Creating Apps for Apple Watch

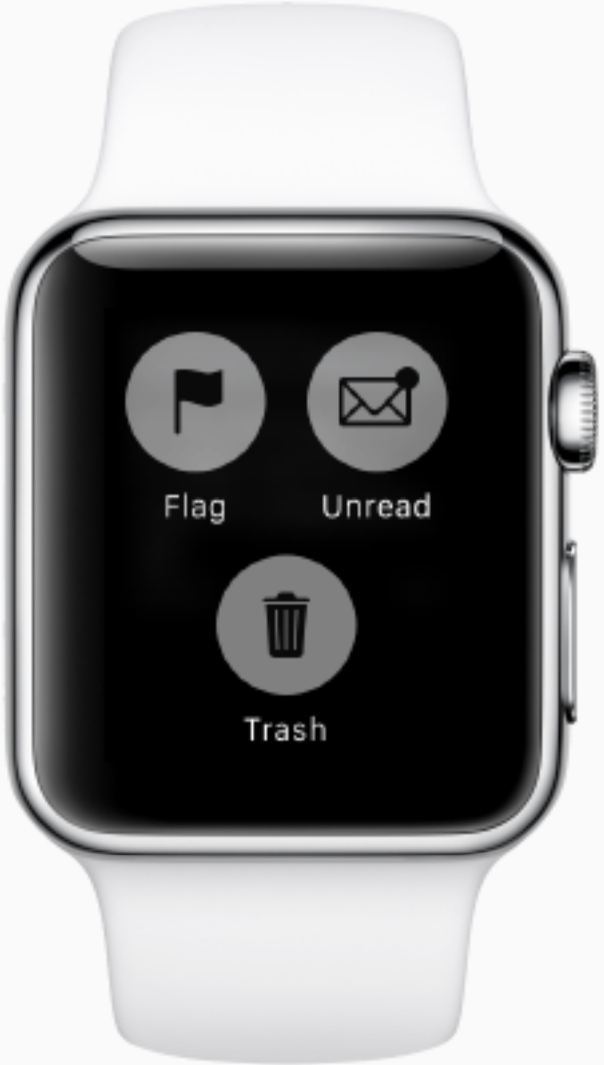
Josh Birnholz

watchOS Basics

- Built for quick (2 second) interactions
- Apple Watch app can’t exist without iOS app
    - Apps installed through the Watch app on the paired iPhone
- Many of the same frameworks…
- …but they can be limited
    - UIKit, but without UIView
- Test on a real device!

