THE UNIVERSITY OF CHICAGO

ADVANCED tvOS APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 9

APPLE TV

APPLE TV

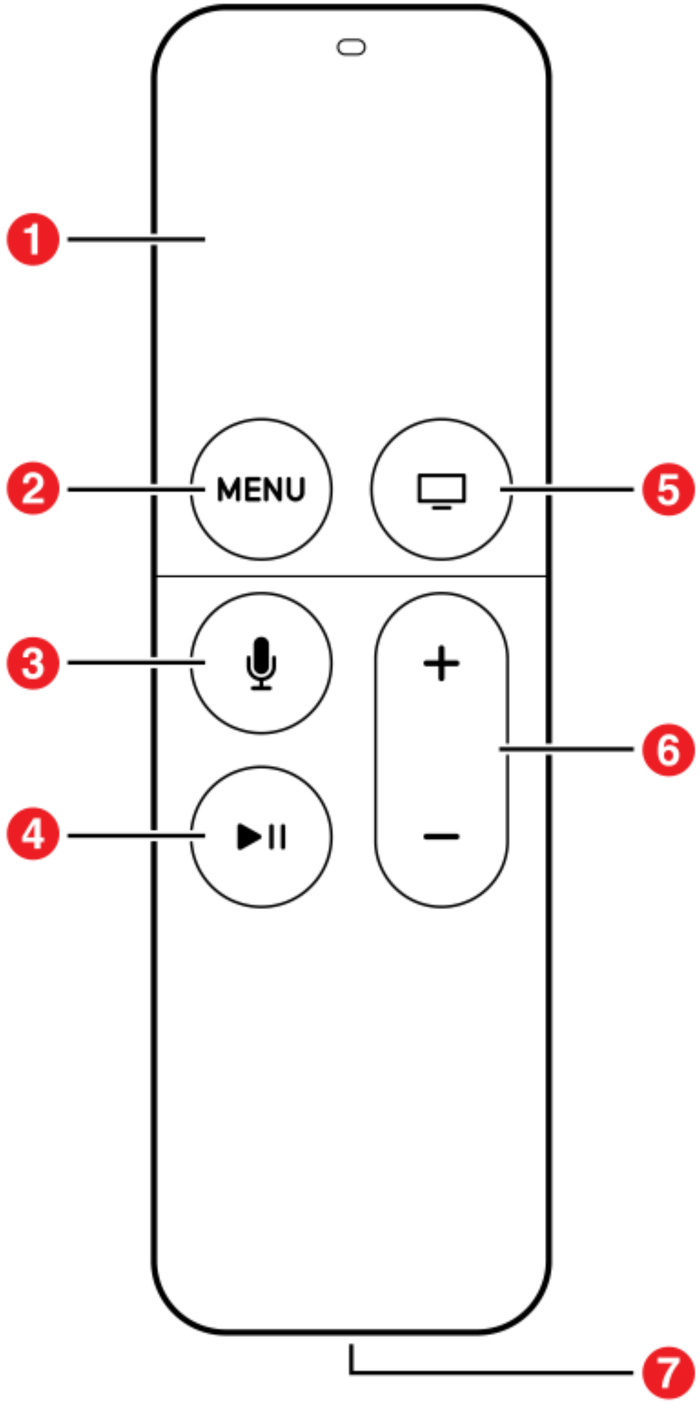
SUBTITLE

- Apple TV has the following hardware specifications
  - 64-bit A8, A10X, A12X Fusion processor
  - 32 GB, 64 GB of storage
  - 2/4 GB of RAM
  - 10/100 Mbps Ethernet
  - WiFi 802.11a/b/g/n/ac
  - 1080p/4K resolution
  - HDMI
  - New Siri Remote / Apple TV Remote

APPLE TV

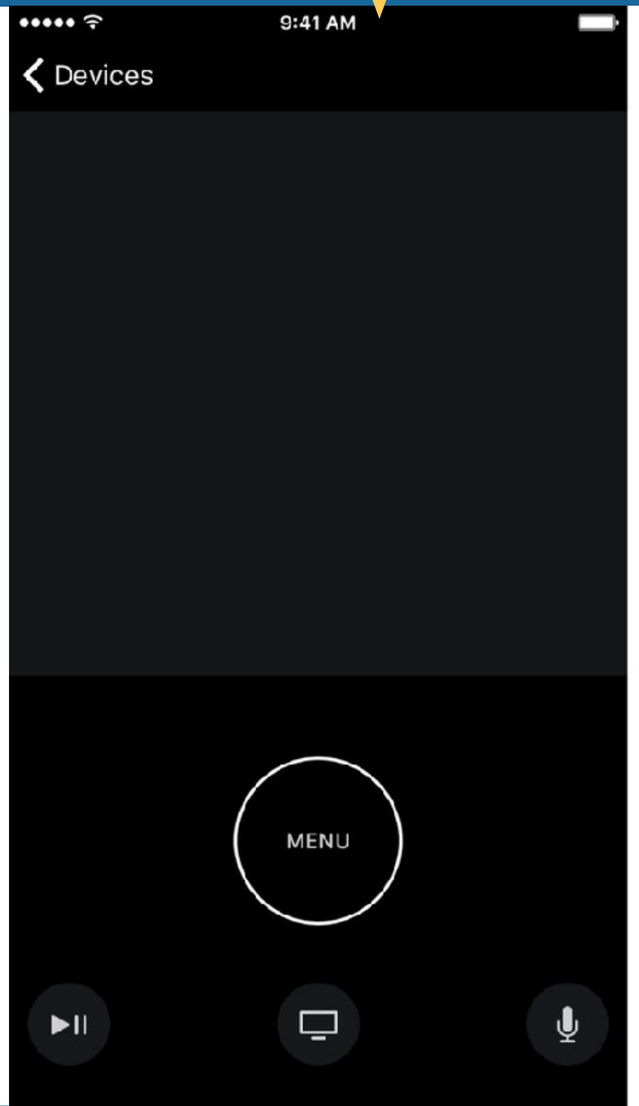
SUBTITLE

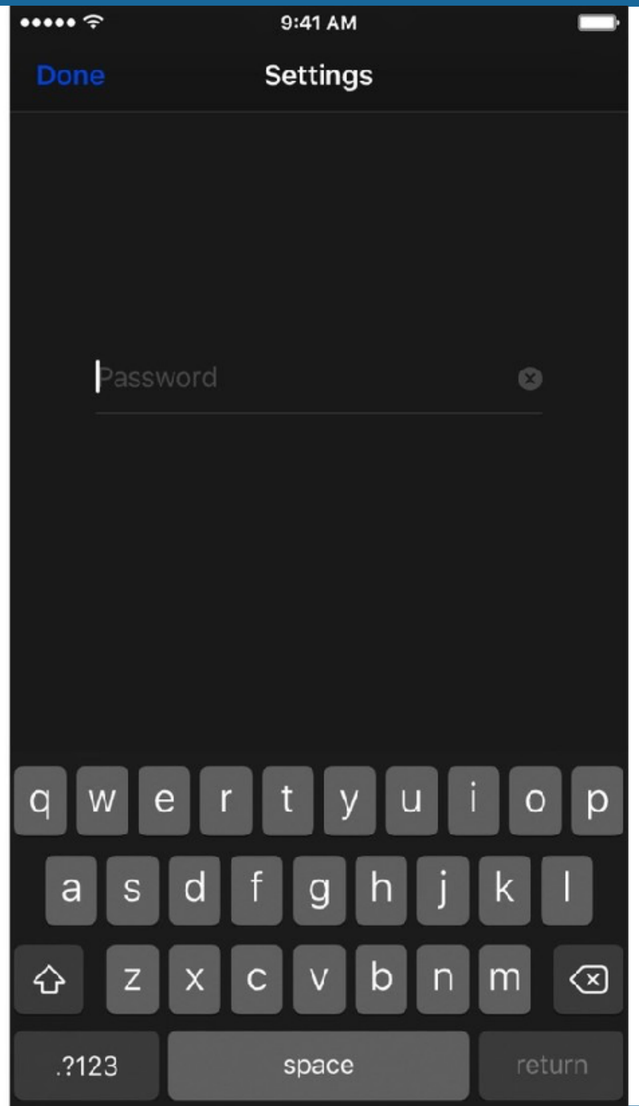
- Two remotes
  - Siri Remote
  - Apple TV remote
  - iPhone App



