



apple WATCH APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 5

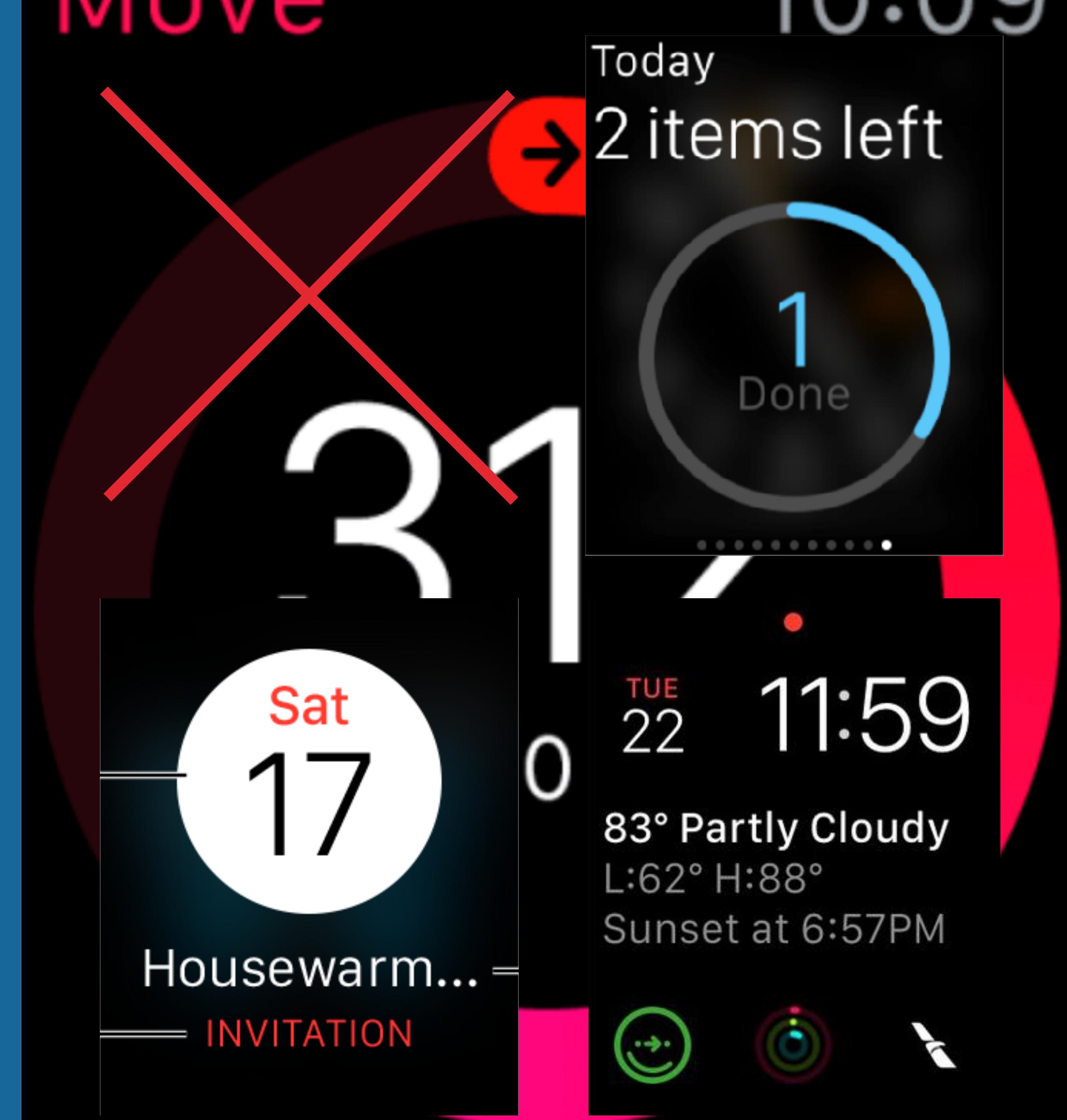
USER INTERACTIONS ON



USER INTERACTIONS

SUBTITLE

- User interactions available
 - WatchKit App
 - Glances
 - Dock
 - Custom notifications
 - Complication (new in 2.0)



USER INTERACTIONS

- What makes up a Watch App?
 - Interface
 - Notification Interface
 - Complication Interface
 - Code for managing all interfaces in extension

The screenshot shows the Xcode interface with a project titled "InterFaces" selected. The project structure is as follows:

- InterFaces (Target)
 - InterFaces (Group)
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - InterFaces WatchKit App (Group)
 - Interface.storyboard
 - Assets.xcassets
 - Info.plist
 - InterFaces WatchKit Extension (Group)
 - InterfaceController.swift
 - ExtensionDelegate.swift
 - NotificationController.swift
 - GlanceController.swift
 - ComplicationController.swift
 - Assets.xcassets
 - Info.plist
 - Supporting Files
 - Products

The code editor on the right displays the content of the `InterfaceController.swift` file:

```
// InterfaceController.swift
// InterFaces WatchKit Extension
// Created by T. Andrew Bi... 2016 The University of Texas at Austin
// Copyright © 2016 The University of Texas at Austin. All rights reserved.

import WatchKit
import Foundation
import WatchConnectivity

class InterfaceController: WKInterfaceController {

    // MARK: - IBOutlets
    @IBOutlet var nameLabel: WKInterfaceLabel

    // MARK: - Connectivity session
    let session = WCSession.default()

    override func awake(withContext context: Any?) {
        super.awake(withContext: context)
        // Set the name label
        nameLabel.setText("Hello, World!")
        // Check if the session is connected
        if session.isConnected {
            session.sendMessage(["text": "Hello, World!"], replyHandler: { [weak self] (reply) in
                self?.nameLabel.setText(reply["text"] as? String)
            }, errorHandler: { [weak self] (error) in
                print("Error: \(error.localizedDescription)")
            })
        }
    }

    @IBAction func tapToInteractive(_ sender: WKInterfaceButton) {
        print("Tap (interactive)")
        let applicationDict = [
            "text": "Hello, World!"
        ]
        session.sendMessage(applicationDict, replyHandler: { [weak self] (reply) in
            self?.nameLabel.setText(reply["text"] as? String)
        }, errorHandler: { [weak self] (error) in
            print("Error: \(error.localizedDescription)")
        })
    }

    @IBAction func tapToApplication(_ sender: WKInterfaceButton) {
        print("Tap (application)")
        do {
            let applicationDict = [
                "text": "Hello, World!"
            ]
            try session.default().sendMessage(applicationDict)
        } catch {
            // Handle errors here
            print(error)
        }
    }

    // MARK: - Lifecycle
    override func awake(withContext context: Any?) {
        super.awake(withContext: context)
        // Set the name label
        nameLabel.setText("Hello, World!")
        // Check if the session is connected
        if session.isConnected {
            session.sendMessage(["text": "Hello, World!"], replyHandler: { [weak self] (reply) in
                self?.nameLabel.setText(reply["text"] as? String)
            }, errorHandler: { [weak self] (error) in
                print("Error: \(error.localizedDescription)")
            })
        }
    }
}
```

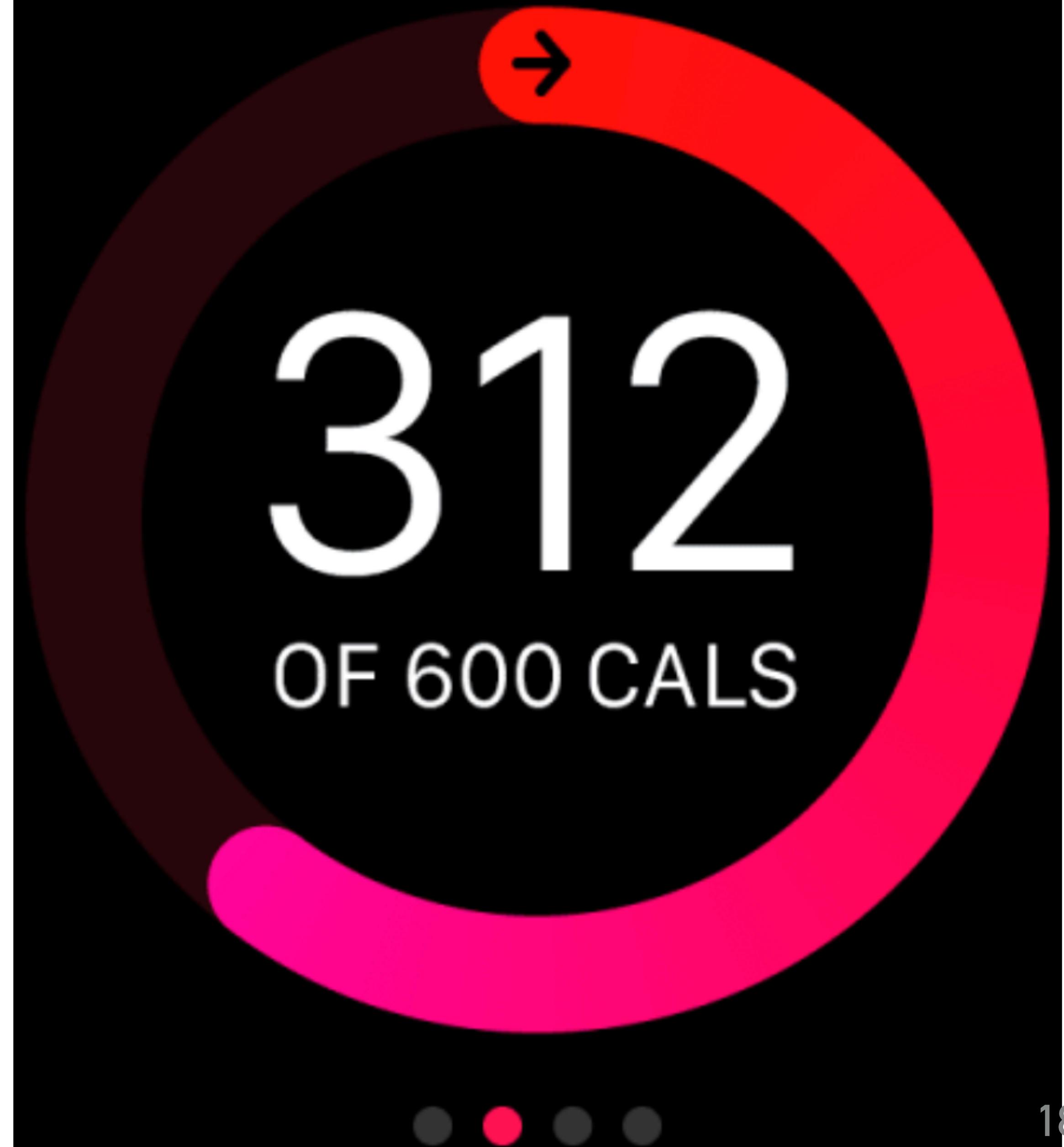
USER INTERACTIONS

SUBTITLE

- WatchKit App
 - Full-app experience which they interact with by opening your app from the Home screen
 - Main way to interact with data
 - Typically, users will only interact with a subset of data/features

Move

10:09



WATCH APP

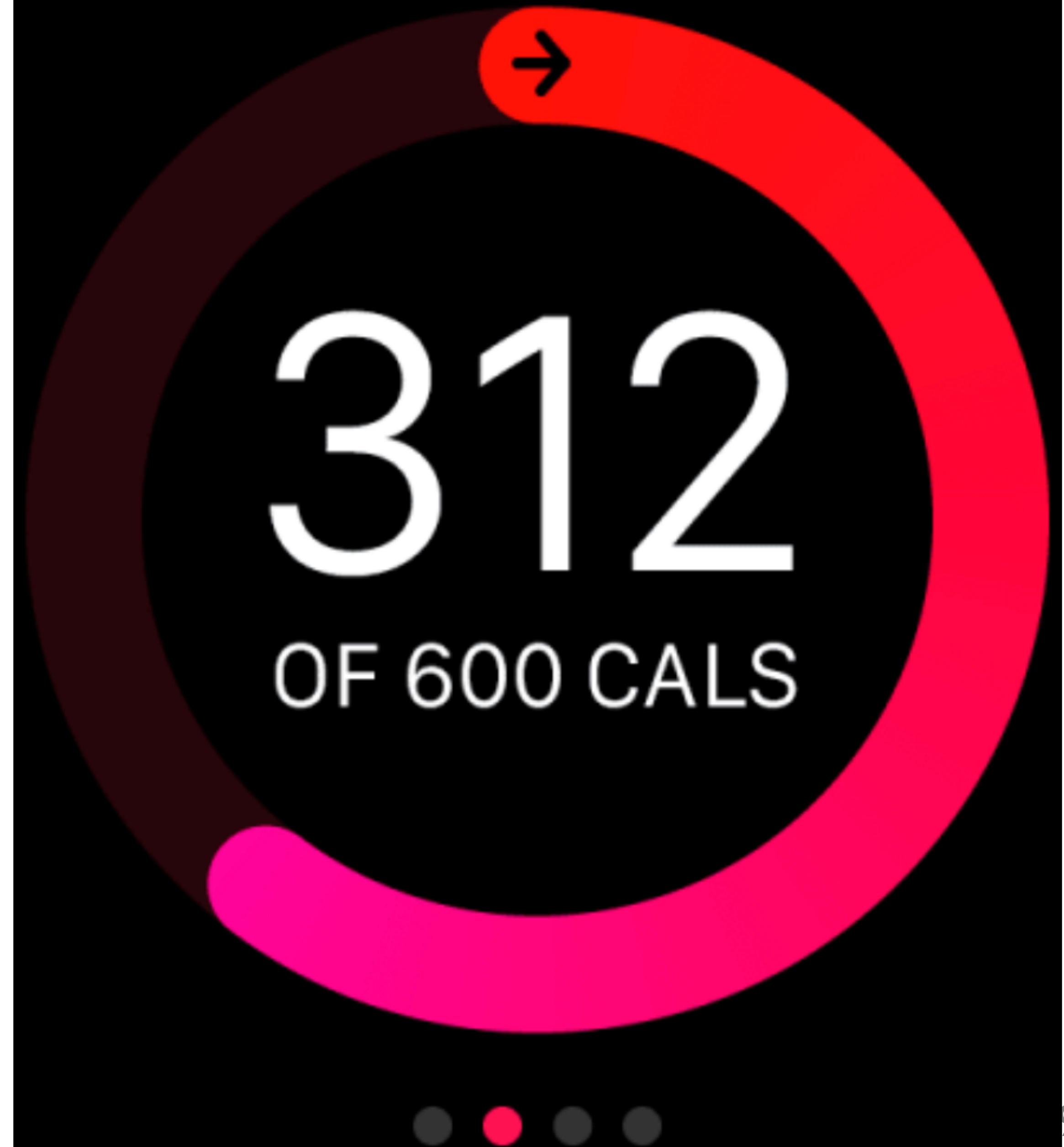
USER INTERACTIONS

SUBTITLE

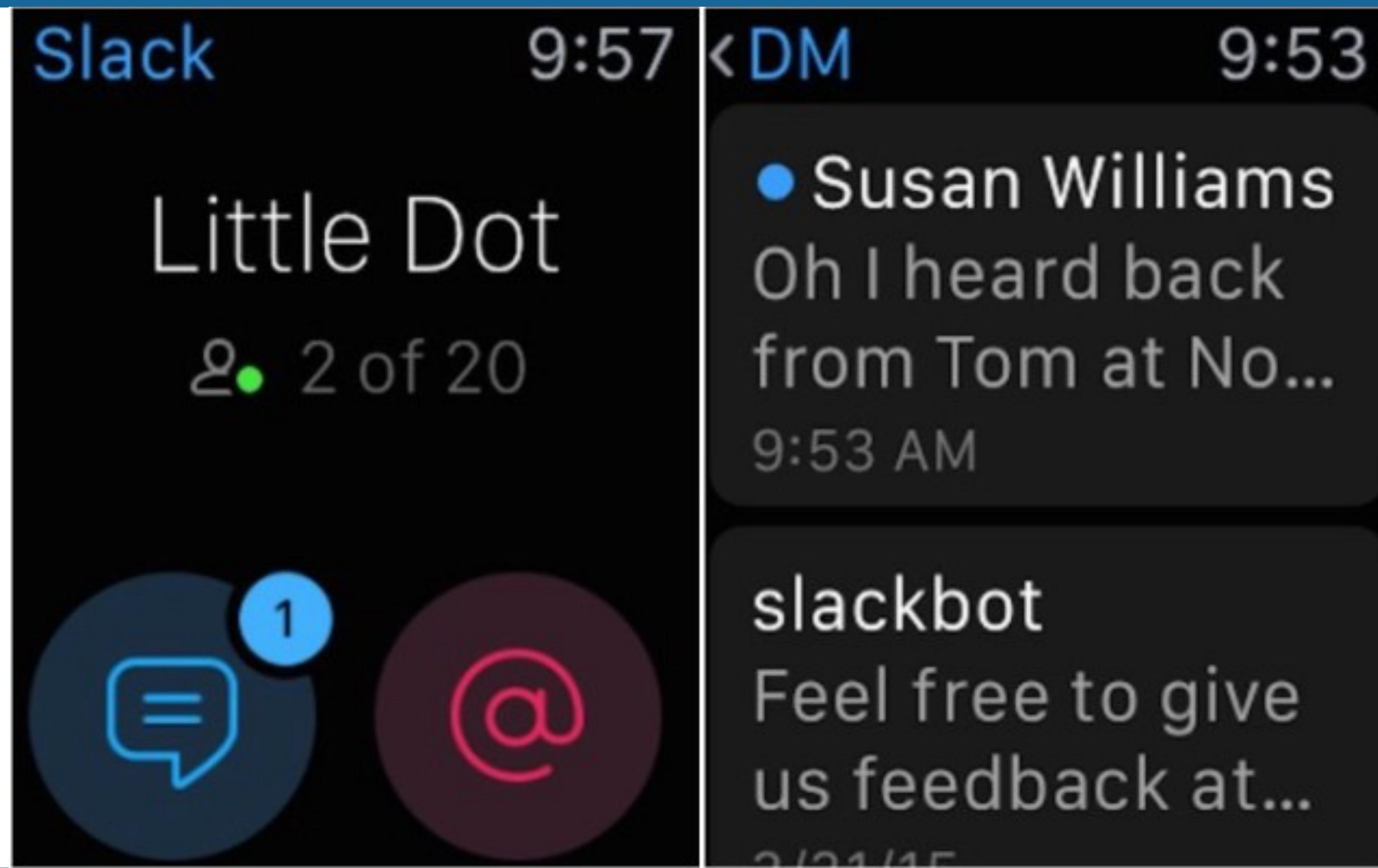
- Watch apps have different navigation and interaction models
 - Health app
 - Music app
 - Game app
 - App app

