Introduction to WatchKit

CS193W - Spring 2016 - Lecture 1

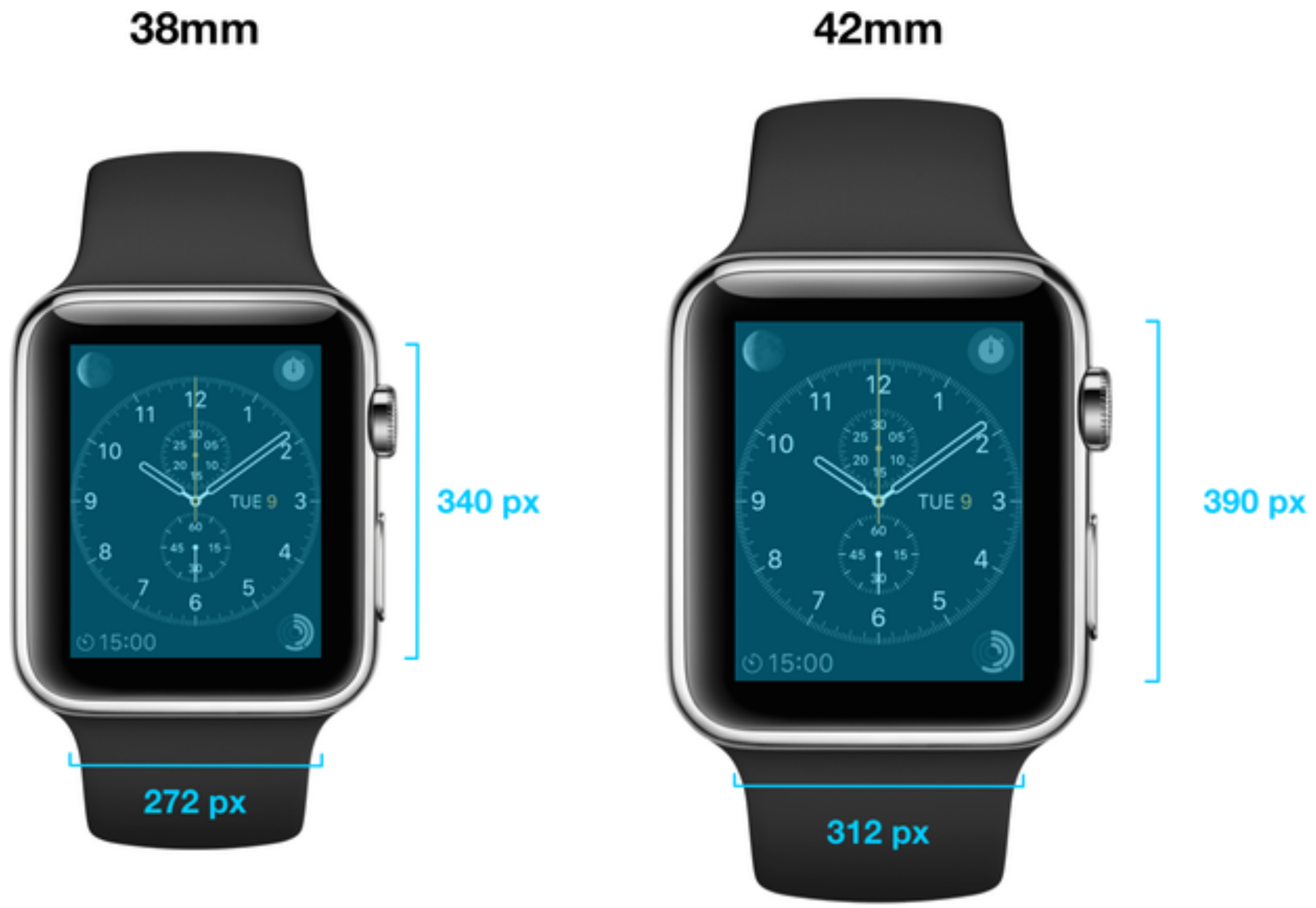
WATCH

Released April 24, 2015

No updates to the hardware yet.

Three collections, over 30 models…

Two sizes

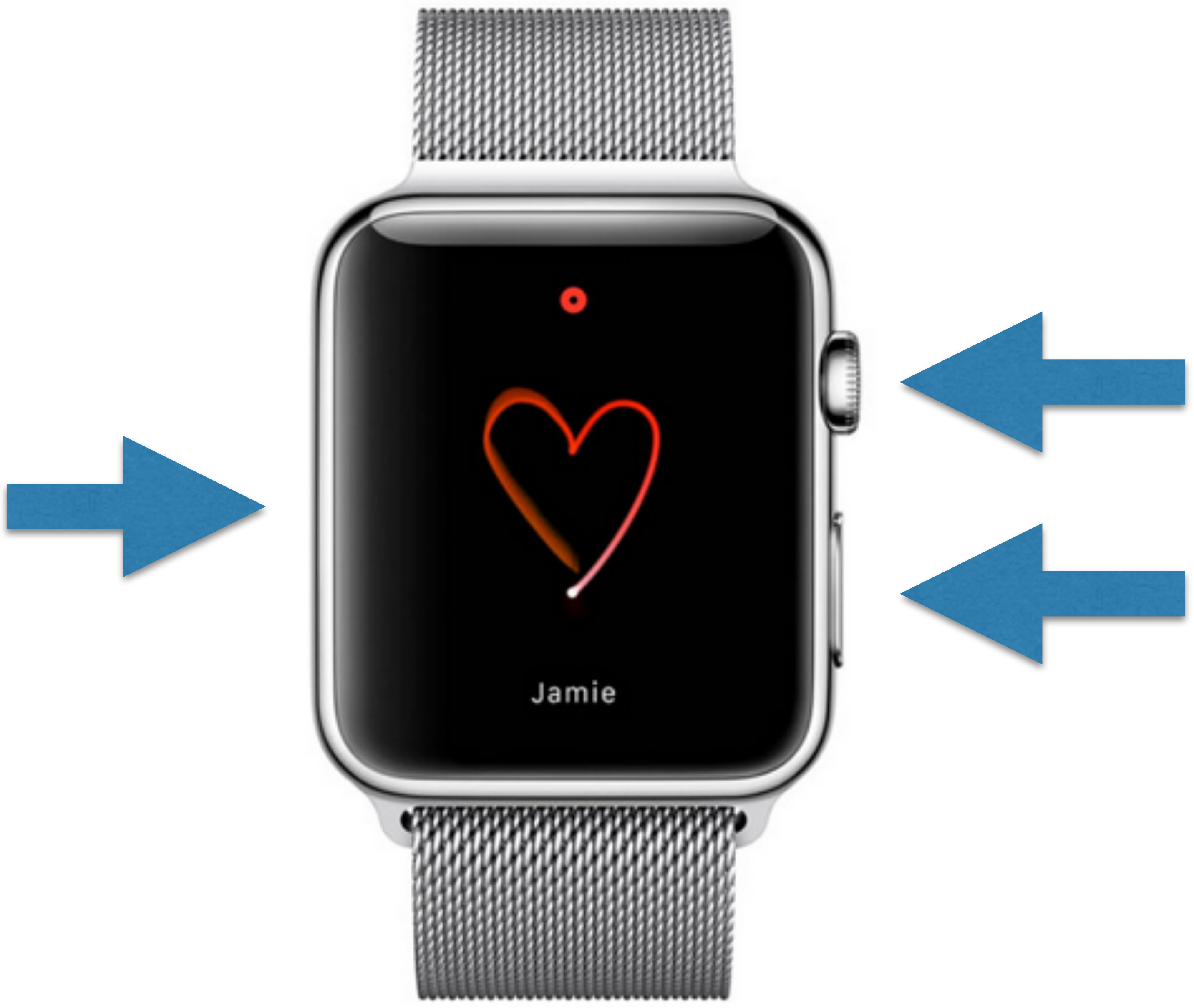


The Screen

- OLED (organic light-emitting diode) screen.
- A black pixel means the pixel is off, i.e. not consuming power.

Physical Controls

Screen with Force Touch sensors



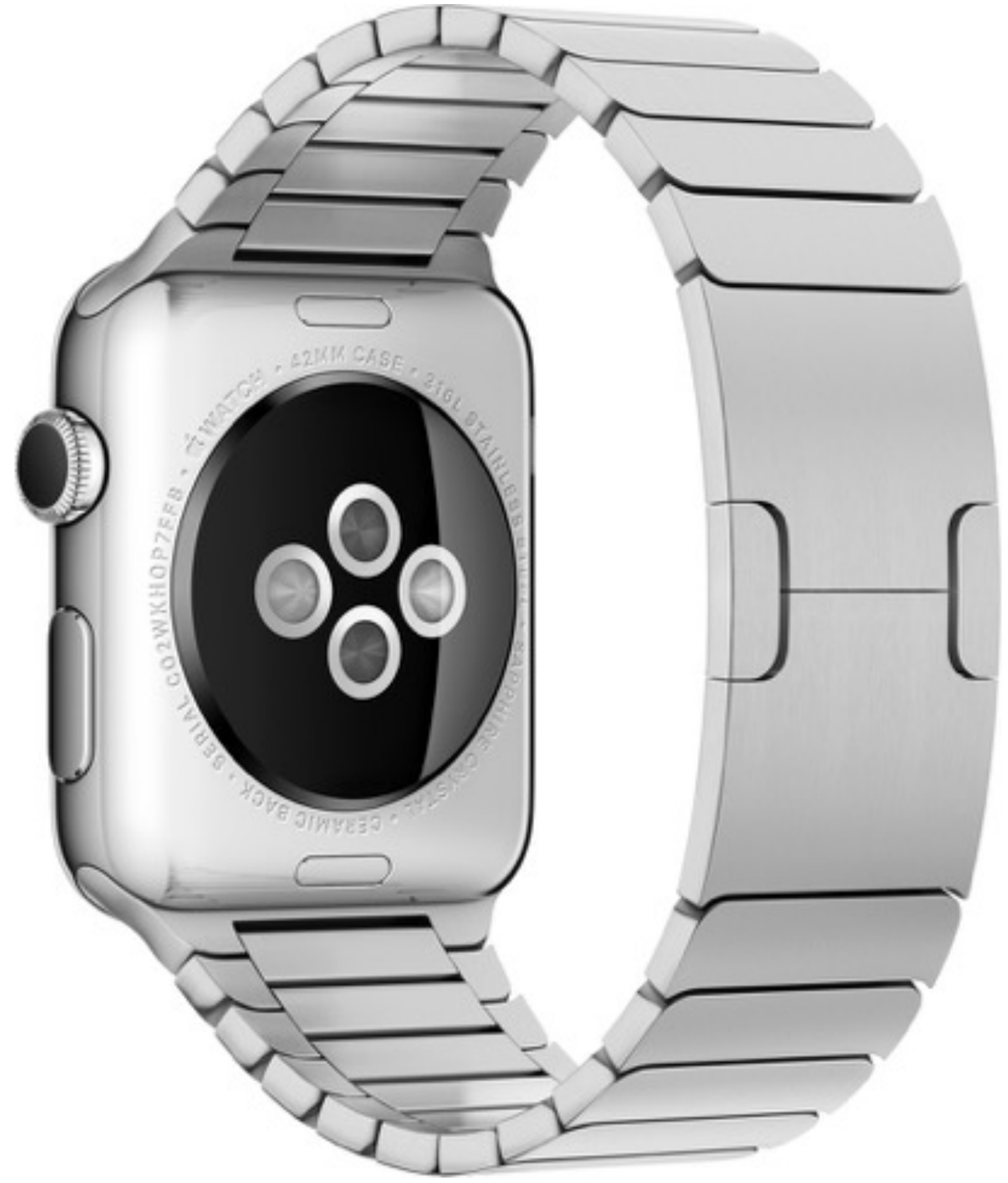
Digital Crown

“The Button”