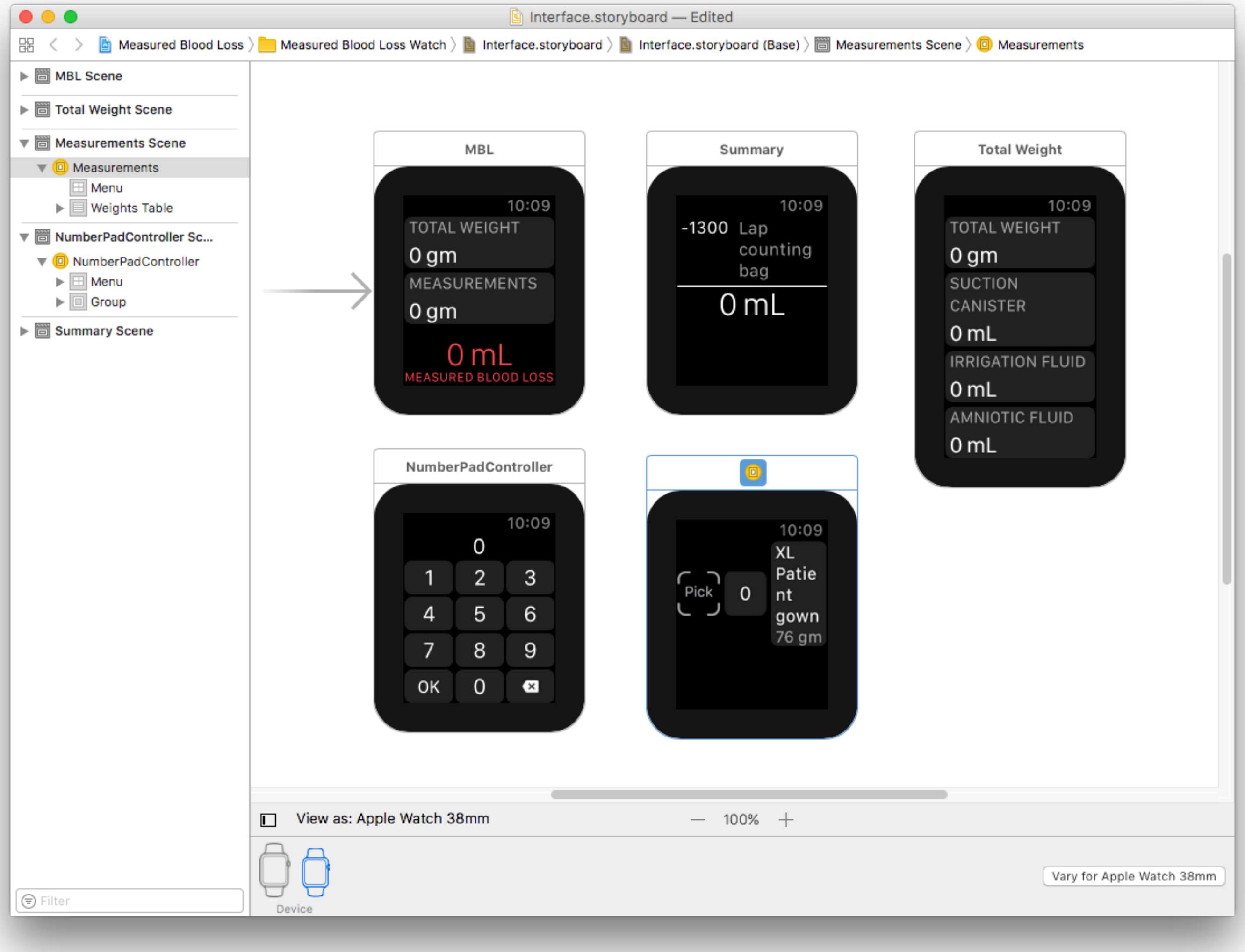
Apple Watch Interactions

- Touch
  - Buttons
  - Gesture Recognizers
  - Scroll
- Force Touch
  - Menus
  - No force touch API
- Digital Crown
  - Scroll
  - Pickers
  - API
- Haptic Feedback
- Accelerometer/Gyroscope