Move

10:09



USER INTERACTIONS



USER INTERACTIONS



USER INTERACTIONS



USER INTERACTIONS



USER INTERACTIONS

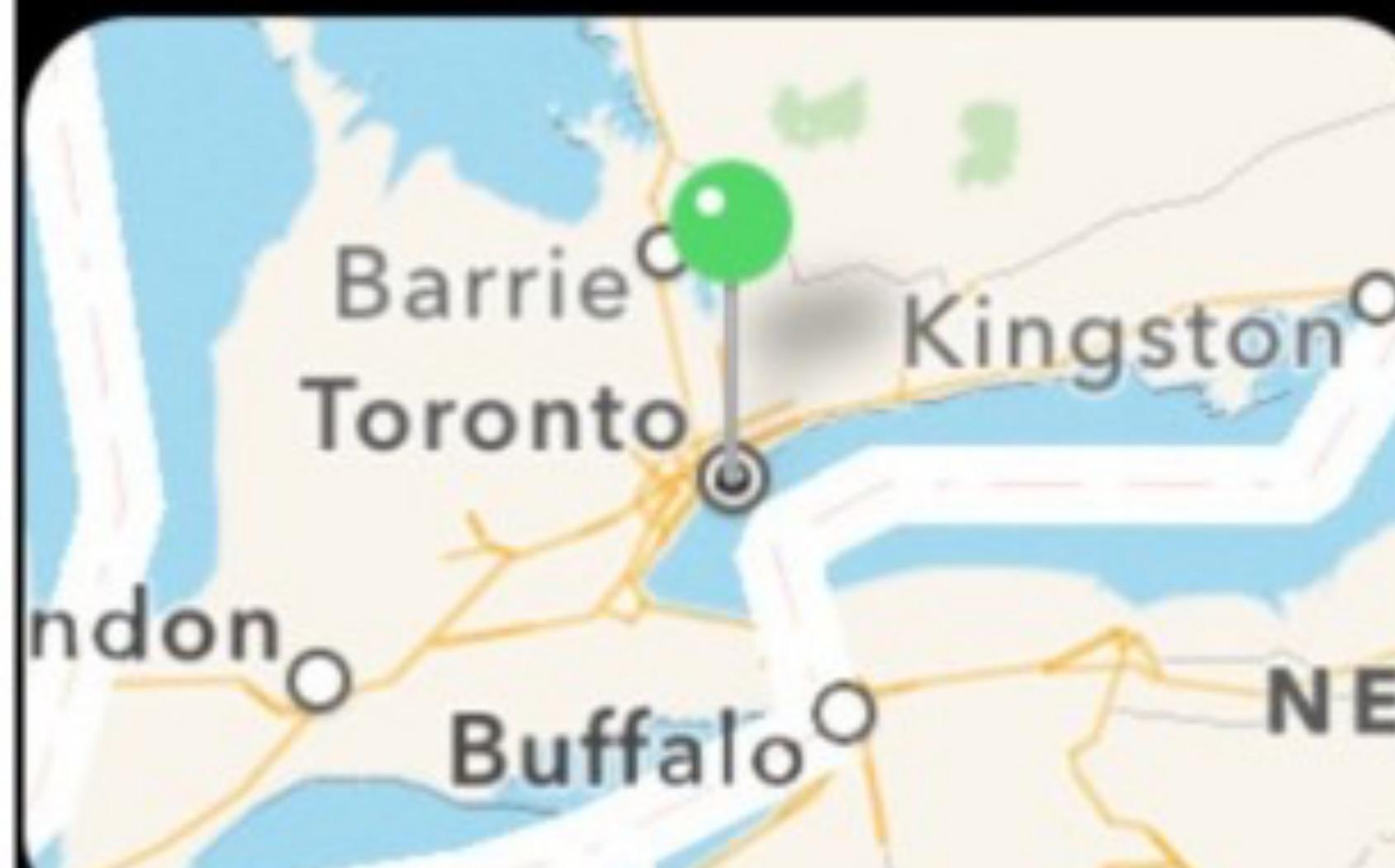
Deliveries 10:09

 Sailor Moon S...
Delivered: Nort...

3 DAYS Seconds and...
In transit: Toro...

8 DAYS 2 11-inch Ma...
Ship date unkn...
Updated a moment ago

< Seconds an... 10:09
3 days
In transit: Toronto, ON
Delivered by Saturday



A map of the Great Lakes region showing the locations of Barrie, Kingston, Toronto, London, and Buffalo. A green dot marks the delivery location in Toronto, ON.

USER INTERACTIONS



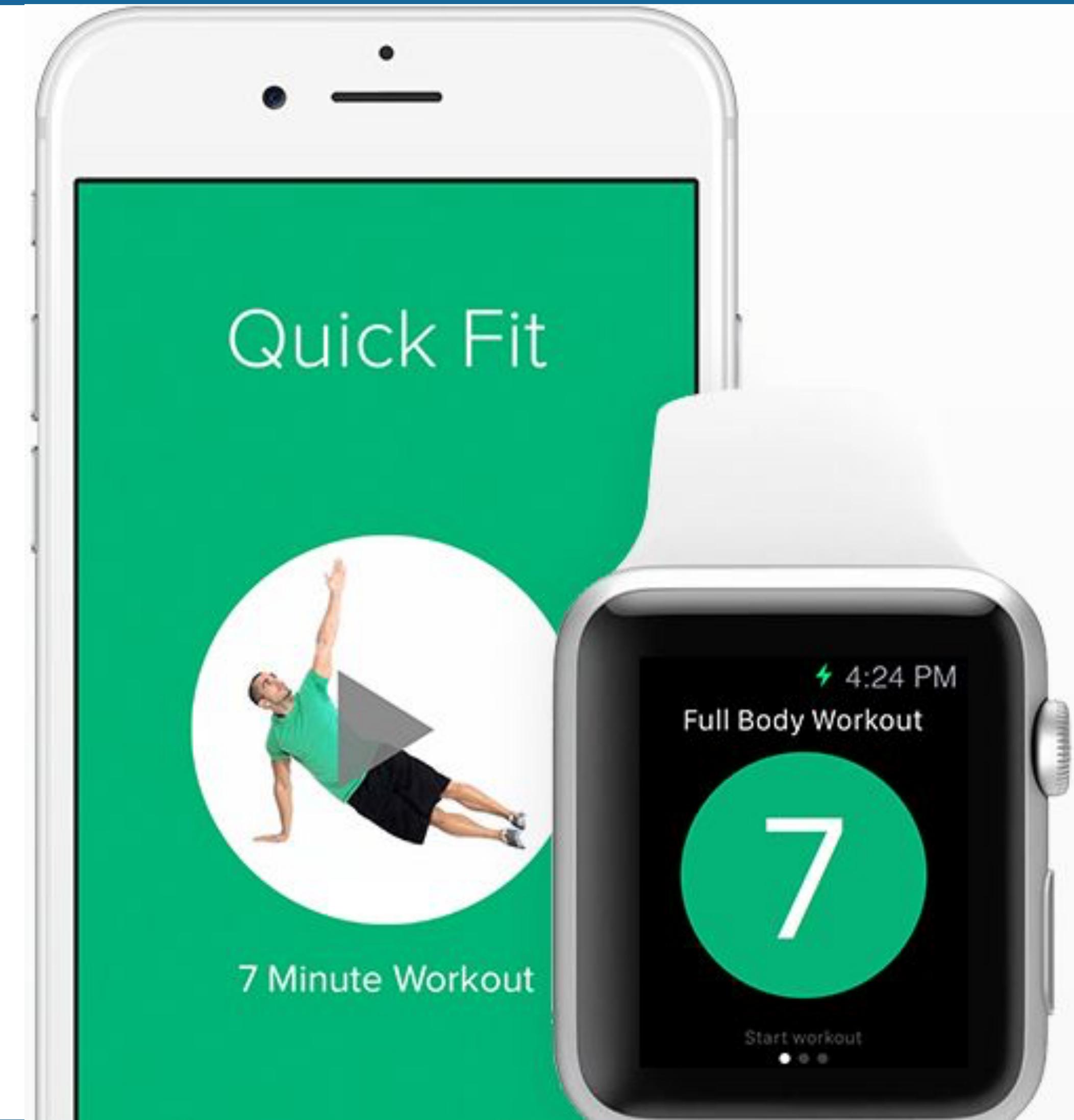
USER INTERACTIONS



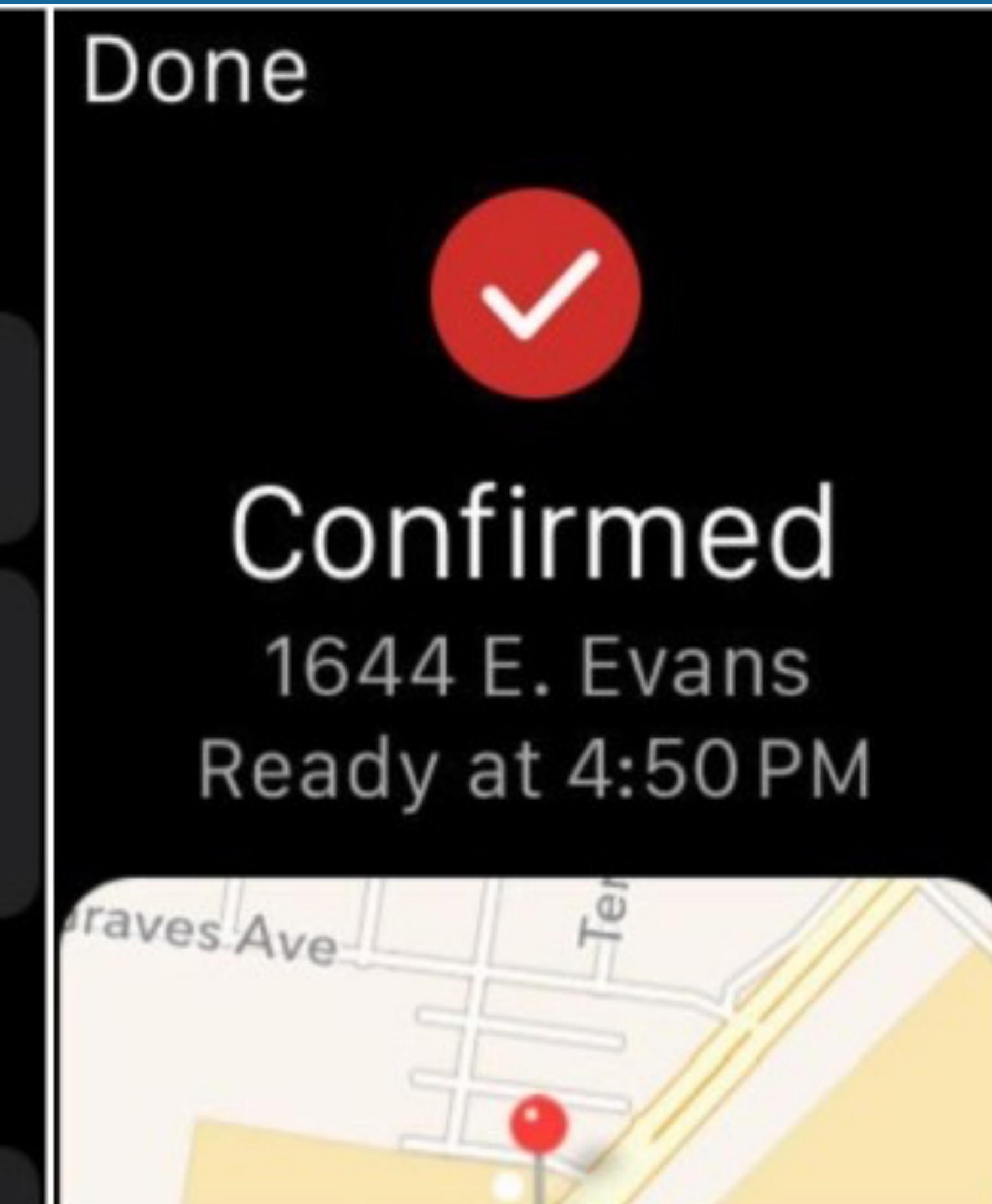
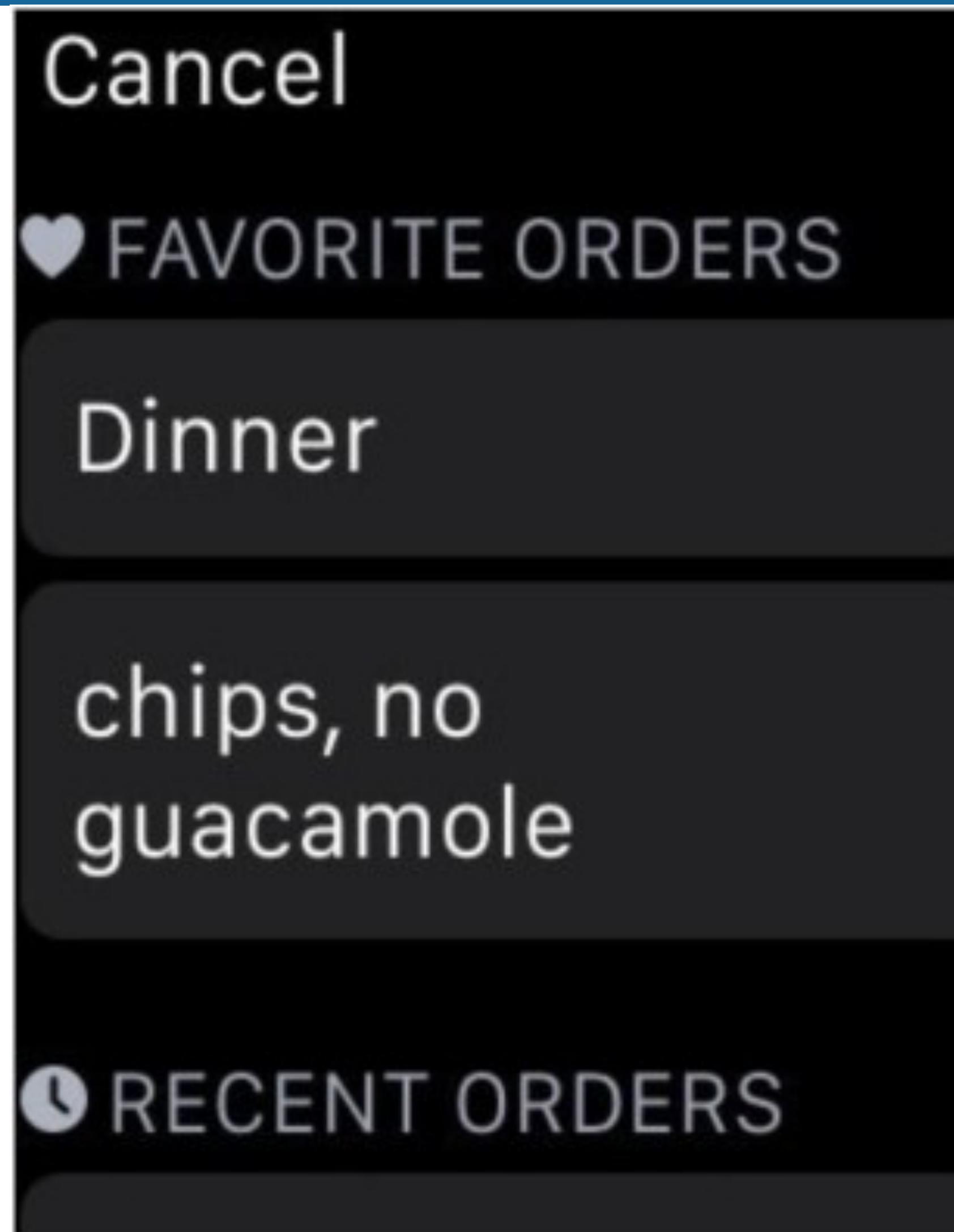
USER INTERACTIONS