APPLE TV

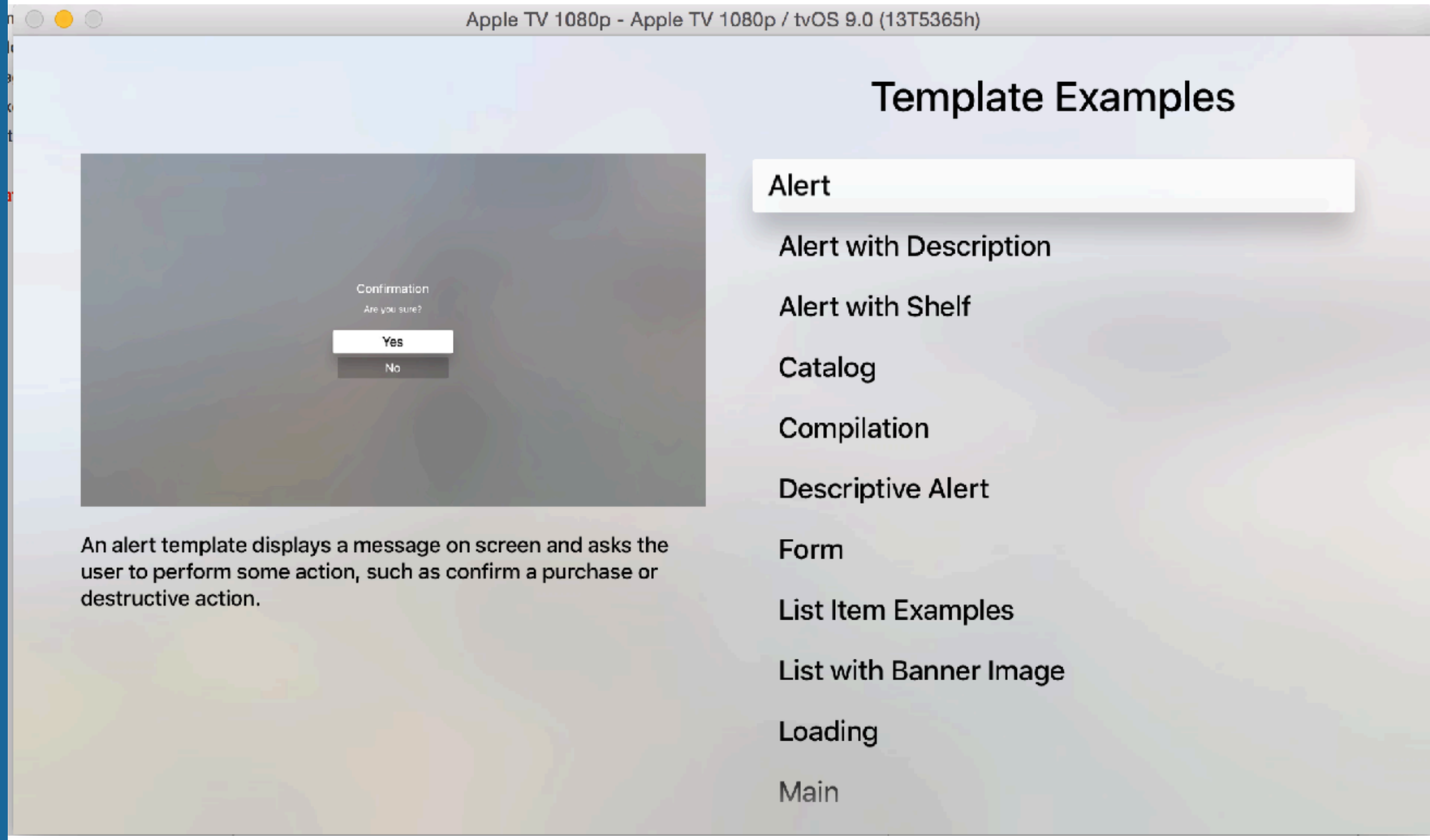
Everything and more but less





APPLE TV

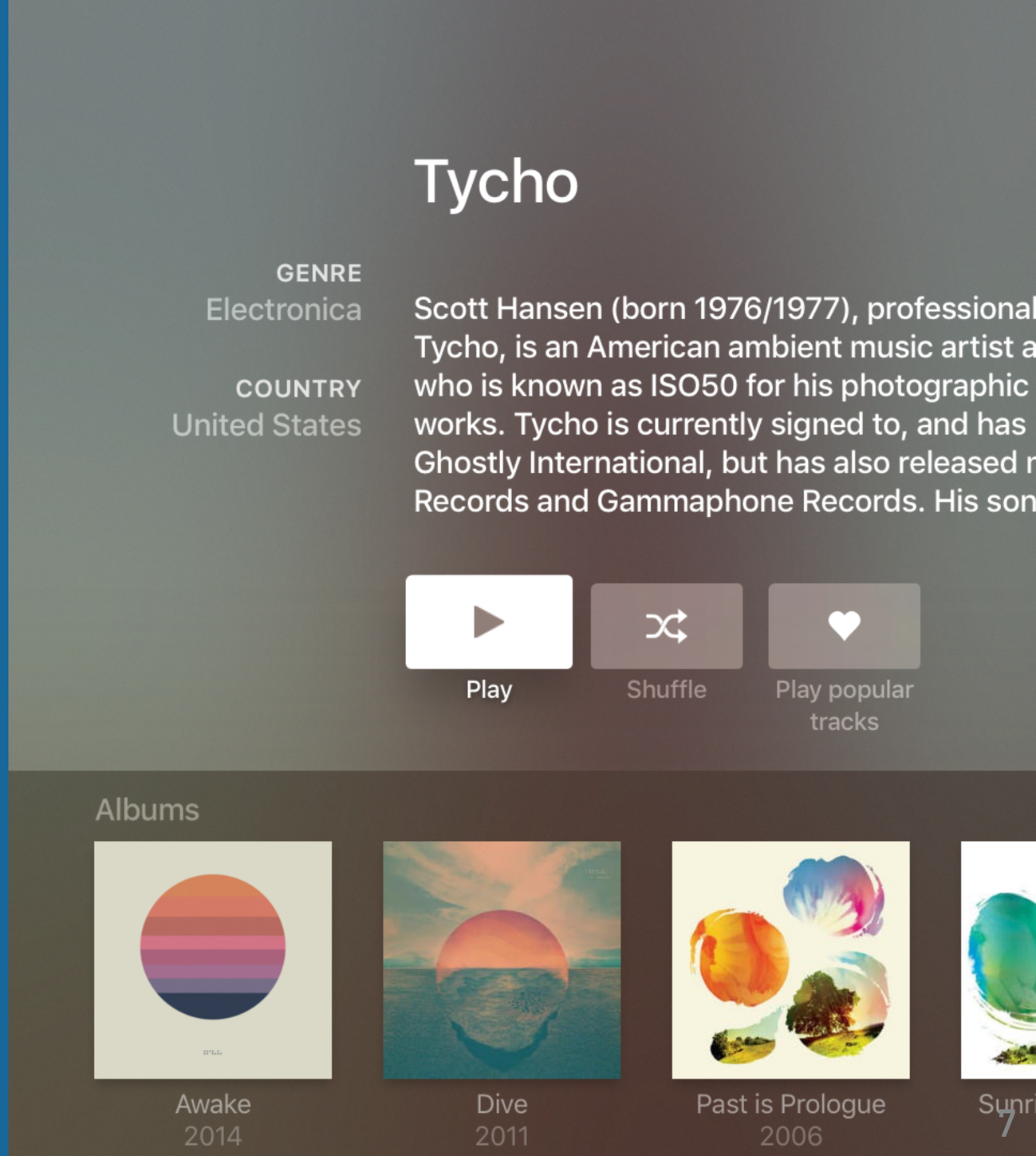
- Two style of apps for Apple TV
  - UIKit
  - TVML



APPLE TV

SUBTITLE

- TVML is a custom markup language used to design apps whose main purpose is to stream media
- Templates define the interface
- Javascript to interface with server providing content



APPLE TV

Tycho

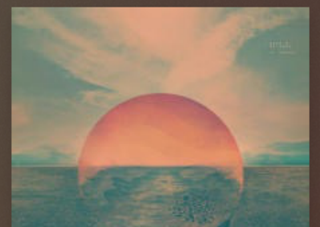
GENRE
Electronica

COUNTRY
United States

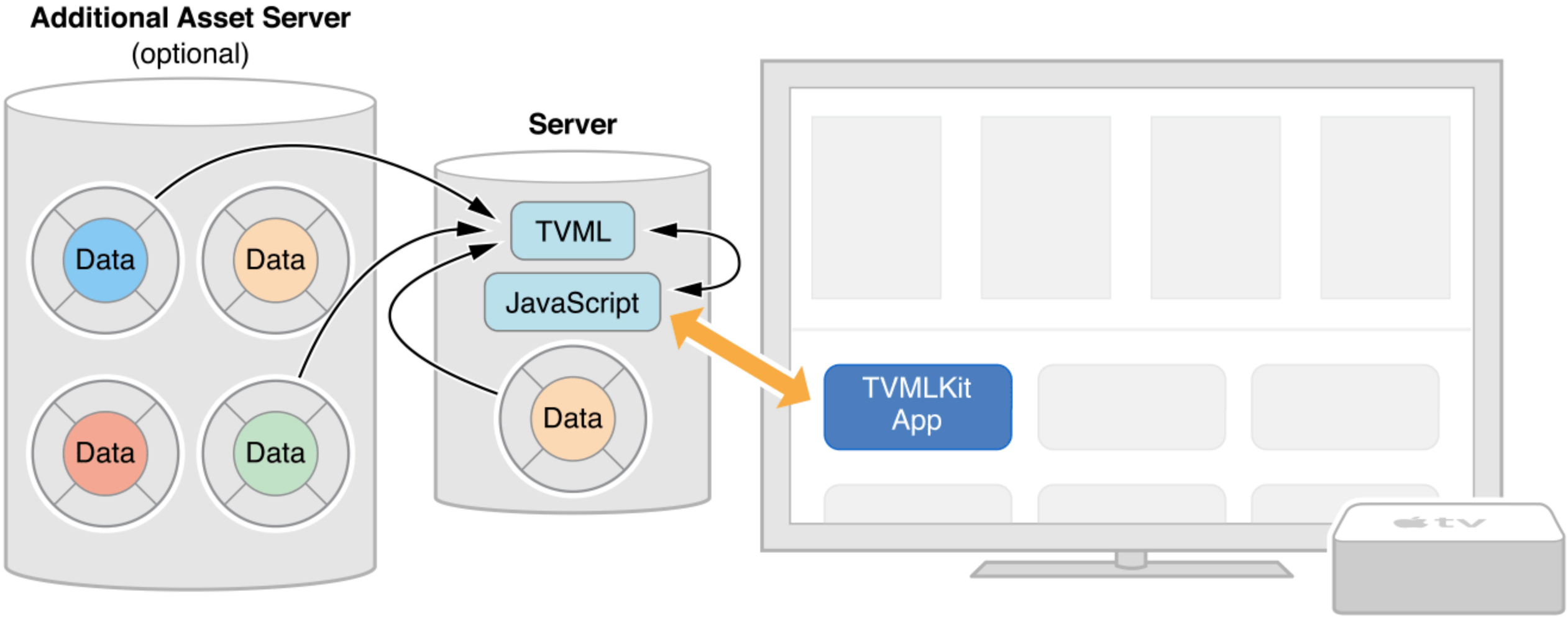
Scott Hansen (born 1976/1977), professionally known as Tycho, is an American ambient music artist and producer, who is known as ISO50 for his photographic and design works. Tycho is currently signed to, and has released under, Ghostly International, but has also released music on Merck Records and Gammaphone Records. His song, Dicta... MORE

Play | Shuffle | Play popular tracks

Albums

APPLE TV


Additional Asset Server
(optional)
Data
Data
Data
Data
Server
TVML
JavaScript
Data
TVMLKit
App
tv

APPLE TV

SUBTITLE

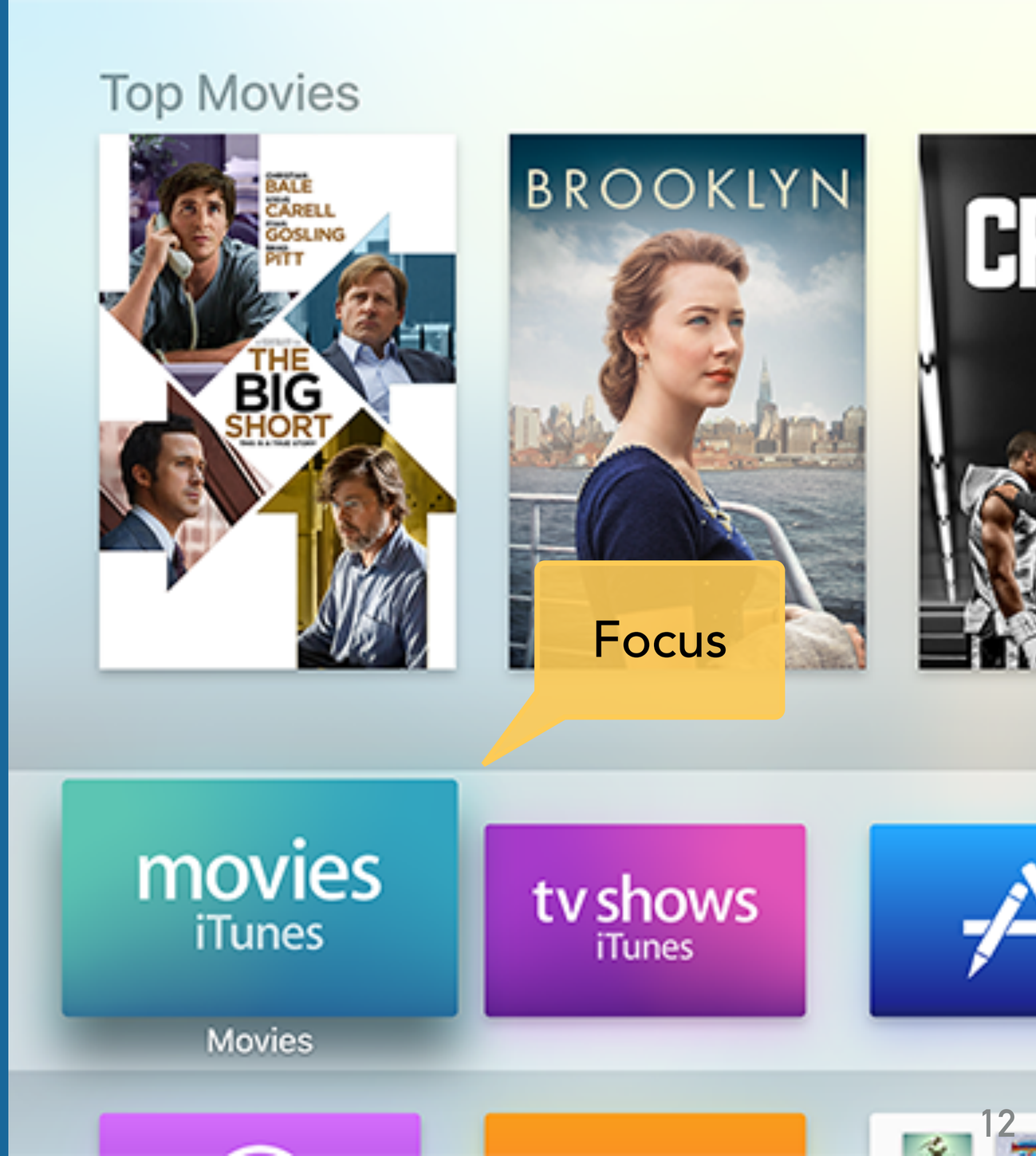
- Top shelf provides preferred placement in top row
- Static or dynamic content