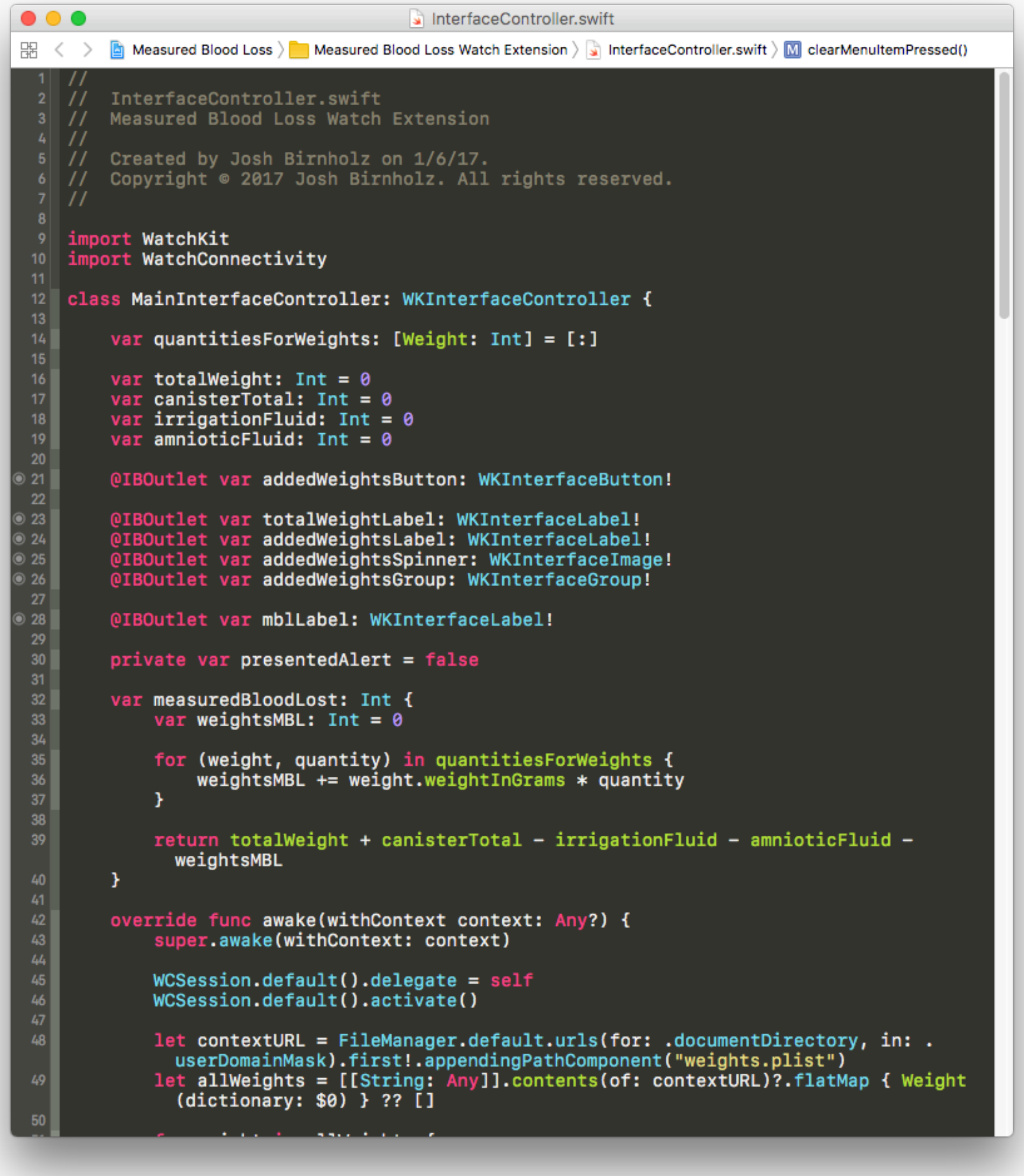


Parts of an Apple Watch App

WatchKit UI

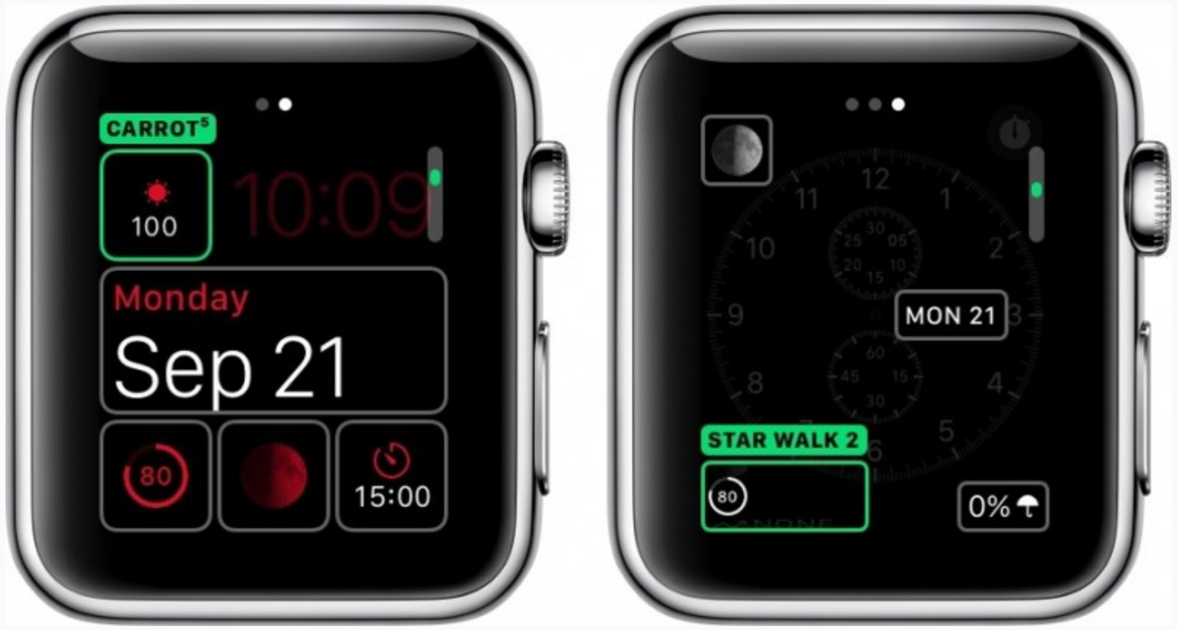


WatchKit Extension

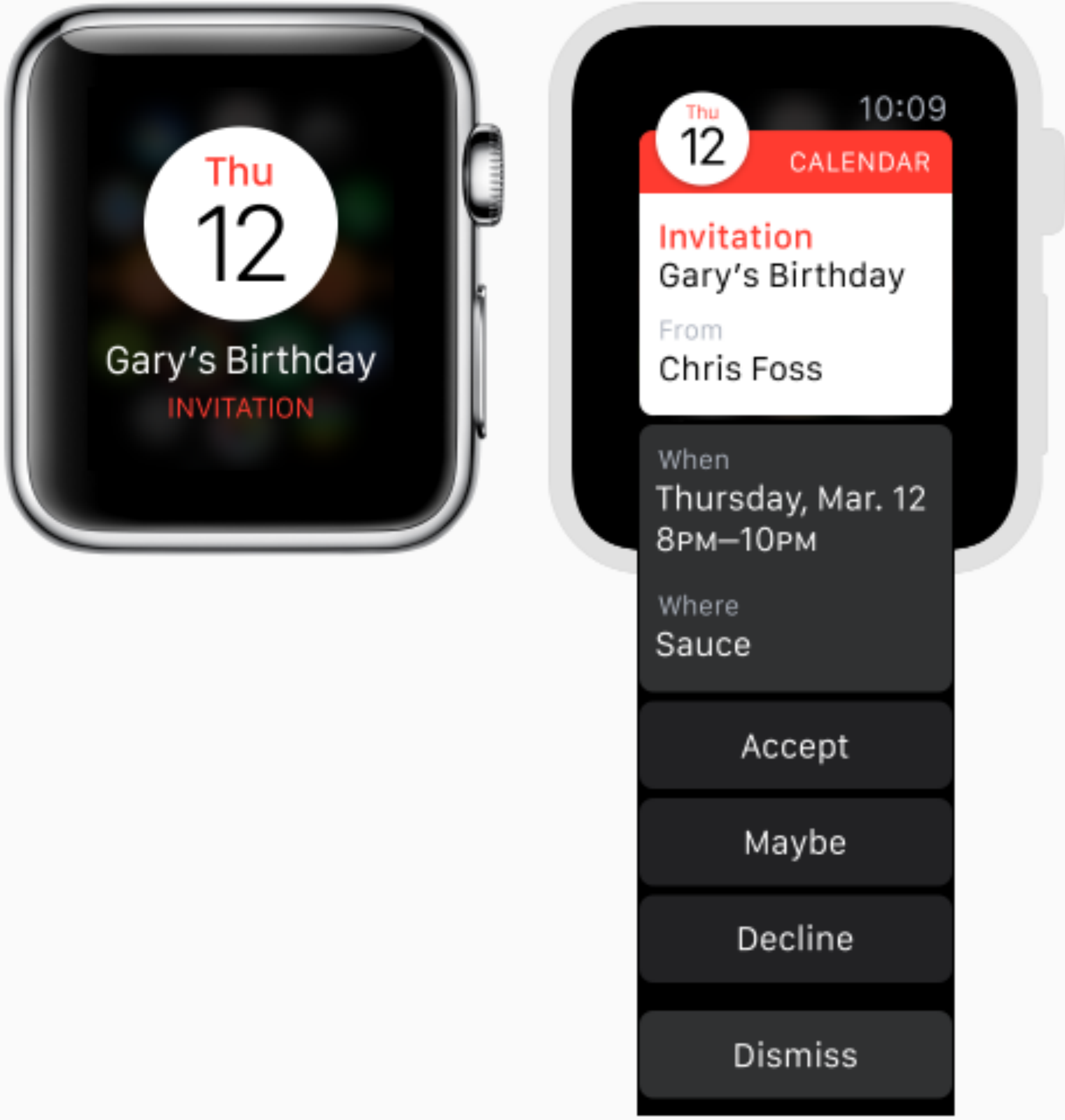

InterfaceController.swift
Measured Blood Loss 〉 Measured Blood Loss Watch Extension 〉 InterfaceController.swift 〉 M clearMenuItemPressed()

```swift
//
//  InterfaceController.swift
//  Measured Blood Loss Watch Extension
//
//  Created by Josh Birnholz on 1/6/17.
//  Copyright © 2017 Josh Birnholz. All rights reserved.
//

import WatchKit
import WatchConnectivity

class MainInterfaceController: WKInterfaceController {

    var quantitiesForWeights: [Weight: Int] = [:]

    var totalWeight: Int = 0
    var canisterTotal: Int = 0
    var irrigationFluid: Int = 0
    var amnioticFluid: Int = 0

    @IBOutlet var addedWeightsButton: WKInterfaceButton!

    @IBOutlet var totalWeightLabel: WKInterfaceLabel!
    @IBOutlet var addedWeightsLabel: WKInterfaceLabel!
    @IBOutlet var addedWeightsSpinner: WKInterfaceImage!
    @IBOutlet var addedWeightsGroup: WKInterfaceGroup!

