: Ready | Today at 10:48 PM

Scene Tests > Sc...App > Int...oard > Int...ase) > Int...ene > Interface Controller > SceneKit Scene

**Scene Tests**

- Scene Tests
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Scene Tests WatchKit App
  - Interface.storyboard
  - Assets.xcassets
  - Info.plist
- Scene Tests WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - SceneKit Scene
  - Main Entry Point
- Interface Controller Scene
  - Interface Controller
    - SpriteKit Scene

SceneKit Scene

10:09

SceneKit Scene

Interface Controller

10:09

SpriteKit Scene

**SceneKit Scene**

- Scene: Scene name
- Antialiasing: None
- Frame Rate: Automatic Per Second
- Rendering:  Default Lighting
- Jittering
- Animations:  Play
- Loop

**View**

- Alpha: 1
- Hidden
- Installed
- Semantic: Unspecified

**Alignment**

- Horizontal: Left
- Vertical: Top

**Size**

- Width: Size To Fit Content
- Height: Size To Fit Content

SceneKit Scene - Displays SceneKit content.

SpriteKit Scene - Displays SpriteKit content.

View as: Apple Watch 42mm | 100%

scene

iPhone 6s + Apple Watch - 42mm | Scene Tests: Ready | Today at 10:48 PM

Scene Tests > Sc...App > Int...oard > Int...ase) > Int...ene > Interface Controller > SceneKit Scene

Scene Tests

- Scene Tests
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Scene Tests WatchKit App
  - Interface.storyboard
  - Assets.xcassets
  - Info.plist
- Scene Tests WatchKit Extension
- Products

Interface Controller Scene

- Interface Controller
  - SceneKit Scene
  - Main Entry Point
- Interface Controller Scene
  - Interface Controller
    - SpriteKit Scene

SceneKit Scene

10:09

SceneKit Scene

Interface Controller

10:09

SpriteKit Scene

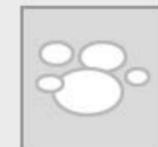
View as: Apple Watch 42mm 100%

SceneKit Scene

- Scene: Scene name
- Antialiasing: None
- Frame Rate: Automatic Per Second
- Rendering:  Default Lighting
- Jittering:
- Animations:  Play
- Loop:
- View
  - Alpha: 1
  - Hidden:
  - Installed:
  - Semantic: Unspecified
  - Alignment
    - Horizontal: Left



**SceneKit Scene** - Displays SceneKit content.



**SpriteKit Scene** - Displays SpriteKit content.



scene



iPhone 6s + Apple Watch - 42mm | Scene Tests: Ready | Today at 10:48 PM

Scene Tests > Sc...App > Int...oard > Int...ase) > Int...ene > Interface Controller > SceneKit Scene

**Scene Tests**

- Scene Tests
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Scene Tests WatchKit App
  - Interface.storyboard
  - Assets.xcassets
  - Info.plist
- Scene Tests WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - SceneKit Scene
  - Main Entry Point
- Interface Controller Scene
  - Interface Controller
    - SpriteKit Scene

SceneKit Scene

10:09

SceneKit Scene

Interface Controller

10:09

SpriteKit Scene

**SceneKit Scene**

- Scene: Scene name
- Antialiasing: None
- Frame Rate: Automatic Per Second
- Rendering:  Default Lighting
- Jittering
- Animations:  Play
- Loop

**View**

- Alpha: 1
- Hidden
- Installed
- Semantic: Unspecified

**Alignment**

- Horizontal: Left
- Vertical: Top

**Size**

- Width: Size To Fit Content
- Height: Size To Fit Content

SceneKit Scene - Displays SceneKit content.

SpriteKit Scene - Displays SpriteKit content.

View as: Apple Watch 42mm | 100%

scene

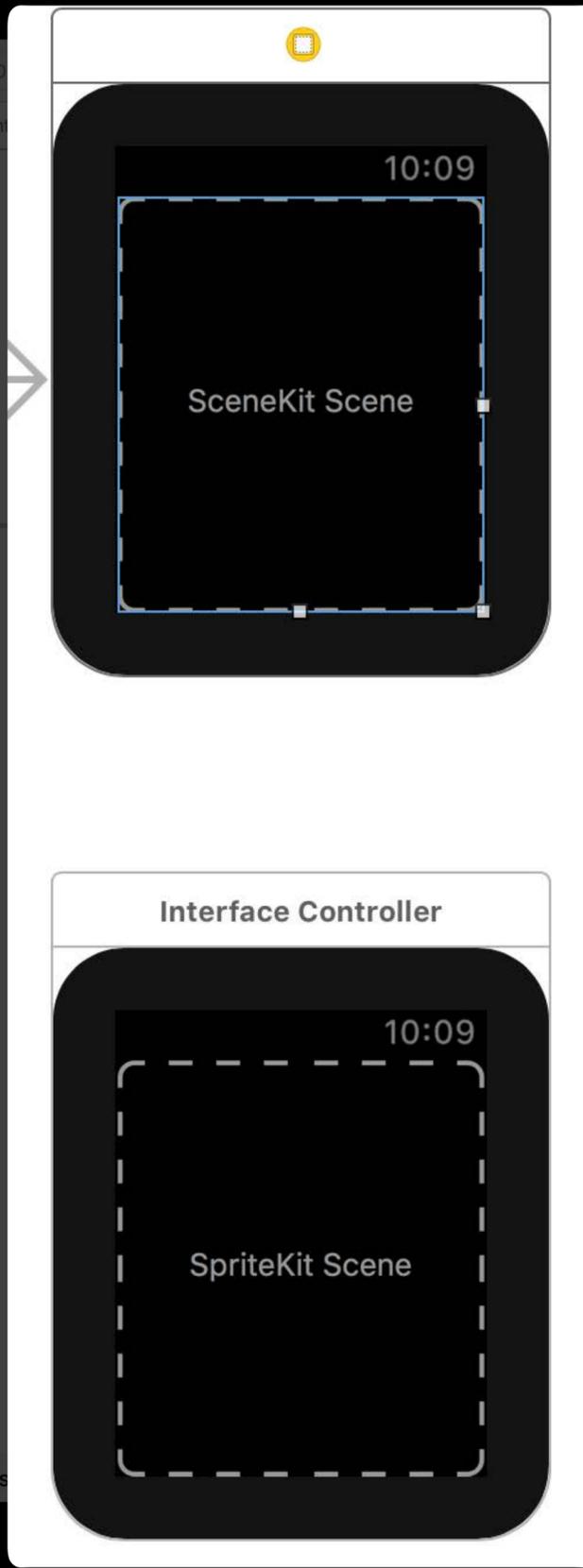
iPhone 6s + Apple Watch - 42mm | Scene Tests: Ready | Today at 10:09