APPLE TV

APPLE TV

SUBTITLE

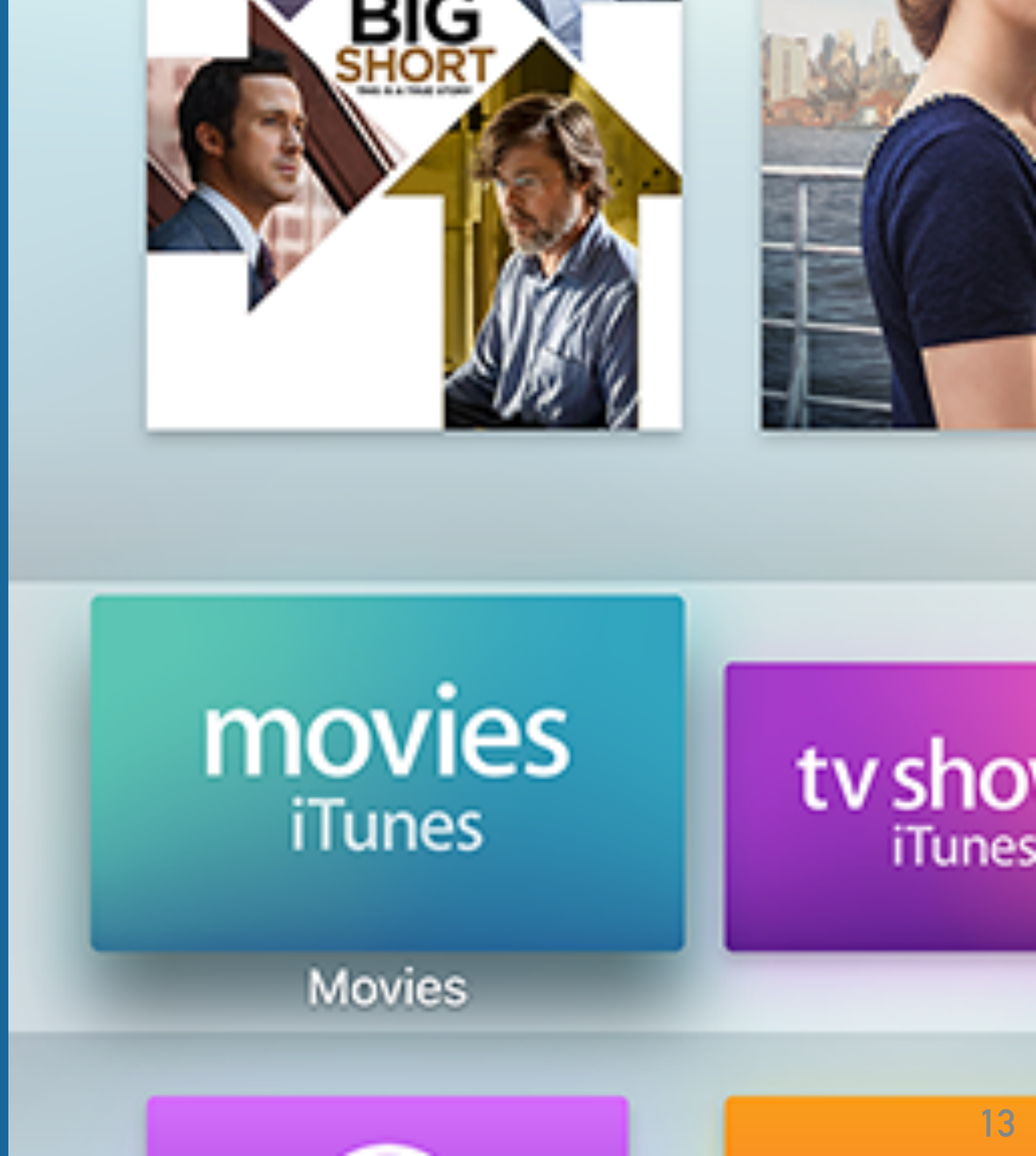
- Focus
  - Navigation
  - UI element is **highlighted** by a user (not selected)



APPLE TV

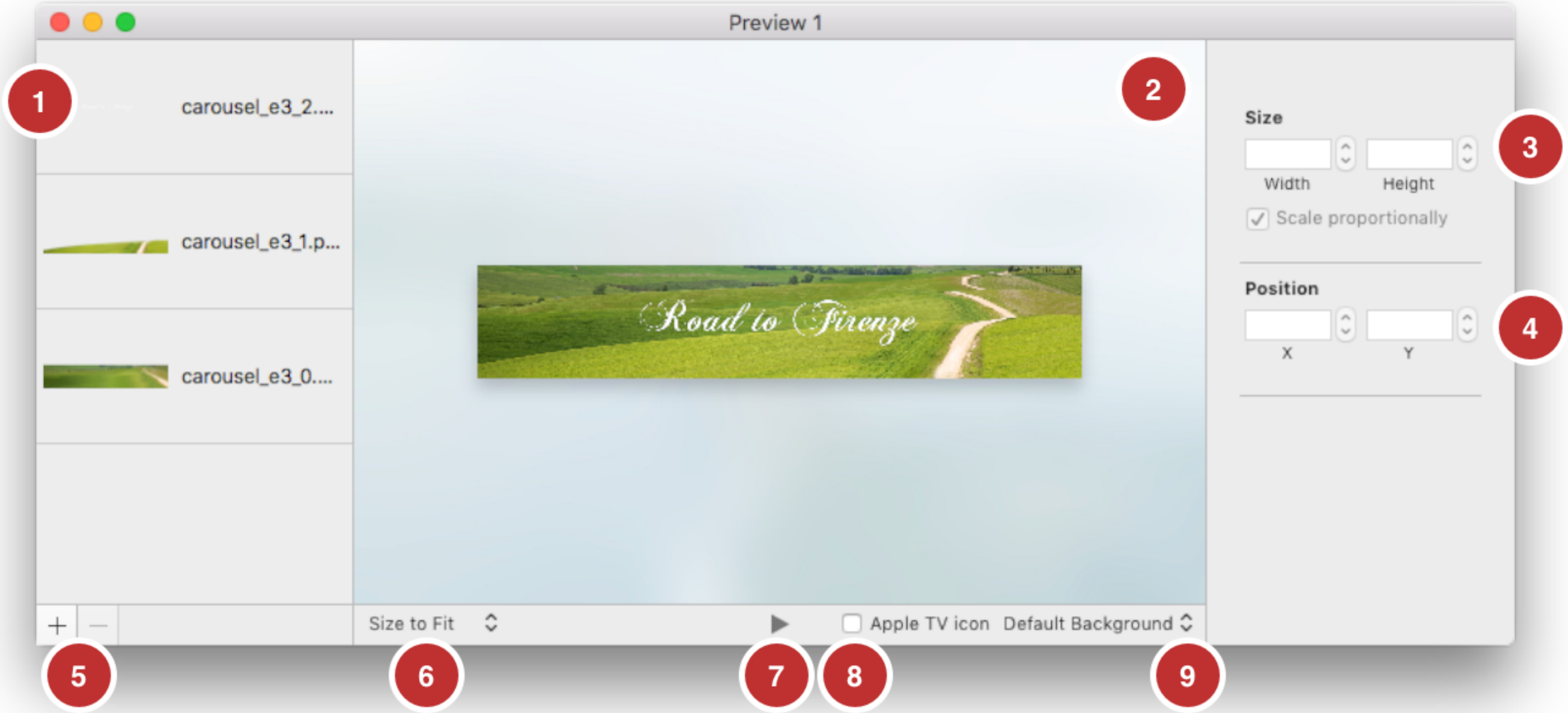
SUBTITLE

- Images can be layered for a single image view
  - UIImageView is customized for tvOS
- "Eye candy" for effect