    @IBOutlet var mblLabel: WKInterfaceLabel!

    private var presentedAlert = false

    var measuredBloodLost: Int {
        var weightsMBL: Int = 0

        for (weight, quantity) in quantitiesForWeights {
            weightsMBL += weight.weightInGrams * quantity
        }

        return totalWeight + canisterTotal - irrigationFluid - amnioticFluid -
          weightsMBL
    }

    override func awake(withContext context: Any?) {
        super.awake(withContext: context)

        WCSession.default().delegate = self
        WCSession.default().activate()

        let contextURL = FileManager.default.urls(for: .documentDirectory, in: .
          userDomainMask).first!.appendingPathComponent("weights.plist")
        let allWeights = [[String: Any]].contents(of: contextURL)?.flatMap { Weight
          (dictionary: $0) } ?? []

```

Complication



Notification


Thu
12
Gary's Birthday
INVITATION

10:09
Thu
12
CALENDAR
Invitation
Gary's Birthday
From
Chris Foss
When
Thursday, Mar. 12
8PM–10PM
Where
Sauce

WatchConnectivity:
WCSession & WCSessionDelegate

Conforming to WCSessionDelegate:

```
func session(_ session: WCSession, activationDidCompleteWith
activationState: WCSessionActivationState, error: Error?)

func sessionDidBecomeInactive(_ session: WCSession) (iOS only)

func sessionDidDeactivate(_ session: WCSession) (iOS only)
```

Activating a session:

```
if WCSession.isSupported() && WCSession.isReachable() {
    let session = WCSession.default()
    session.delegate = self
    session.activate()
}
```

Using a session:

```
if session.activationState == .activated && session.isPaired &&
  session.isWatchAppInstalled {
    ...
}
```

Ways of Communicating

- Application Context
    - Sends a dictionary of information (plist types)
    - Only the newest information is kept
    - Old data is overwritten

Ways of Communicating

- Transfer User Info
  - Sends a dictionary of information (plist types)
  - All information is kept
- Transfer File
  - Supply the URL of the file to transfer
  - The counterpart get a callback with the local URL of the file after it has been transferred