Scene Tests

- Scene Tests
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Scene Tests WatchKit App
  - Interface.storyboard
  - Assets.xcassets
  - Info.plist
- Scene Tests WatchKit Extension
- Products

Interface Controller Scene

- Interface Controller
  - SceneKit Scene
  - Main Entry Point
- Interface Controller Scene
  - Interface Controller
    - SpriteKit Scene



SceneKit Scene

- Scene: Scene name
- Antialiasing: None
- Frame Rate: Automatic Per Second
- Rendering:  Default Lighting
- Jittering:
- Animations:  Play
- Loop:

View

- Alpha: 1
- Hidden:
- Installed:
- Semantic: Unspecified

Alignment

- Horizontal: Left
- Vertical: Top

Size

- Width: Size To Fit Content
- Height: Size To Fit Content

SceneKit Scene - Displays SceneKit content.

SpriteKit Scene - Displays SpriteKit content.

scene

iPhone 6s + Apple Watch - 42mm | Scene Tests: Ready | Today at 10:48 PM

Scene Tests > Sc...App > Int...oard > Int...ase) > Int...ene > Interface Controller > SceneKit Scene

**Scene Tests**

- Scene Tests
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Scene Tests WatchKit App
  - Interface.storyboard
  - Assets.xcassets
  - Info.plist
- Scene Tests WatchKit Extension
- Products

**Interface Controller Scene**

- Interface Controller
  - SceneKit Scene
  - Main Entry Point
- Interface Controller Scene
  - Interface Controller
    - SpriteKit Scene

SceneKit Scene

10:09

SceneKit Scene

Interface Controller

10:09

SpriteKit Scene

**SceneKit Scene**

- Scene: Scene name
- Antialiasing: None
- Frame Rate: Automatic Per Second
- Rendering:  Default Lighting
- Jittering
- Animations:  Play
- Loop

**View**

- Alpha: 1
- Hidden
- Installed
- Semantic: Unspecified

**Alignment**

- Horizontal: Left
- Vertical: Top

**Size**

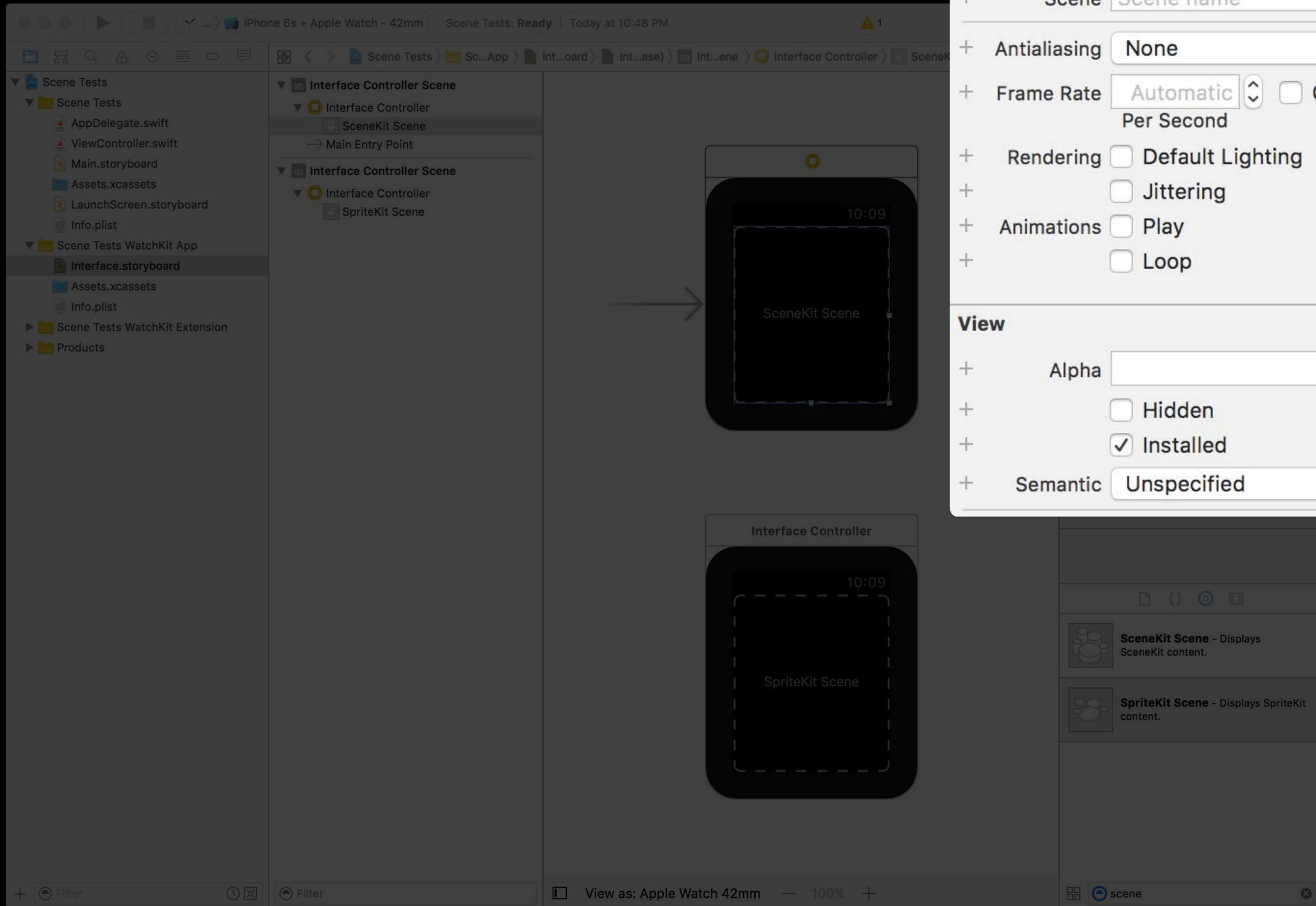
- Width: Size To Fit Content
- Height: Size To Fit Content

SceneKit Scene - Displays SceneKit content.