USER INTERACTIONS



USER INTERACTIONS



GLANCES

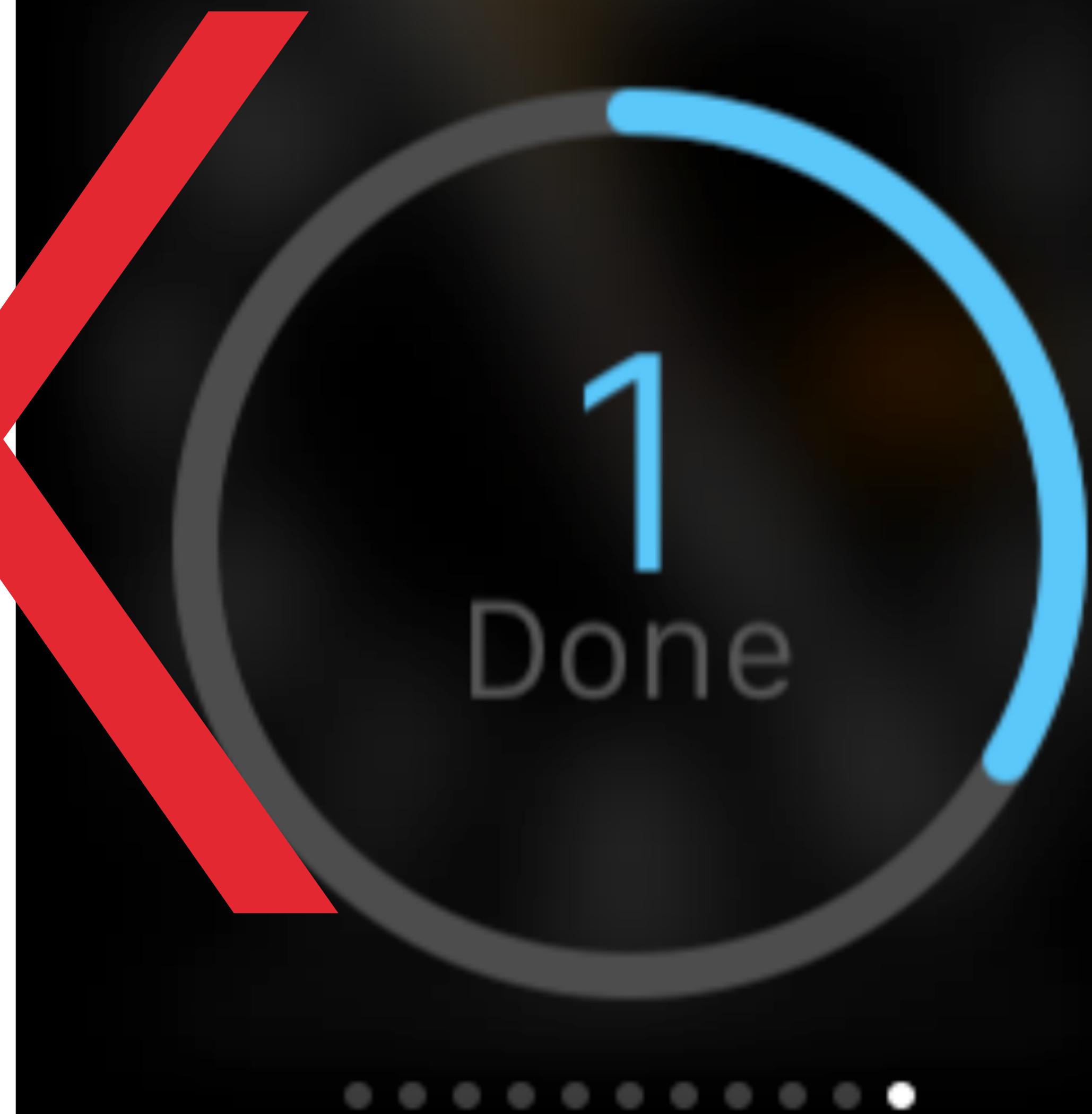


USER INTERACTIONS

SUBTITLE

- Glances
 - A focused interface that you use to display your app's most important information
 - Glances are nonscrolling and read-only
 - Cannot contain buttons, switches, or other interactive controls
 - Tapping a glance launches your WatchKit app
 - “Not to be used as an app launcher”

Today
2 items left



USER INTERACTIONS

GLANCES HAS BEEN REPLACED BY THE DOCK



NOTIFICATIONS

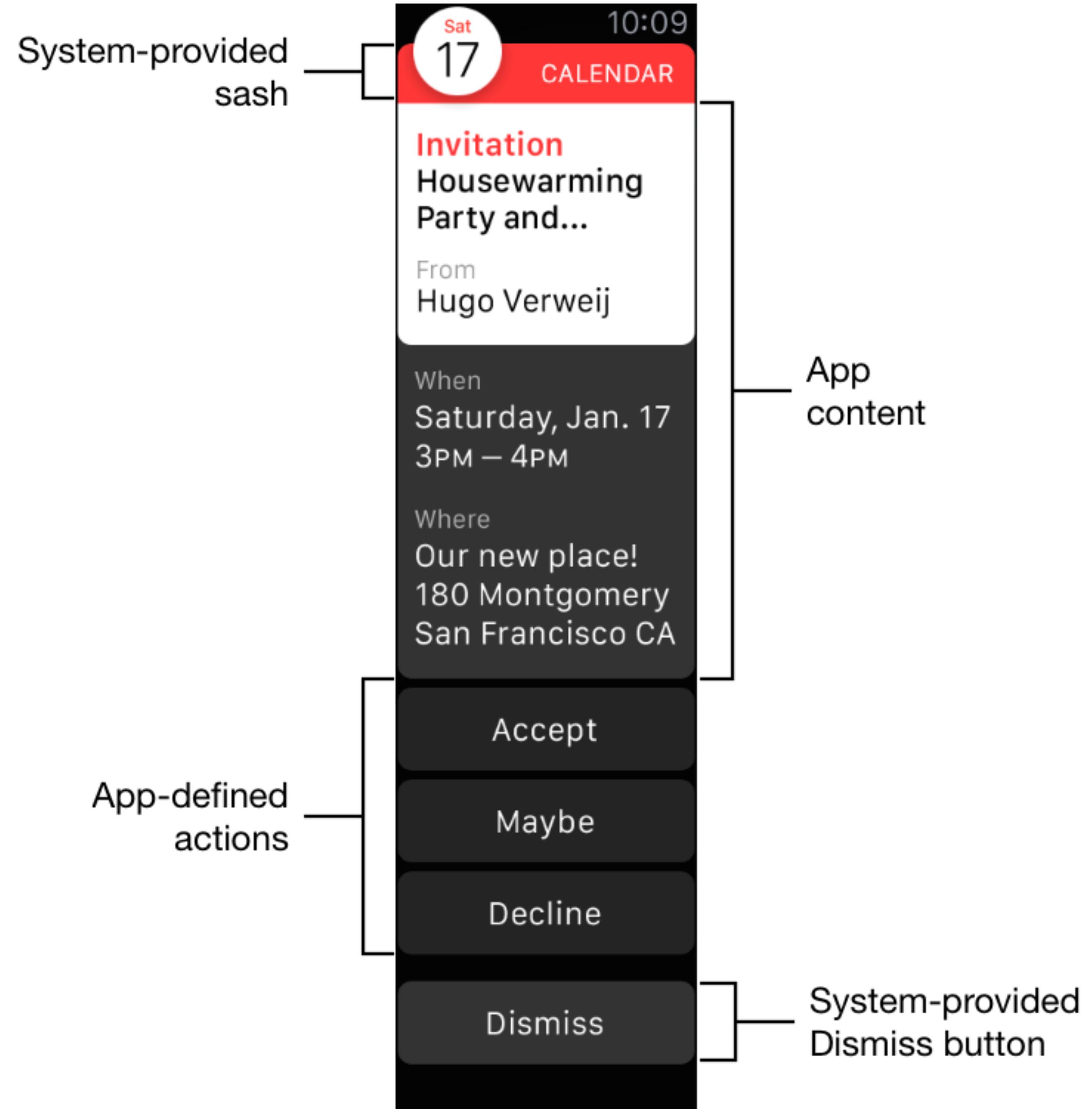
USER INTERACTIONS

- Custom Notifications
 - Works with iPhone to display local and remote notifications
 - Providing custom notification interfaces
 - Short
 - Long



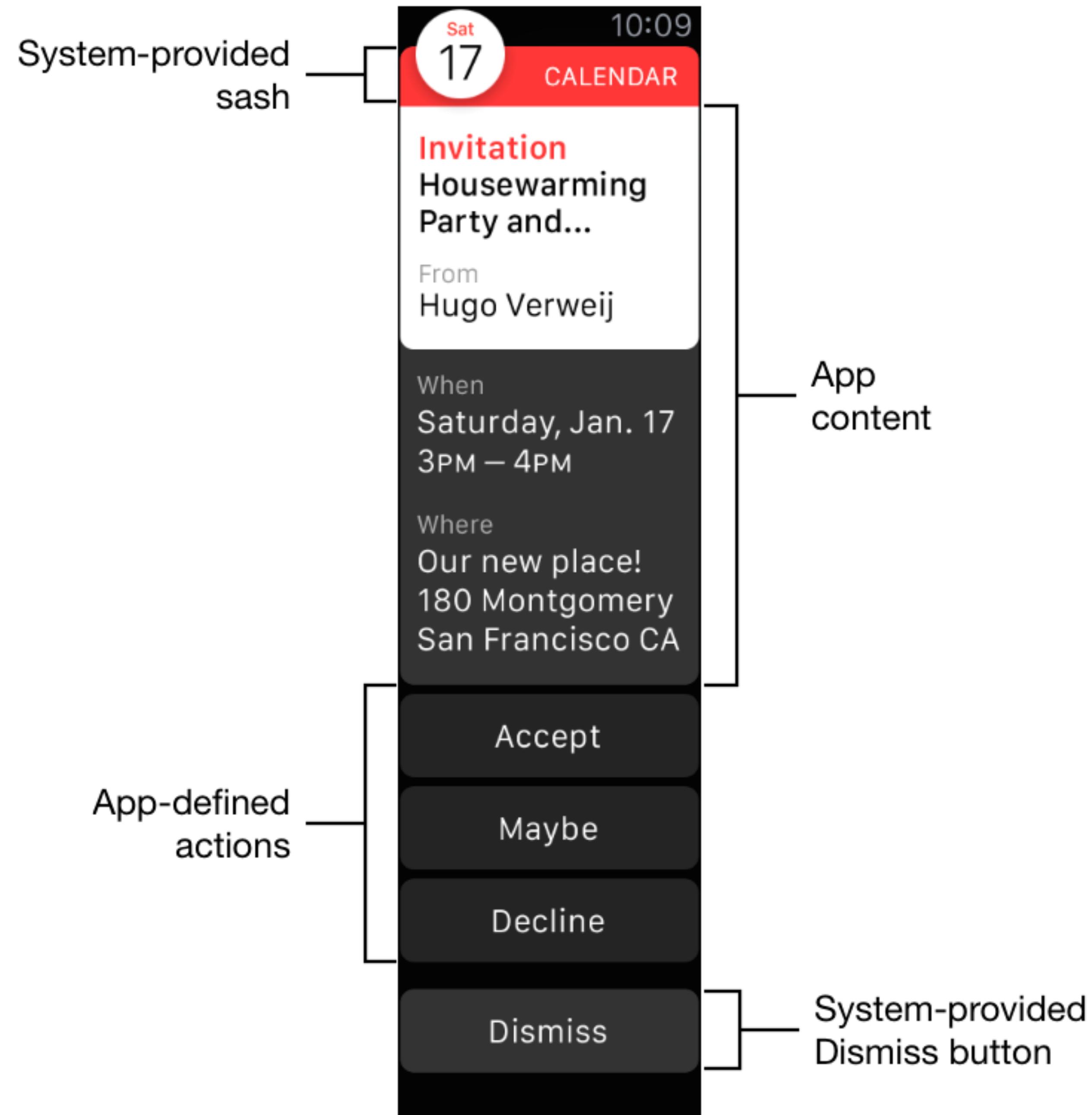
USER INTERACTIONS

- Long looks notifications
- Actionable notifications to increase functionality



USER INTERACTIONS

- watchOS works with UNNotification framework
- Schedule and handle location notification on watch using extension
 - Time, location based
- Handle remote notifications delivered to watch