On the Flip Side

Heart Rate Monitor

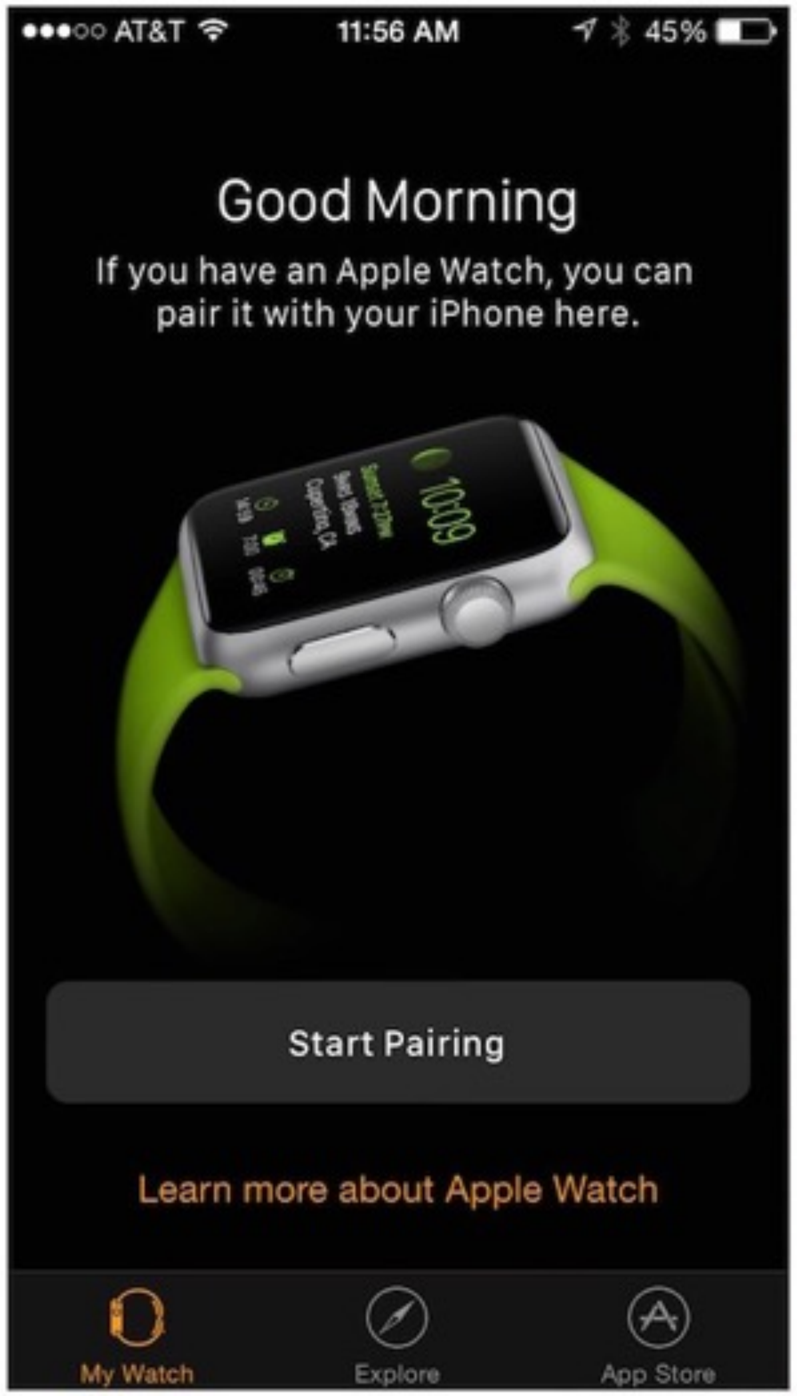
Taptic Engine


WATCH · 42MM CASE · 316L STAINLESS STEEL · SAPPHIRE CRYSTAL · CERAMIC BACK · SERIAL CO2WKHOP7EF8

Managing Apple Watch apps

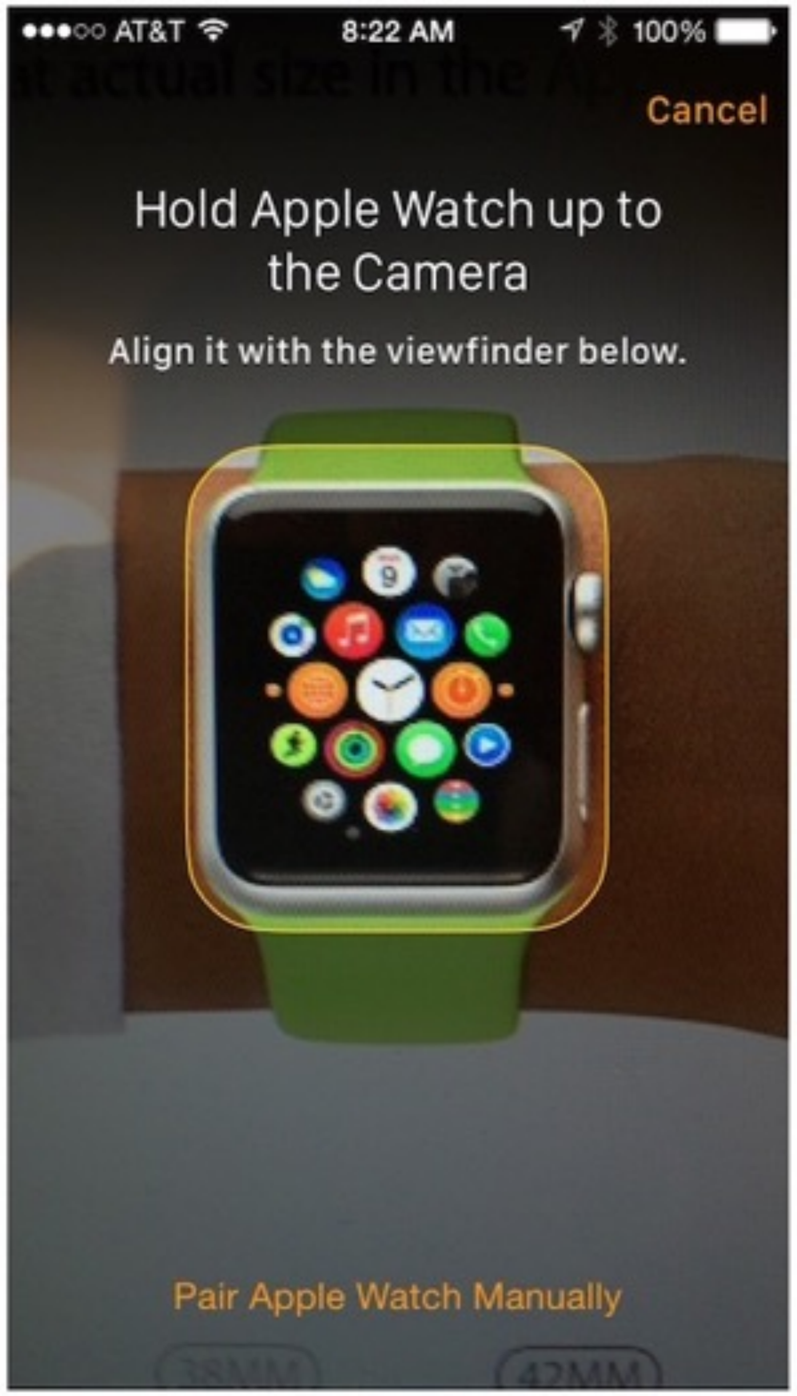
- Not a standalone product. Requires iPhone accessible via Bluetooth or Wifi
- Users with WatchKit-enabled iPhone apps are prompted on their phone to install Apple Watch apps
- The Apple Watch companion app allows you to configure what is on your watch

Pairing Your Apple Watch



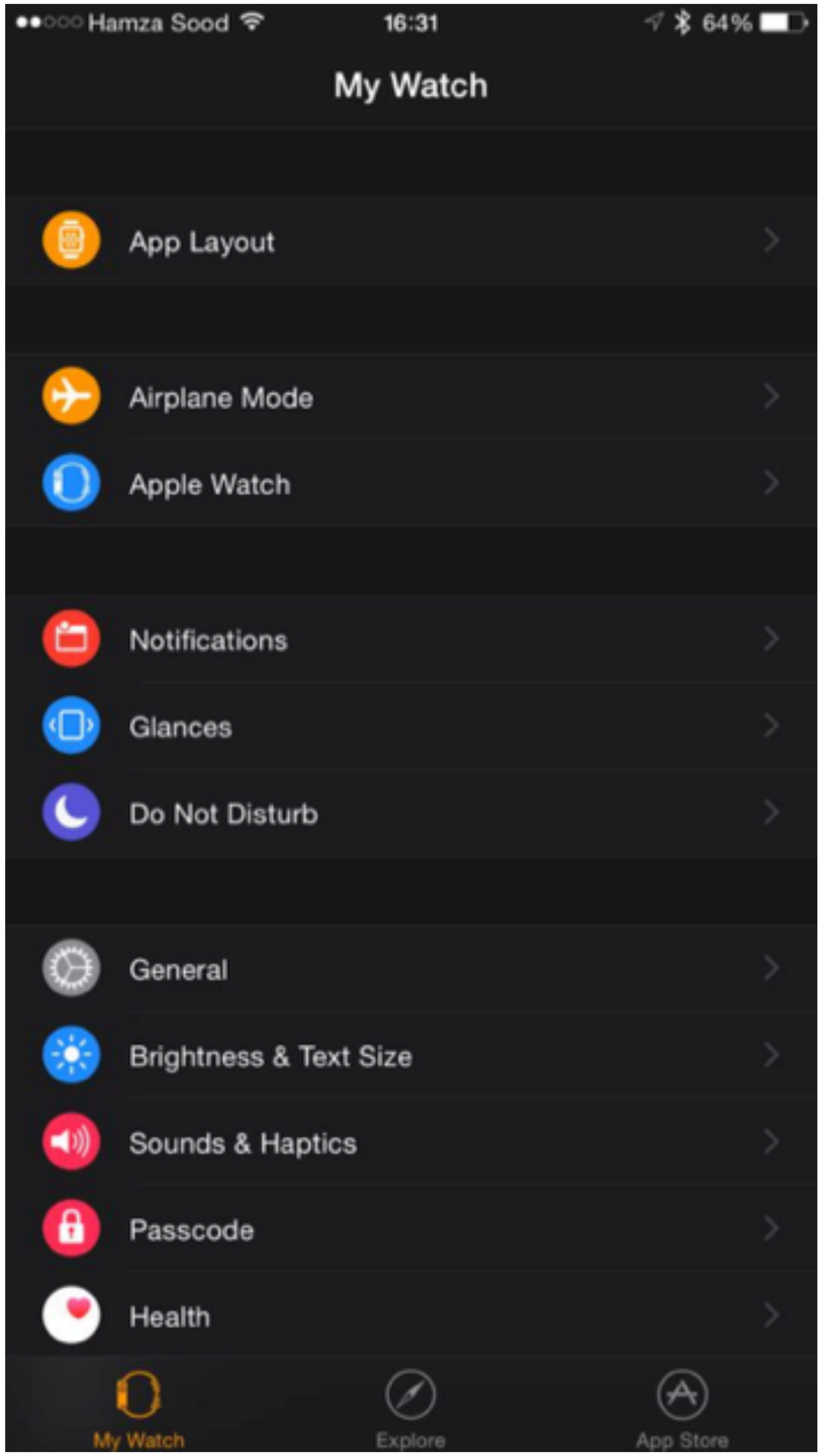
Good Morning

If you have an Apple Watch, you can pair it with your iPhone here.



Hold Apple Watch up to the Camera

Align it with the viewfinder below.


●●○○○ Hamza Sood 16:31 64%

My Watch

App Layout

Airplane Mode

Apple Watch

Notifications

Glances

Do Not Disturb

General

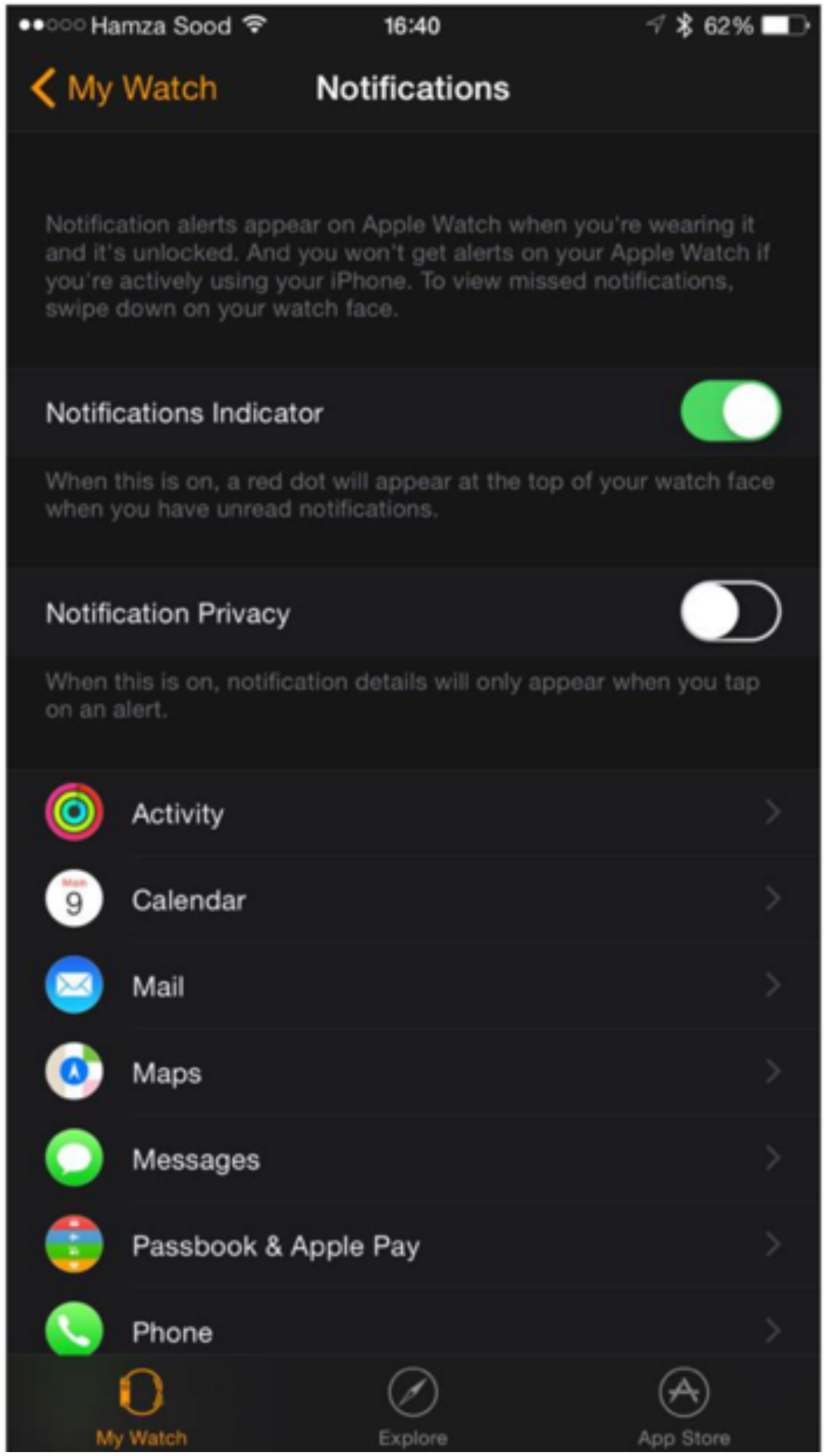
Brightness & Text Size

Sounds & Haptics

Passcode

Health

My Watch Explore App Store


●●○○○ Hamza Sood 16:40 62%

My Watch Notifications

Notification alerts appear on Apple Watch when you're wearing it and it's unlocked. And you won't get alerts on your Apple Watch if you're actively using your iPhone. To view missed notifications, swipe down on your watch face.

Notifications Indicator: On

When this is on, a red dot will appear at the top of your watch face when you have unread notifications.

Notification Privacy: Off

When this is on, notification details will only appear when you tap on an alert.

Activity

Calendar

Mail

Maps

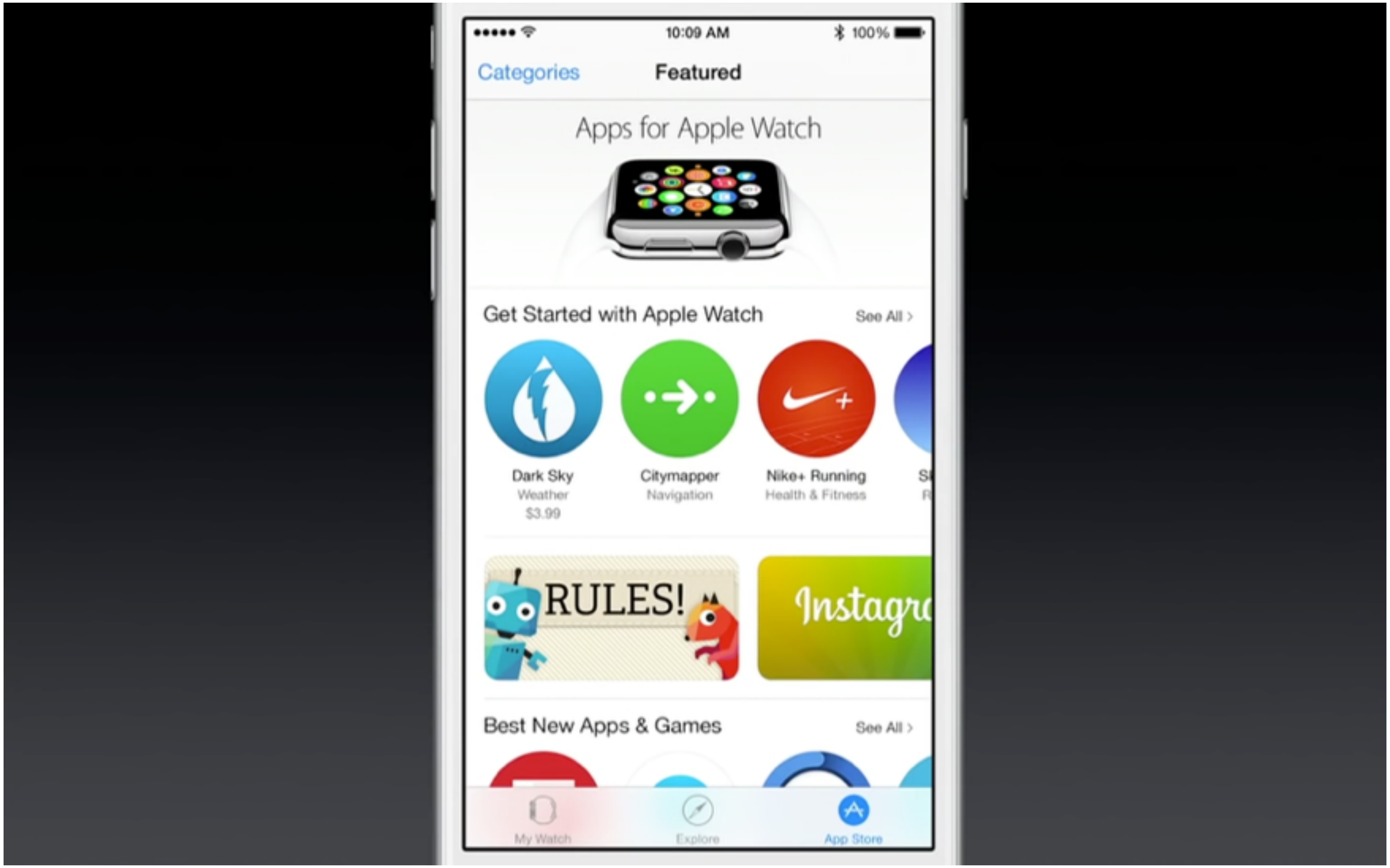
Messages

Passbook & Apple Pay

Phone

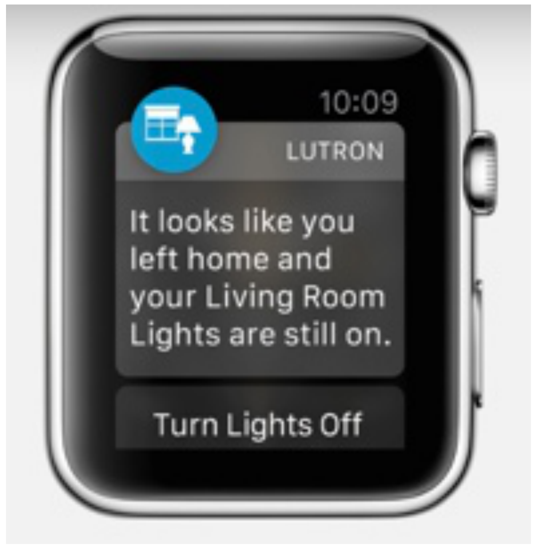
My Watch Explore App Store

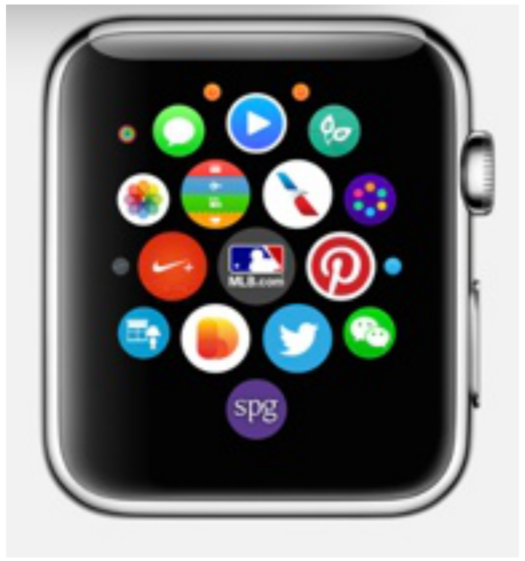
Apple Watch App Store

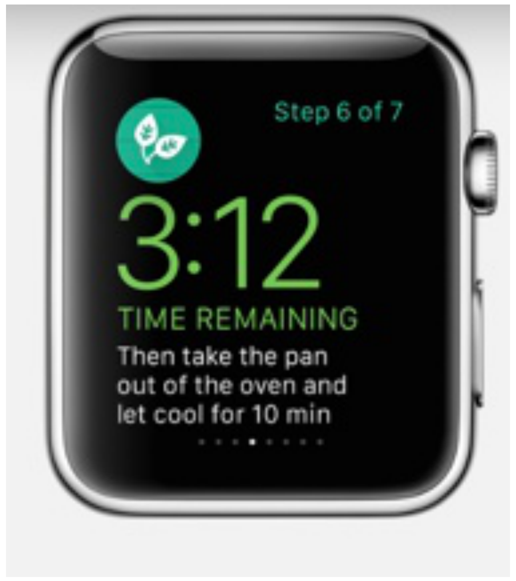


Interfaces

Glances


10:09
LUTRON
It looks like you left home and your Living Room Lights are still on.
Turn Lights Off