SpriteKit Scene - Displays SpriteKit content.

View as: Apple Watch 42mm | 100%

scene



### SceneKit Scene

+ Scene

+ Antialiasing

+ Frame Rate   Custom  
Per Second

+ Rendering  Default Lighting

+  Jittering

+ Animations  Play

+  Loop

---

### View

+ Alpha

+  Hidden

+  Installed

+ Semantic

# Displaying the Content

NEW

# Displaying the Content

NEW

Prepare the scene with SpriteKit or SceneKit APIs

# Displaying the Content

NEW

Prepare the scene with SpriteKit or SceneKit APIs

Present a SpriteKit scene with function on WKInterfaceSKScene

```
func present(_ : SKScene?)  
func present(_ : SKScene?, transition : SKTransition)
```

# Displaying the Content

NEW

Prepare the scene with SpriteKit or SceneKit APIs

Present a SpriteKit scene with function on WKInterfaceSKScene

```
func present(_ : SKScene?)  
func present(_ : SKScene?, transition : SKTransition)
```

Set the SceneKit scene property on WKInterfaceSCNScene

```
var scene : SCNScene
```

# Your watchOS 3 API Toolkit

Glanceable, Actionable, Responsive

Gesture recognizers

Digital Crown rotation

Improved table navigation

New Notifications Framework

SceneKit and SpriteKit integration

# Quick Interaction Techniques

Tom Witkin watchOS Engineer

# Quick Interaction Techniques



# Quick Interaction Techniques

Actionable complications



# Quick Interaction Techniques

Actionable complications

Glanceable notifications



# Quick Interaction Techniques

Actionable complications

Glanceable notifications

Simple navigation



# Quick Interaction Techniques

Actionable complications

Glanceable notifications

Simple navigation

Drop wrist



# Actionable Complications







10:09

CAL 240/400  
MIN 13/30  
HOUR 3/12



68°

# Why Complications



# Why Complications

Highly glanceable



# Why Complications

Highly glanceable

Frequent updates



# Why Complications

Highly glanceable

Frequent updates

Keeps app in memory



# Why Complications

Highly glanceable

Frequent updates

Keeps app in memory

Accelerates user into your app







9:41 AM



# Face Gallery

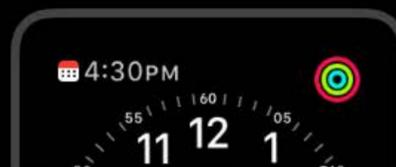
## Mickey Mouse and Minnie Mouse



## Numerals



## Utility



My Watch



Face Gallery



App Store



Search



9:41 AM



< My Watch

Modular

10:09

CAL 500/500  
MIN 30/30  
HOUR 12/12

72°

### Complications

Top Left

Workout

Middle

Activity

Bottom Left

Heartrate

Bottom Middle

Activity

Bottom Right

Weather

Set as current Watch Face



My Watch



Face Gallery



App Store



Search



9:41 AM



< My Watch

Modular



10:09

CAL 500/500

MIN 30/30

HOUR 12/12



72°

### Complications

Top Left

Workout

Timer

Weather

Weather Conditions

Workout

World Clock: Cupertino

World Clock: New York

Middle

Activity



My Watch



Face Gallery



App Store



Search

More Information

“Adding Complications to the Gallery”

[developer.apple.com/library/prerelease/content](https://developer.apple.com/library/prerelease/content)



# Glanceable Notifications

# Why Notifications



# Why Notifications

Customizable



# Why Notifications

Customizable

Displays valuable information



# Why Notifications

Customizable

Displays valuable information

Removes the necessity to launch your app



# Why Notifications

Customizable

Displays valuable information

Removes the necessity to launch your app

Now supports:

- SpriteKit
- SceneKit
- Inline video





SpriteKit



SceneKit



SpriteKit



SceneKit



SpriteKit



SceneKit



10:09

CAL 240/400  
MIN 13/30  
HOUR 3/12



68°



10:09

CAL 240/400  
MIN 13/30  
HOUR 3/12



68°



10:09

MYPHOTOS

Ivy Calhoun has  
shared 10 new  
photos.