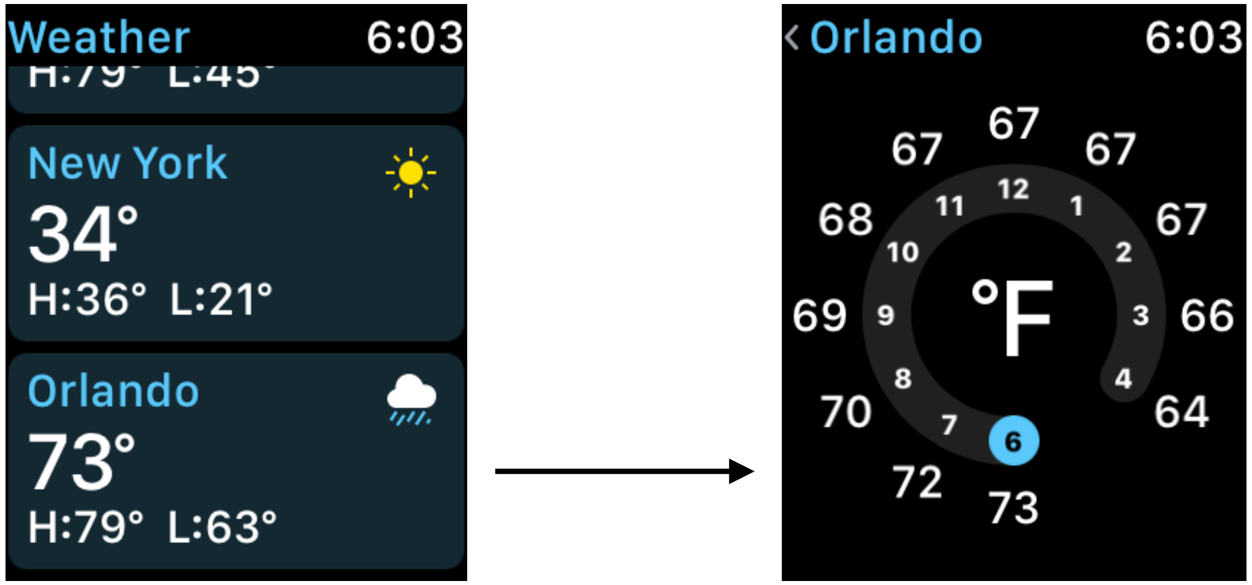
Ways of Communicating

- Send Message
  - Send a message to the counterpart, supplying the message, a reply handler, and an error handler
  - Kinds of messages:
    - Dictionary (plist types)
    - Data

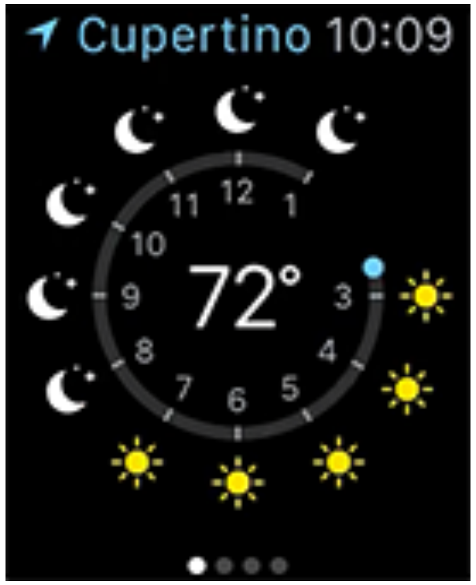
Meet `WKInterfaceController`

- Part of WatchKit
- Instances never created directly
  - Instead, they are pushed or presented via storyboard
- Root `WKInterfaceGroup` contains other elements
- `context` property
- Title