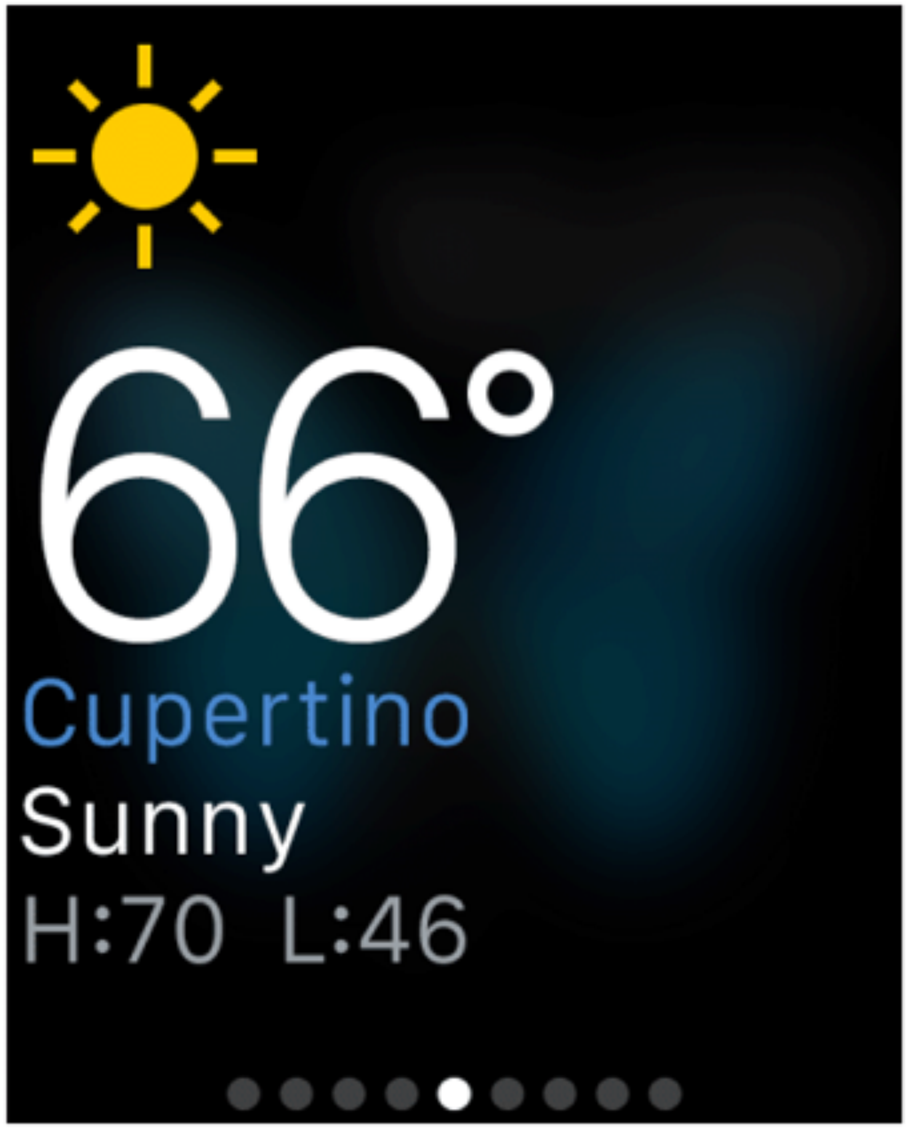


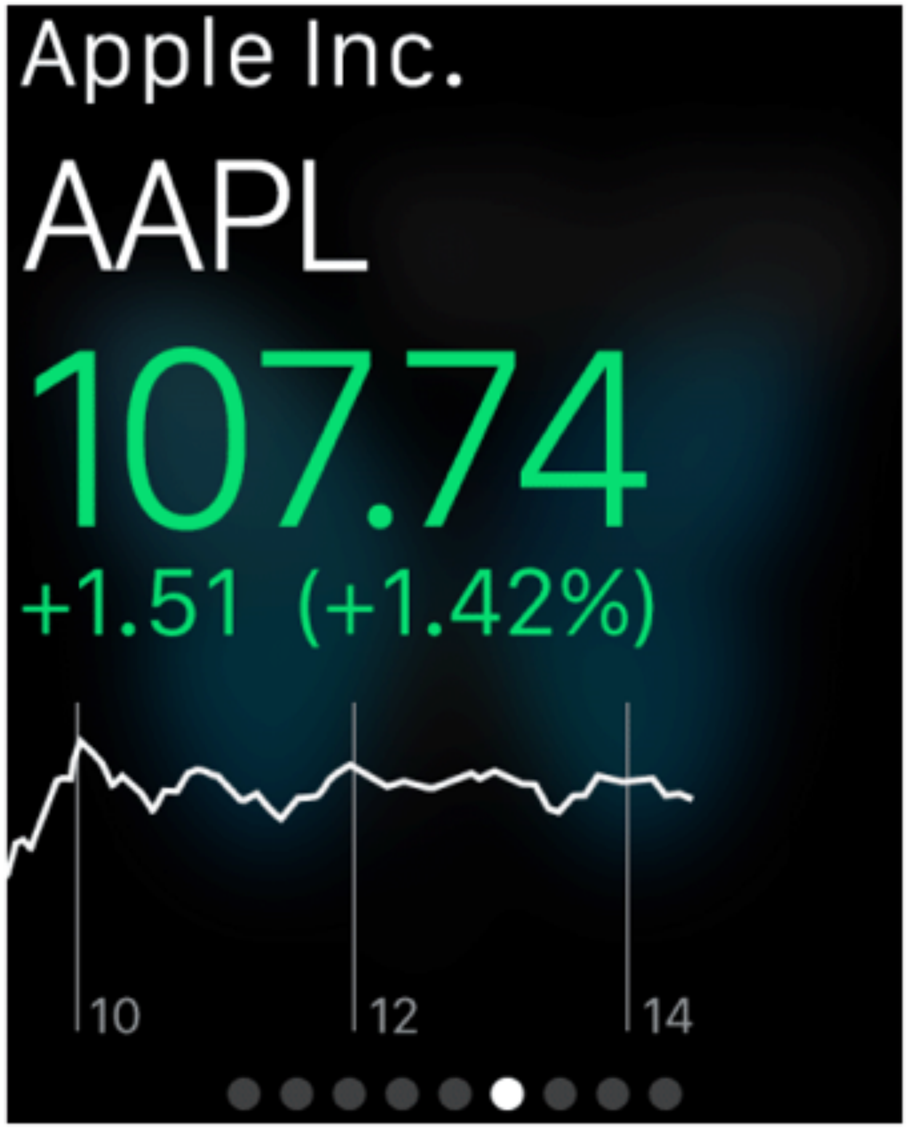


Apps

Notifications

Glances

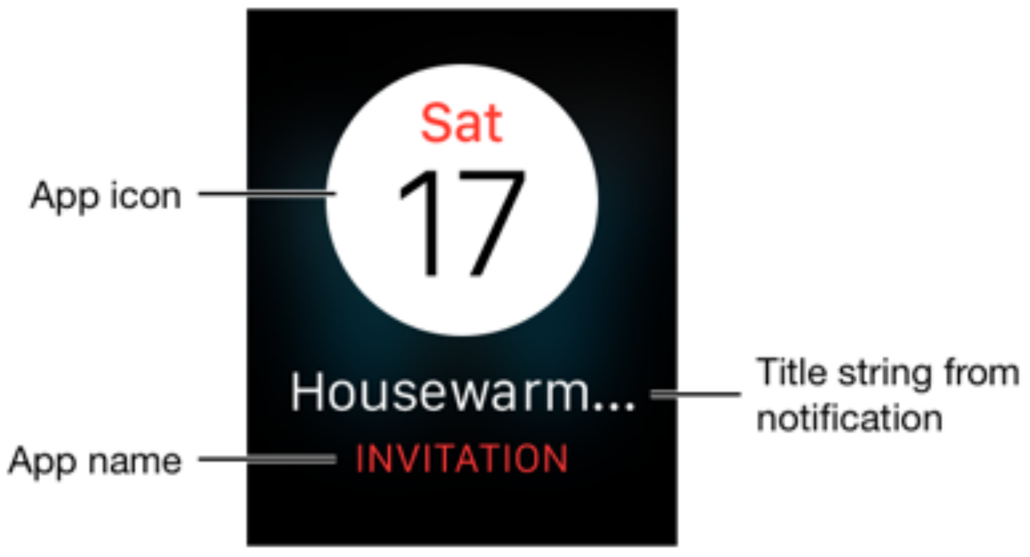




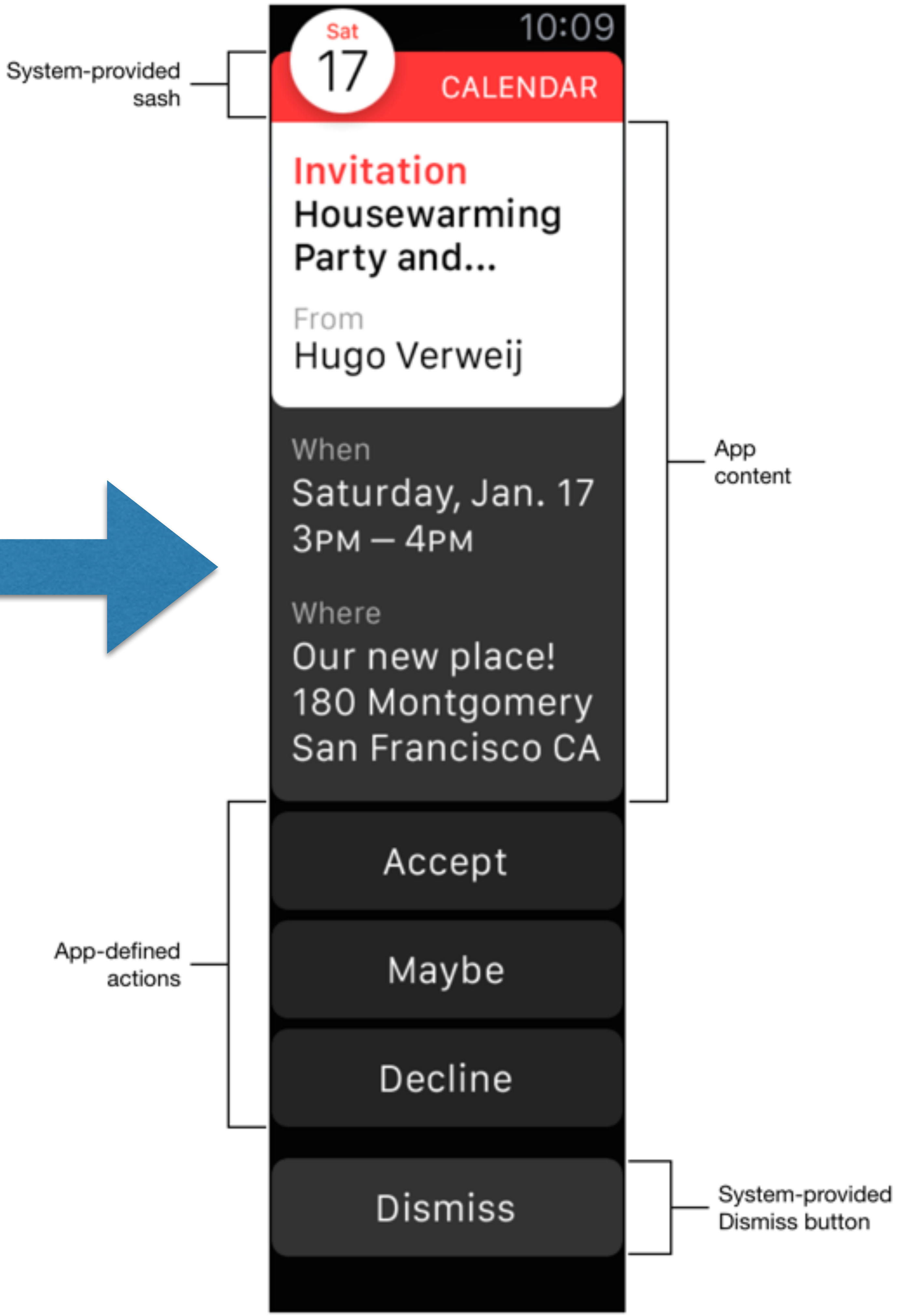
Screen-sized, static, at-a-glance info. One per app.

Notifications

Short
Look



Long
Look



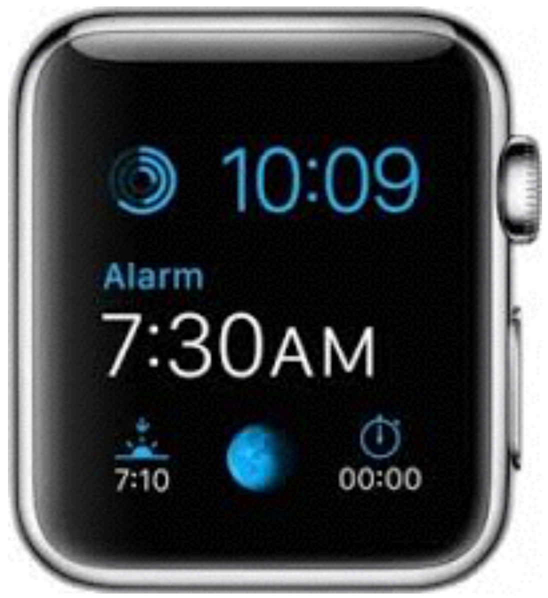
Invitation
Housewarming Party and...