View

Dismiss

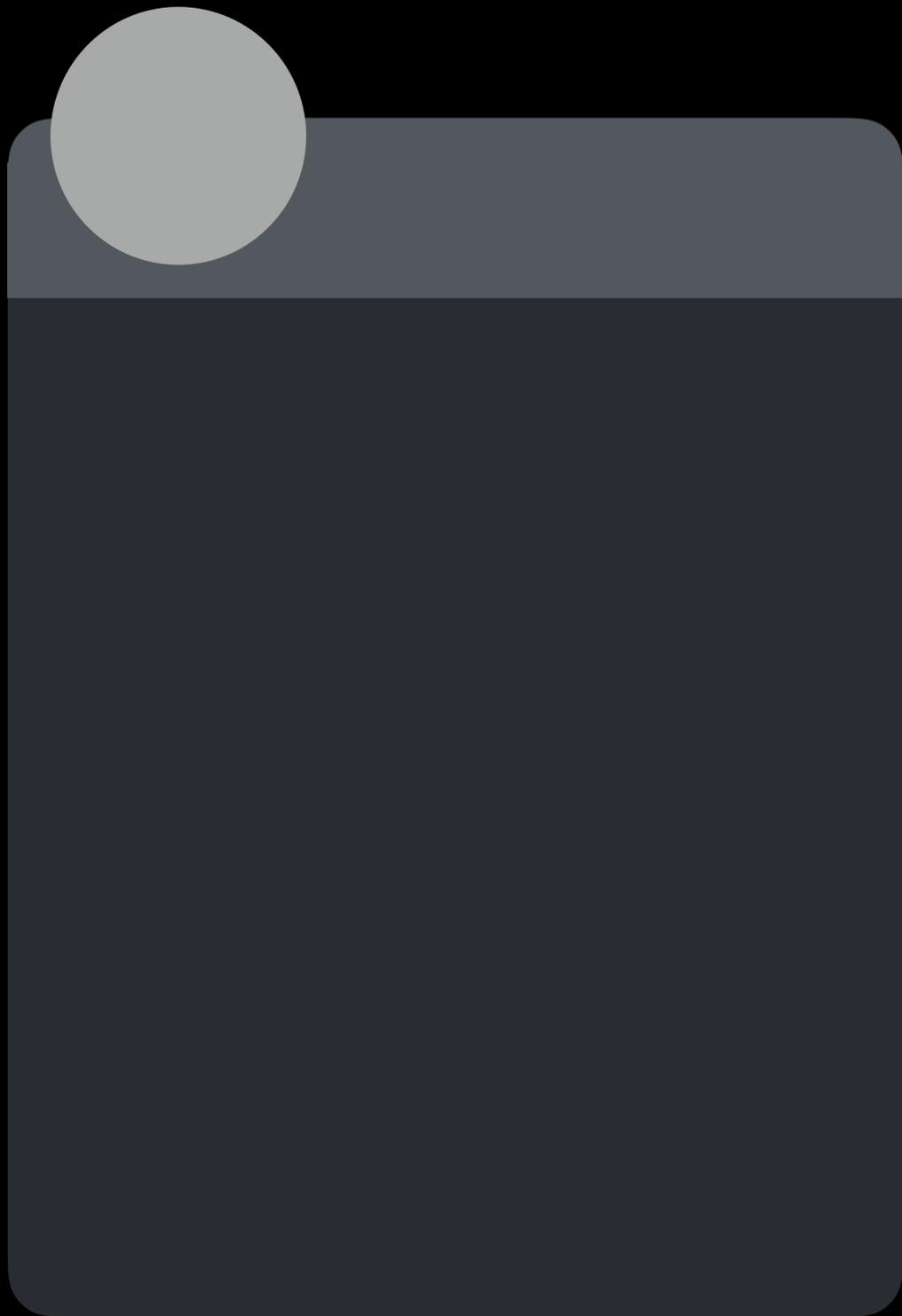


10:09

MYPHOTOS



Ivy Calhoun has shared 10 new





WKInterfaceSKScene



WKInterfaceSKScene

WKInterfaceLabel



10:09

MYPHOTOS

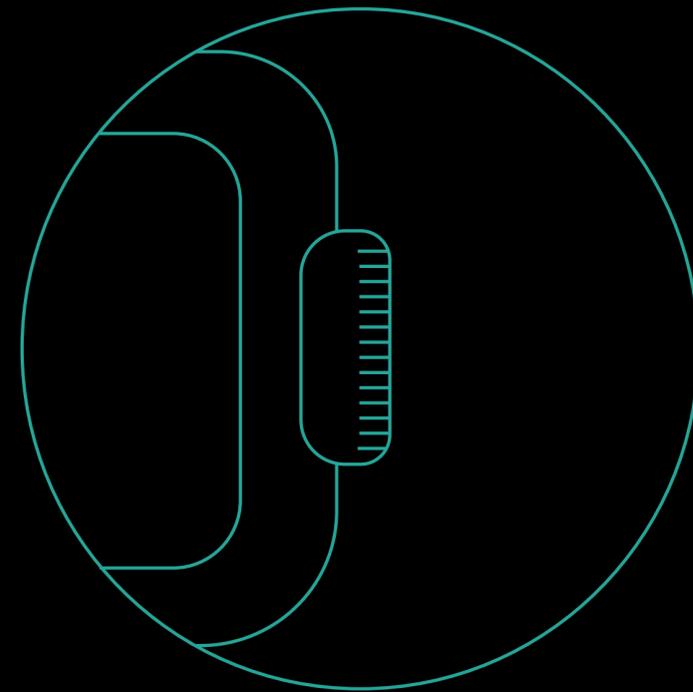


Ivy Calhoun has shared 10 new

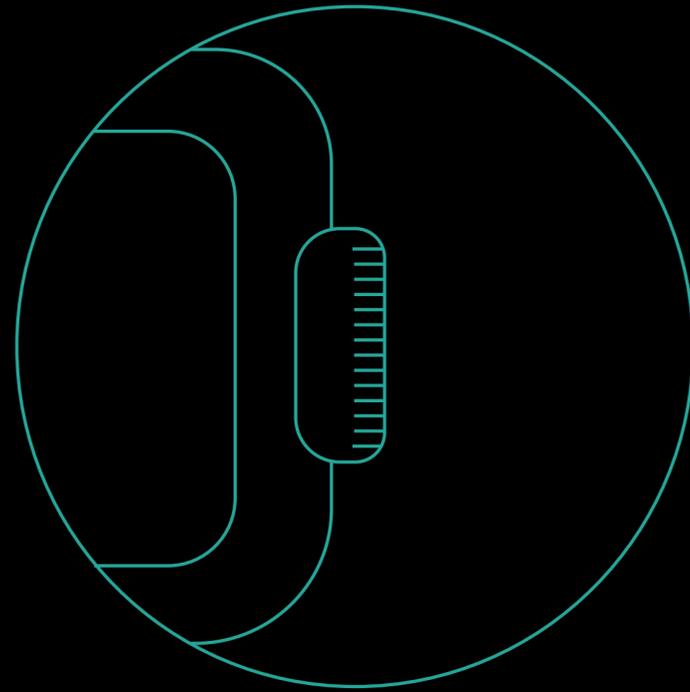
# Simple Navigation



Gestures



Digital Crown



Digital Crown









**MyRun** 10:09

DIST 1 mile  
PACE 7'50"  
ALT 108 feet

DIST 2 miles  
PACE 7'52"  
ALT 152 feet

DIST 2 miles



MyRun 10:09

ALT 32 feet

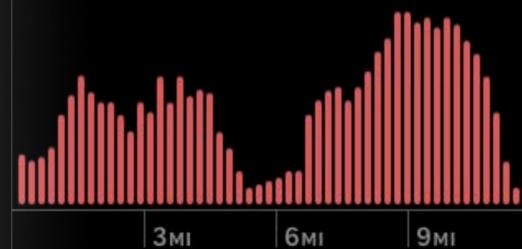
DIST 7 miles  
PACE 8'10"  
ALT 31 feet

DIST 8 miles  
PACE 8'14"  
ALT 133 feet



**MyRun**  
June 15, 2016

10:09

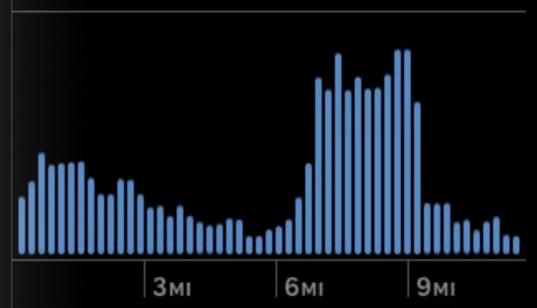


MAX ALTITUDE  
**294FT**



MyRun  