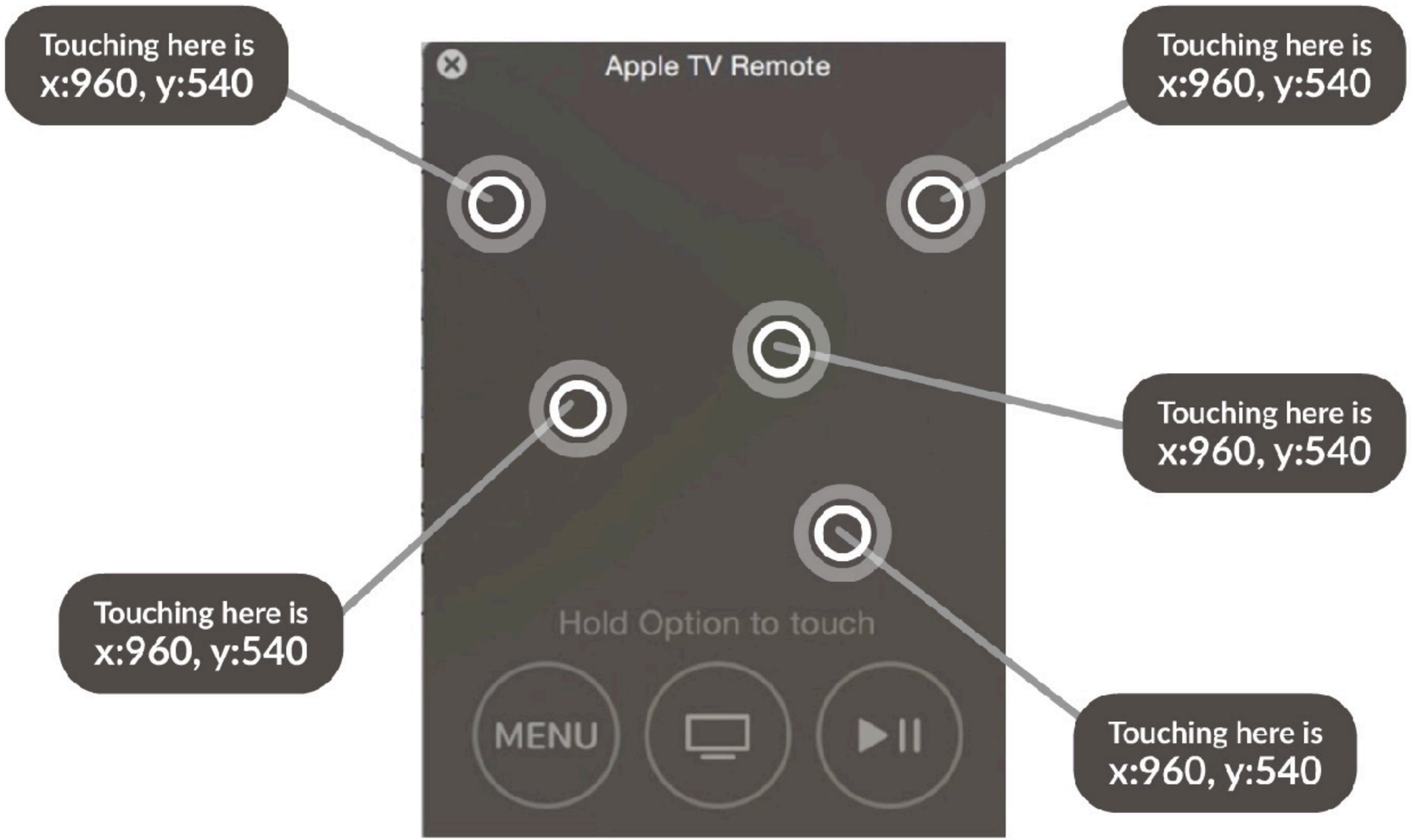
APPLE TV

Xcode tools



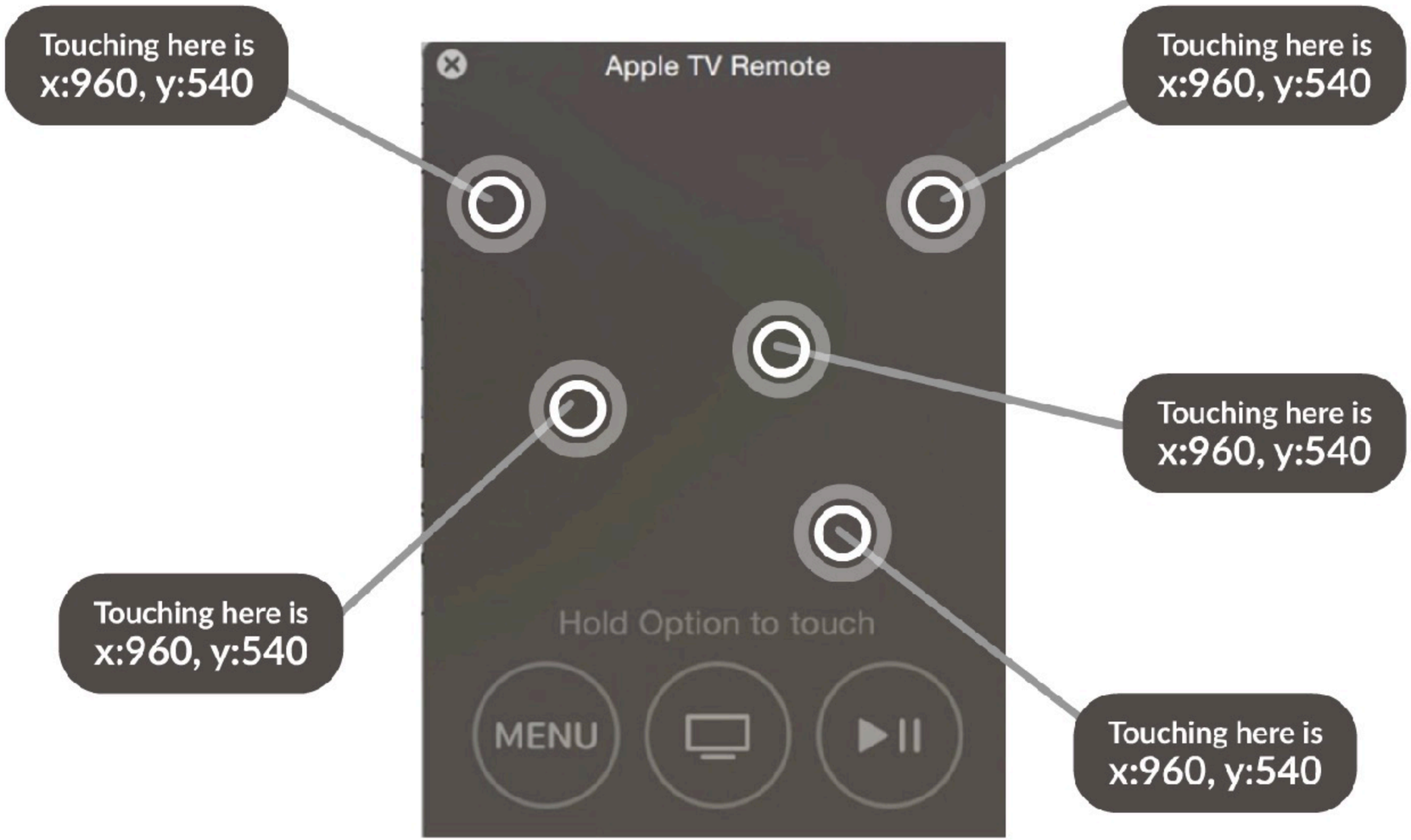
APPLE TV

- UIKit implementation is familiar
- Interaction is very different



APPLE TV

- Users can only interact is specific places accessible by remote
- Multi-users
  - Remote
  - iPhone
  - GamePad



APPLE TV

- Local storage is extremely limited (500KB)
  - Highly unstable (tvOS will clear cache aggressively)
- Apps can only be 4GB
  - Majority of storage is for apps and streaming content

APPLE TV

- Strategies
  - Download and cache
  - On-demand resources (Apple provided or self-hosted)
  - iCloud (CloudKit)

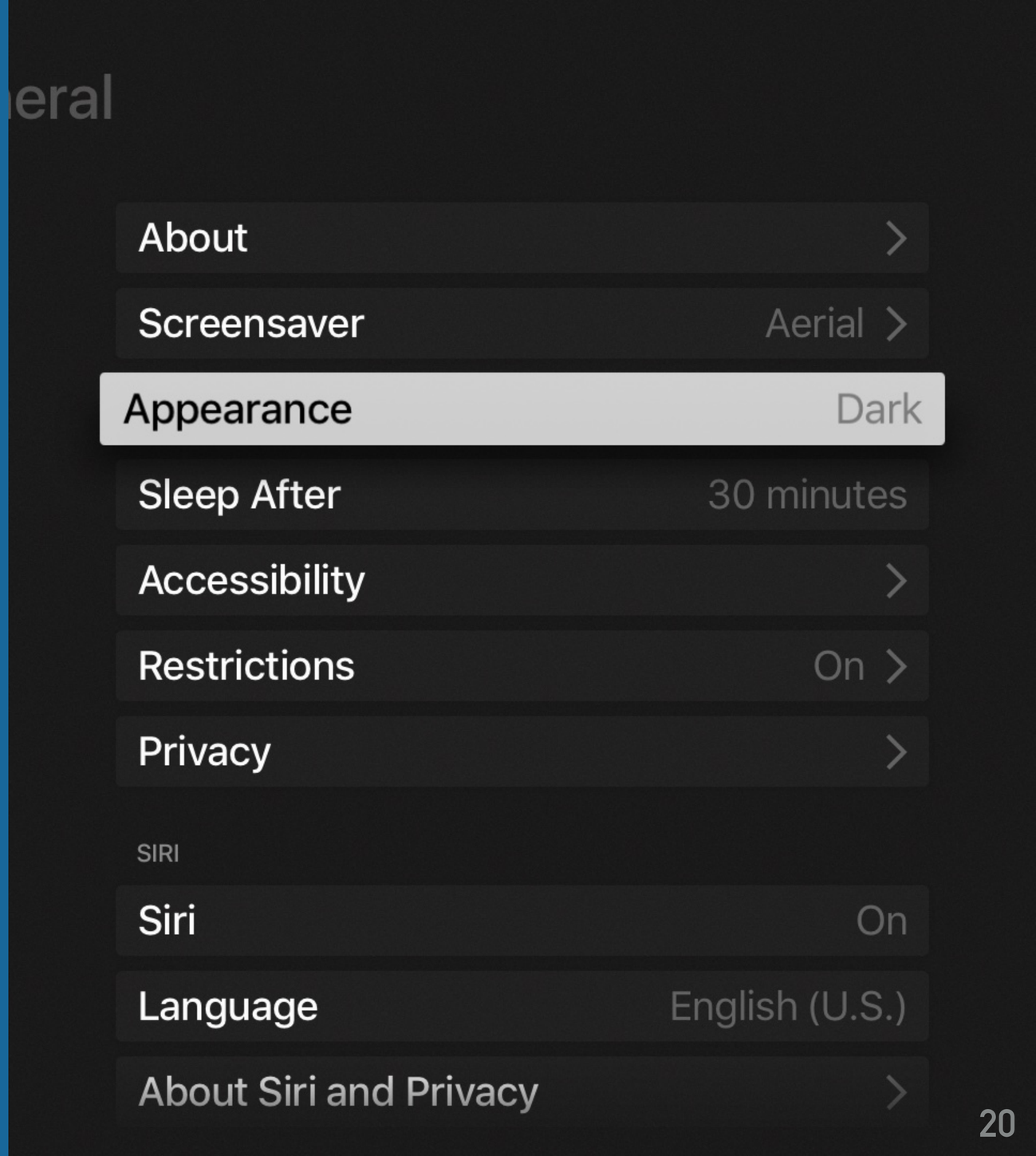
APPLE TV

- tvOS 10.0+ has dark and light themes
  - Think about your icons and UI

APPLE TV

SUBTITLE

- `UIUserInterfaceStyle` property in your apps info.plist to either Dark or Automatic



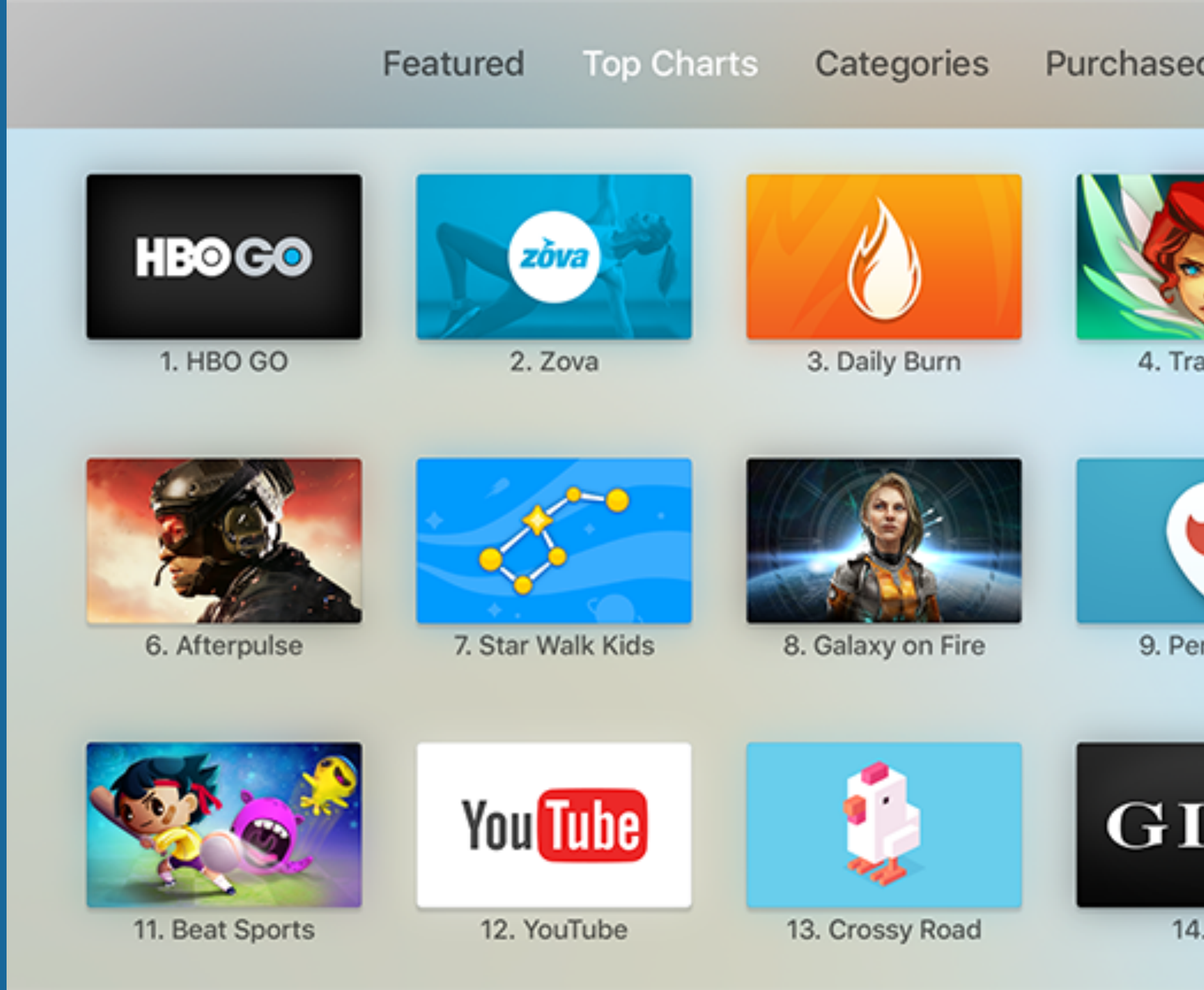
APPLE TV

```
let light = UITraitCollection(userInterfaceStyle: .light)
let dark = UITraitCollection(userInterfaceStyle: .dark)

UIButton.forTraitCollection(light).setTitleColor(.red(), for: [])
UIButton.forTraitCollection(dark).setTitleColor(.blue(), for: [])
```