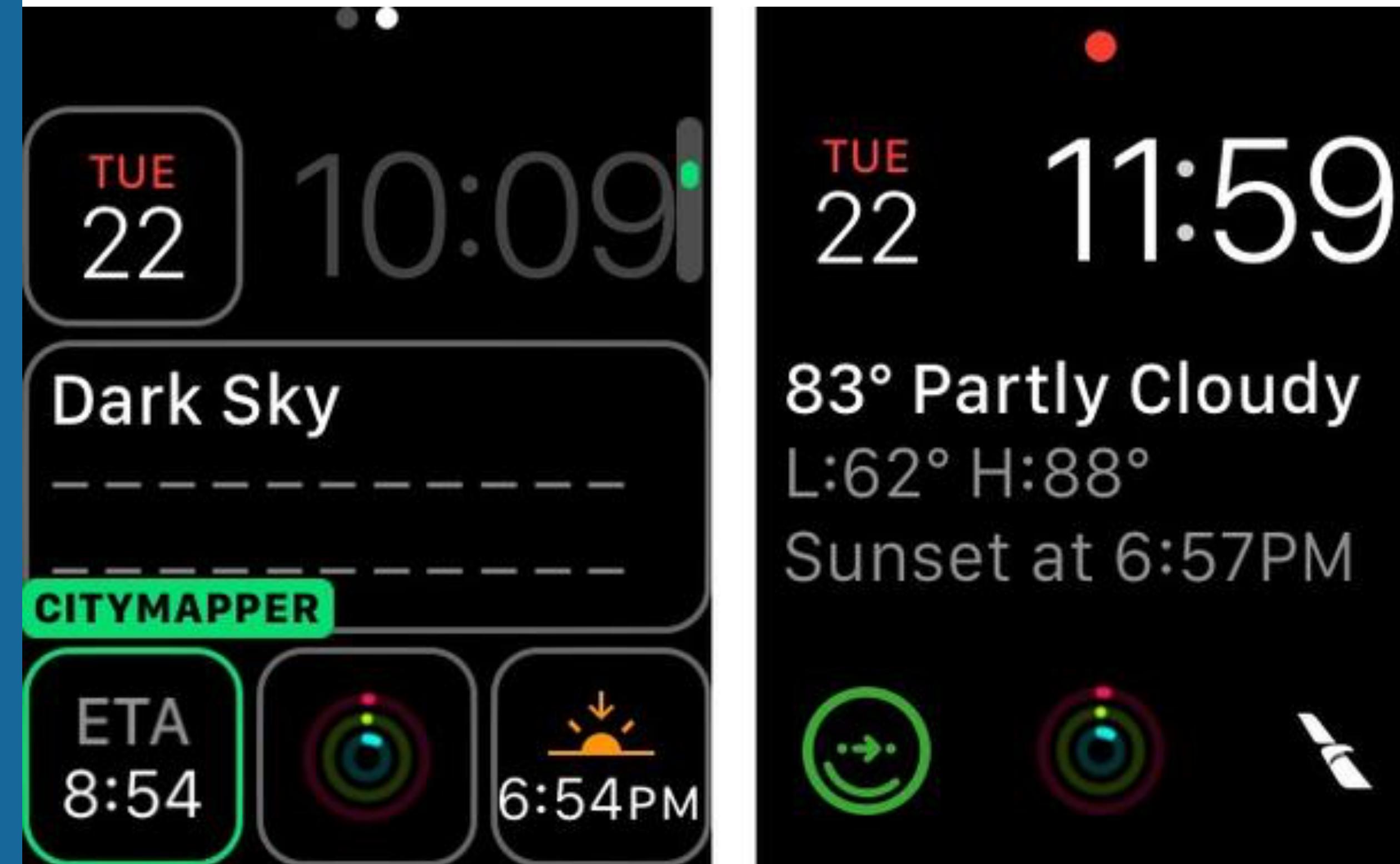


COMPLICATIONS

USER INTERACTIONS

COMPLICATIONS SUBTITLE

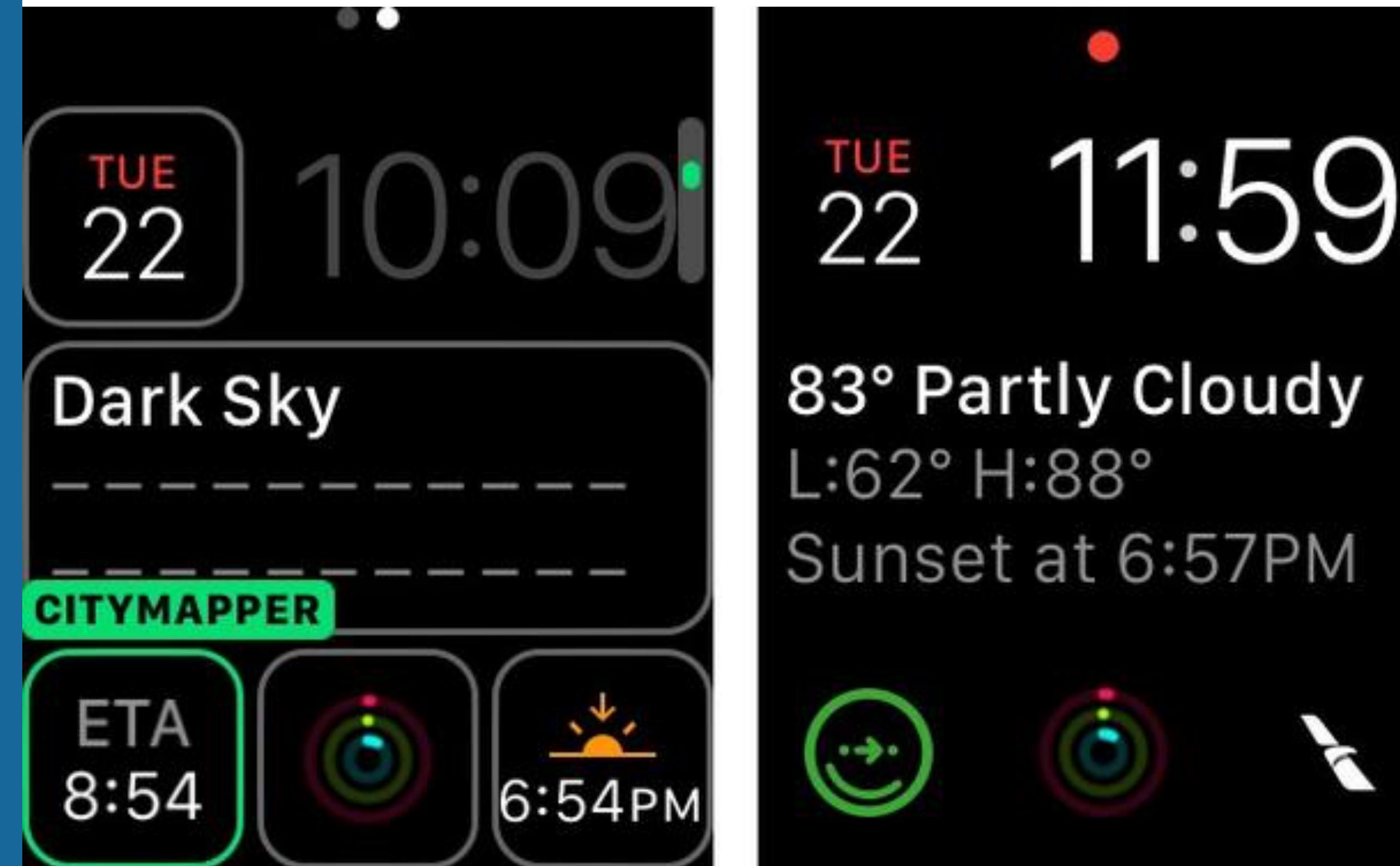
- Small visual elements that appear on the watch face
- Customizable
- Always visible



USER INTERACTIONS

COMPLICATIONS SUBTITLE

- The term *complication* comes from watch making, where the addition of features added complexity to the watch construction



USER INTERACTIONS



USER INTERACTIONS



USER INTERACTIONS

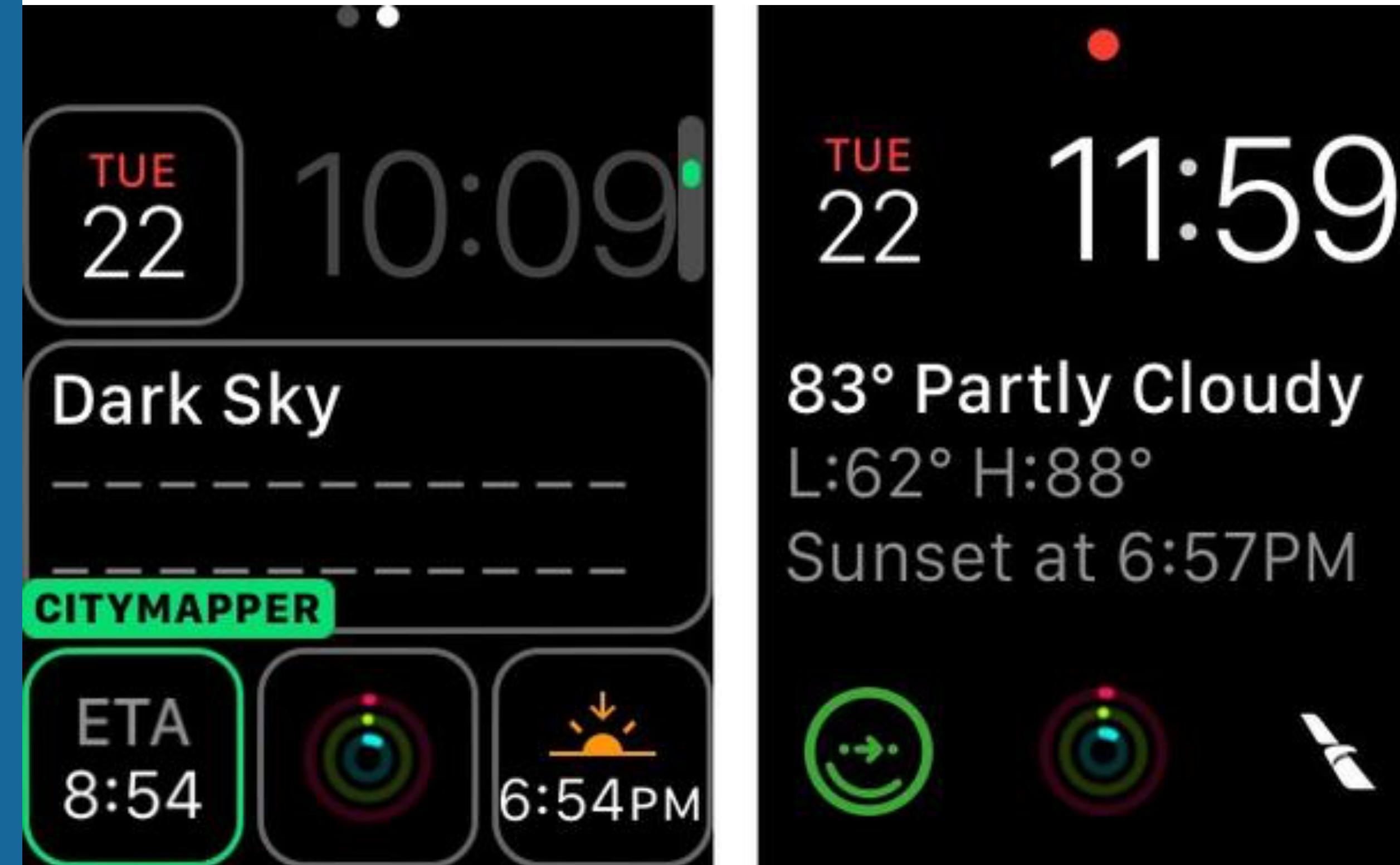


USER INTERACTIONS

COMPLICATIONS

SUBTITLE

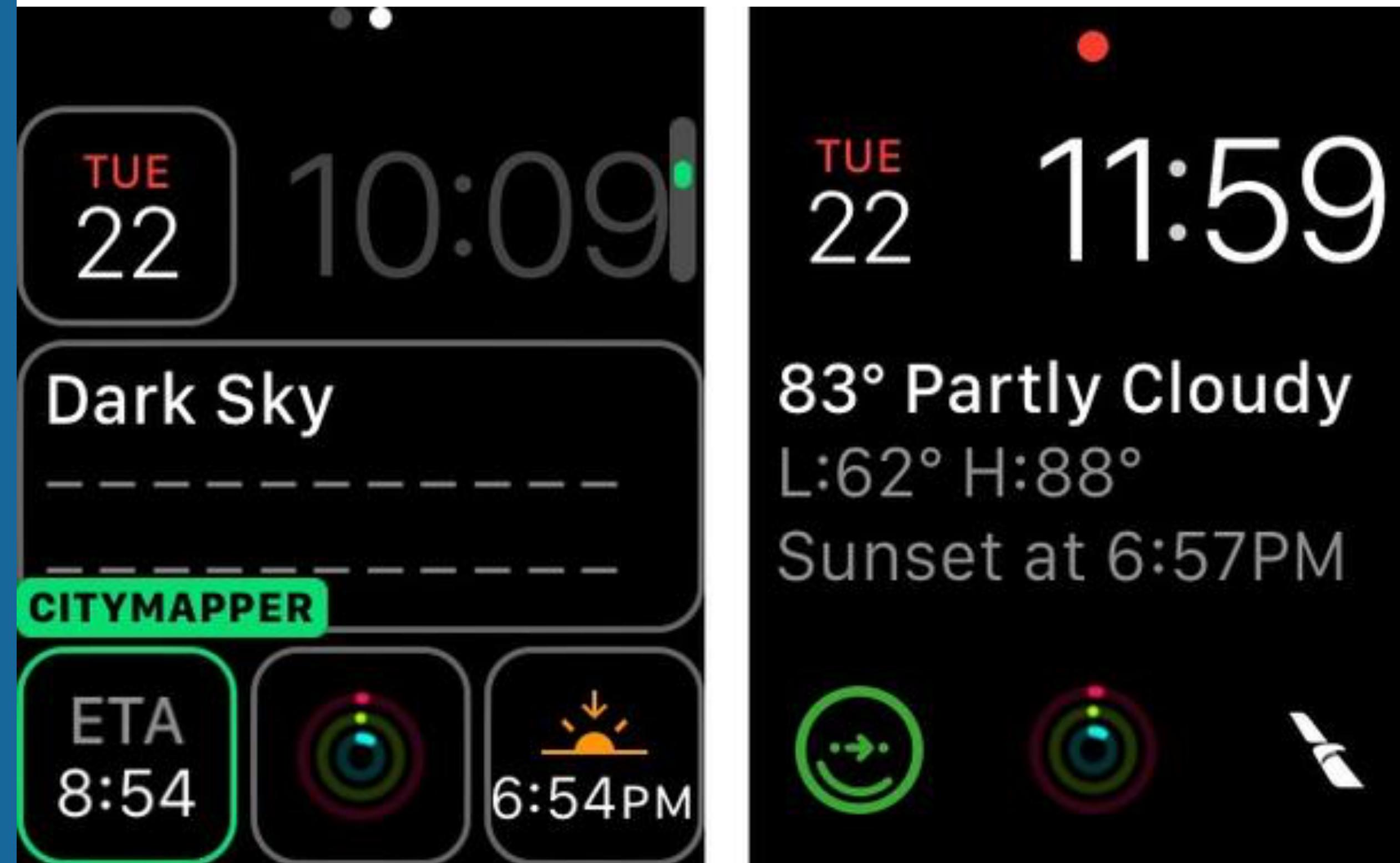
- Number of complications varies depending on the watch face
- At least 2



USER INTERACTIONS

COMPLICATIONS

- Apps whose complications are selected on the face get VIP treatment
 - Stays in memory; fast launch
 - Receives more time to execute background tasks
 - Receive background updates (at least 2x/hour)



USER INTERACTIONS

COMPLICATIONS

Apple recommends that all Watch apps include a complication, even if that complication only acts as a button to launch the app. For information about complications and how to implement them, see [Complication Essentials](#).



USER INTERFACES ON WATCHOS

USER INTERACTIONS



Glanceable



Actionable

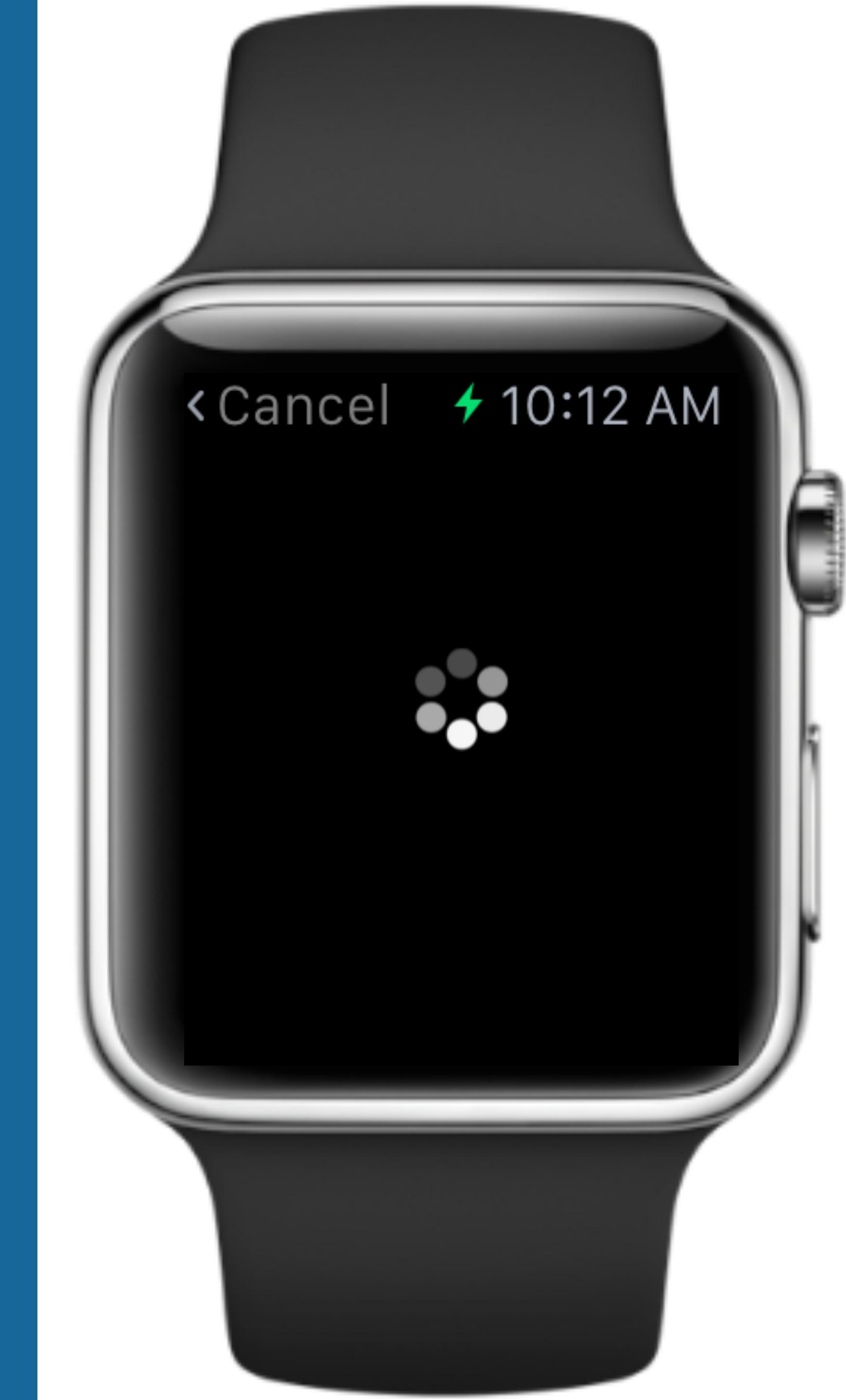


Responsive

- Design for quick interactions

USER INTERACTIONS

- Users will not be interested in seeing this interface in your application



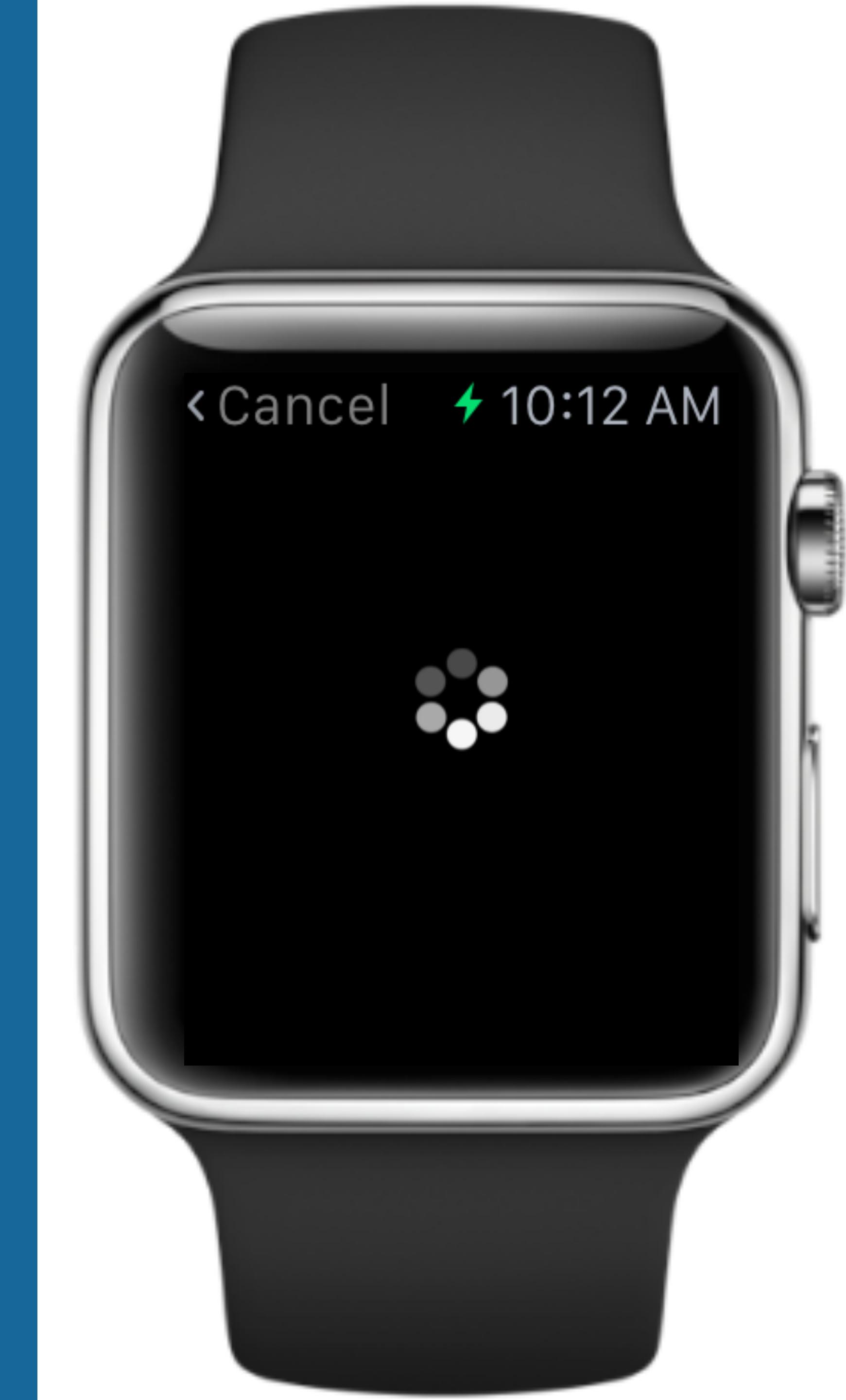
USER INTERACTIONS

- How long is a quick interaction?