June 15, 2016

10:09

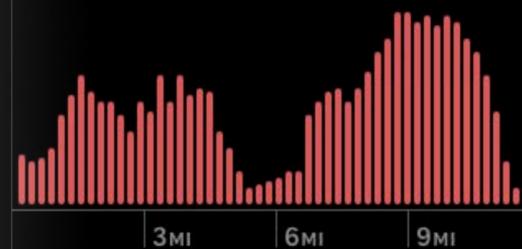


AVG. PACE  
7'54" MI



MyRun  
June 15, 2016

10:09

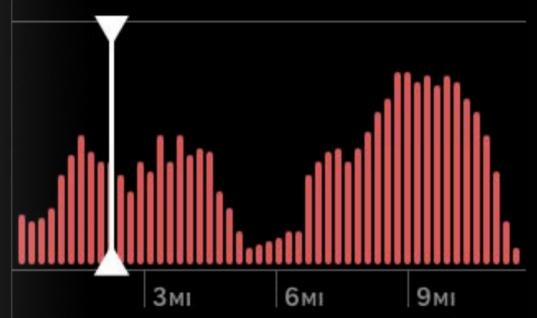


MAX ALTITUDE  
294FT



MyRun  
June 15, 2016

10:09

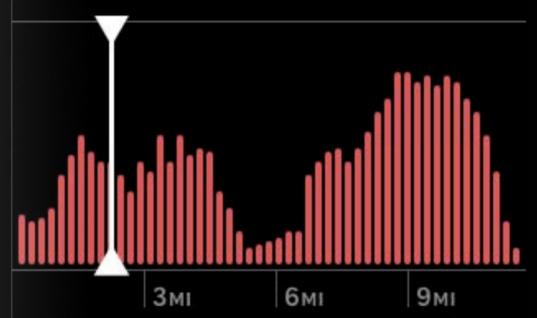


ALTITUDE  
142FT



MyRun  
June 15, 2016

10:09



ALTITUDE  
142FT

Drop Wrist







---

Keeping Your Watch App Up to Date

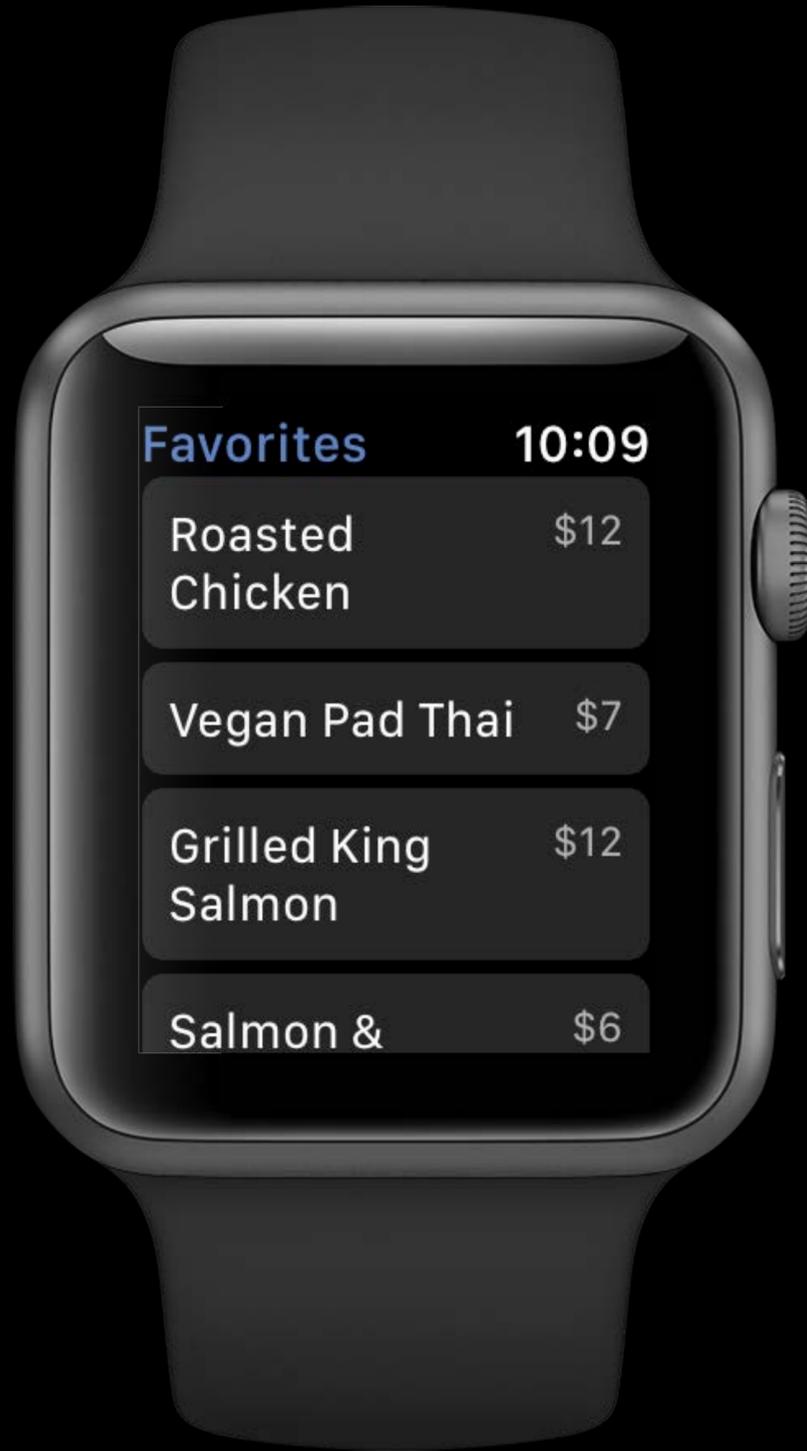
Mission

Thursday 9:00AM

---

Responding to User Action

Instant Feedback



**Favorites** 10:09

Roasted Chicken \$12

Vegan Pad Thai \$7

Grilled King Salmon \$12

Salmon & \$6



< Purchase

10:09

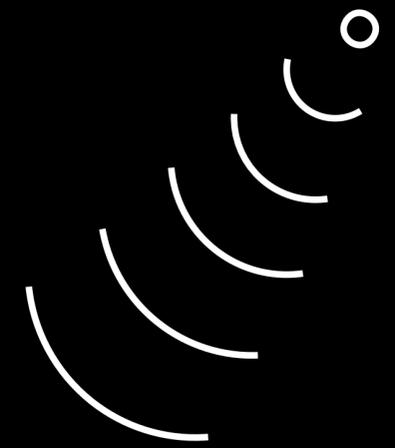
Roasted  
Chicken

\$12

Buy









<

10:09



Your order is on  
its way!