- Dynamically change your appearance

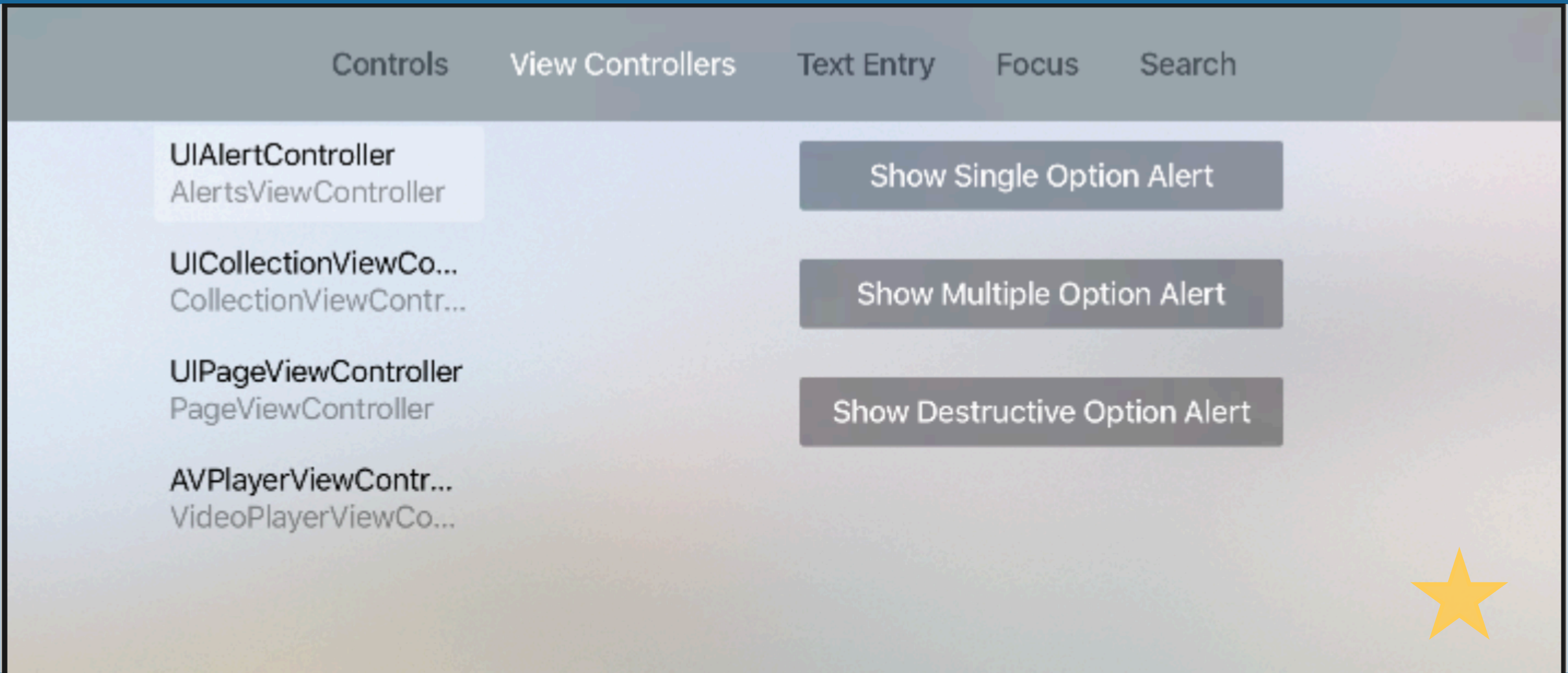
APPLE TV

- How to purchase tvOS apps
  - Directly from App Store on AppleTV
  - Universal purchasing enabled on linked iOS/tvOS app