From
Hugo Verweij

When
Saturday, Jan. 17
3PM – 4PM

Where
Our new place!
180 Montgomery
San Francisco CA

Complications



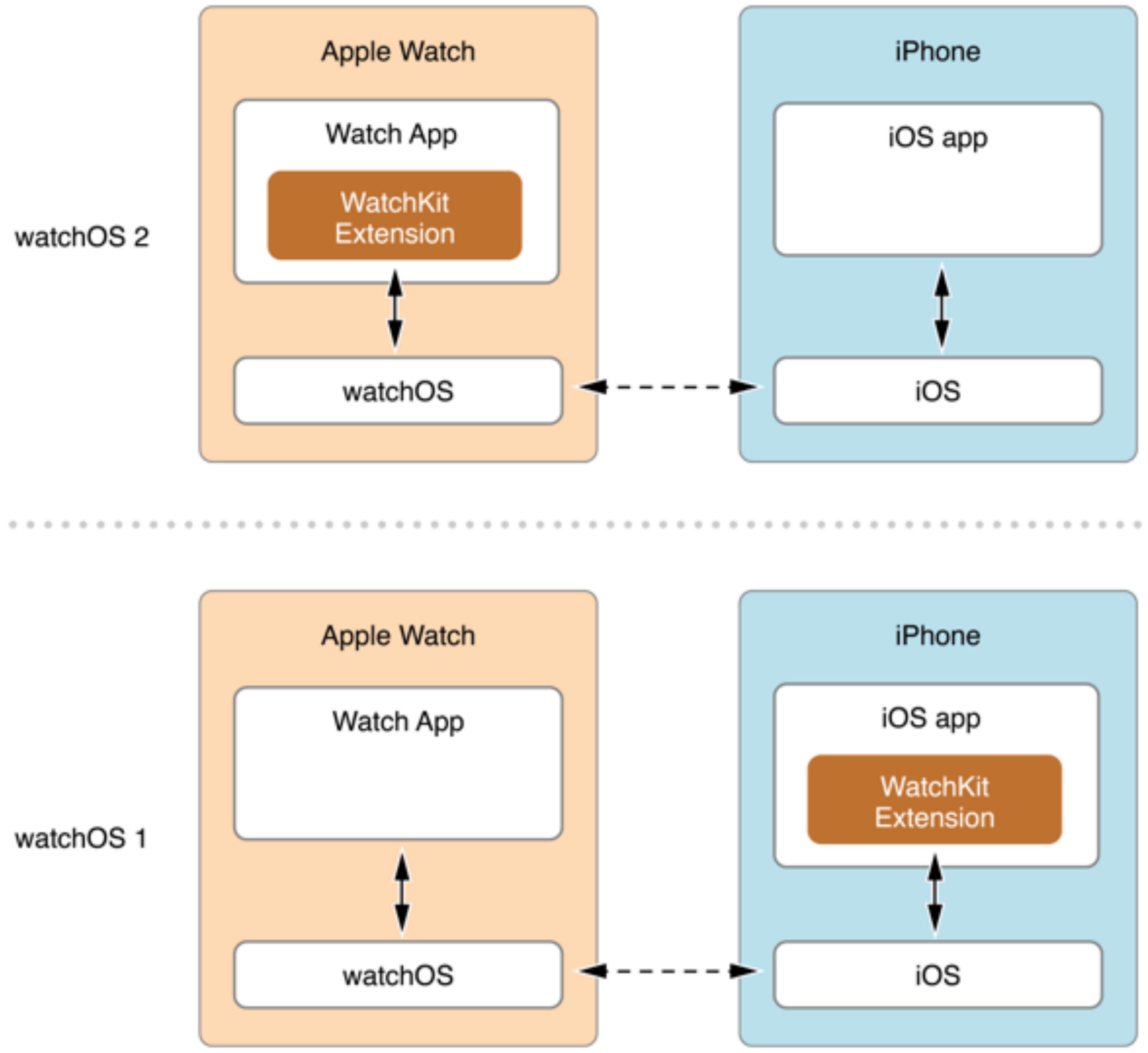
WatchKit

- Apple’s framework to create apps for the WATCH
- Included in iOS 8.2
- Allows you to create apps that are *extensions* of (and embedded in) iPhone apps.
- Most classes start with WKInterface, i.e. WKInterfaceLabel.

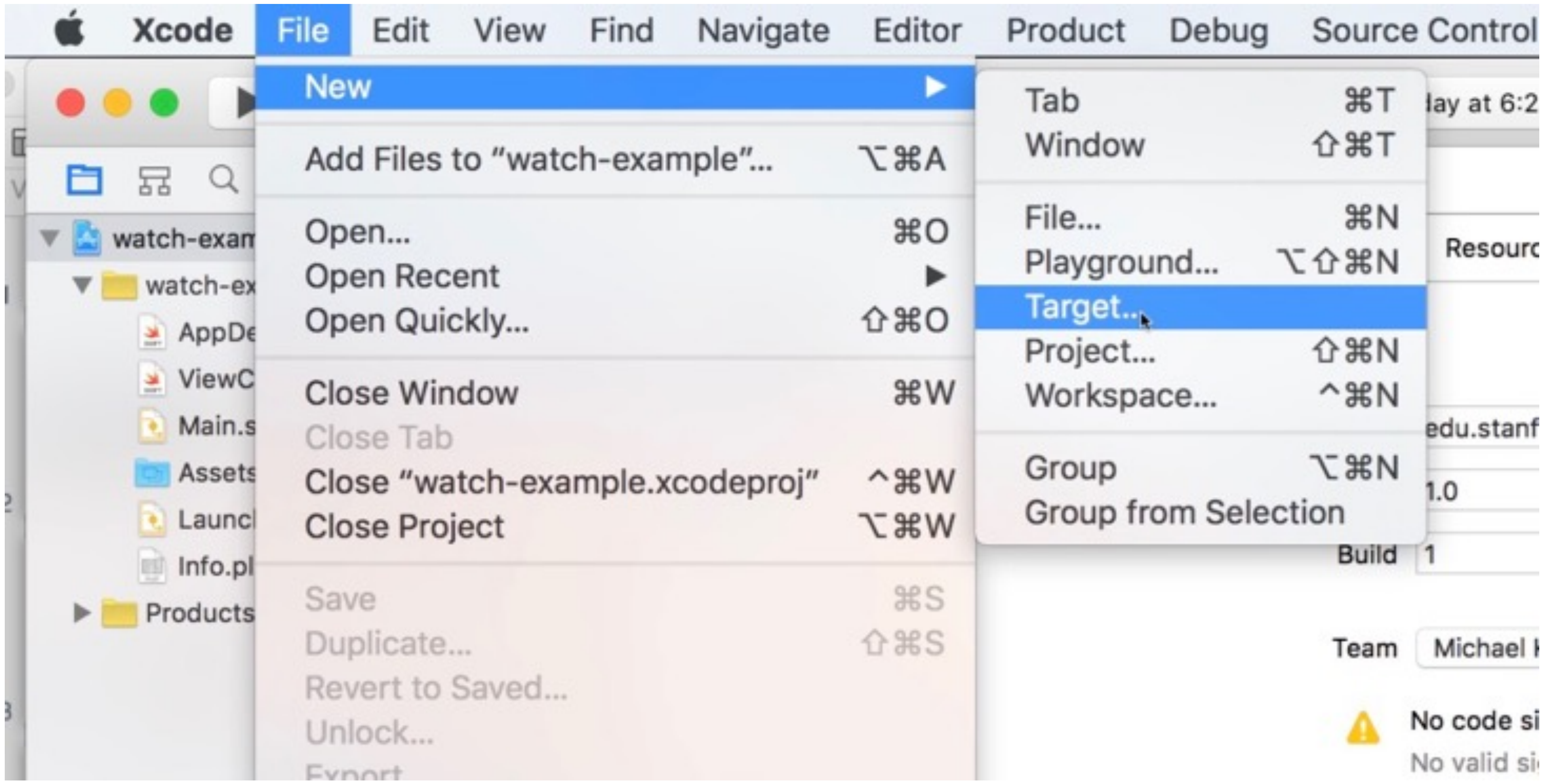
watchOS 2

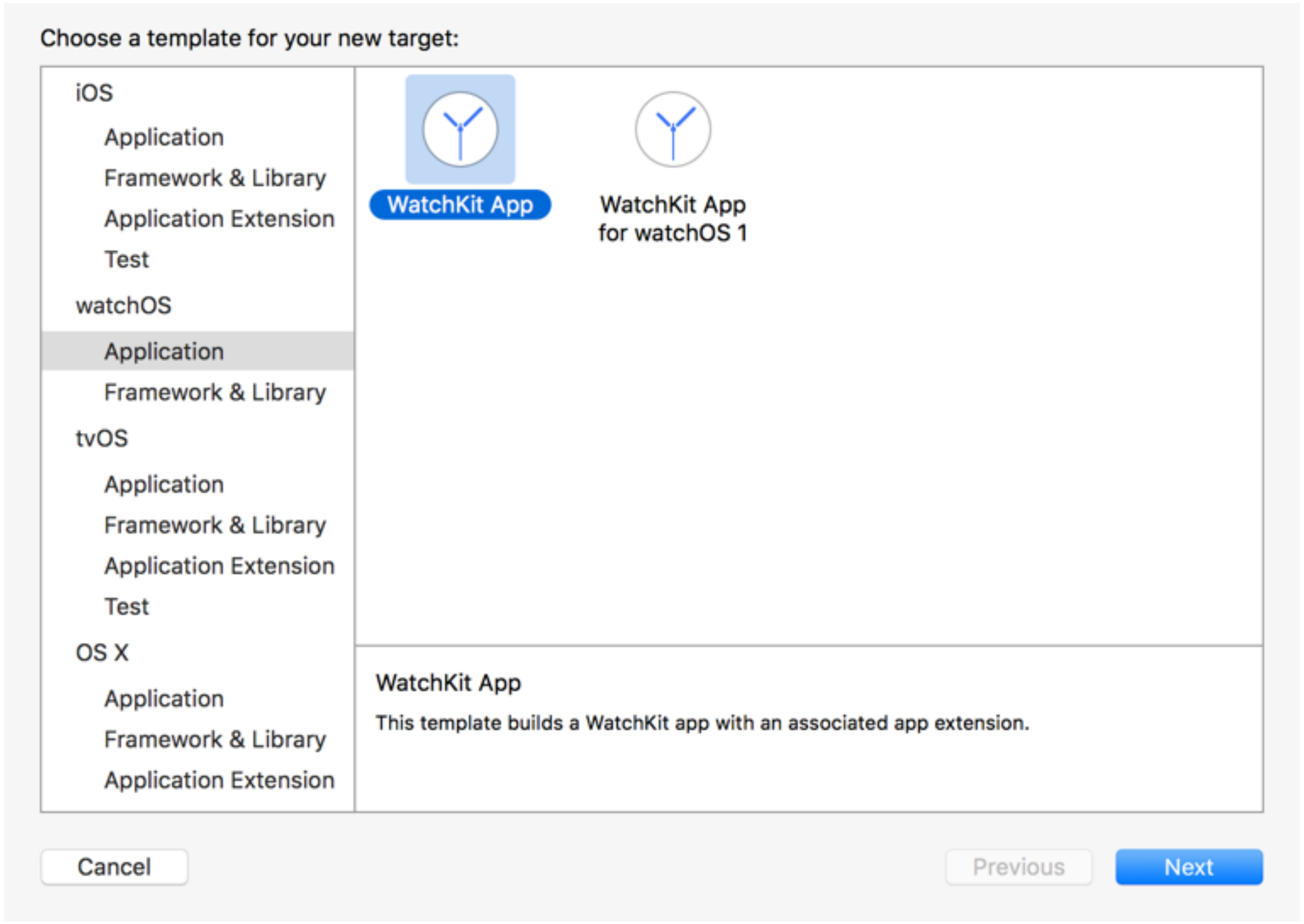
- A *major* update
- App extension runs on the watch, not the phone
- Watch connectivity framework
- Ability to create complications
- More access to hardware - accelerometer, heart rate monitor, haptic feedback