. 2 seconds

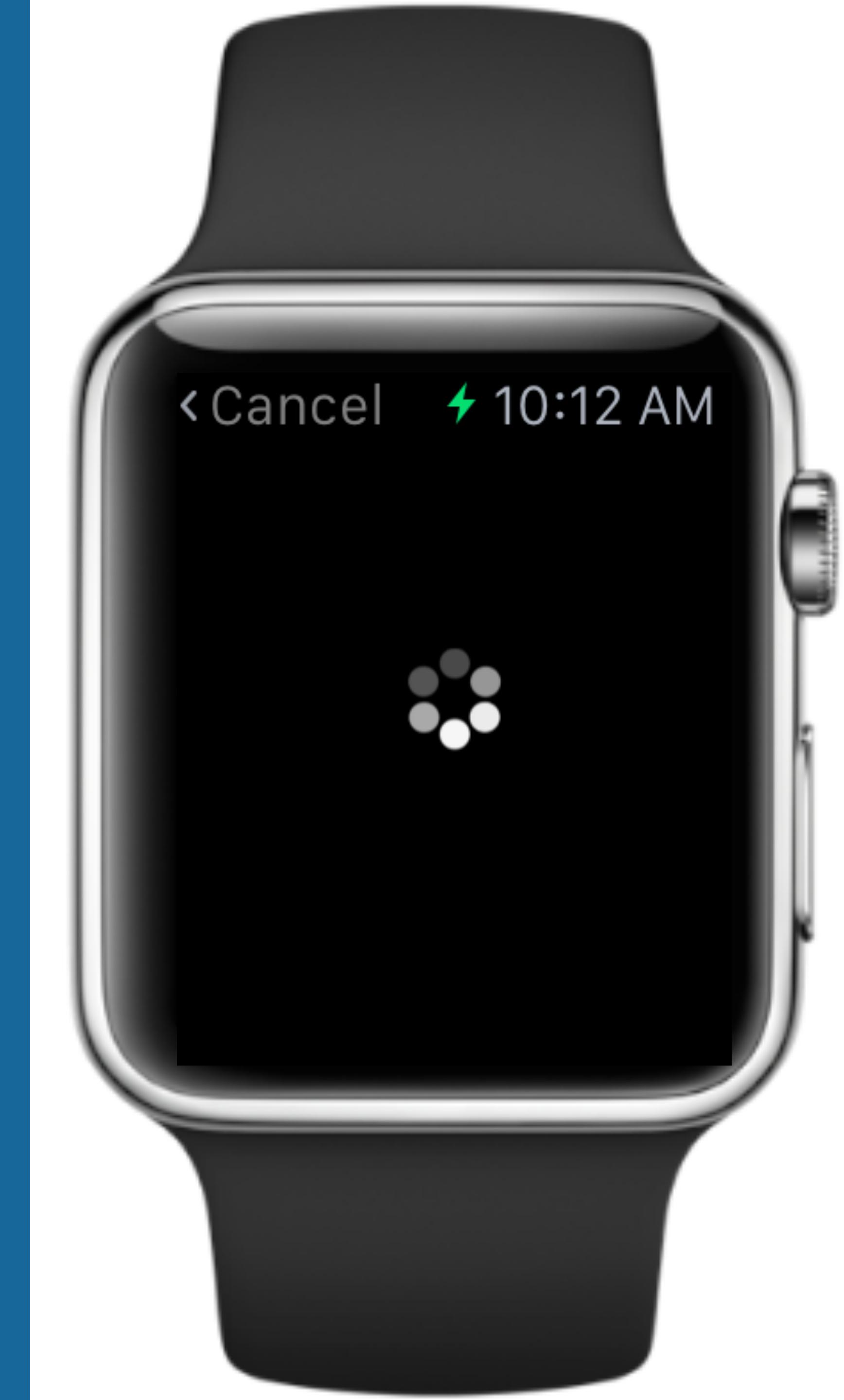
USER INTERACTIONS

- Design for quick and fast interactions
- Compliment your iOS app
- Be very conscious of load times



USER INTERACTIONS

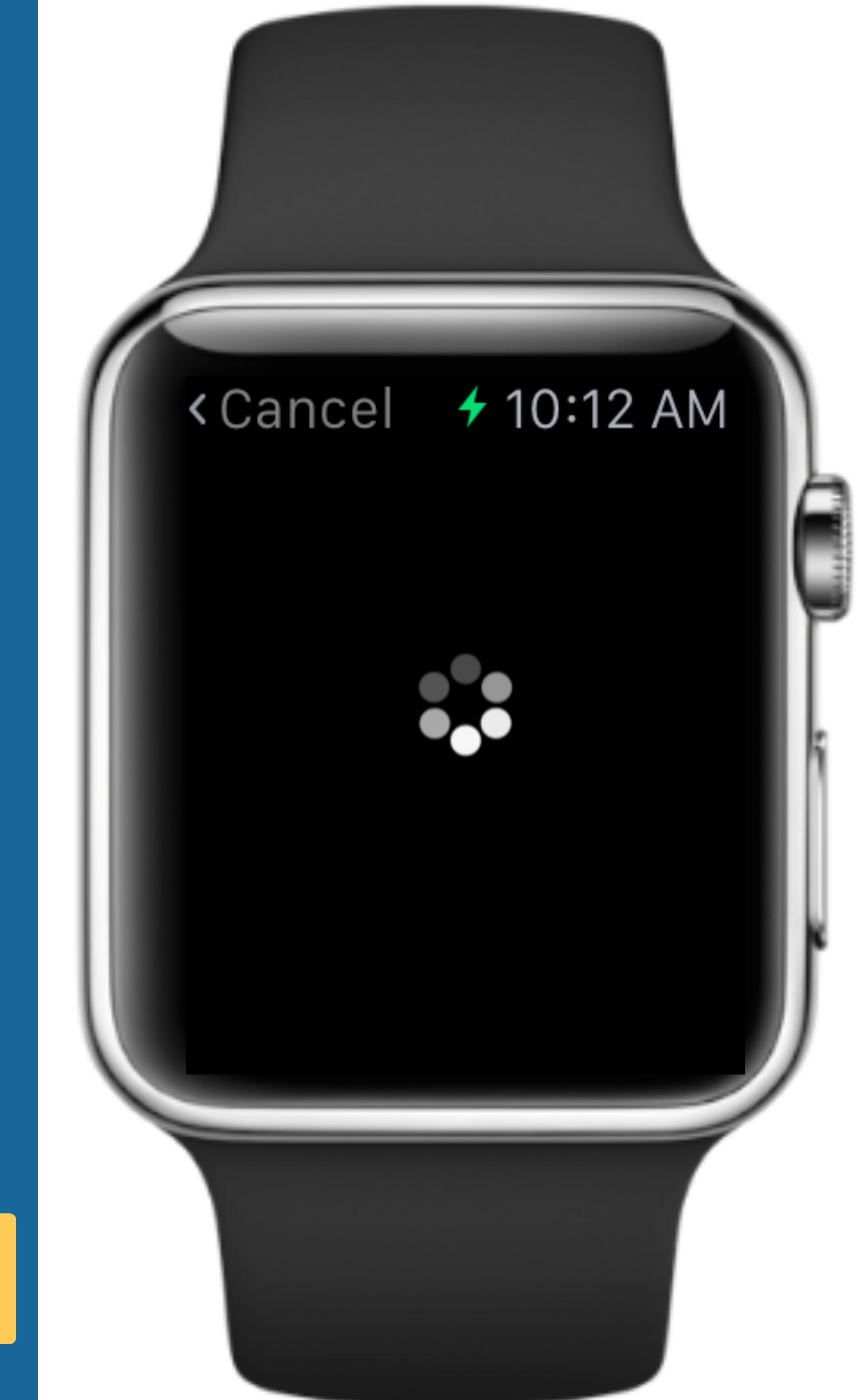
- watchOS3 additions to support quick interactions
 - Gesture recognizers
 - Digital crown rotation



USER INTERACTIONS

- watchOS additions to support displaying and updating information
 - Improved table navigation (vertical paging)
 - Support for new Notifications Framework
 - SceneKit and SpriteKit integration

YES, SPRITEKIT



CONFIGURING XCODE PROJECTS FOR DEPENDENT



CONFIGURING XCODE PROJECTS

- Create a new iOS Watch App
 - Create a single view controller iOS app



WatchKit



CONFIGURING XCODE PROJECTS

- What kind of watch app
 - Requires an iOS app for dependent
- Add a new WatchKit app target
 - Configures the bundles and initial resources for your WatchKit app and WatchKit extension
 - Those bundles are then delivered as part of your iOS app



WatchKit



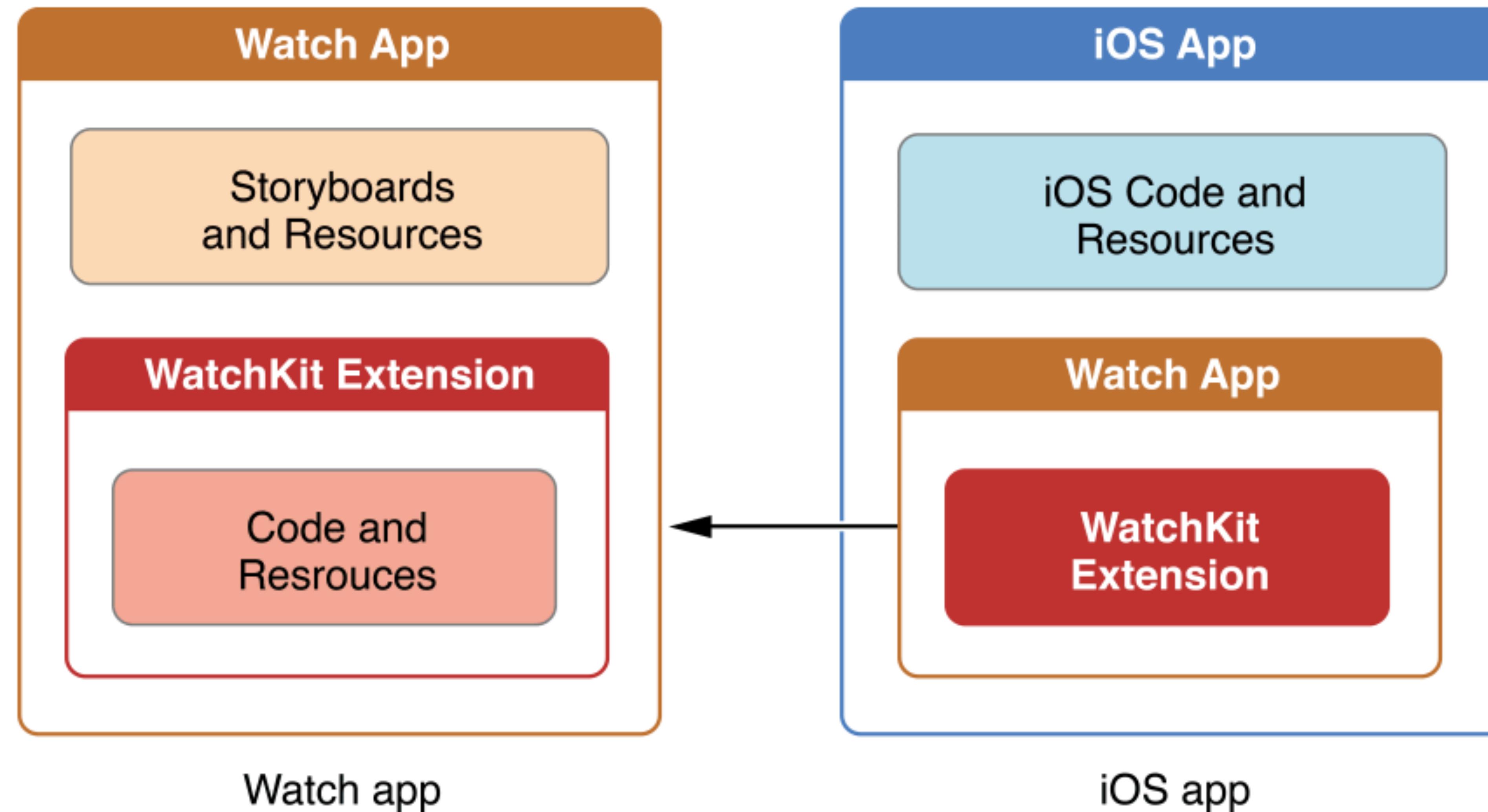
CONFIGURING XCODE PROJECTS FOR WATCH

- Adding a Watch app target to your Xcode project
 - Creates two new executables
 - Updates your project's build dependencies.
 - Creates build schemes for building and debugging the interfaces
 - Main Watch app interface, glances, notifications, and complications
- Building your iOS app builds all three executables (the iOS app, Watch app, and WatchKit extension) and packages them together inside the iOS app's bundle

CONFIGURING XCODE PROJECTS

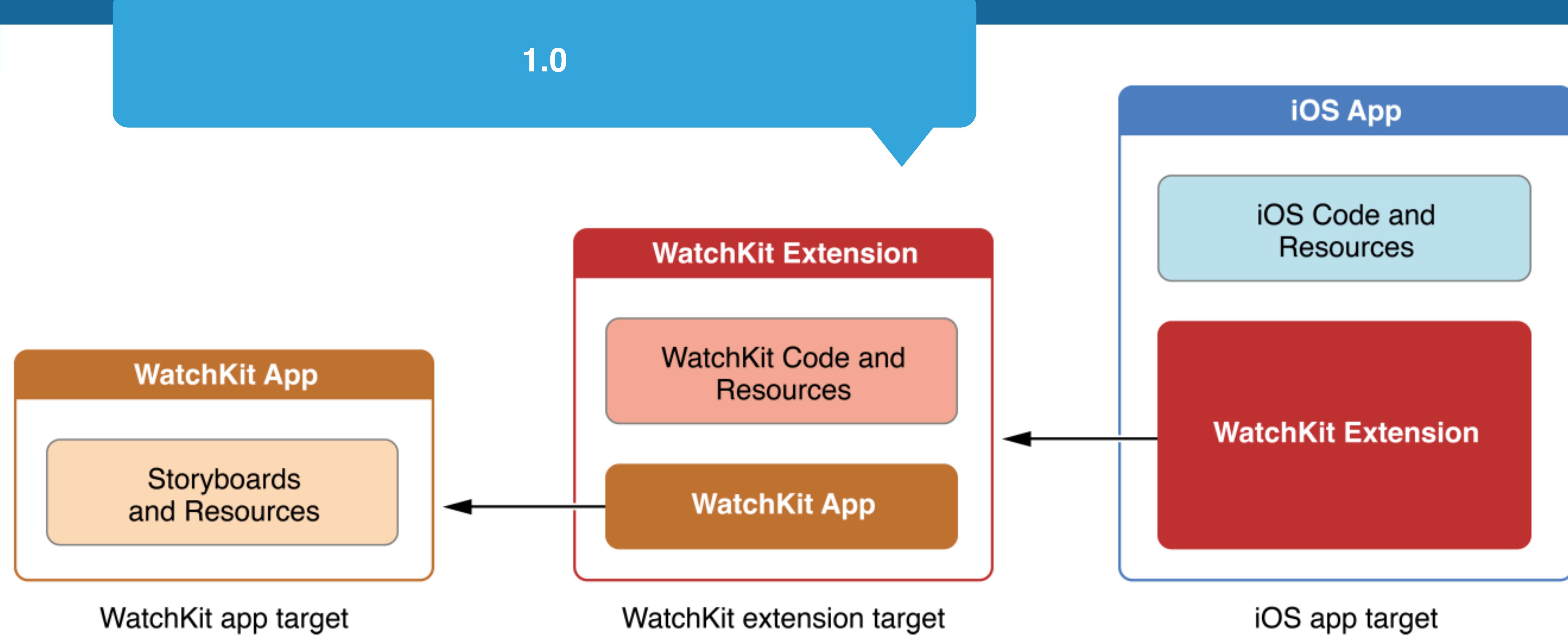
- Building your iOS app builds all three executables
 - iOS app
 - Watch app
 - WatchKit extension
- Packaged together inside the iOS app's bundle