Choose Item





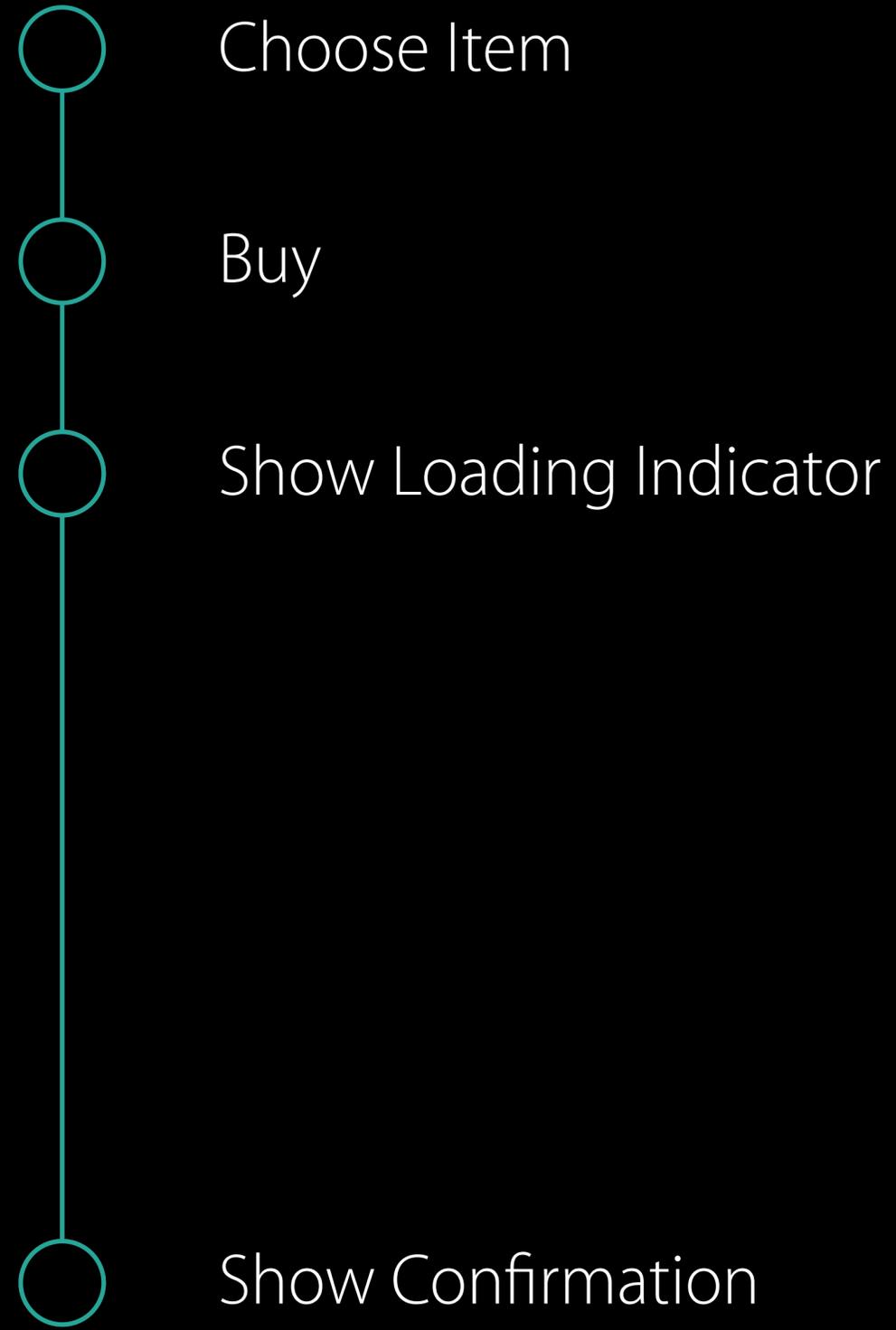
Choose Item



Buy



Show Loading Indicator







< Purchase

10:09

Roasted  
Chicken

\$12

Buy







10:09

CAL 240/400  
MIN 13/30  
HOUR 3/12



68°



10:09

CAL 240/400  
MIN 13/30  
HOUR 3/12



68°

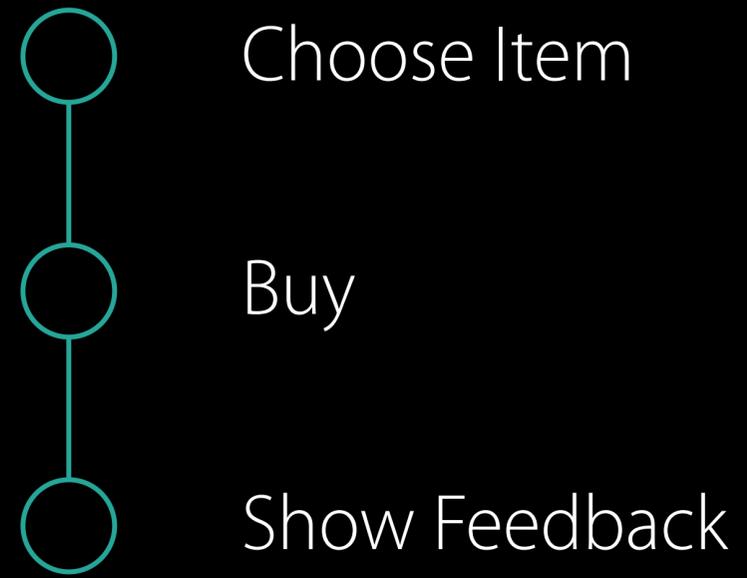


Choose Item



Choose Item

Buy



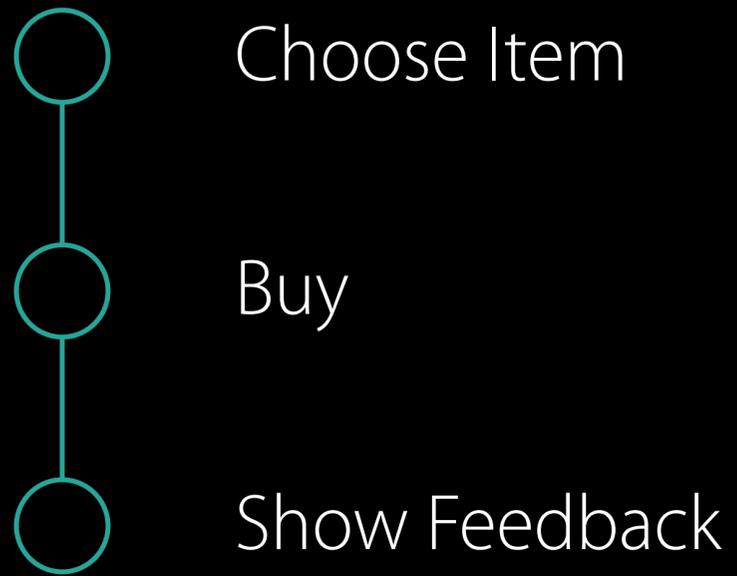


Choose Item

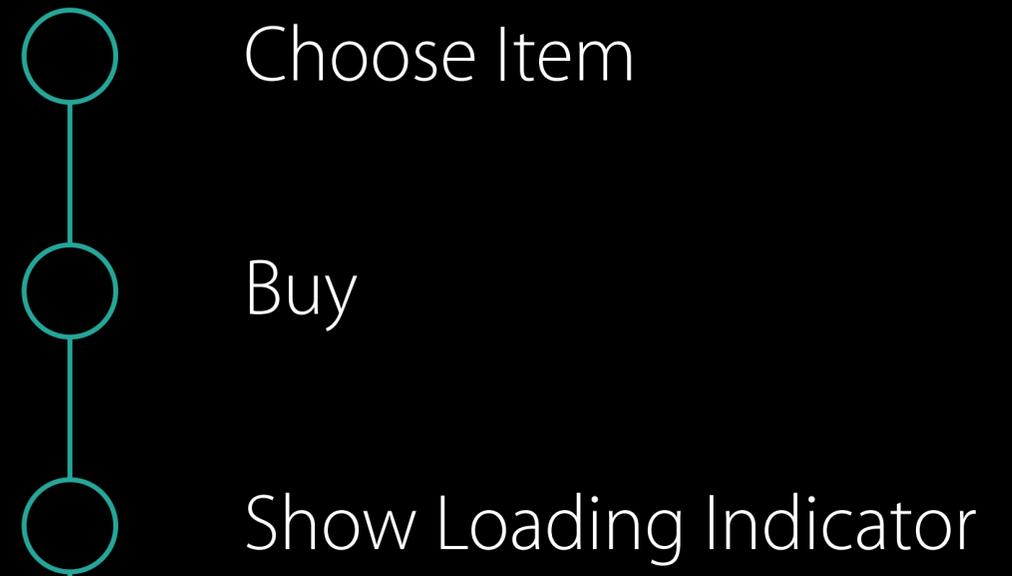
Buy

Show Feedback

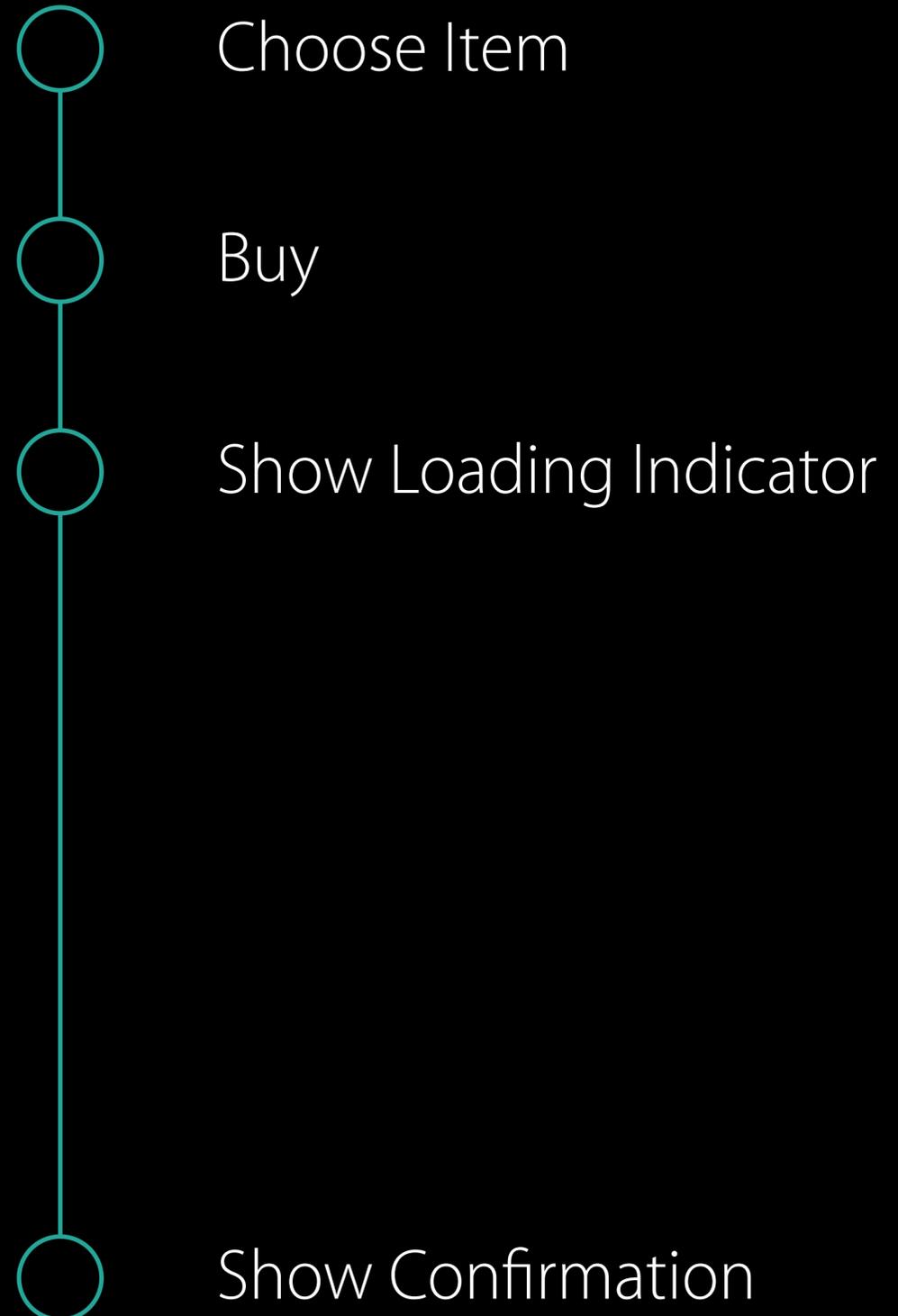
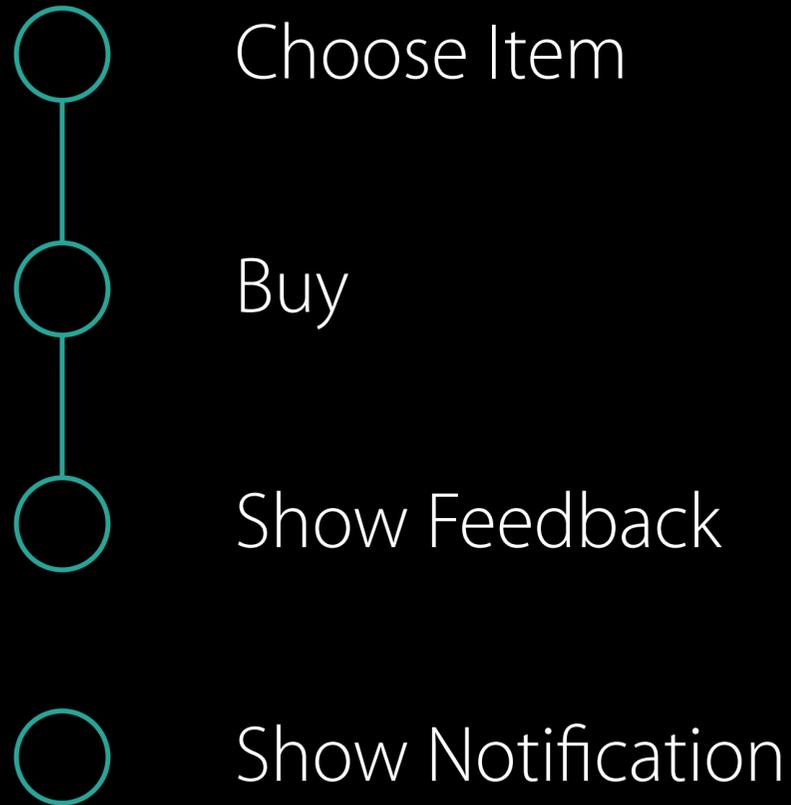
Show Notification



Show Notification



Show Confirmation



# Processing After Dropping Wrist

# Processing After Dropping Wrist

Background URLSession