TVOS CATALOG

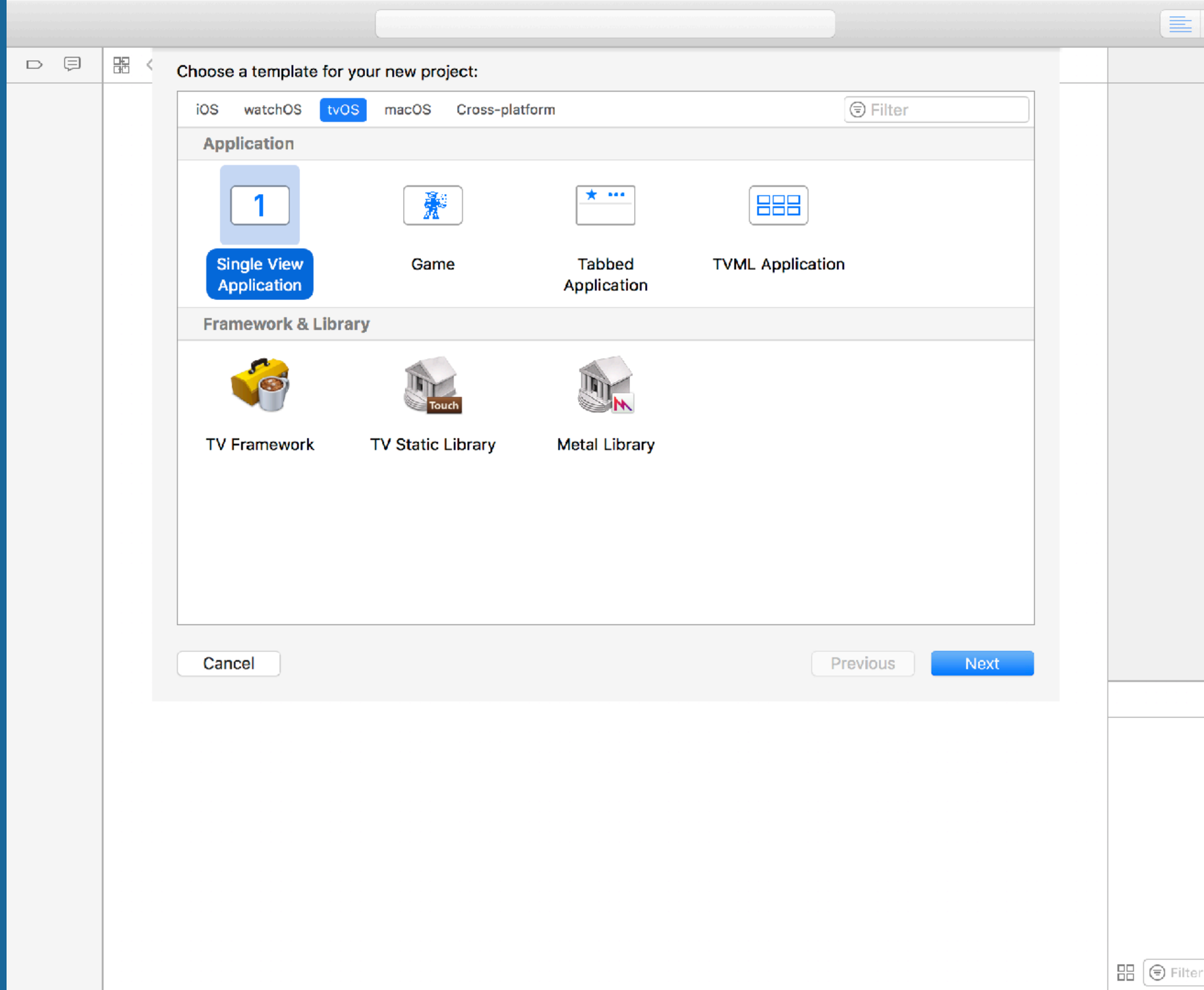
TVOS CATALOG



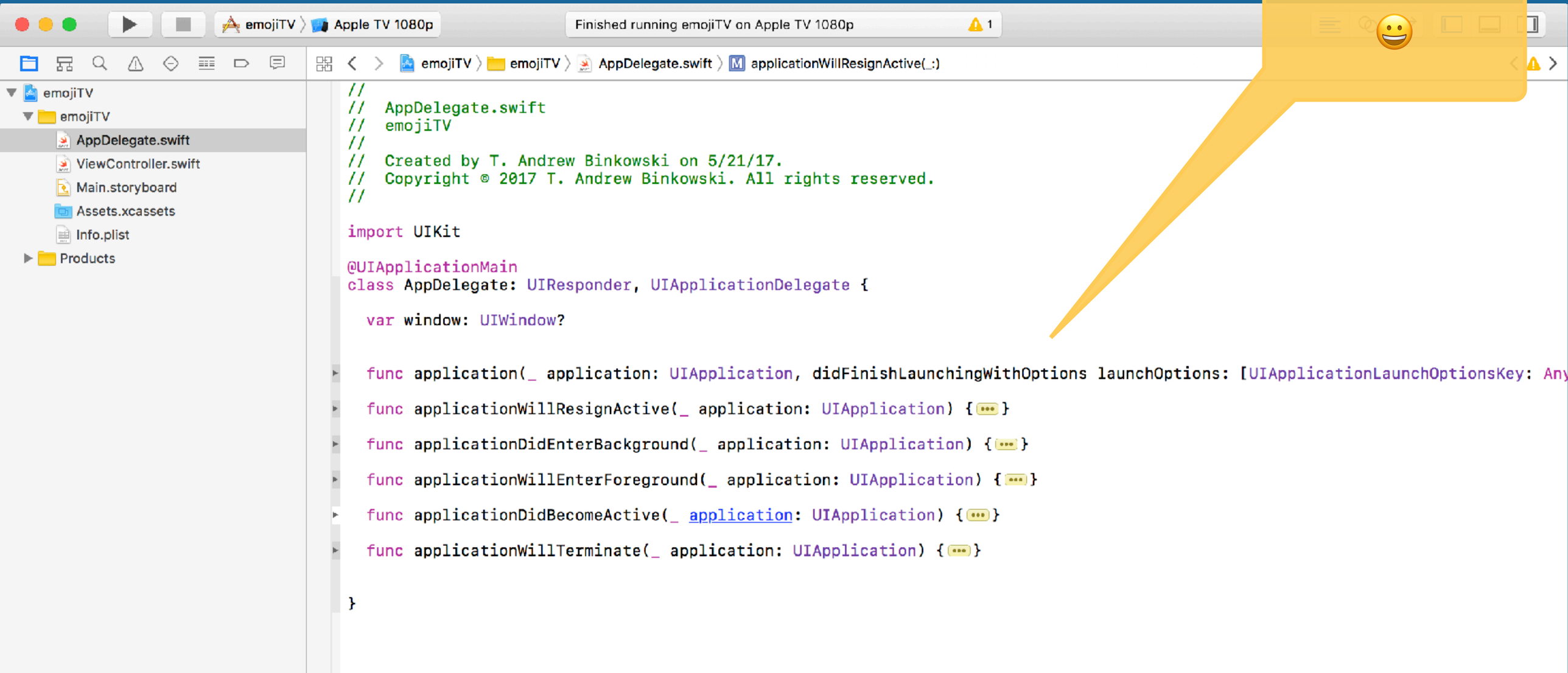
TVOS APPS

TVOS

- Xcode Templates



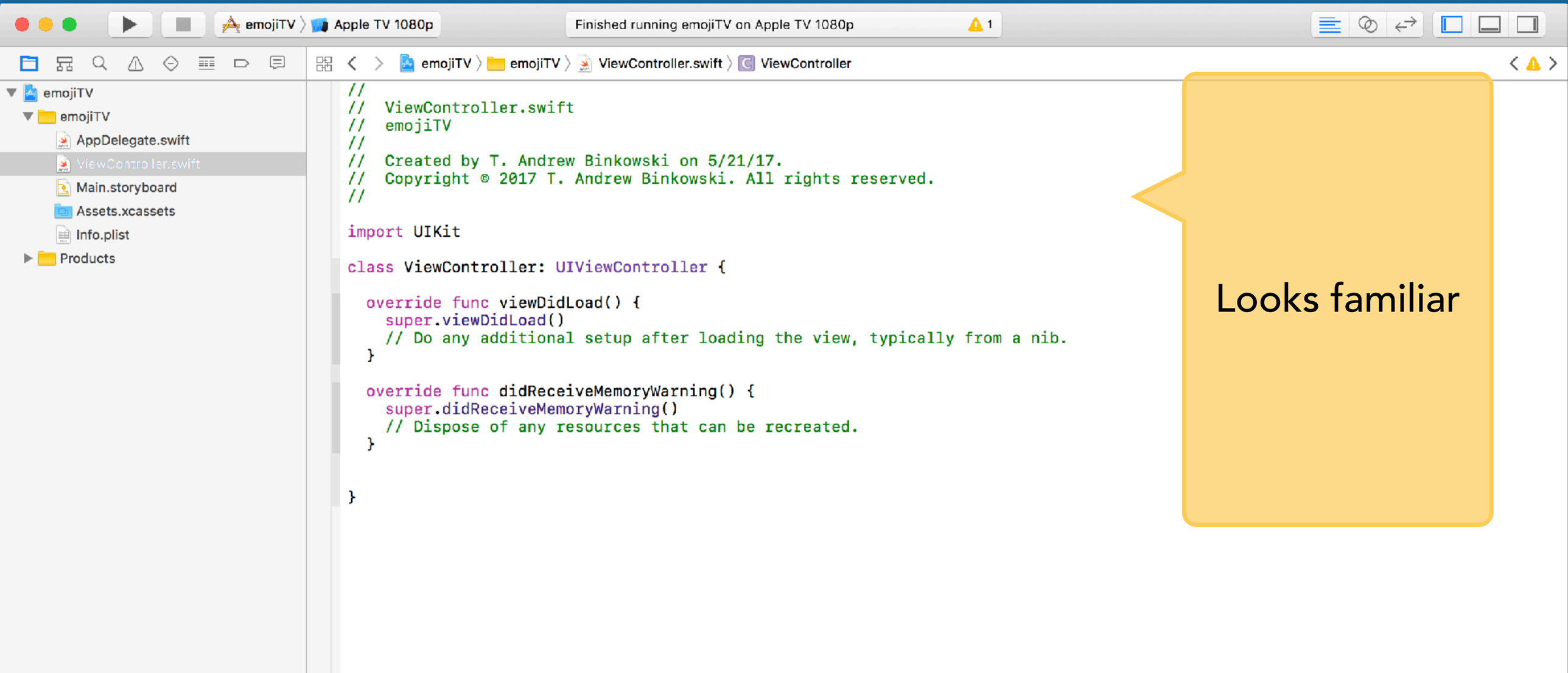
TVOS



Looks familiar

😀

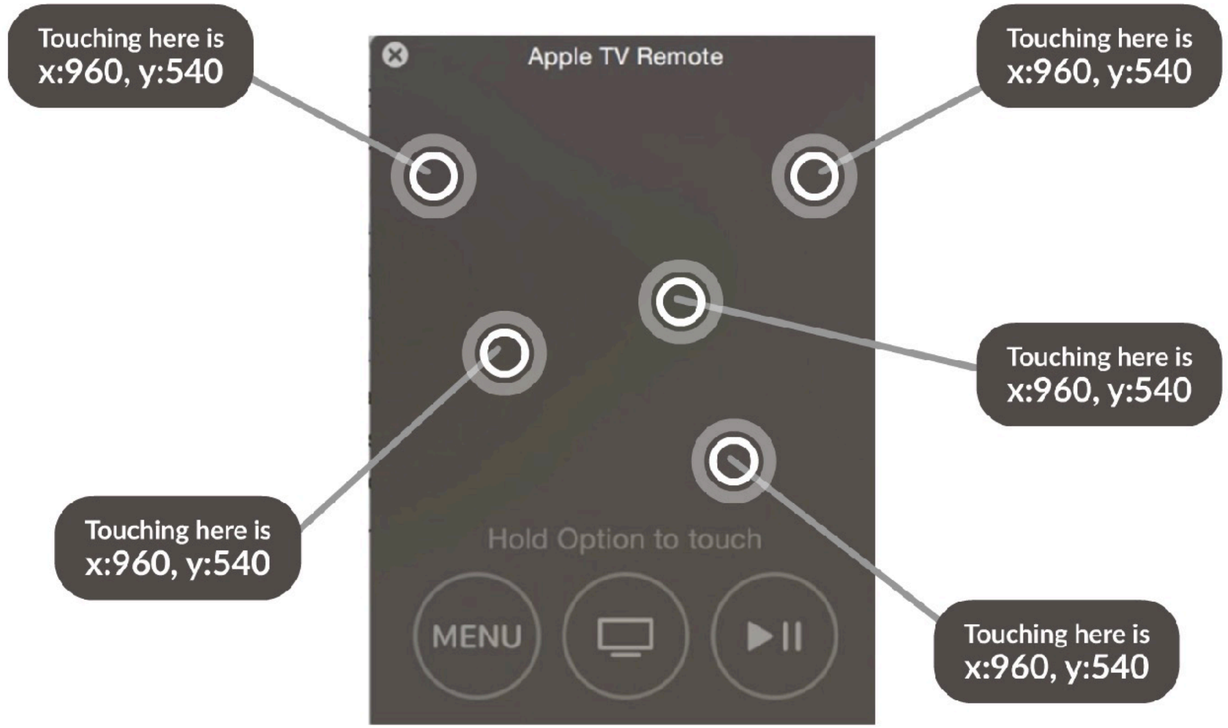
TVOS


emojiTV 〉 Apple TV 1080p
Finished running emojiTV on Apple TV 1080p
1
emojiTV 〉 emojiTV 〉 ViewController.swift 〉 ViewController
emojiTV
emojiTV
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
Info.plist
Products
//
// ViewController.swift
// emojiTV
//
// Created by T. Andrew Binkowski on 5/21/17.
// Copyright © 2017 T. Andrew Binkowski. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
Looks familiar

TOUCHES AND PRESSES

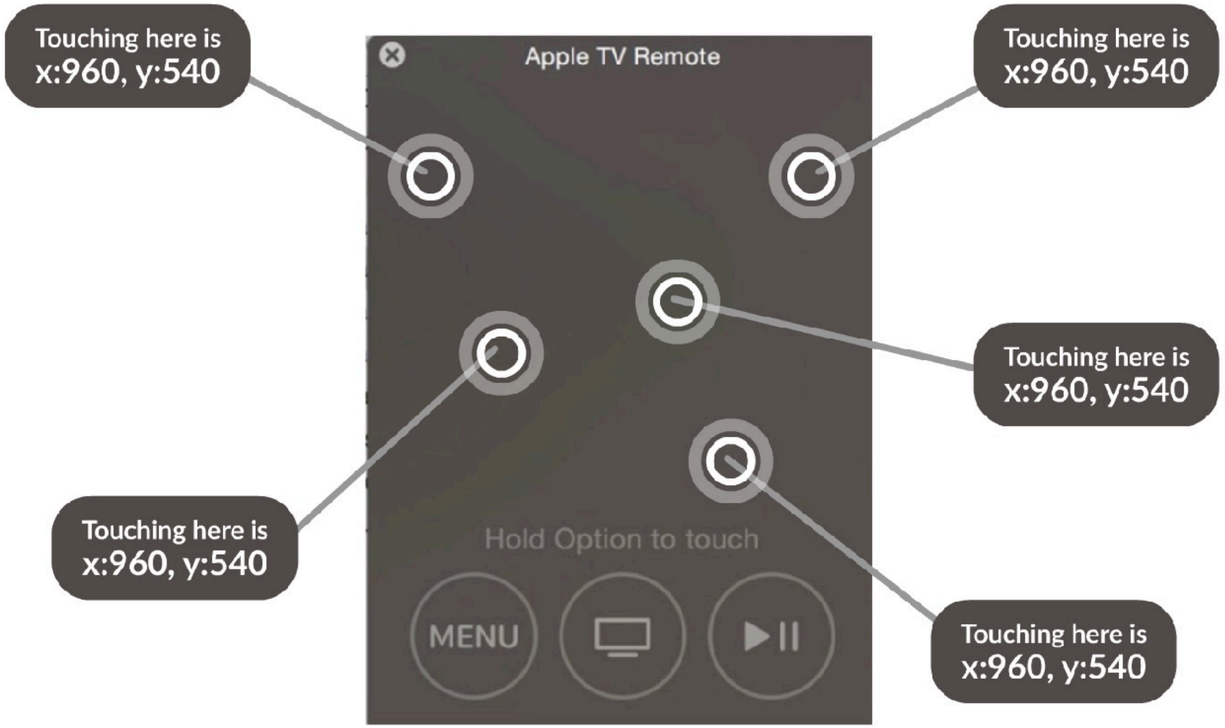
TOUCHES AND PRESSES

- UITouch represents contact between the user's finger and the touch surface
  - Just like the iOS version
  - UITouchTypeIndirect
- Begins centered in the focused view
  - No absolute coordinates on the touch surface