watchOS 1 vs watchOS 2



Creating a WatchKit App

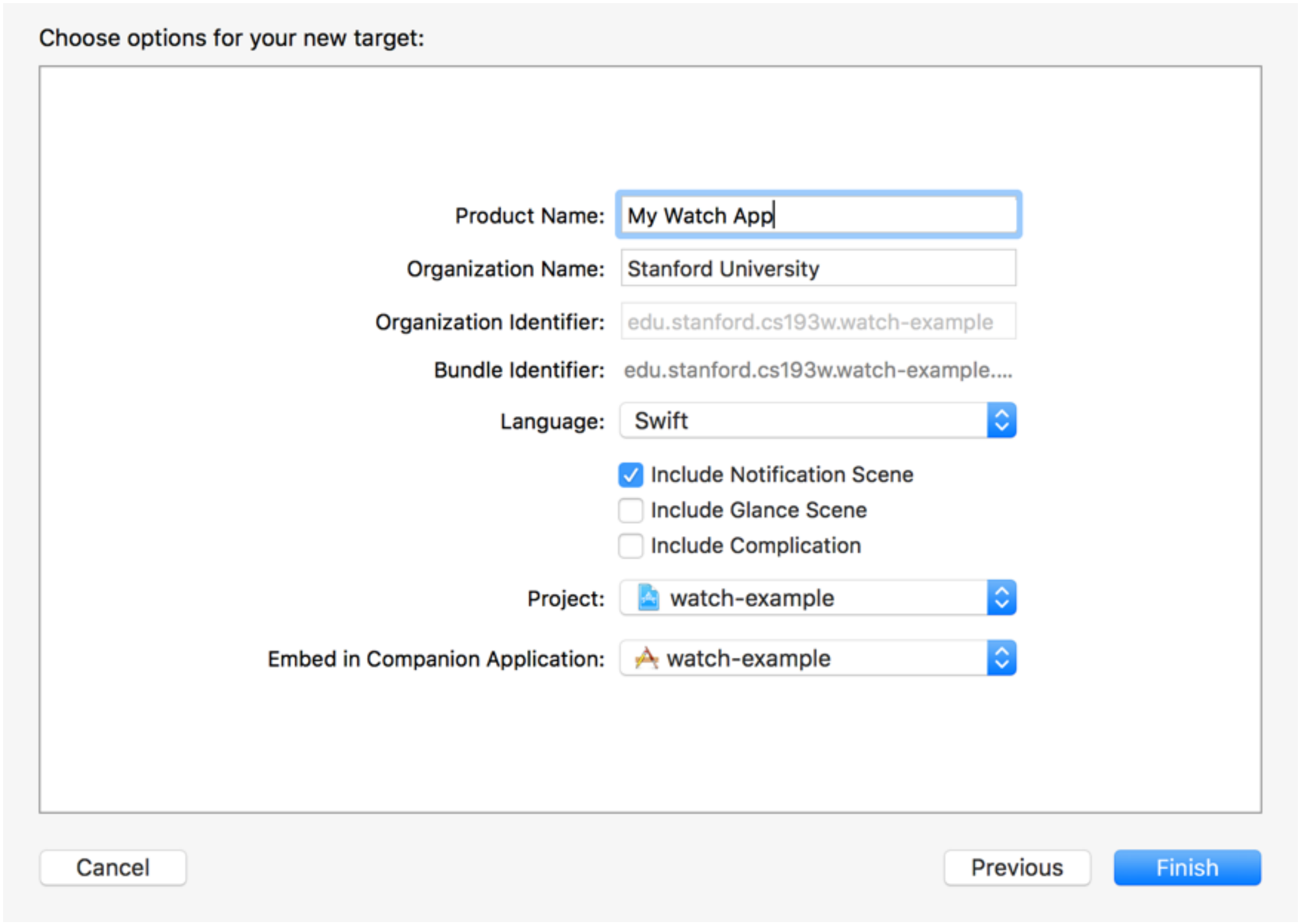




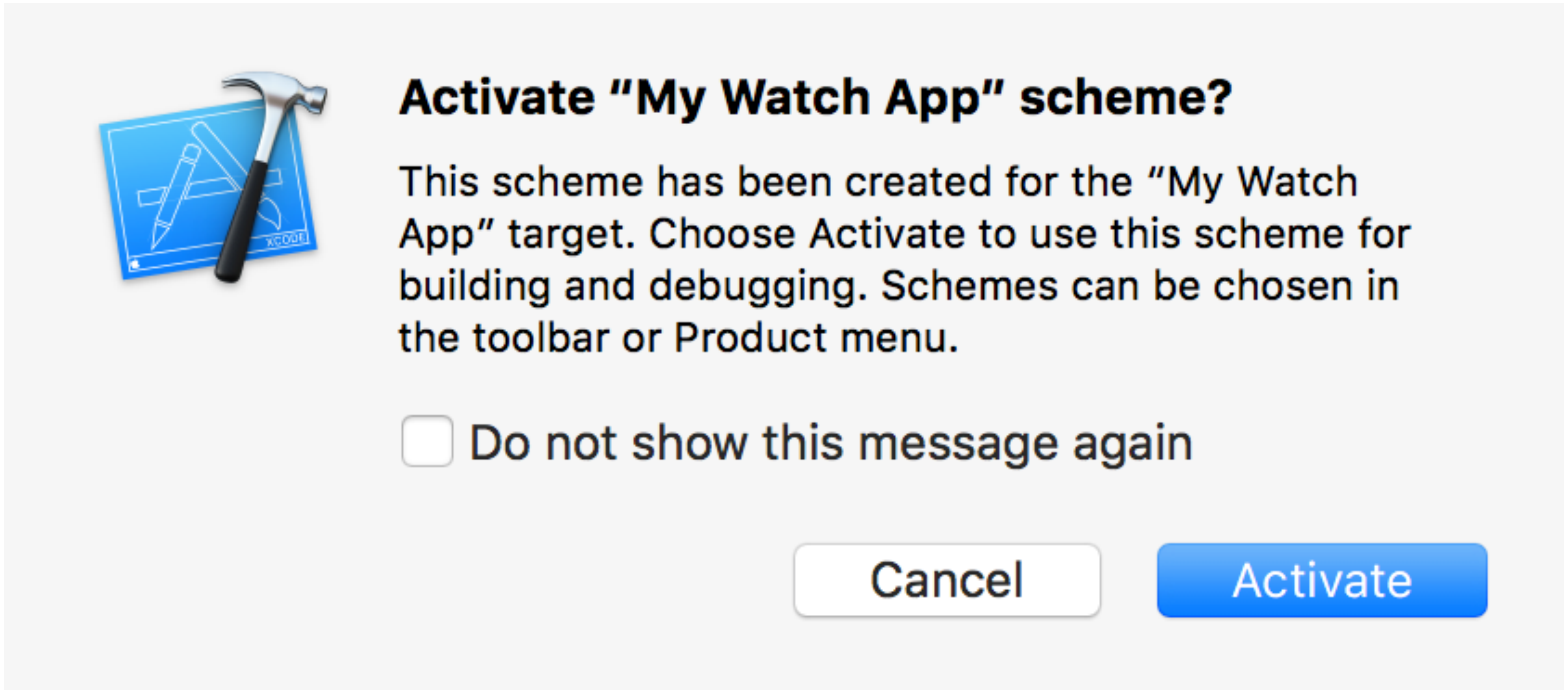
Choose a template for your new target:

WatchKit App

This template builds a WatchKit app with an associated app extension.


Choose options for your new target:

Product Name:	My Watch App
Organization Name:	Stanford University
Organization Identifier:	edu.stanford.cs193w.watch-example
Bundle Identifier:	edu.stanford.cs193w.watch-example....
Language:	Swift
	[✓] Include Notification Scene
	[ ] Include Glance Scene
	[ ] Include Complication
Project:	watch-example
Embed in Companion Application:	watch-example



Activate “My Watch App” scheme?

This scheme has been created for the “My Watch App” target. Choose Activate to use this scheme for building and debugging. Schemes can be chosen in the toolbar or Product menu.

☐ Do not show this message again

WatchKit App Scheme

- Xcode will generate a new scheme for you, which will allow you to run the WatchKit app on the simulator
- You can run the iPhone app at the same time, and debug both at the same time!

Two Targets

- Watch App
    - Contains the storyboard
- Watch Extension
    - Contains the code

- Both can contain assets

The Storyboard

- All interface objects for an interface are specified in the storyboard
- You cannot use alloc/init to create new interface objects

Layout

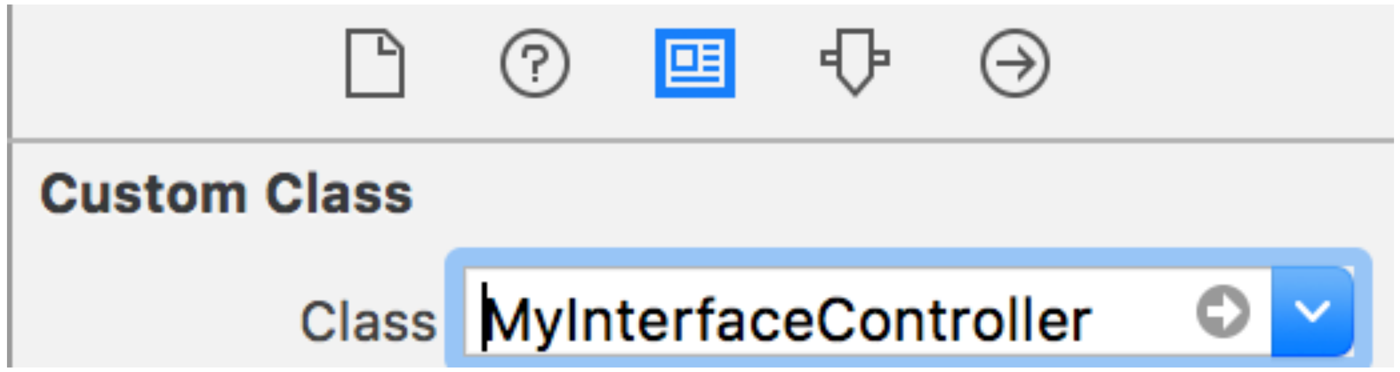
- Unlike UIViews, you cannot access the frame of an interface element
- Elements laid out top-to-bottom
- Hiding an element closes the gap between elements

WKInterfaceController

- Analogous to UIViewController
- Manage the content in a storyboard scene
  - Assign values to interface controls
  - Respond to user inputs (target-action)
  - Change appearance of interface controls

Specifying the WKInterfaceController Subclass

- Make sure to specify the subclass in the storyboard



Custom Class

Class: MyInterfaceController

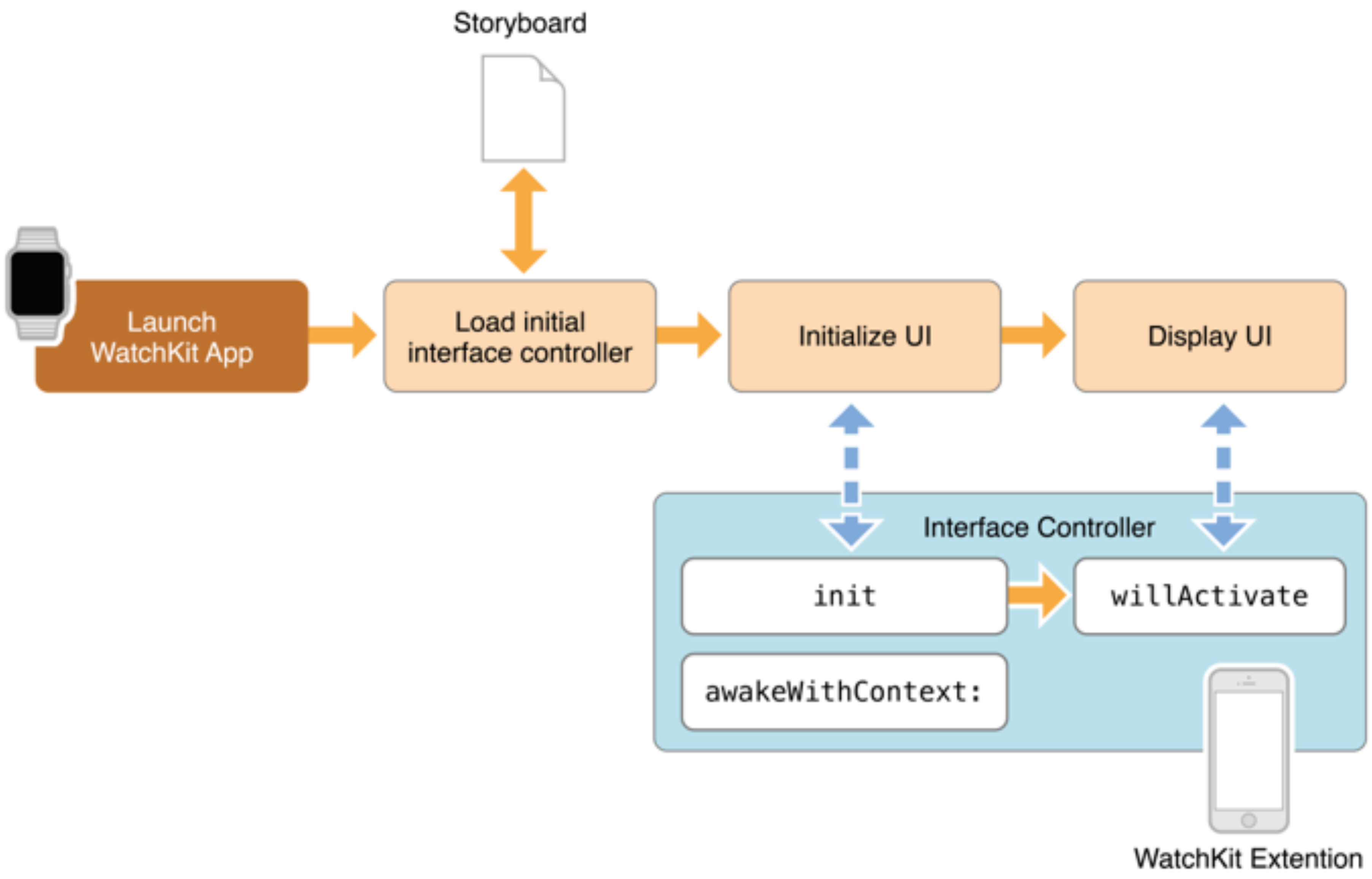
- Then you can connect the storyboard to your class implementation to set up `IBOutlets` and `IBActions`

Initializing WKInterfaceController

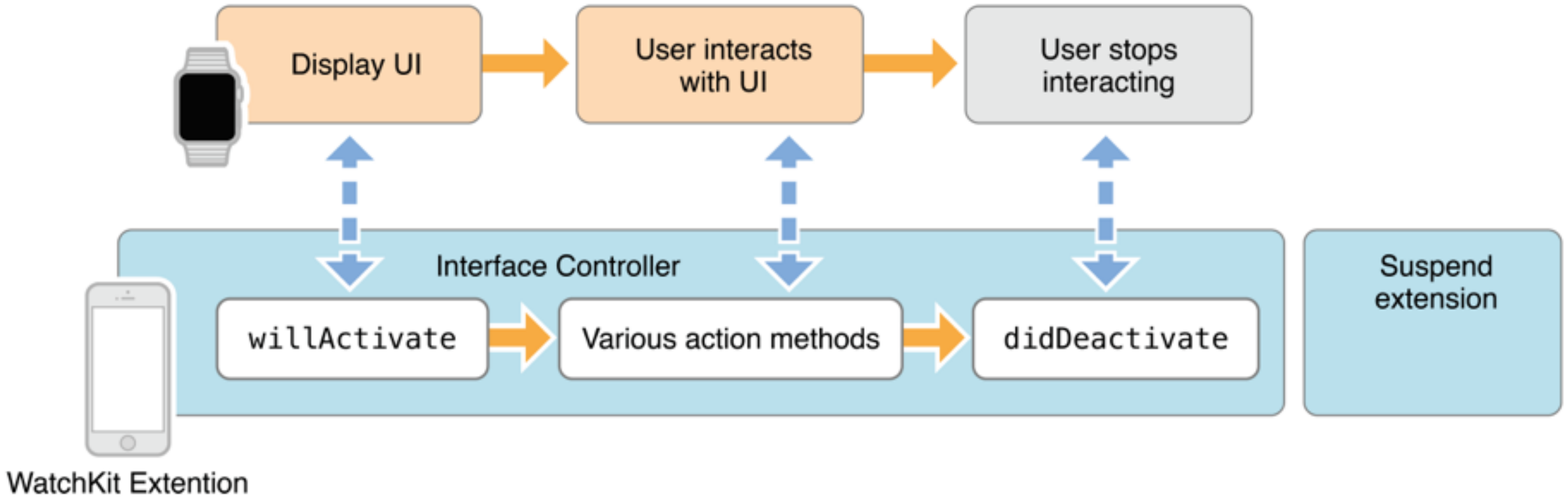
```
func awakeWithContext(_ context: AnyObject?)
```

`context` is passed in by the previous interface controller

Launching a WatchKit app



Lifecycle of an interface controller

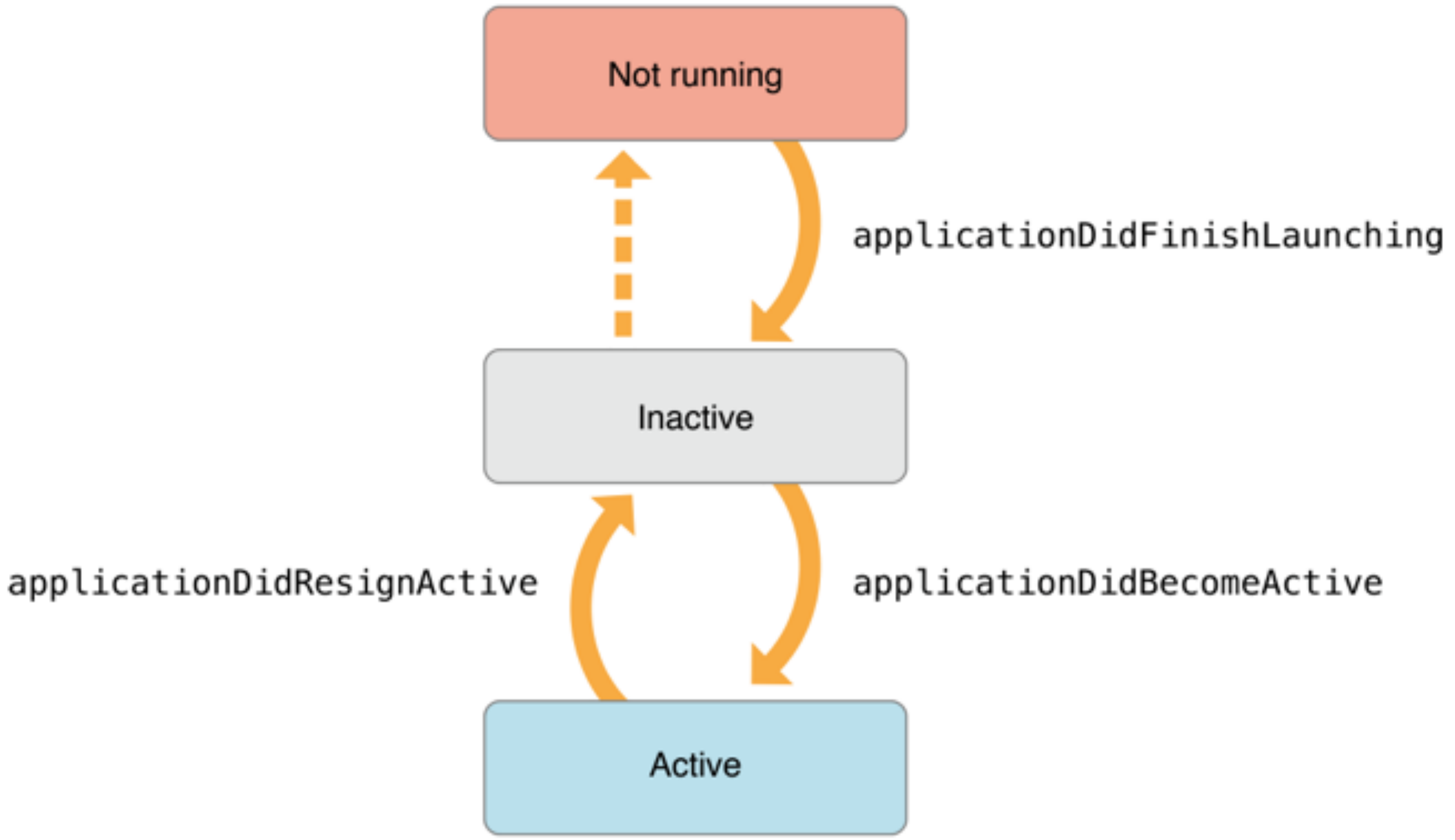


WKExtension

- Analogous to UIApplication
- **WKExtension.sharedExtension()** - analogous to UIApplication.sharedApplication()
- **WKExtension.rootInterfaceController** - analogous to UIWindow.rootViewController
- **delegate** - a WKExtensionDelegate - analogous to UIApplicationDelegate