# Processing After Dropping Wrist

Background NSURLSession

Background task assertion

# Processing After Dropping Wrist

Background NSURLSession

Background task assertion



2 seconds



Focus on quick interactions



Focus on quick interactions

Use complications and notifications



Focus on quick interactions

Use complications and notifications

Create glanceable UIs with SceneKit / SpriteKit



Focus on quick interactions

Use complications and notifications

Create glanceable UIs with SceneKit / SpriteKit

Simplify navigation



Focus on quick interactions

Use complications and notifications

Create glanceable UIs with SceneKit / SpriteKit

Simplify navigation

Don't make the user wait



More Information

<https://developer.apple.com/wwdc16/211>

# Related Sessions

---

What's New in watchOS 3

WWDC 2016 Videos

---

Introduction to Notifications

WWDC 2016 Videos

---

Designing Great Apple Watch Experiences

Presidio

Wednesday 1:40PM

---

Keeping Your Watch App Up to Date

Mission

Thursday 9:00AM

---

Architecting for Performance on watchOS 3

Mission

Thursday 3:00PM

---

# Labs

---

WatchKit and ClockKit Lab

Frameworks Lab B

Wednesday 12:00PM

---

WatchKit and Background Tasks Lab

Frameworks Lab C

Thursday 10:30AM

---

WatchKit and WatchConnectivity Lab

Frameworks Lab B

Friday 2:00PM

---



W

W

D

C

1

6