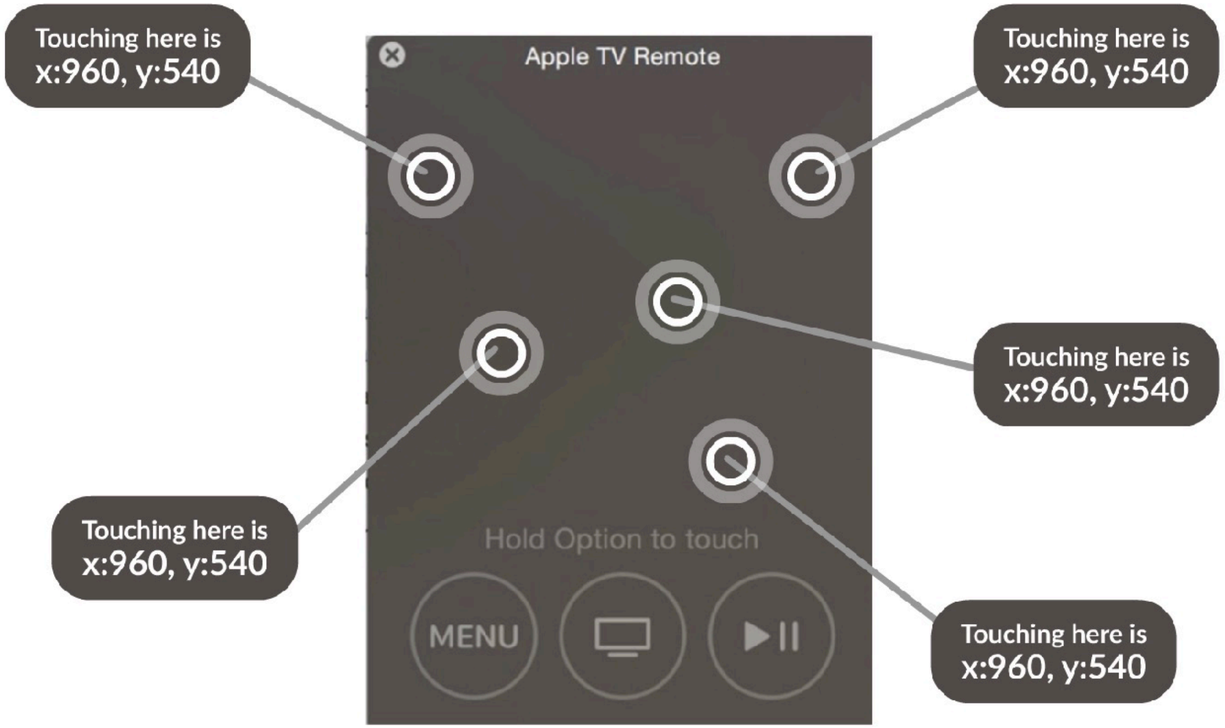
TOUCHES AND PRESSES

- Do not want a mouse/pointer type interaction



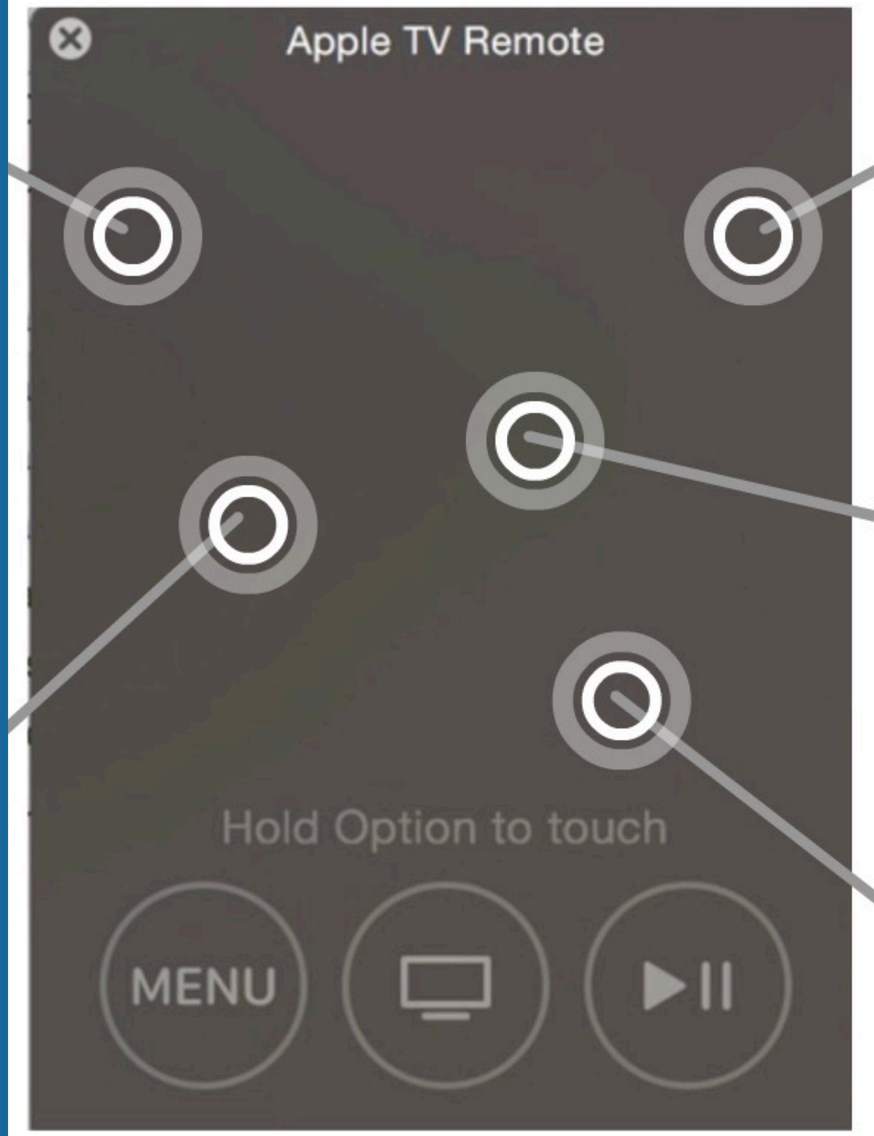
TOUCHES AND PRESSES

- All UIView methods respond
  - TouchesBegan
  - TouchesEnded
  - …



TOUCHES AND PRESSES

- UIPress represents the up or down state of a physical button
- May be pressure-sensitive
- UIGestureRecognizer



TOUCHES AND PRESSES

```
func pressesBegan(_ presses: Set<UIPress>, with event: UIPressesEvent?) {}
func pressesChanged(_ presses: Set<UIPress>, with event: UIPressesEvent?) {}
func pressesEnded(_ presses: Set<UIPress>, with event: UIPressesEvent?) {}
func pressesCancelled(_ presses: Set<UIPress>, with event: UIPressesEvent?) {}
```

TOUCHES AND PRESSES

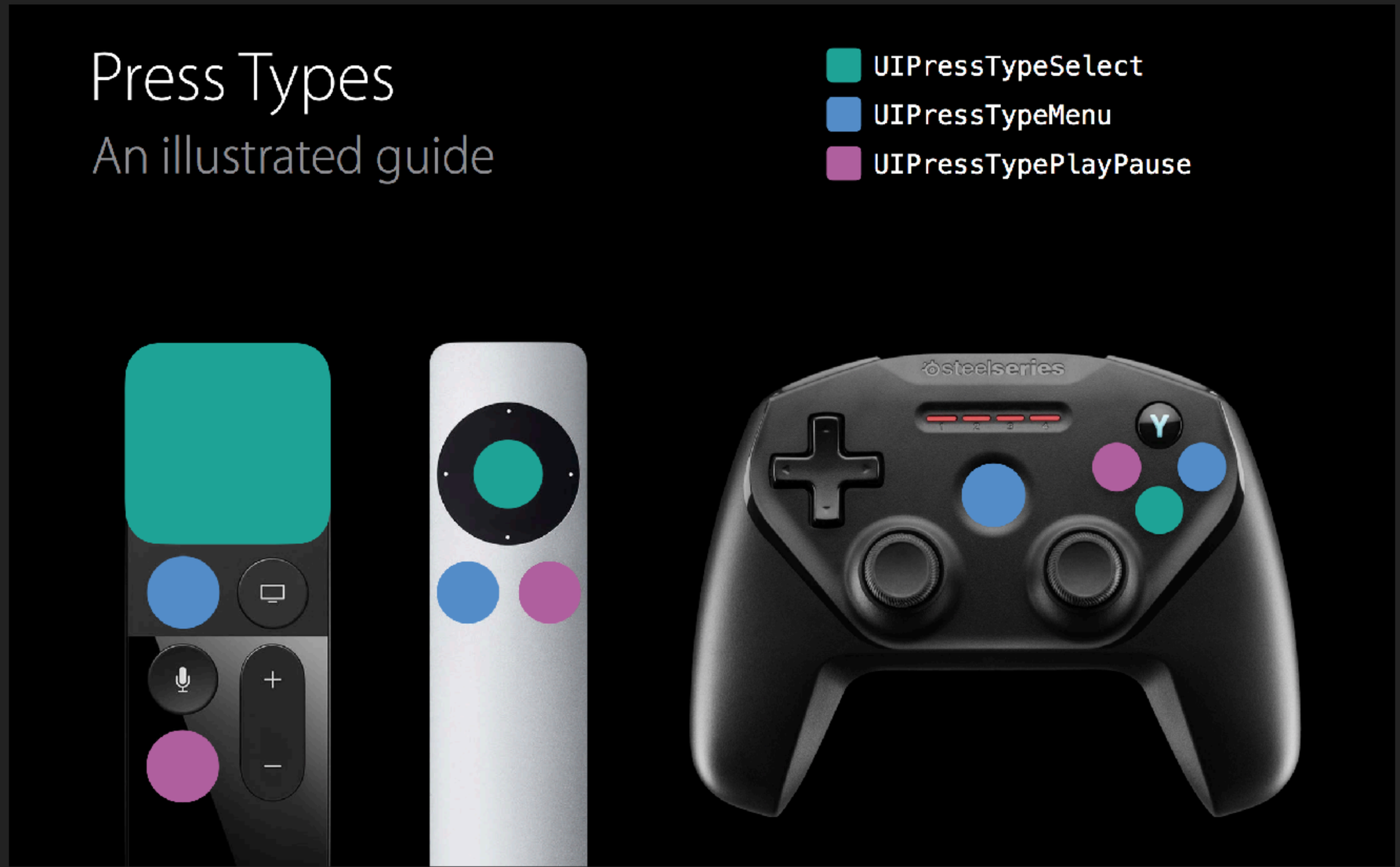
Press Types

An illustrated guide

UIPressTypeSelect

TOUCHES AND PRESSES

TOUCHES AND PRESSES



TOUCHES AND PRESSES

Press Types

An illustrated guide

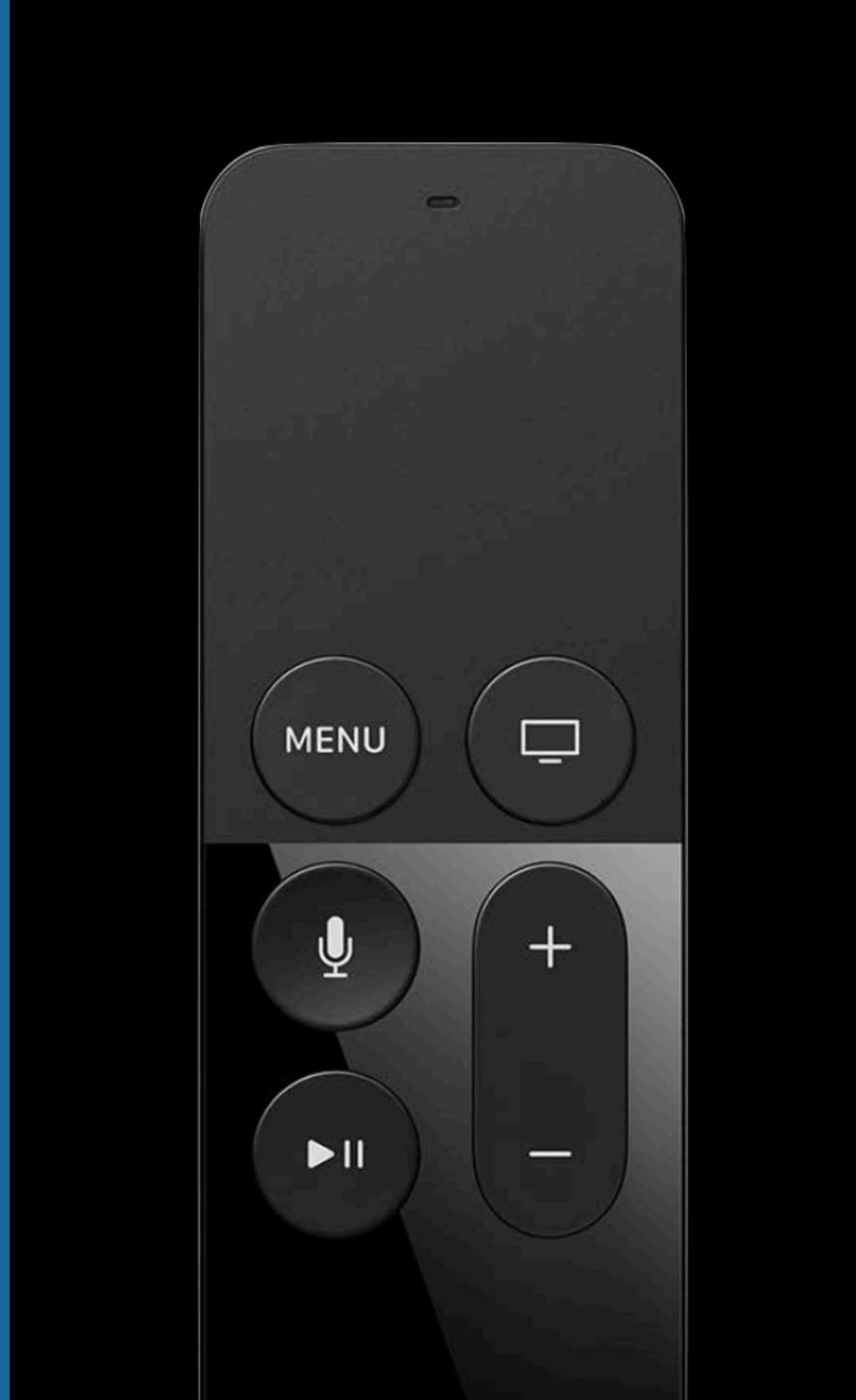
- UIPressTypeUpArrow
- UIPressTypeDownArrow
- UIPressTypeLeftArrow
- UIPressTypeRightArrow

TOUCHES AND PRESSES

```
// Tap on Play-Pause button.
let playPauseTap = UITapGestureRecognizer(target: self, action:
   #selector(MyClass.handlePlayPause(_:)))
playPauseTap.allowedPressTypes = [ UIPressType.playPause.rawValue ]

// Long-press on Select button.
let selectLongPress = UILongPressGestureRecognizer(target: self, action:
   #selector(MyClass.handleSelectLongPress(_:)))
selectLongPress.allowedPressTypes = [ UIPressType.select.rawValue ]

// Double-tap on Select button.
let selectDoubleTap = UITapGestureRecognizer(target: self, action:
   #selector(MyClass.handleSelectDoubleTap(_:)))
selectDoubleTap.allowedPressTypes = [ UIPressType.select.rawValue ]
selectDoubleTap.numberOfTapsRequired = 2
```