WKExtensionDelegate

- Gets callbacks when the application changes state

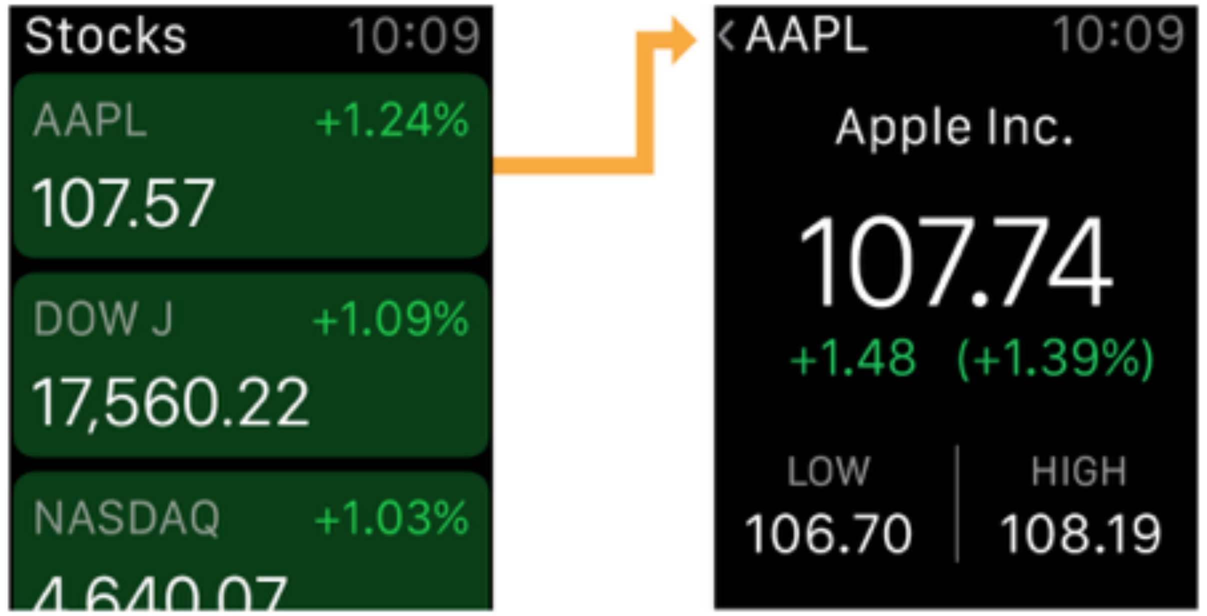

Not running
applicationDidFinishLaunching
Inactive
applicationDidResignActive
applicationDidBecomeActive
Active

State Change Callbacks

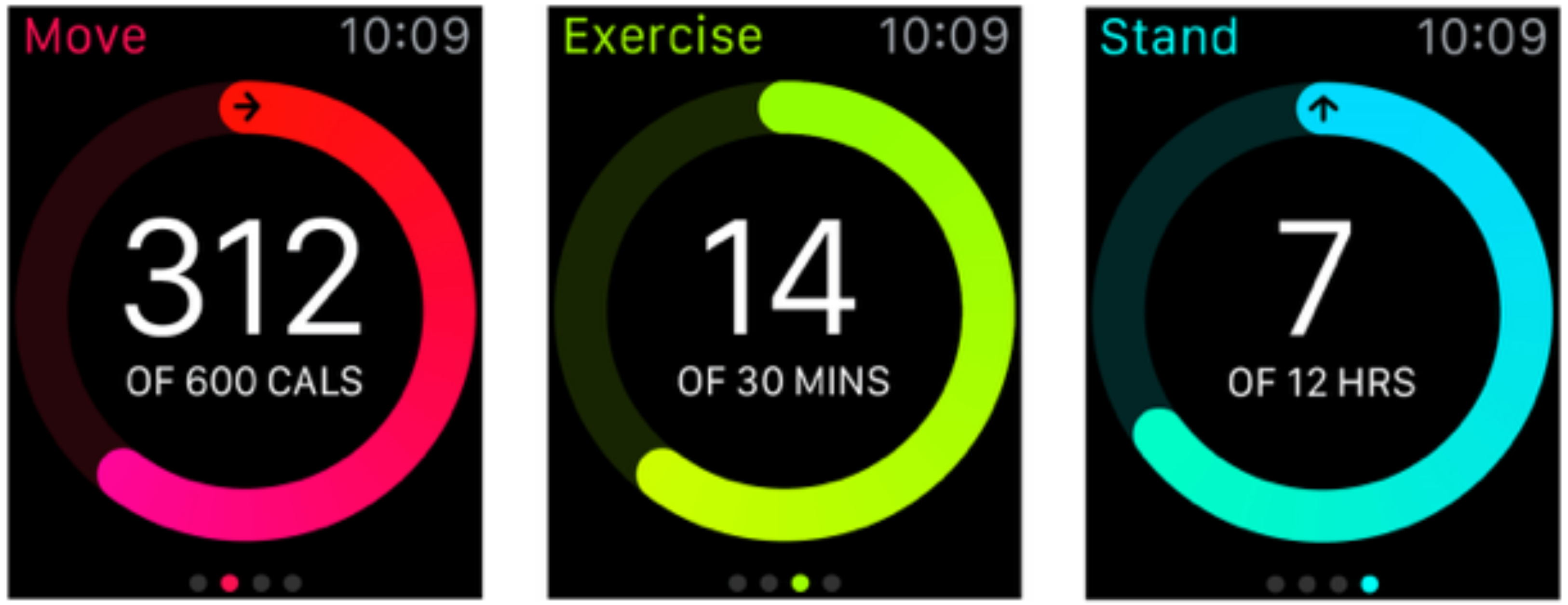
- **applicationDidFinishLaunching** - Called after the launch cycle has finished and before the app interface is active.
- **applicationDidBecomeActive** - The WatchKit app is now visible and processing events.
- **applicationWillResignActive** - The WatchKit app is exiting. Note that this might not be called - e.g. if the app crashes, or the watch runs out of power.

Interface Navigation

Hierarchical



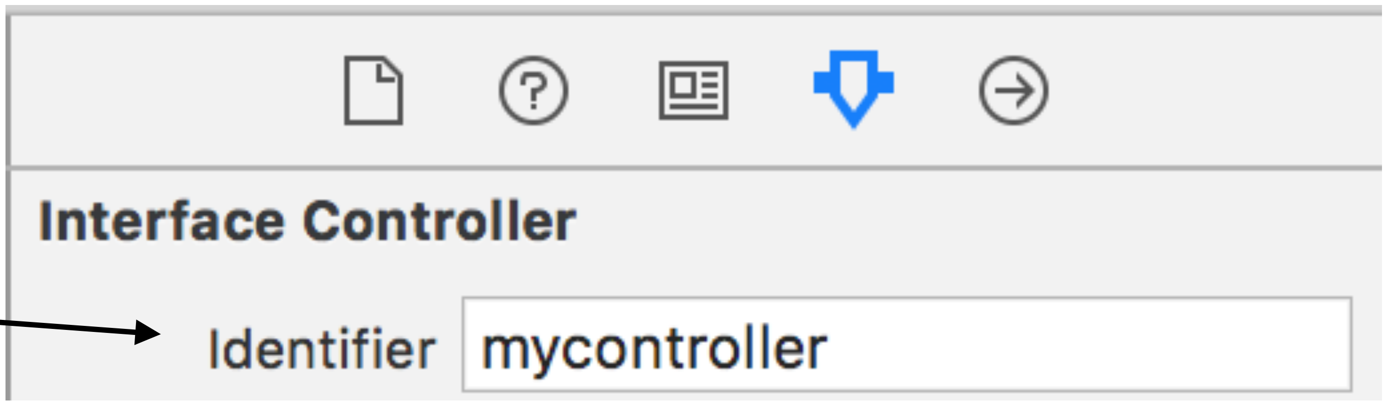
Page-based



WKInterfaceController: Managing a Navigation Interface

```
func presentControllerWithName(_ name: String,
                               context context: AnyObject?)
```

“name”


Interface Controller

Identifier: mycontroller

```
func popController()

func popToRootController()
```

Managing a Page-Based Interface

```
class func reloadRootControllersWithNames(_ names: [String],
                                          contexts contexts: [AnyObject]?)
```

```
func becomeCurrentPage()
```

Modals

Single interface controller or collection of controllers

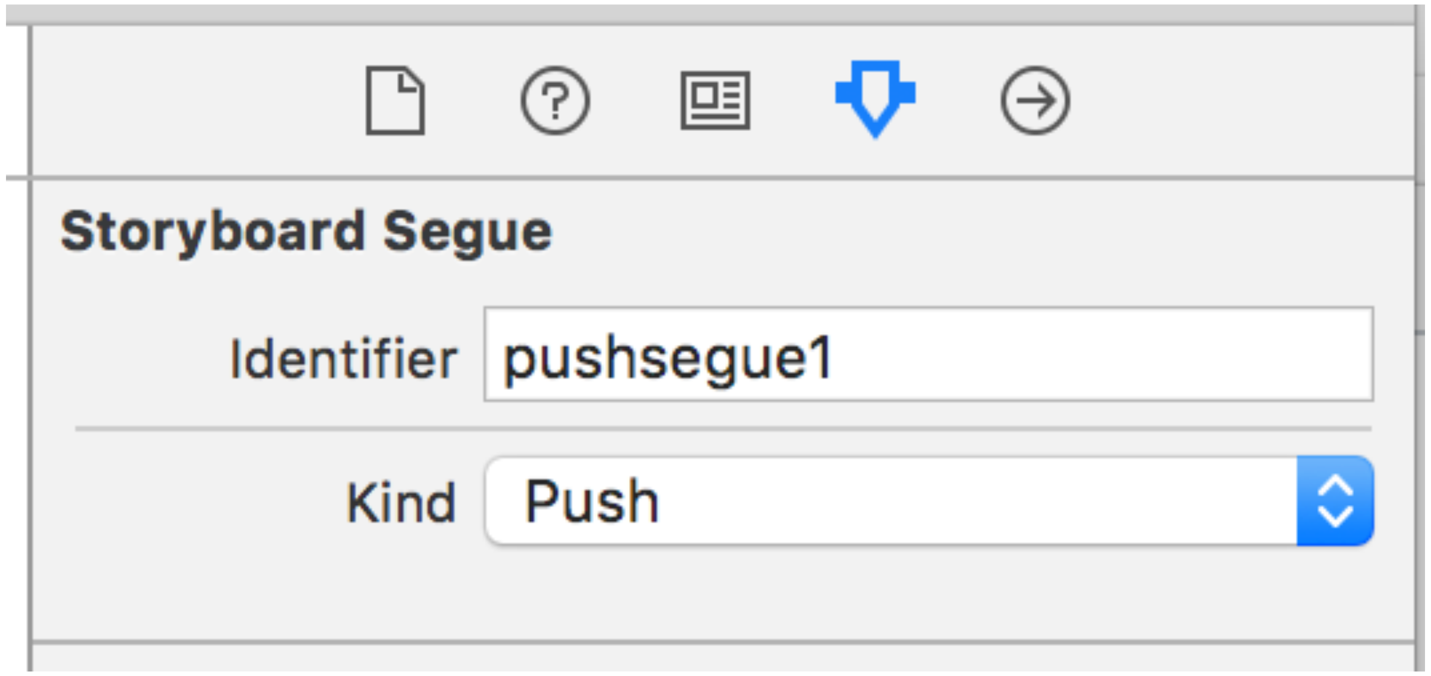
```
func presentControllerWithName(_ name: String,
                               context context: AnyObject?)
```

```
func presentControllerWithNames(_ names: [String],
                               contexts contexts: [AnyObject]?)
```

```
func dismissController()
```

Segues

- Transitions between interface controllers can also be set up in the storyboard
- Ctrl-click and drag to set up segues between interface controllers
- Make sure to create an identifier for each segue



Storyboard Segue

Identifier: pushsegue1

Kind: Push

Passing Context Using Segues

```
func contextForSegueWithIdentifier(_ segueIdentifier: String) -> AnyObject?

func contextsForSegueWithIdentifier(_ segueIdentifier: String) -> [AnyObject]?
```

Interface Objects

WKInterface**Button**
WKInterface**Date**
WKInterface**Group**
WKInterface**Image**
WKInterface**Label**
WKInterface**Map**
WKInterface**Picker**
WKInterface**Separator**
WKInterface**Slider**
WKInterface**Switch**
WKInterface**Table**
WKInterface**Timer**

Interface Objects are *not* views.

They are *proxies* that control the views

WKInterfaceObject API - Hiding and Showing

```
func setHidden(_ hidden: Bool)
```

```
func setAlpha(_ alpha: CGFloat)
```

Constant Sizing

```
func setWidth(_ width: CGFloat)

func setHeight(_ height: CGFloat)
```

Relative Sizing

```
func setRelativeWidth(_ width: CGFloat,
    withAdjustment adjustment: CGFloat)

func setRelativeHeight(_ height: CGFloat,
      withAdjustment adjustment: CGFloat)
```

`width` is value between 0.0 and 1.0

object width = (container width * `width`) + `adjustment`

Content-based Sizing

```
func sizeToFitWidth()

func sizeToFitHeight()
```

The width is set to the width of the content, e.g. the width of a text in a label.

The width is never wider than the container’s width.

Alignment

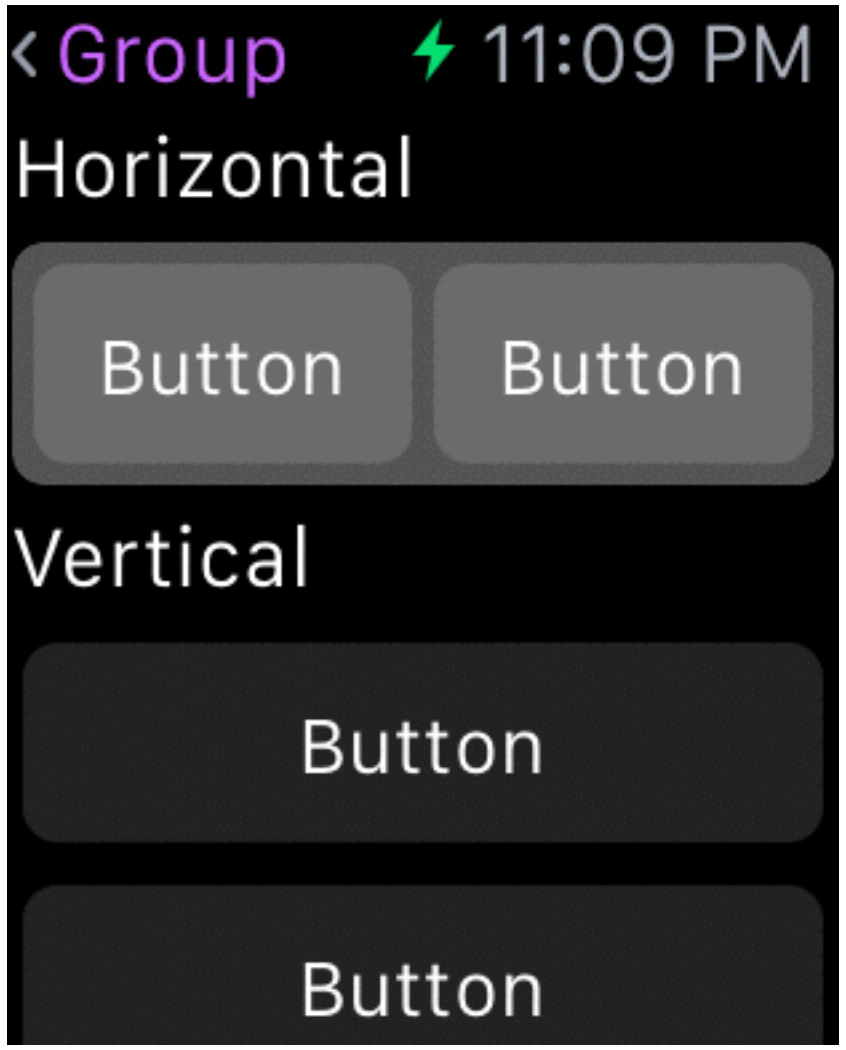
```
func setHorizontalAlignment(_ horizontalAlignment:
WKInterfaceObjectHorizontalAlignment)

(Left / Center / Right)


func setVerticalAlignment(_ verticalAlignment:
WKInterfaceObjectVerticalAlignment)


(Top / Center / Bottom)
```