CONFIGURING XCODE PROJECTS

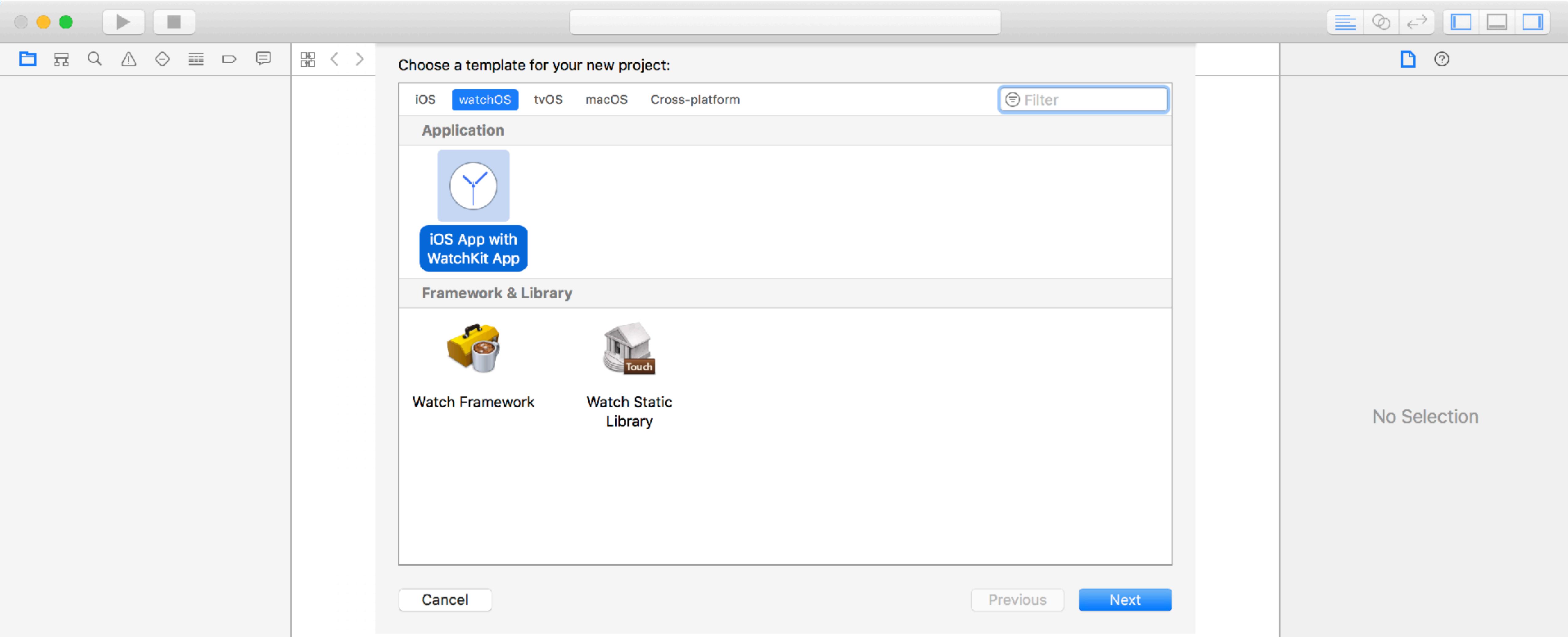


CONFIGURING XCODE PROJECTS

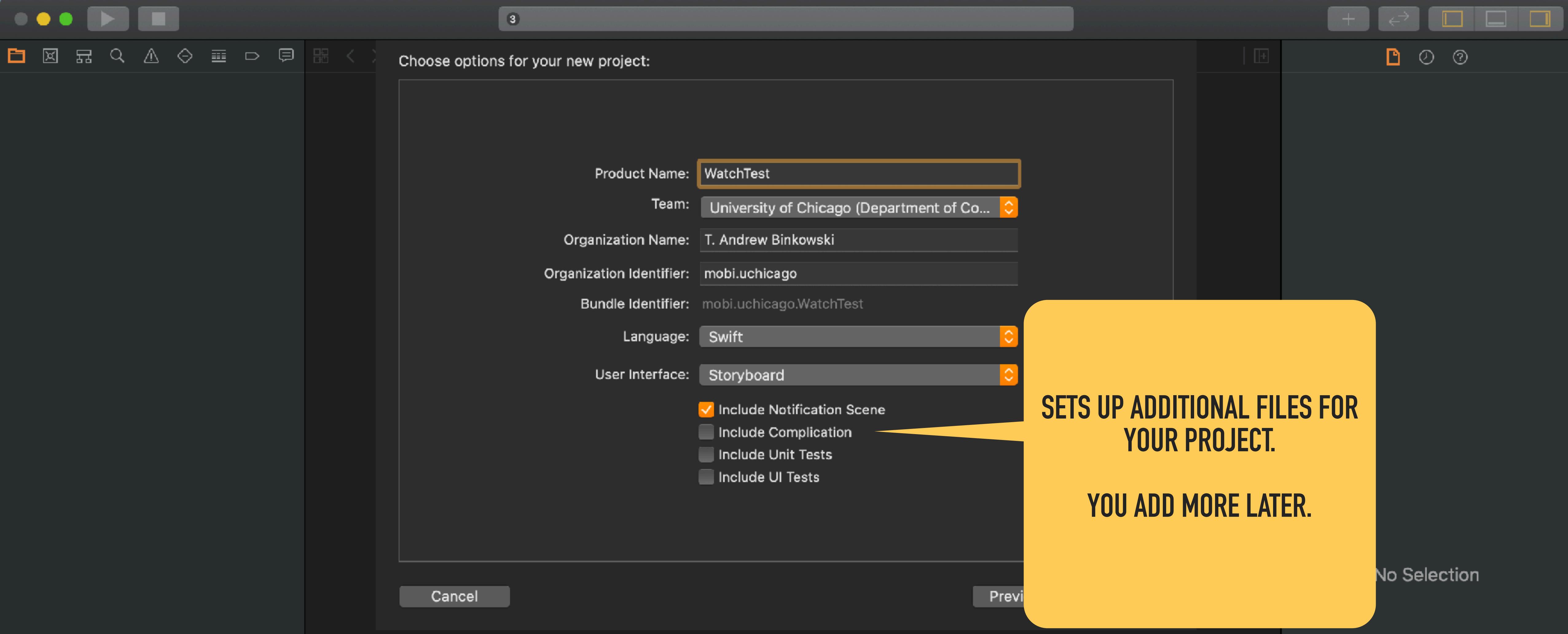
1.0



CONFIGURING XCODE PROJECTS FOR WATCH



CONFIGURING XCODE PROJECTS FOR WATCH



CONFIGURING XCODE PROJECTS FOR WATCH

The screenshot shows the Xcode interface with the project 'WatchTest' selected in the left sidebar. The main area displays the 'Info' tab of the target configuration. A green callout bubble points to the 'WatchTest' target in the 'TARGETS' section of the 'PROJECT' sidebar, containing the text: 'Executables created for a watch project'. The 'Identity' section shows the display name 'WatchTest', bundle identifier 'mobi.uchicago.WatchTest', version '1.0', and build '1'. The 'Deployment Info' section specifies the target as 'iOS 13.4' and devices as 'iPhone' and 'iPad'. The 'Main Interface' is set to 'Main'. Under 'Device Orientation', 'Portrait' is checked, while 'Upside Down', 'Landscape Left', and 'Landscape Right' are also selected. The 'Status Bar Style' is set to 'Default'. The right sidebar contains sections for 'Identity and Type', 'Project Document', and 'Text Settings'.

WatchTe...chKit App > Andrew's Apple Watch via Andrew Binkowski's iPhone XS 3 Waiting for Device

WatchTest

WatchTest

AppDelegate.swift

SceneDelegate.swift

ViewController.swift

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

WatchTest WatchKit App

Interface.storyboard

Assets.xcassets

Info.plist

WatchTest WatchKit Extension

InterfaceController.swift

ExtensionDelegate.swift

NotificationController.swift

ComplicationController.swift

Assets.xcassets

Info.plist

PushNotificationPayload.apns

Products

General Signing & Capabilities Resource Tags

Info Build Settings Build Phases Build Rules

PROJECT

WatchTest

TARGETS

WatchTest

WatchTest WatchK...

WatchTest WatchK...

Display Name WatchTest

Bundle Identifier mobi.uchicago.WatchTest

Version 1.0

Build 1

Deployment Info

Target Device

iOS 13.4

iPhone

iPad

Mac (requires macOS 10.15)

Main Interface Main

Device Orientation

Portrait

Upside Down

Landscape Left

Landscape Right

Status Bar Style Default

Hide status bar

Requires full screen

Supports multiple windows

+

Identity and Type

Name WatchTest

Location Absolute

WatchTest.xcodeproj

Full Path /Users/tabinkowski/Google Drive/g-Teaching/uchicago.mobi/uchicago.mobi-courses/mpcs51032/mpcs51032-2020-spring-playground/session-5/WatchTest/WatchTest.xcodeproj

Project Document

Project Format Xcode 9.3-compatible

Organization T. Andrew Binkowski

Class Prefix

Text Settings

Indent Using Spaces

Widths Tab 2 Indent 2

Wrap lines

CONFIGURING XCODE PROJECTS FOR WATCH

The screenshot shows the Xcode interface with a project named "WatchTest". The left sidebar displays the project structure, including files like AppDelegate.swift, SceneDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, Info.plist, and WatchTest WatchKit App. A green callout bubble points from the "Products" section to the "WatchTest WatchKit Extension" folder, which contains InterfaceController.swift, ExtensionDelegate.swift, NotificationController.swift, ComplicationController.swift, Assets.xcassets, Info.plist, and PushNotificationPayload.apns.

The main area shows the "General" tab of the WatchTest target. The "Identity" section includes:

- Display Name: WatchTest
- Bundle Identifier: mobi.uchicago.WatchTest
- Version: 1.0
- Build: 1

The "Deployment Info" section shows the target is set to "WatchKit Extension" and "Device" is set to "iPhone". It also lists supported devices: iPhone, iPad, and Mac (requires macOS 10.15). The "Main Interface" is set to "Main".

The "Text Settings" section includes:

- Indent Using: Spaces
- Widths: Tab 2, Indent 2
- Wrap lines: checked

A green callout bubble contains the text: "WatchKitExtension runs on watch".

CONFIGURING XCODE PROJECTS

The screenshot shows the Xcode interface with a WatchKit project named "WatchTest". The General tab is selected in the top navigation bar. A callout bubble highlights the "WatchKit App runs on watch" note and the "No code files!" message.

Identity and Type

- Name: WatchTest
- Location: Absolute
- WatchTest.xcodeproj
- Full Path: /Users/tabinkowski/Google Drive/g-Teaching/uchicago.mobi/uchicago.mobi-courses/mpcs51032/mpcs51032-2020-spring/mpcs51032-2020-spring-playground/session-5/WatchTest/WatchTest.xcodeproj

Project Document

- Project Format: Xcode 9.3-compatible
- Organization: T. Andrew Binkowski
- Class Prefix:

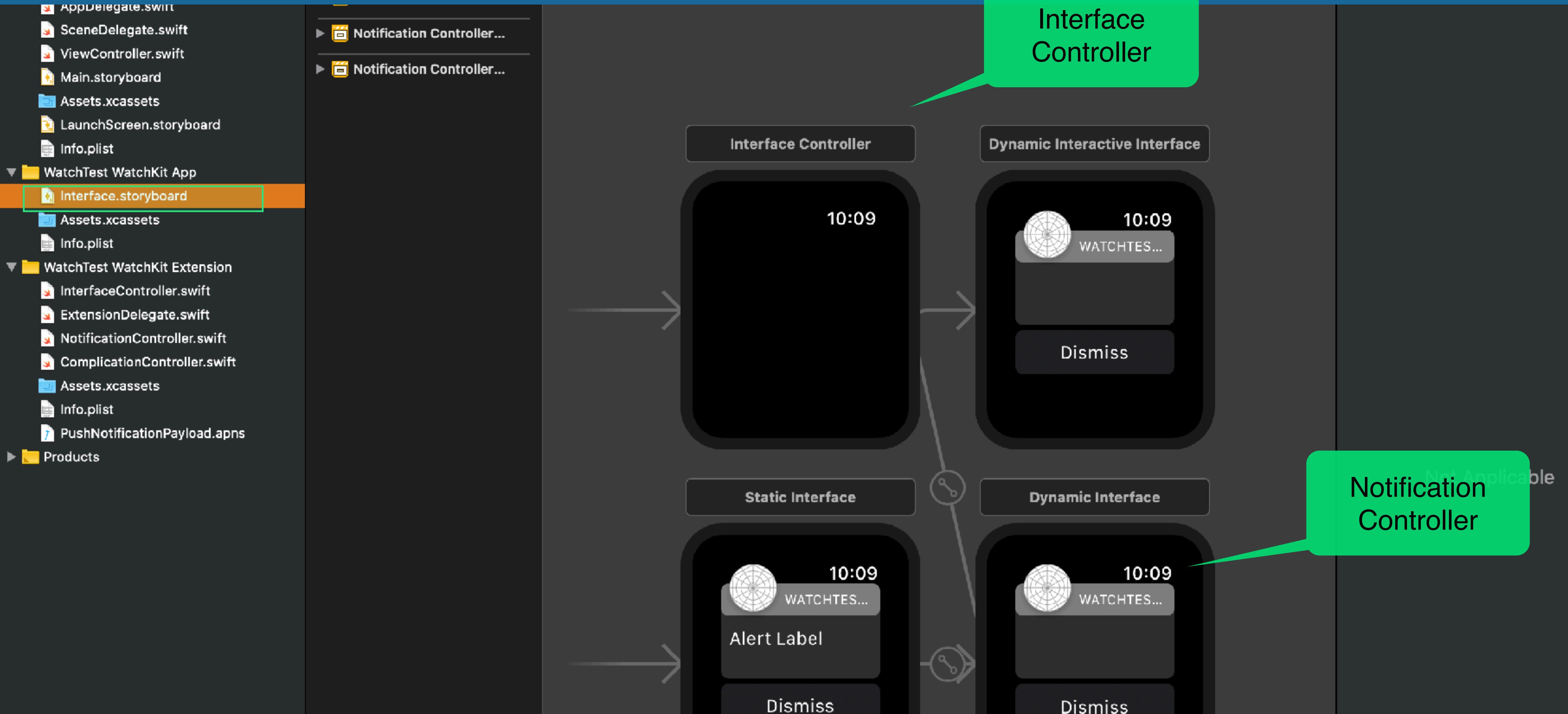
Text Settings

- Indent Using: Spaces
- Widths: Tab 2, Indent 2
- Wrap lines: checked

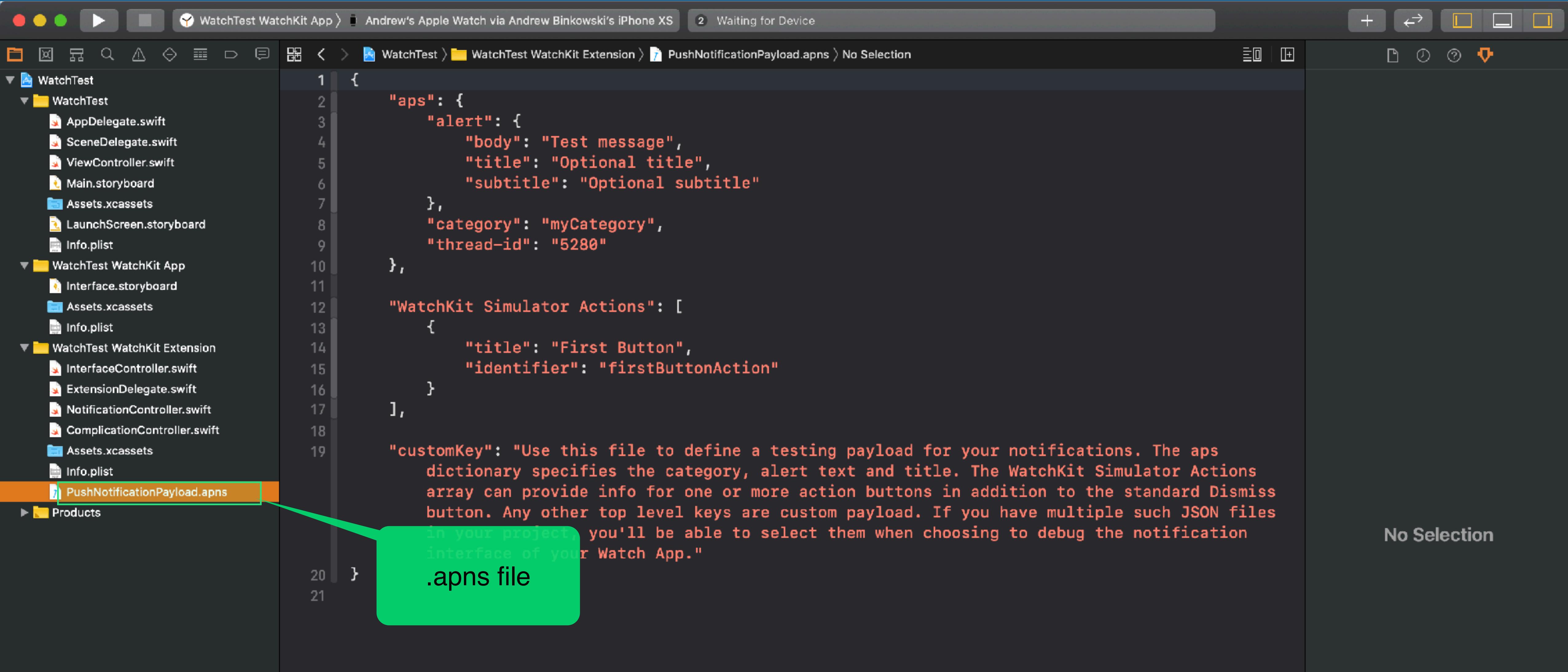
General Tab Configuration

- PROJECT**: WatchTest
- TARGETS**: WatchTest (selected)
- Identity**:
 - Display Name: WatchTest
 - Bundle Identifier: mobi.uchicago.WatchTest
 - Version: 1.0
- Deployment Info**:
 - iOS 13.4
 - iPhone
 - iPad (selected)
 - Mac (requires macOS 10.15)
- Main Interface**: Main
- Device Orientation**:
 - Portrait (selected)
 - Upside Down
 - Landscape Left
 - Landscape Right
- Status Bar Style**: Default
 - Hide status bar
 - Requires full screen
 - Supports multiple windows

CONFIGURING XCODE PROJECTS



CONFIGURING XCODE PROJECTS

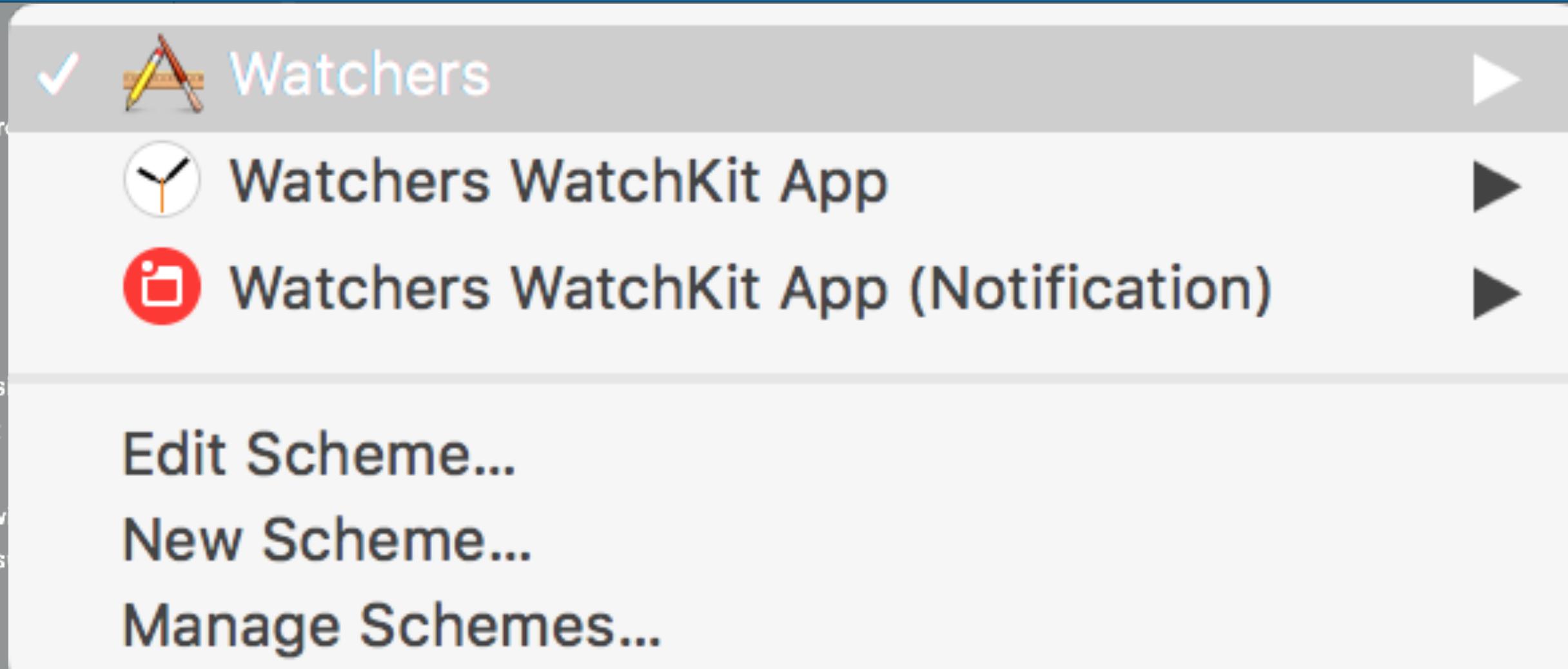


The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure:
 - WatchTest
 - WatchTest (containing: AppDelegate.swift, SceneDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist)
 - WatchTest WatchKit App (containing: Interface.storyboard, Assets.xcassets, Info.plist)
 - WatchTest WatchKit Extension (containing: InterfaceController.swift, ExtensionDelegate.swift, NotificationController.swift, ComplicationController.swift, Assets.xcassets, Info.plist)
 - Products
- Editor:** Displays the content of the `PushNotificationPayload.apns` file.
- Toolbar:** Shows the current target as `WatchTest WatchKit App`, connected to `Andrew's Apple Watch via Andrew Binkowski's iPhone XS`, and the status `Waiting for Device`.
- Bottom Status Bar:** Shows "No Selection".

```
1  {
2      "aps": {
3          "alert": {
4              "body": "Test message",
5              "title": "Optional title",
6              "subtitle": "Optional subtitle"
7          },
8          "category": "myCategory",
9          "thread-id": "5280"
10     },
11
12     "WatchKit Simulator Actions": [
13         {
14             "title": "First Button",
15             "identifier": "firstButtonAction"
16         }
17     ],
18
19     "customKey": "Use this file to define a testing payload for your notifications. The aps
20         dictionary specifies the category, alert text and title. The WatchKit Simulator Actions
21         array can provide info for one or more action buttons in addition to the standard Dismiss
button. Any other top level keys are custom payload. If you have multiple such JSON files
in your project, you'll be able to select them when choosing to debug the notification
interface of your Watch App."
22     .apns file
23 }
```

CONFIGURING XCODE PROJECTS

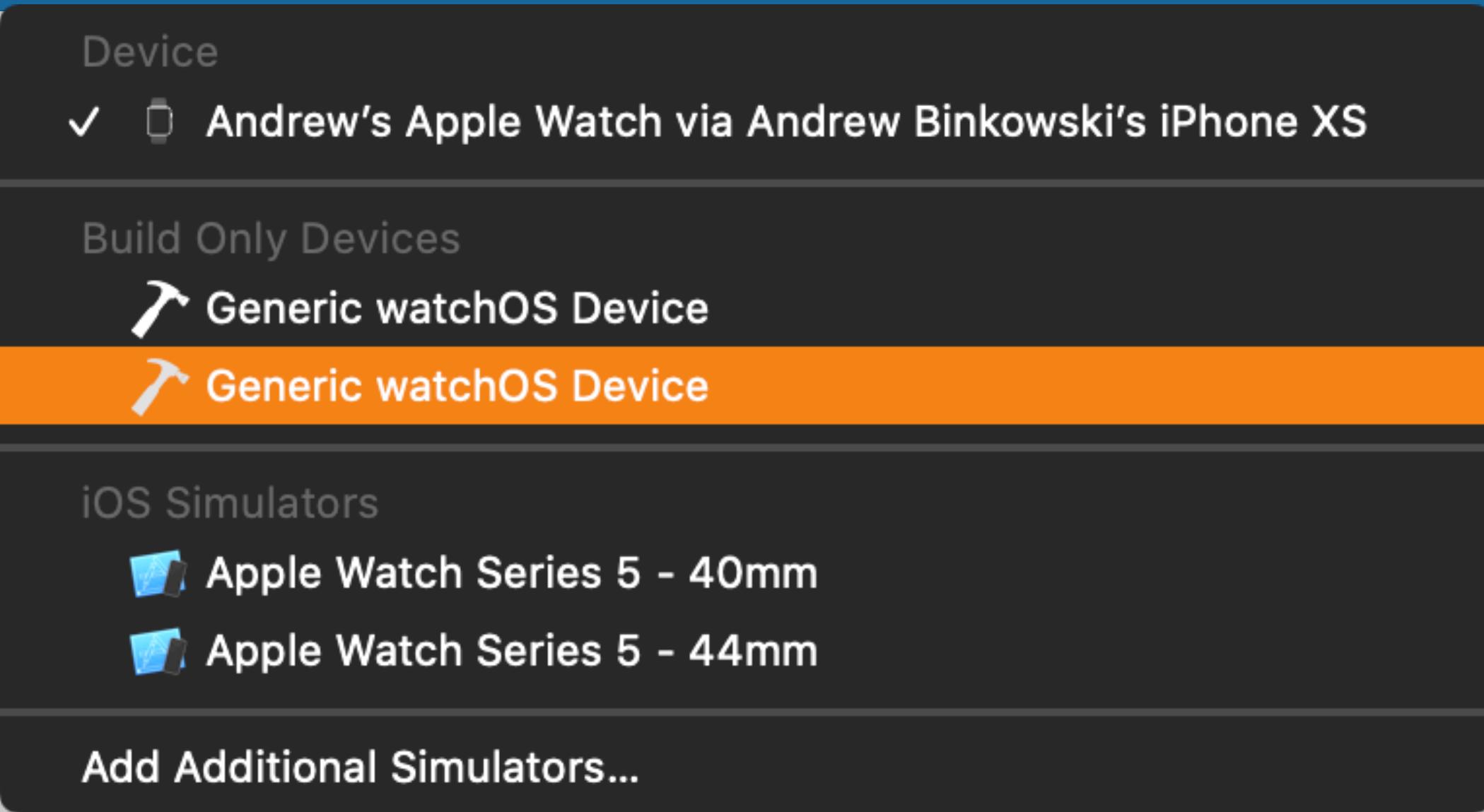


The screenshot shows the Xcode interface with the following details:

- Scheme Selector:** The "Watchers" scheme is selected.
- Project Structure:** The left sidebar shows the project structure with files like Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, WatchTest WatchKit App (containing Interface.storyboard, Assets.xcassets, Info.plist), WatchTest WatchKit Extension (containing InterfaceController.swift, ExtensionDelegate.swift, NotificationController.swift, ComplicationController.swift, Assets.xcassets, Info.plist, PushNotificationPayload.apns), and Products.
- Schemes List:** The "Watchers" scheme is listed twice:
 - Watchers WatchKit App:** Represented by a yellow circle icon.
 - Watchers WatchKit App (Notification):** Represented by a red square icon.
- Action Buttons:** To the right of each scheme entry are three buttons: a white triangle (play), a black triangle (stop), and a black triangle with a dot (refresh).
- Text Area:** A large text area displays code snippets related to scene transitions and notifications. It includes comments explaining the purpose of each method.
- Status Bar:** At the bottom right, it says "No Selection".

```
25    // This occurs shortly after the scene enters the background, or when its session is discarded.
26    // Release any resources associated with this scene that can be re-created the next time the
27    // scene connects.
28    // The scene may re-connect later, as its session was not necessarily discarded (see
29    // `application:didDiscardSceneSessions` instead).
30
31    func sceneDidBecomeActive(_ scene: UIScene) {
32        // Called when the scene has moved from an inactive state to an active state.
33        // Use this method to restart any tasks that were paused (or not yet started) when the scene
34        // was inactive.
35
36    func sceneWillResignActive(_ scene: UIScene) {
37        // Called when the scene will move from an active state to an inactive state.
38        // This may occur due to temporary interruptions (ex. an incoming phone call).
39
```

CONFIGURING XCODE PROJECTS



The screenshot shows the Xcode interface with a project tree on the left and a code editor on the right. A context menu is open over some code, specifically targeting line 18.

```
13 Device
14 ✓ Andrew's Apple Watch via Andrew Binkowski's iPhone XS
15
16 options connectionOptions:
17
18 Build Only Devices
19
20 Generic watchOS Device
21
22 Generic watchOS Device
23
24 iOS Simulators
25
26 Apple Watch Series 5 - 40mm
27
28 Apple Watch Series 5 - 44mm
29
30 Add Additional Simulators...
31
32
33
34
35
36
37
38
```

The menu items are:

- Device
- ✓ Andrew's Apple Watch via Andrew Binkowski's iPhone XS
- Build Only Devices
 - Generic watchOS Device
 - Generic watchOS Device
- iOS Simulators
 - Apple Watch Series 5 - 40mm
 - Apple Watch Series 5 - 44mm
- Add Additional Simulators...

Below the menu, the code editor shows Swift code for a scene delegate:

```
// Release any resources associated with this scene that can be re-created the next time the
// scene connects.
// The scene may re-connect later, as its session was not necessarily discarded (see
// `application:didDiscardSceneSessions` instead).
}

func sceneDidBecomeActive(_ scene: UIScene) {
    // Called when the scene has moved from an inactive state to an active state.
    // Use this method to restart any tasks that were paused (or not yet started) when the scene
    // was inactive.
}

func sceneWillResignActive(_ scene: UIScene) {
    // Called when the scene will move from an active state to an inactive state.
    // This may occur due to temporary interruptions (ex. an incoming phone call).
}
```

On the far right, there is a "No Selection" message.

WORST APP
EVER



WORST WATCH APP EVER



WatchInterface WatchKit App > iPhone 6 Finished running mobi.uchicago.WatchInterface.watchkitextension on iPhone 6

WatchInterface.xcodeproj Debug

Main →

Glance →

WKInterfaceLabel

View properties

You won't need a designer fo this one

You won't need a designer fo this one

No Debug Session

Interface Controller Scene

Interface Controller

HeaderLabel

Main Entry Point

Glance Interface Contr...

Static Notification Inter...

Notification Controller...