Two Navigation Styles



Hierarchical



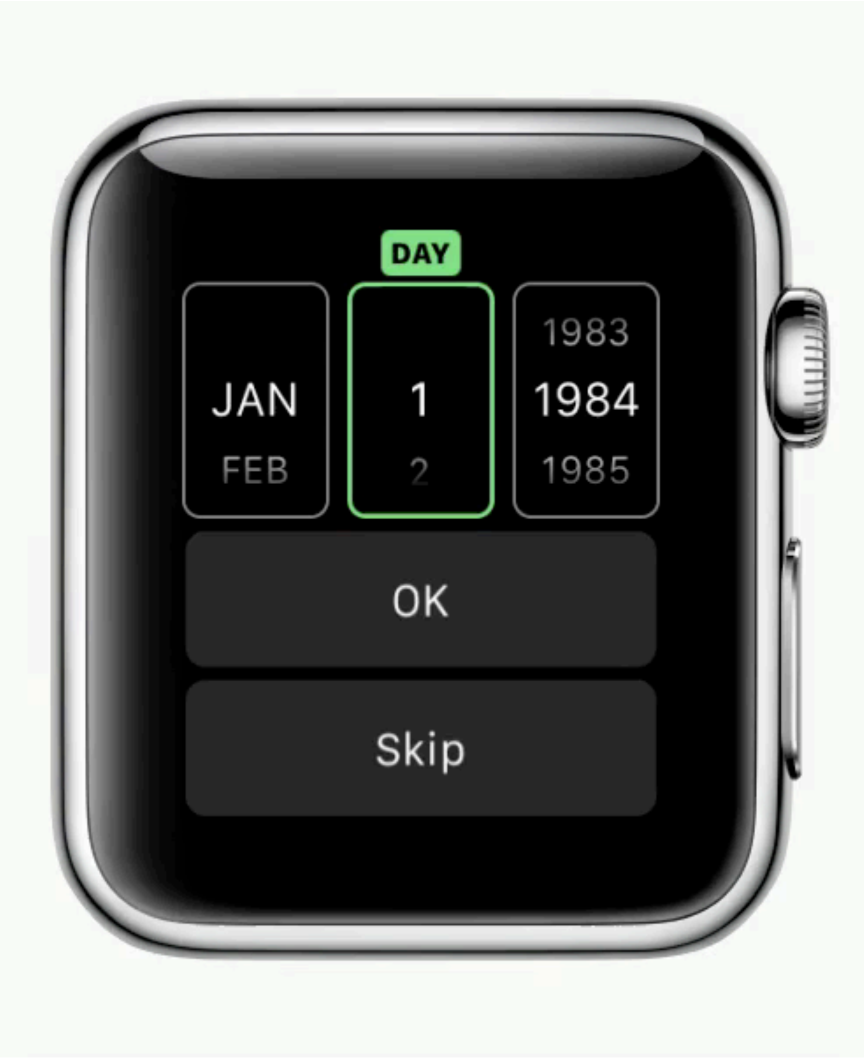
Page-based

Can’t mix and match!