Simple Interface Objects

Groups

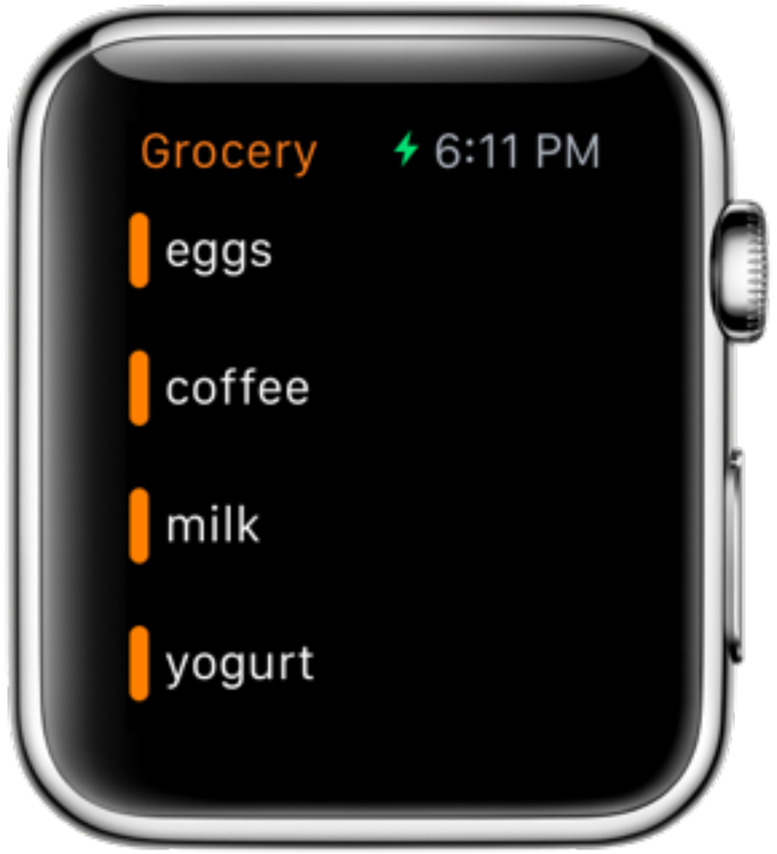
WKInterfaceGroup

- Groups are used to layout elements horizontally or vertically
- Has attributes for specifying margins and spacing among group elements
- Can display an image or solid color as a background
- Has a configurable corner radius for its background and content



Use Groups instead of Images when Possible

- Groups are an efficient alternative to images and should be used when possible
- Groups are the closest thing to `UIViews` in WatchKit

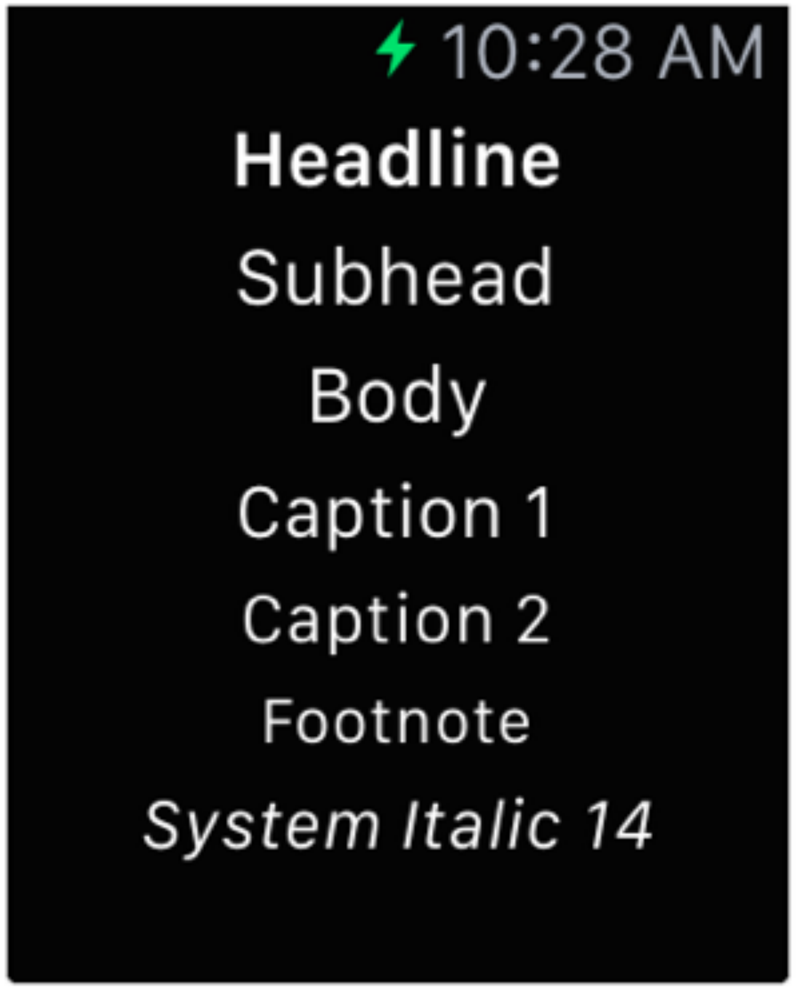


WKInterfaceGroup API

```
func setBackgroundColor(_ color: UIColor?)


func setBackgroundImage(_ image: UIImage?)

func setBackgroundImageData(_ imageData: NSData?)

func setBackgroundImageNamed(_ imageName: String?)


func setCornerRadius(_ cornerRadius: CGFloat)

func setContentInset(_ contentInset: UIEdgeInsets)
```

Labels

WKInterfaceLabel



Like `NSLabel`. Uses `NSAttributedString`.

WKInterfaceLabel API

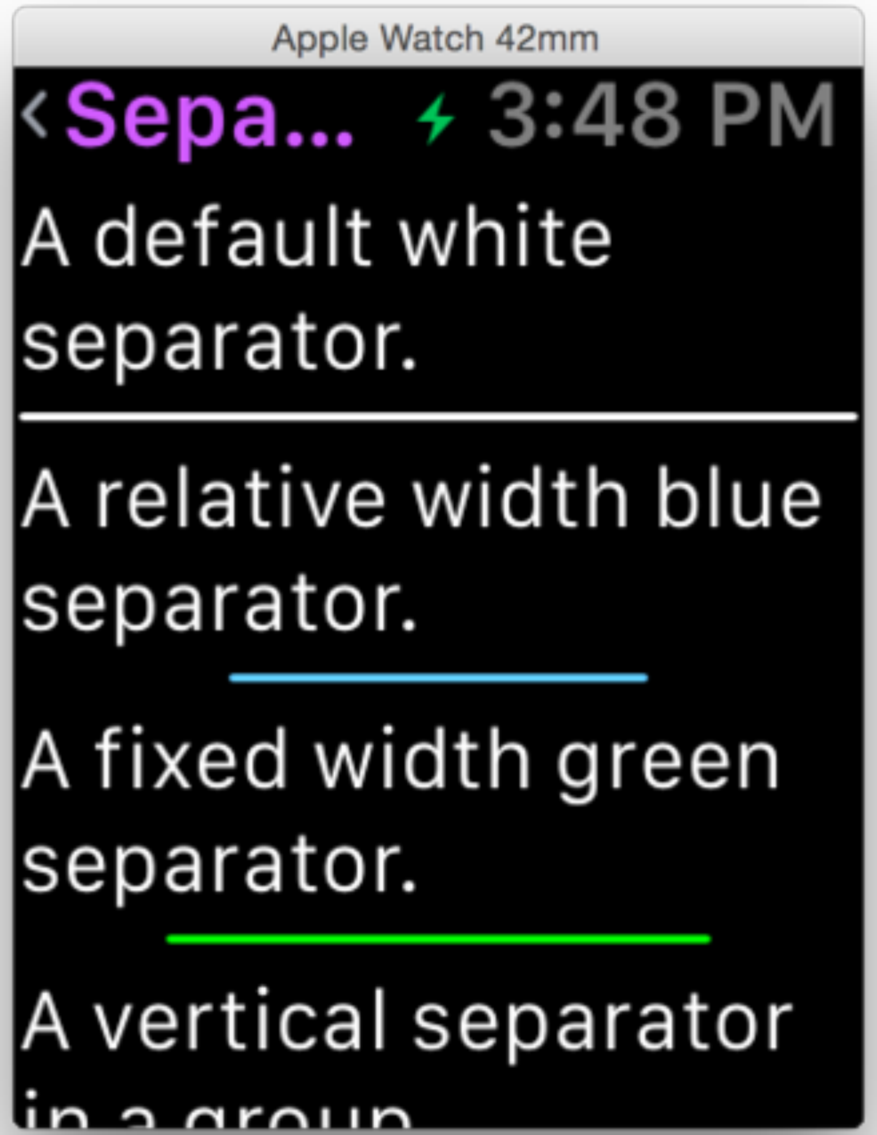
```
func setText(_ text: String?)

func setAttributedText(_
attributedText: NSAttributedString?)


func setTextColor(_ color: UIColor?)
```

Fonts

- By default, labels use the System font (San Francisco)
- You can use a custom font by using an `NSAttributedString` with the `NSFontAttributeName` key set to the `UIFont` you want to use
- If you want to include a custom font, include it in both the Watch App bundle and the Watch Extension bundle and set the `UIAppFonts` key in both targets' `Info.plist` files to specify the font

WKInterfaceSeparator

- Can change color, width



A default white separator.

A relative width blue separator.

A fixed width green separator.

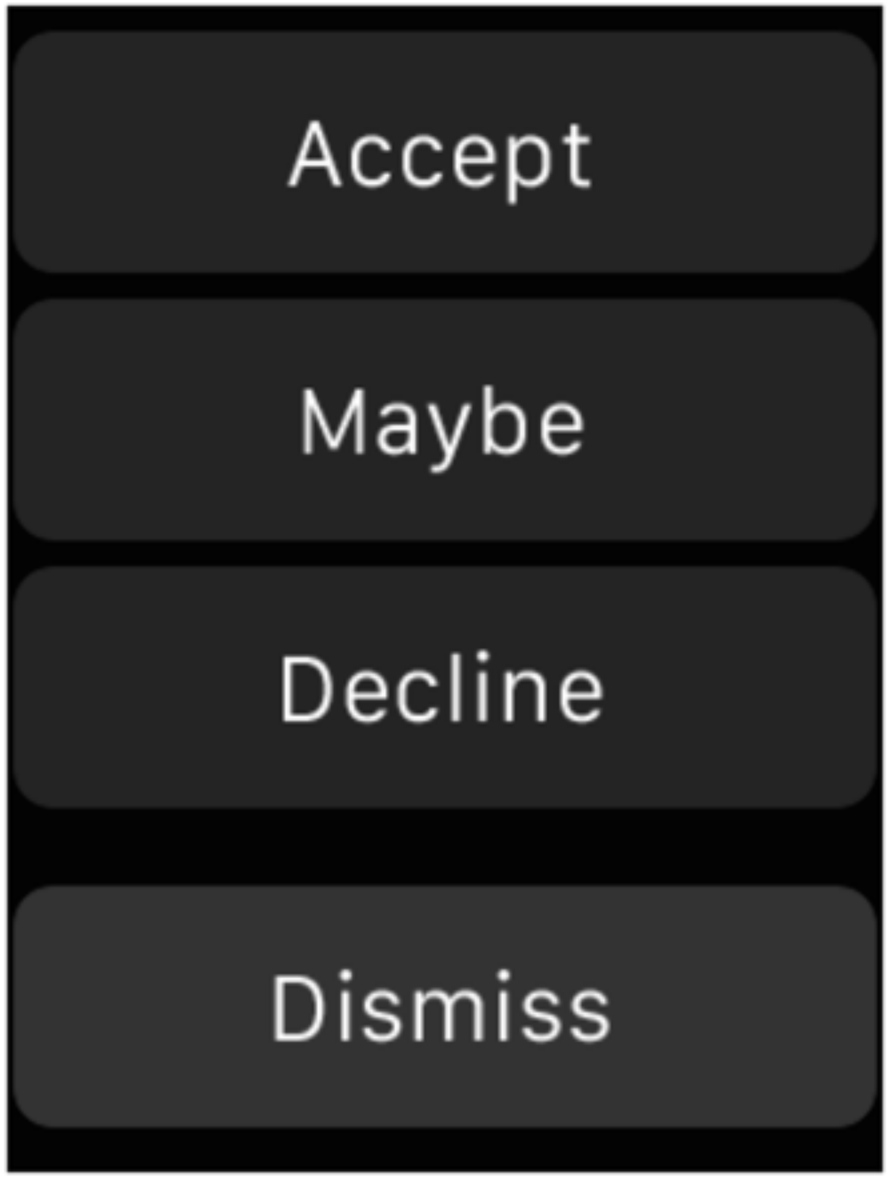
A vertical separator

WKInterfaceSeparator API

```
func setColor(_ color: UIColor?)
```

WKInterfaceButton

- Can contain a group or label
- Can have background color or image

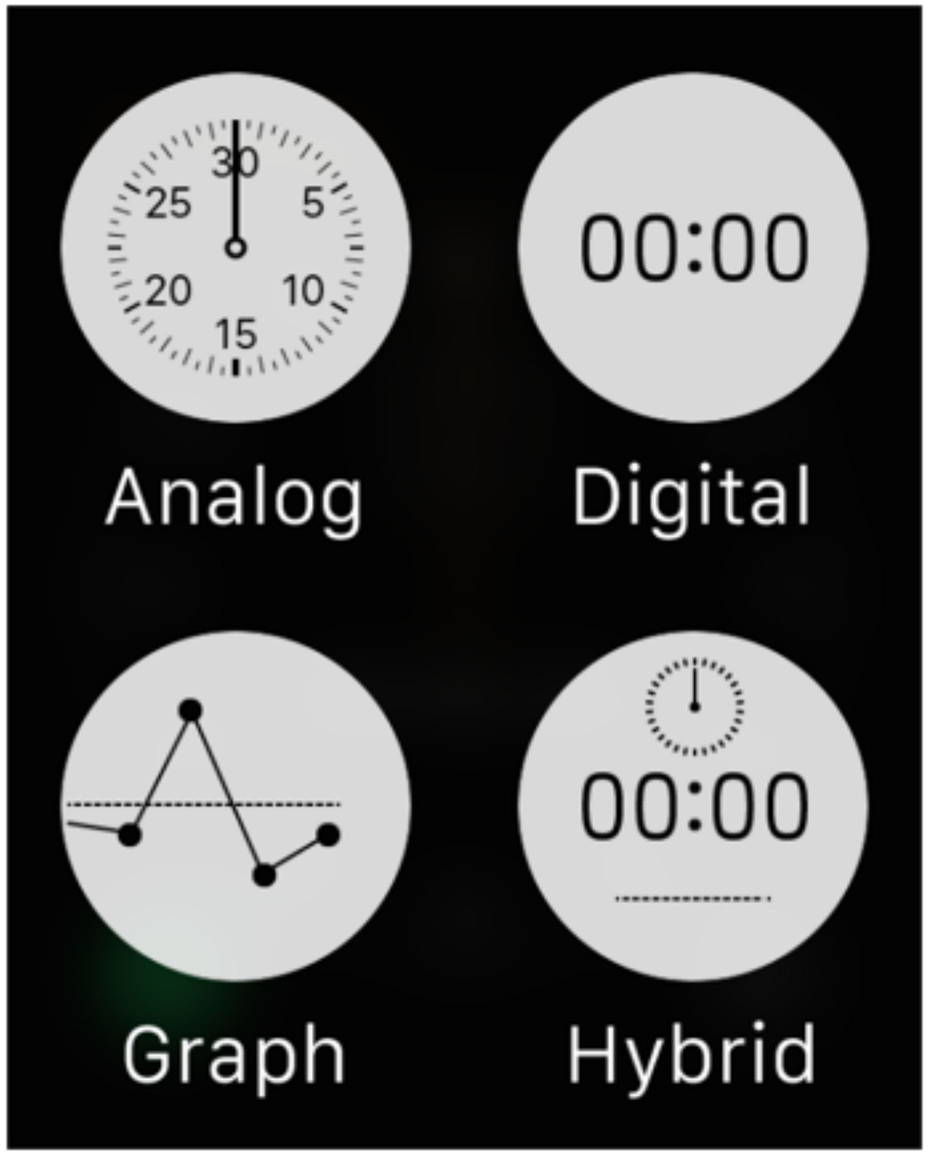


WKInterfaceButton API

```
func setTitle(_ title: String?)

func setAttributedTitle(_ attributedTitle:
NSAttributedString?)



func setBackgroundColor(_ color: UIColor?)

func setBackgroundImage(_ image: UIImage?)

func setBackgroundImageData(_ imageData: NSData?)

func setBackgroundImageNamed(_ imageName: String?)



func setEnabled(_ enabled: Bool)
```