Label

Text: HeaderLabel

Text Color: Default

Font: Body

Min Scale: 1

Alignment: Center

Lines: 1

View

Alpha: 1

Hidden:

Installed:

Position

Horizontal: Center

Vertical: Center

Size

Width: Size To Fit Content

Height: Size To Fit Content

Glance Interface Controller

Glance Interface

Group

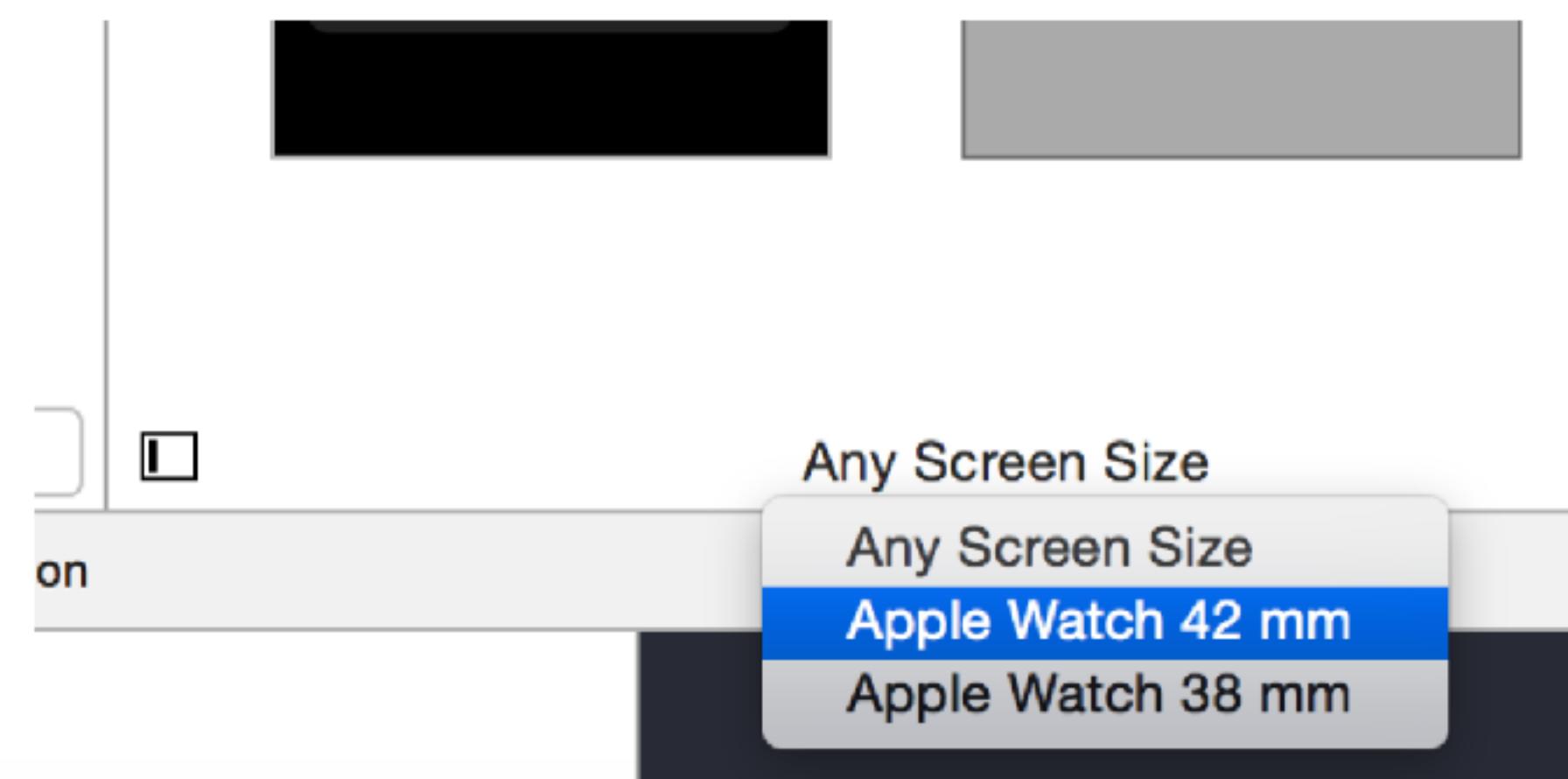
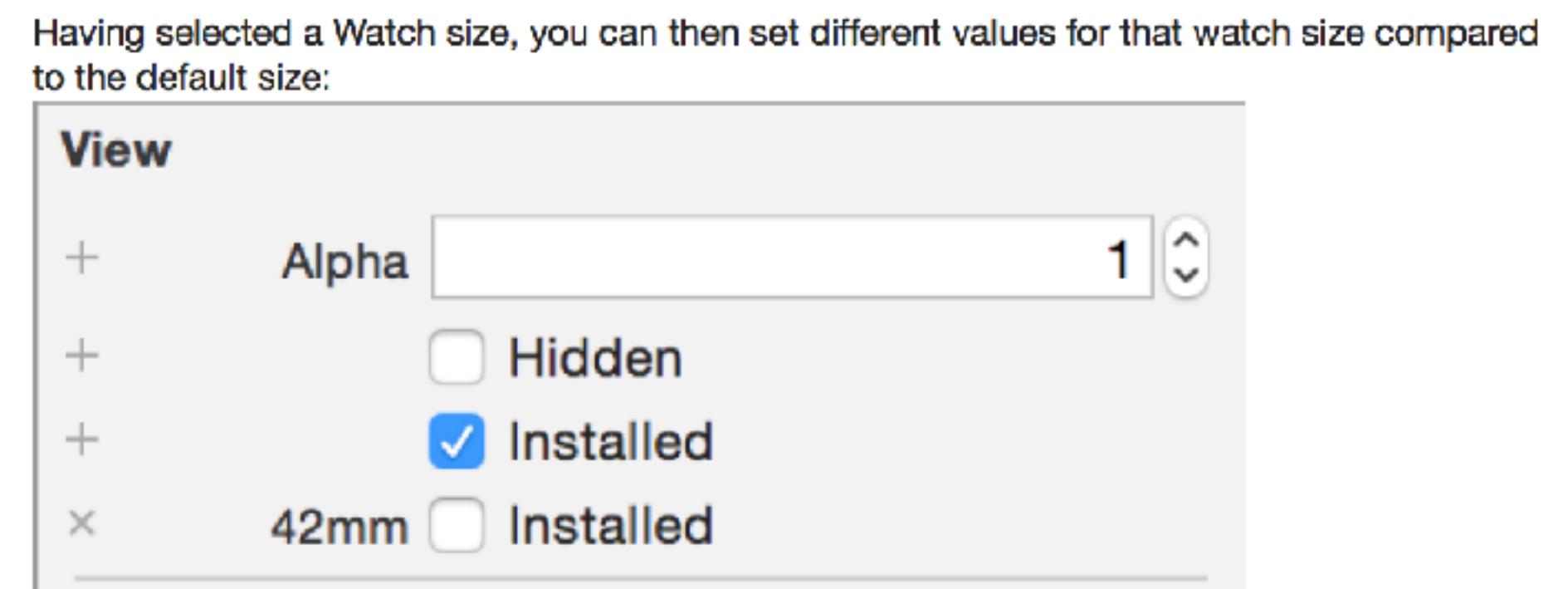
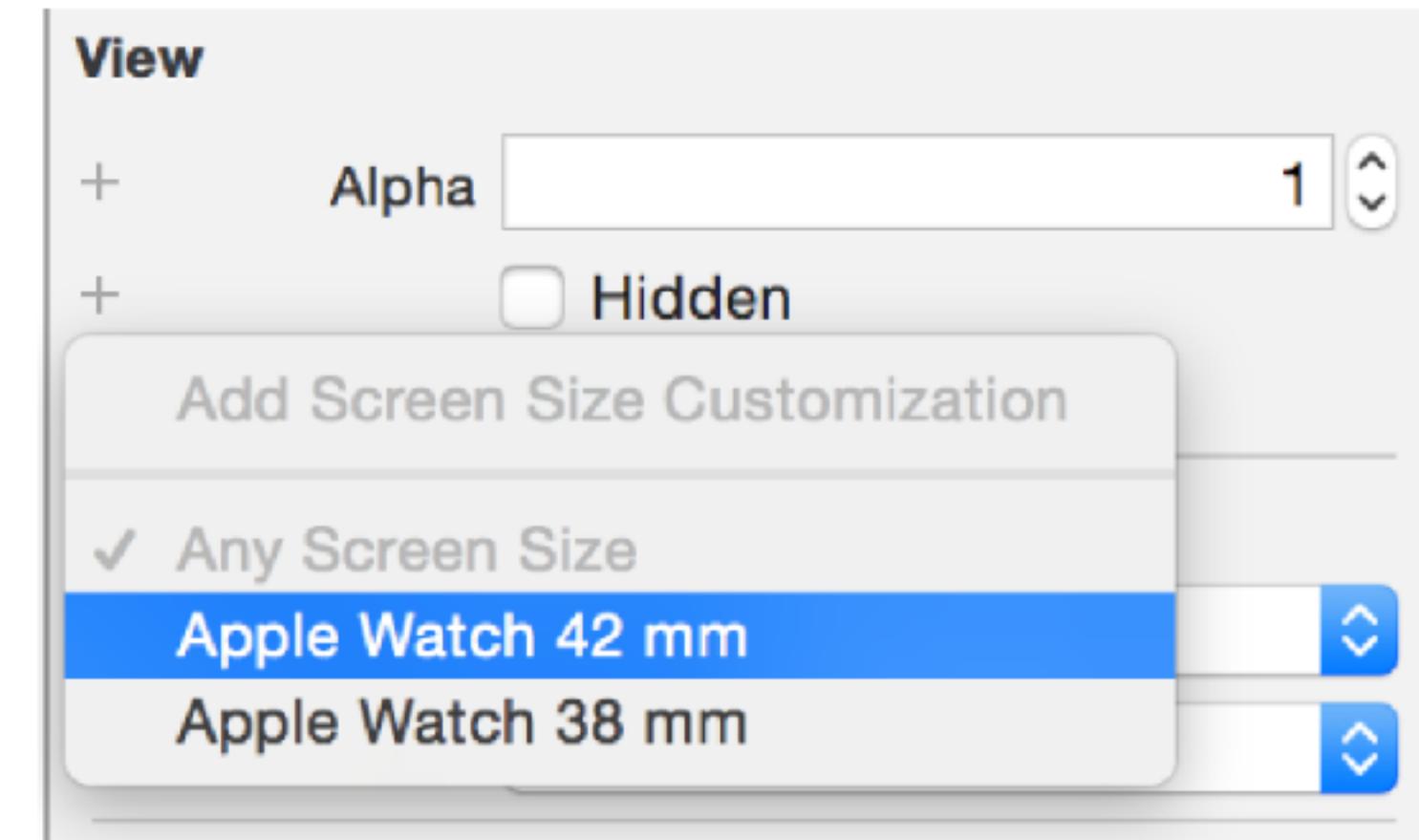
Group

WORST WATCH APP EVER



SUBTITLE

- Installed vs. hidden
 - Every interface object has to be created and installed with app
 - Can NOT create programmatically
 - Animation API between states
- Use case
 - Include/exclude interface object based on size of watch



WORST WATCH APP EVER



WatchInterface > Wat...App > Inte...oard > Inte...ase) > Inte...ene > Interface Controller > Lbl HeaderLabel

Interface Controller Scene

Interface Controller

HeaderLabel

Main Entry Point

Glance Interface Contr...

Static Notification Inter...

Notification Controller...

Main

HeaderLabel

12:00

Glance Interface

Group

Label

+ Text HeaderLabel

+ Text Color Default

+ Font Body

+ Min Scale 1

+ Alignment

+ Lines 1

View

+ Alpha 1

+ Hidden

+ Installed

Position

+ Horizontal Left

+ Vertical Bottom

Size

Width Size To Fit Content

Height Size To Fit Content

This screenshot shows the WatchKit interface editor in Xcode. The top navigation bar displays the project structure: WatchInterface > Wat...App > Inte...oard > Inte...ase) > Inte...ene > Interface Controller > Lbl HeaderLabel. The left sidebar lists the Interface Controller Scene, which contains the Interface Controller object, with HeaderLabel selected. The main canvas shows a watch face with a black header bar at the top displaying the time '12:00'. A blue rectangular area below it is labeled 'HeaderLabel'. A large grey arrow points from the word 'Main' to this blue area. Below the watch face, there is a preview of a 'Glance Interface' card titled 'Glance Interface' containing a single item labeled 'Group'. The right side of the interface editor shows the 'HeaderLabel' properties in the Attributes Inspector. The 'Label' section includes settings for Text (HeaderLabel), Text Color (Default), Font (Body), Min Scale (1), and Alignment. The 'View' section includes settings for Alpha (1), Hidden (unchecked), and Installed (checked). The 'Position' section includes settings for Horizontal (Left) and Vertical (Bottom). The 'Size' section includes settings for Width (Size To Fit Content) and Height (Size To Fit Content).

WORST WATCH APP EVER



WatchInterface > Wat...App > Inte...oard > Inte...ase) > Inte...ene > Interface Controller > Lbl HeaderLabel

Interface Controller Scene

Interface Controller

Lbl HeaderLabel

Main Entry Point

Glance Interface Contr...

Static Notification Inter...

Notification Controller...

Main

HeaderLabel

12:00

Glance Interface

Group

Label

+ Text HeaderLabel

+ Text Color Default

+ Font Body

+ Min Scale 1

+ Alignment ---

+ Lines 1

View

+ Alpha 1

+ Hidden

+ Installed

Position

+ Horizontal Right

+ Vertical Top

Size

Width Size To Fit Content

Height Size To Fit Content

This screenshot shows the WatchKit interface editor in Xcode. The top navigation bar displays the file structure: WatchInterface > Wat...App > Inte...oard > Inte...ase) > Inte...ene > Interface Controller > Lbl HeaderLabel. The left sidebar lists the scene hierarchy: Interface Controller Scene, Interface Controller, Lbl HeaderLabel (selected), Main Entry Point, Glance Interface Contr..., Static Notification Inter..., and Notification Controller... A large arrow labeled "Main" points from the scene list to a preview window. The preview window shows a watch face with a black header bar containing the text "12:00" and a blue main area containing the text "HeaderLabel". On the right side, the "HeaderLabel" properties are detailed in the Attributes Inspector. The "Label" section includes Text (HeaderLabel), Text Color (Default), Font (Body), Min Scale (1), Alignment (centered), and Lines (1). The "View" section includes Alpha (1), Hidden (unchecked), and Installed (checked). The "Position" section shows Horizontal alignment set to Right and Vertical alignment set to Top. The "Size" section indicates both Width and Height are set to "Size To Fit Content". Below the preview, another smaller window titled "Glance Interface" shows a dark screen with a single item labeled "Group".

WORST WATCH APP EVER



The screenshot shows the Xcode interface with the WatchKit storyboard and code editor. The storyboard on the left displays a single interface controller with a header label labeled "headerLabel". The code editor on the right shows the corresponding Swift code:

```
13 class InterfaceController: WKInterfaceController {
14
15     @IBOutlet weak var headerLabel: WKInterfaceLabel!
16
17     override func awakeWithContext(context: AnyObject?) {
18         super.awakeWithContext(context)
19
20         // Configure interface objects here.
21         self.headerLabel.setText("Worst Watch App Ever")
22     }
23 }
```

The line `@IBOutlet weak var headerLabel: WKInterfaceLabel!` is highlighted with a red box, and the line `self.headerLabel.setText("Worst Watch App Ever")` is also highlighted with a red box.

WORST WATCH APP EVER



The screenshot shows the Xcode interface with the following components:

- Left Sidebar:** Shows file navigation with items like "Glance Interface Contr...", "Static Notification Inter...", and "Notification Controller...".
- Central Area:** Displays a preview of the Apple Watch interface. The watch face shows "12:00" at the top, "Apple Watch 42mm" in the center, a battery icon with "9:28 PM" below it, and the text "Worst Watch App...".
- Code Editor:** Shows the Swift code for the "InterfaceController.swift" file. The code includes imports for WatchKit and Foundation, defines a class "InterfaceController" that inherits from "WKInterfaceController", and overrides the "awakeWithContext" method to set the header label text to "interFACES".
- Right Sidebar:** Includes sections for "Location", "Full Path", "Target Membership", "Text Settings", and "Status".

A yellow callout bubble with a black border and white text is positioned on the right side of the screen, containing the text:

HARDWARE>LOCK IS THE EQUIVALENT OF LOWERING YOUR WRIST

WORST WATCH APP EVER



WatchInterface Watch App Interface storyboard Interface Controller headerLabel

Interface Controller Scene

Interface Controller headerLabel → Main Entry Point

Glance Interface Contr...

Static Notification Inter...

Notification Controller...

Apple Watch 42mm

⚡ 9:32 PM

Worst Watch App Ever

Main

headerLabel

12:00

Glance Interface

Group

Label

- + Text headerLabel
- + Text Color Default
- + Font Body
- + Min Scale 0.5
- + Alignment
- + Lines 1

View

- + Alpha 1
- + Hidden
- + Installed

Position

- + Horizontal Left
- + Vertical Top

Size

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

This screenshot shows the Apple Watch interface editor in Xcode. The project navigation bar at the top lists the WatchInterface target, Watch App folder, Interface storyboard, Interface Controller scene, Interface Controller, and headerLabel. The left sidebar shows the Interface Controller Scene with its components: Interface Controller, headerLabel, Glance Interface Controller, Static Notification Interface, and Notification Controller. The main canvas displays two interface cards. The top card is the main Interface Controller, showing a black header with the time 12:00 and a blue body containing a white label with the text "headerLabel". The bottom card is a Glance Interface, showing a black header with the time 9:32 PM and a blue body with the text "Worst Watch App Ever". A large arrow labeled "Main" points from the glance interface to the main interface. The right-hand panel is the Attribute Inspector for the selected "headerLabel" text field, showing its properties: Text (headerLabel), Text Color (Default), Font (Body), Min Scale (0.5, highlighted in red), Alignment (Left), Lines (1), View (Alpha 1, Hidden unchecked, Installed checked), Position (Horizontal Left, Vertical Top), and Size (Width and Height both set to "Size To Fit Content").

WORST WATCH APP EVER



WatchInterface.xcodeproj Debug

WatchInterface > W...pp > Int...rd > Int...se) > Int...ne > Interface Controller > Lbl Right-Top

Interface Controller Scene

Interface Controller

- Lbl Left-Top
- Lbl Center-Top
- Lbl Right-Top**
- Lbl Center-Bottom

Main Entry Point

Glance Interface Contr...

Static Notification Inter...

Notification Controller...

Main →

Any Screen Size

Text Color: Default

Font: Body

Min Scale: 1

Alignment: Center

Lines: 1

View

Alpha: 1

Hidden:

Installed:

Position

Horizontal: Right

Vertical: Top

Size

Width: Size To Fit Content

Height: Size To Fit Content



apple WATCH APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 5