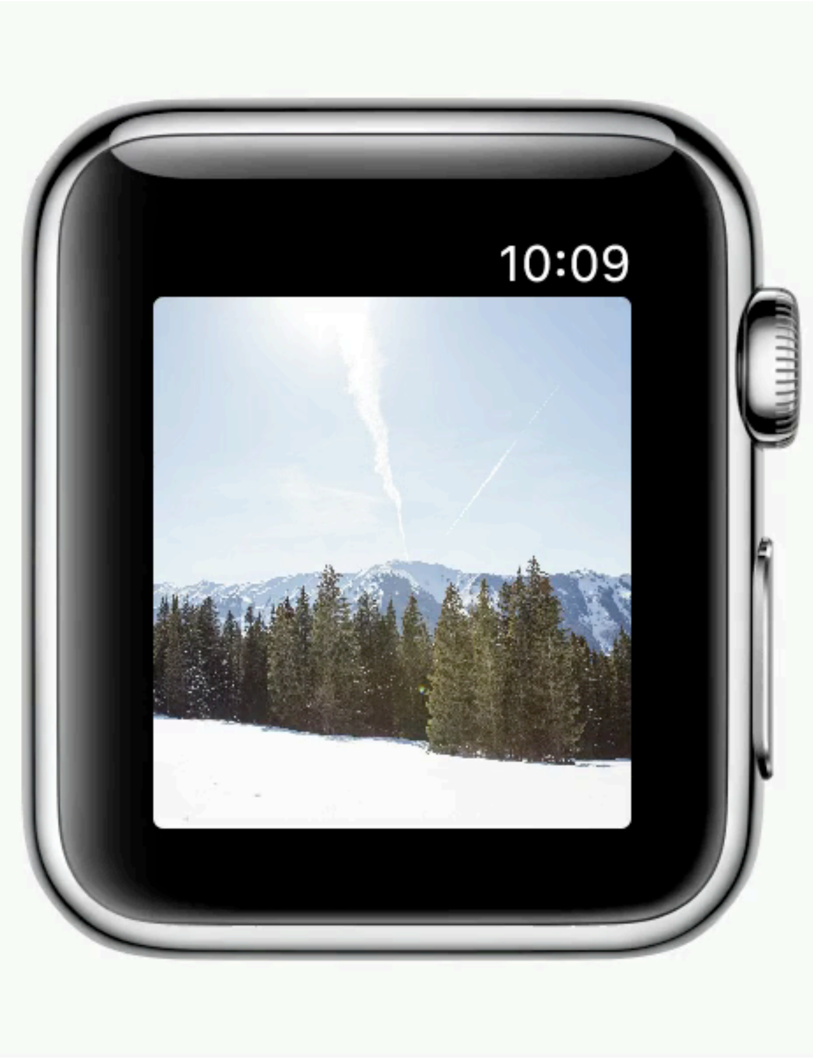
Presenting Interface Controllers

- Standard system controllers:
  - Alerts
  - Text input (speech/scribble)
  - Video/audio playback
  - Audio recording
  - PassKit
- Custom controllers:
  - Storyboard segue
  - `pushController(withName:context:)`
  - `presentController(withName:context:)`
  - `presentController(withNames:contexts:)`

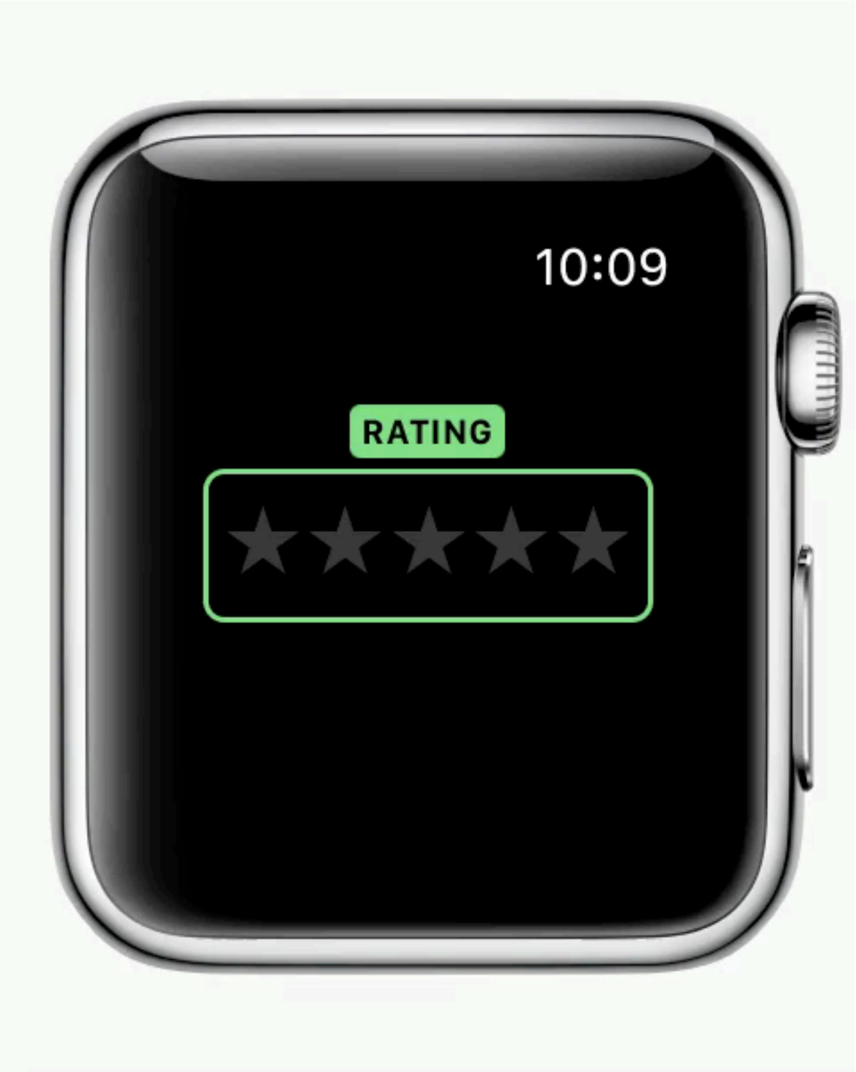
WKInterfacePicker



List



Stack



Sequence