Context Menus

- Activated by a Force Touch
- 1-4 actions

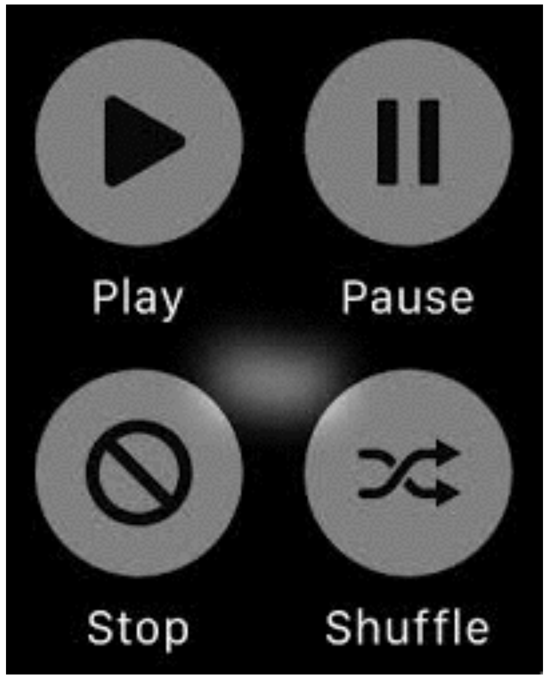


Configuring Context Menus

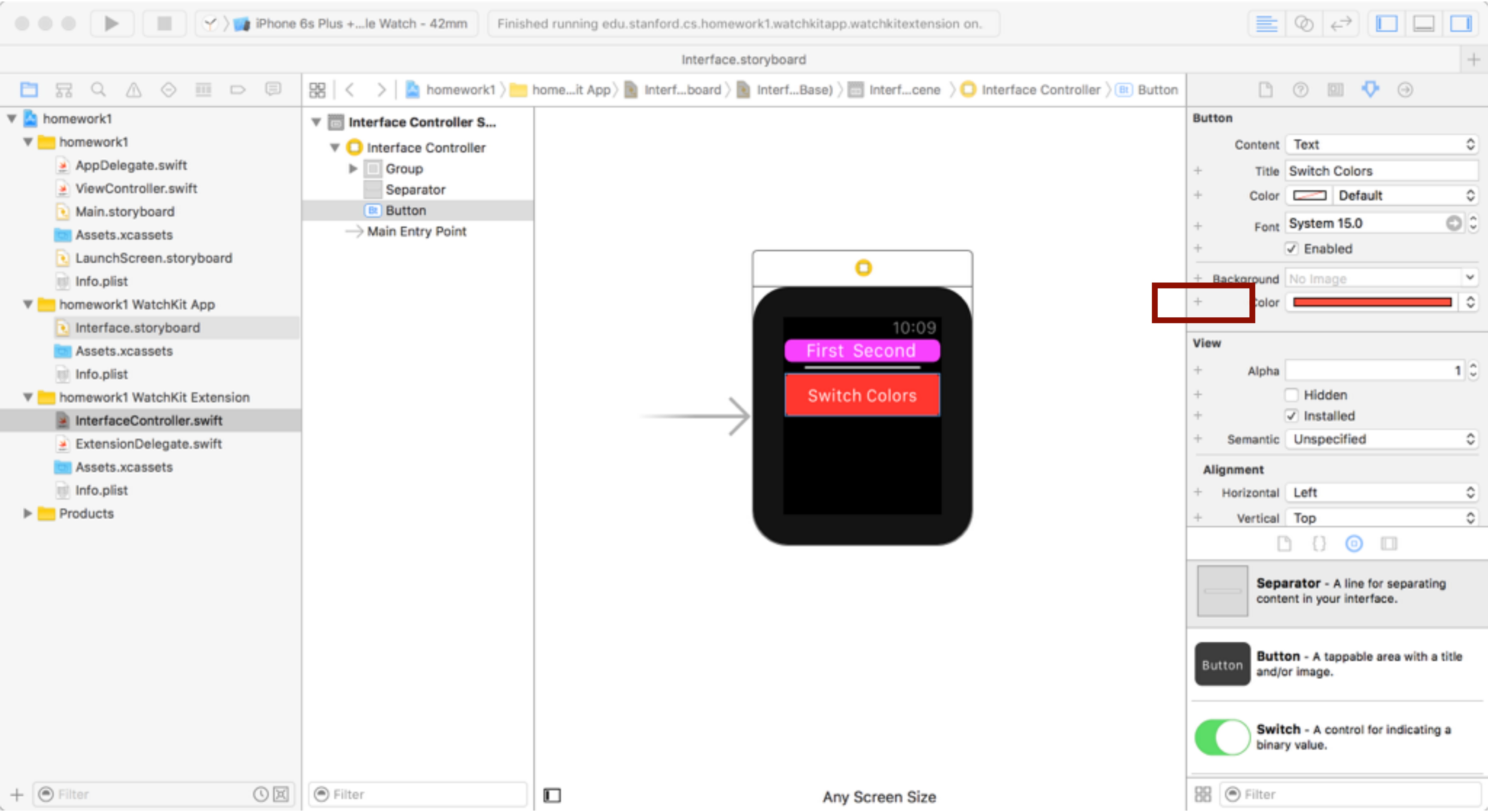
```
func addMenuItemWithImageNamed(_ imageName: String,
                               title title: String,
                              action action: Selector )

func addMenuItemWithImage(_ image: UIImage,
                          title title: String,
                         action action: Selector )

func addMenuItemWithItemIcon(_ itemIcon: ,
                             title title: String,
                            action action: Selector )

func clearAllMenuItems()
```

WAMenuItemIcon

```
enum  : Int {
   case Accept ,        // checkmark
   case Add ,           // '+'
   case Block ,         // circle w/ slash
   case Decline ,       // 'x'
   case Info ,          // 'i'
   case Maybe ,         // '?'
   case More ,          // '...'
   case Mute ,          // speaker w/ slash
   case Pause ,         // pause button
   case Play ,          // play button
   case Repeat ,        // looping arrows
   case Resume ,        // circular arrow
   case Share ,         // share icon
   case Shuffle ,       // swapped arrows
   case Speaker ,       // speaker icon
   case Trash ,         // trash icon
};
```



Customizing UI for different device sizes



Button

Content	Text
Title	Switch Colors
Color	Default
Font	System 15.0
	✓ Enabled
Background	No Image
Color	🟥

View

Alpha	1
	Hidden
	✓ Installed
Semantic	Unspecified

Alignment

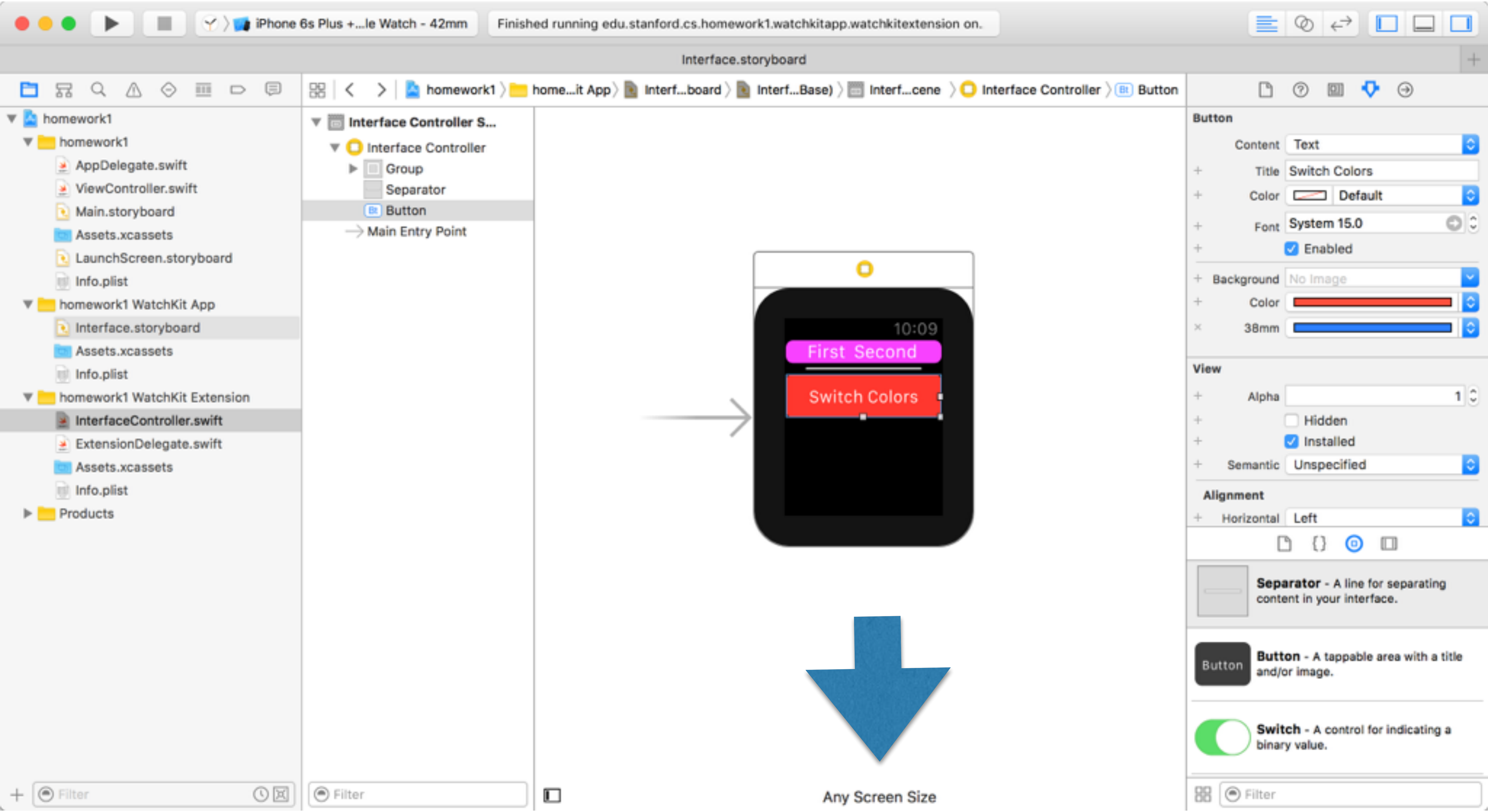
Horizontal	Left
Vertical	Top

Separator - A line for separating content in your interface.

Button - A tappable area with a title and/or image.

Switch - A control for indicating a binary value.

Customizing UI for different device sizes



Finished running edu.stanford.cs.homework1.watchkitapp.watchkitextension on.

Button

Content: Text
Title: Switch Colors
Color: Default
Font: System 15.0
✓ Enabled
Background: No Image
Color: 🟥
38mm: 🟦

View

Alpha: 1
Hidden
✓ Installed
Semantic: Unspecified

Alignment

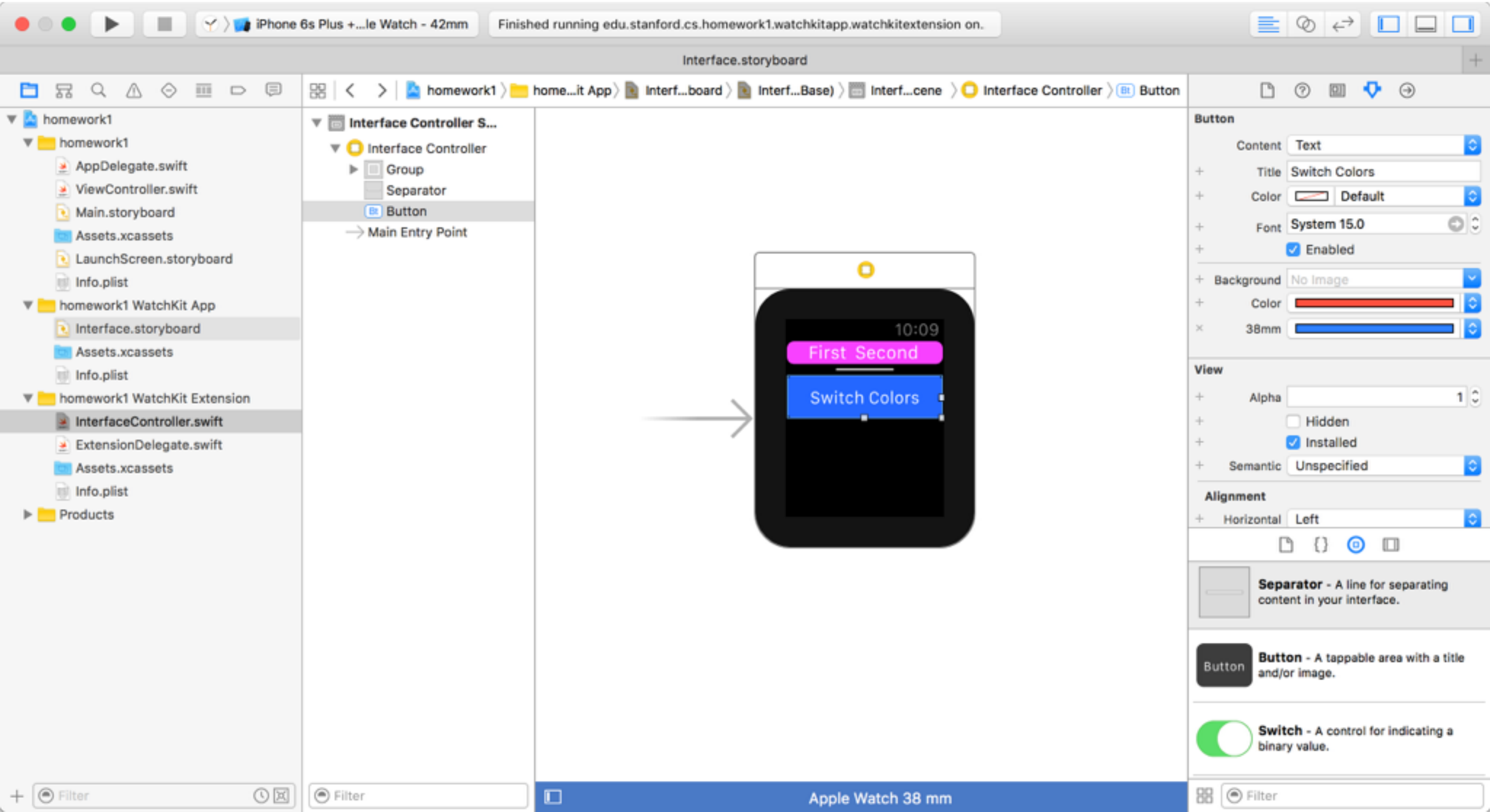
Horizontal: Left

Separator - A line for separating content in your interface.

Button - A tappable area with a title and/or image.

Switch - A control for indicating a binary value.

Customizing UI for different device sizes



Button

Content	Text
Title	Switch Colors
Color	Default
Font	System 15.0
	✓ Enabled
Background	No Image
Color	🟥
38mm	🟦

View

Alpha	1
	Hidden
	✓ Installed
Semantic	Unspecified

Alignment

Horizontal	Left

Separator - A line for separating content in your interface.

Button - A tappable area with a title and/or image.

Switch - A control for indicating a binary value.

Readings

- https://developer.apple.com/watchkit/
  - Developer guide
  - Human Interface Guidelines
- Be careful when searching Google; the wrong documentation may show up.
- watchOS 2:
  - https://developer.apple.com/library/**watchos**/documentation/WatchKit/Reference/WKInterfaceController_class/
- watchOS 1:
  - https://developer.apple.com/library/**ios**/documentation/WatchKit/Reference/WKInterfaceController_class/