TOUCHES AND PRESSES

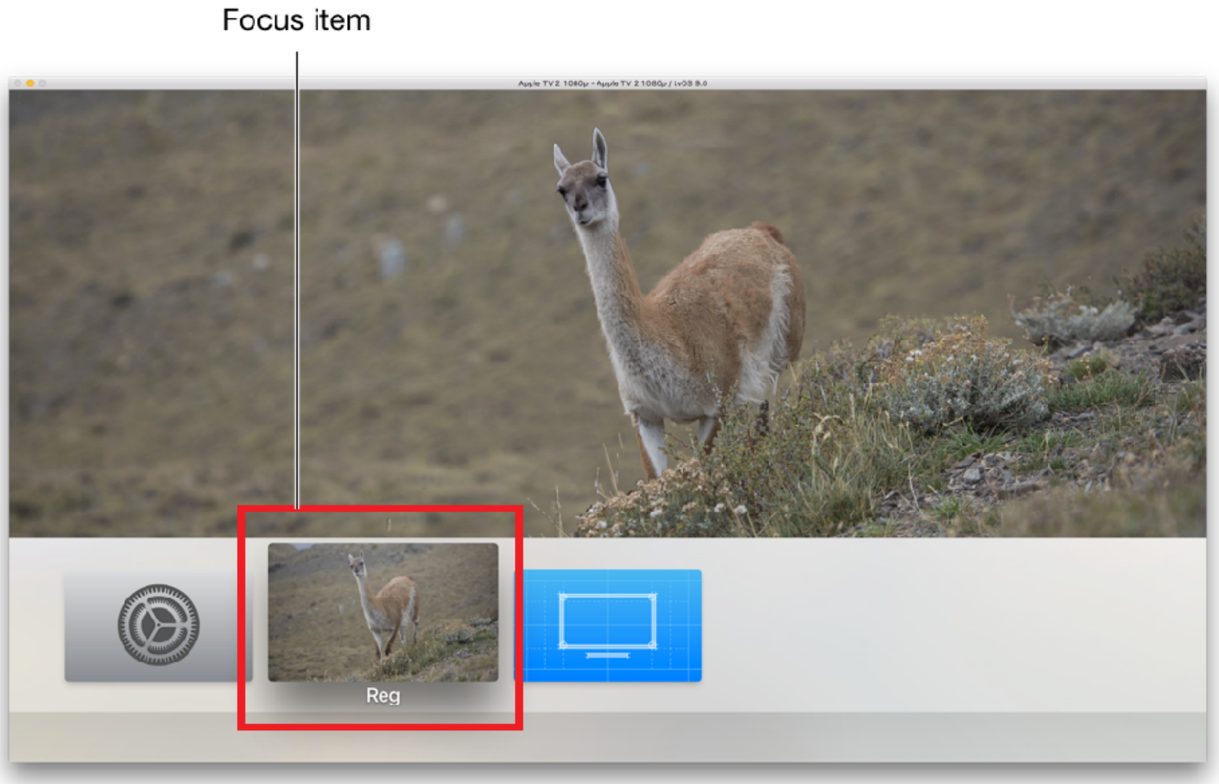
- Users must be able to exit your app using the Menu button
- This is a requirement that App Review specifically looks for
- The event must reach `UIApplication pressesEnded:withEvent:` in order for the app to exit



FOCUS

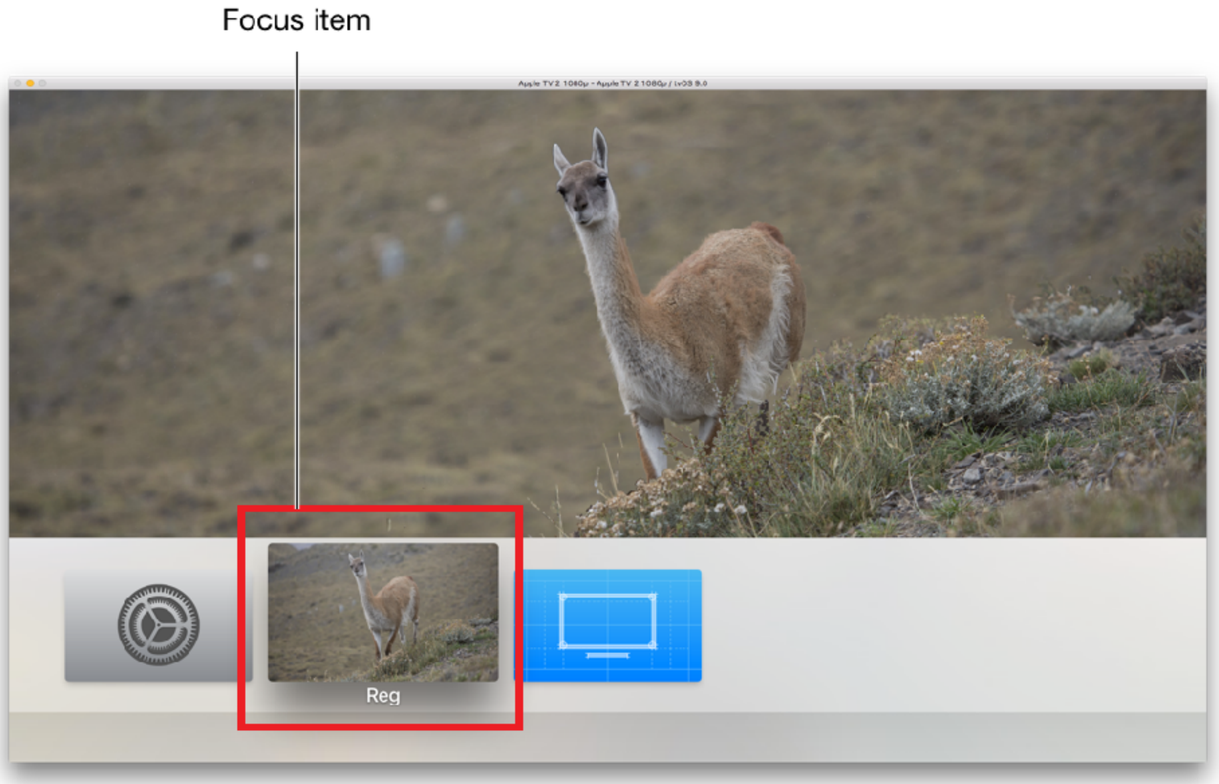
TVOS APPS

- Users interact with AppleTV using remote, game controller, iOS app
- As users navigate to items they are "focused"
  - Highlighted effects
- Focused items receive "action" when triggered



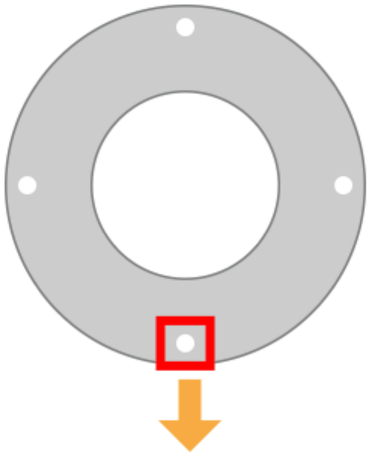
TVOS APPS

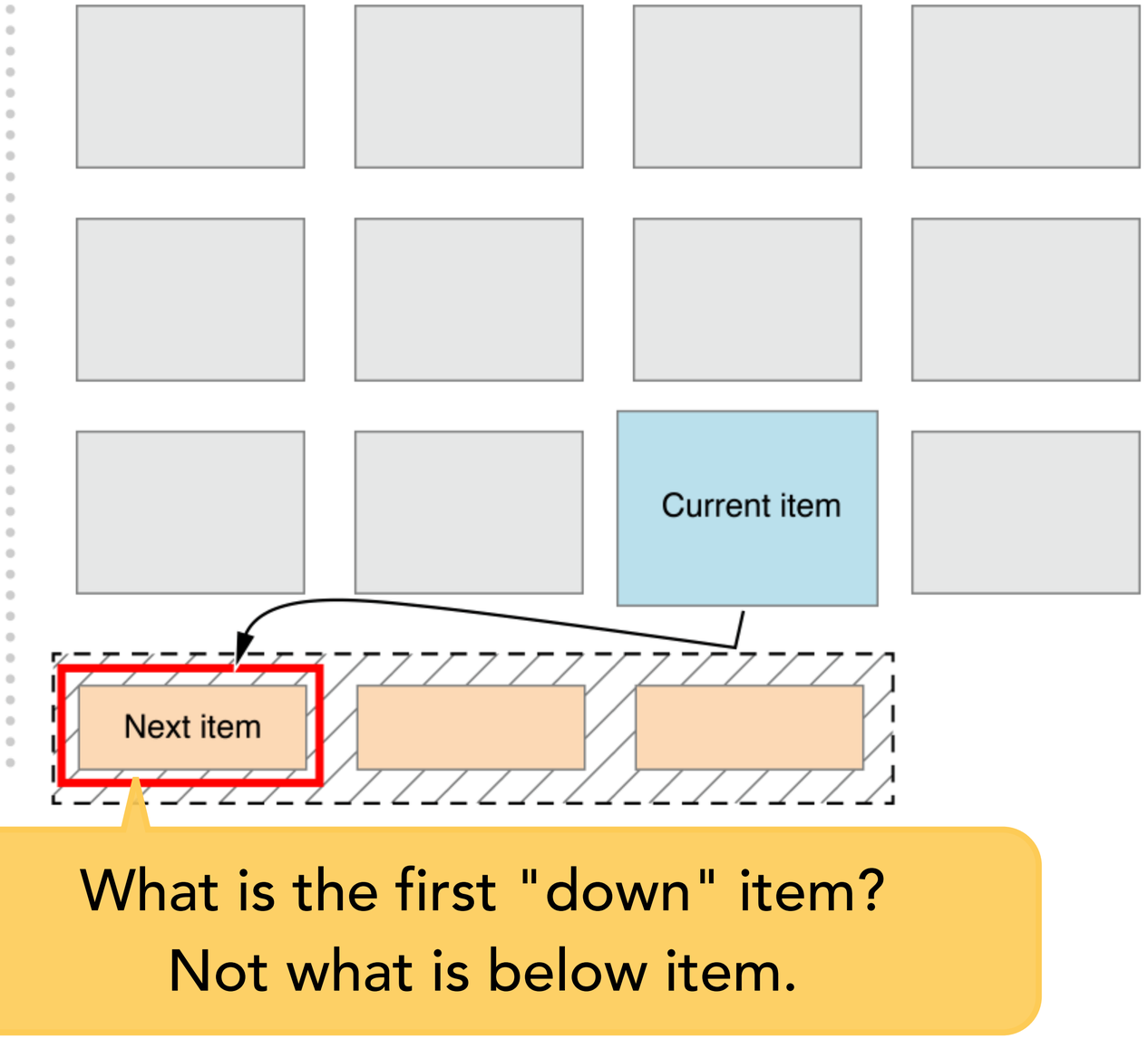
- Focus based interfaces are provided automatically (for the most part)
  - Views that make sense (eg. buttons)
- Cannot move focus programmatically
  - User in control



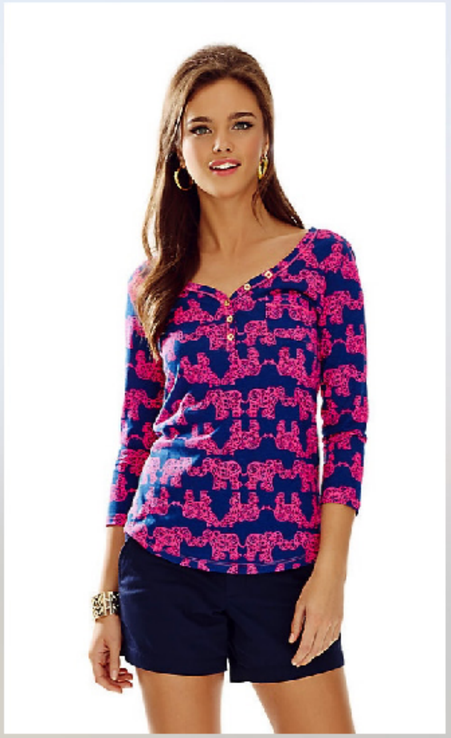
TVOS APPS

- The focus engine only responds to user movements
- Movements are restricted to prevent wandering around the screen





TVOS APPS

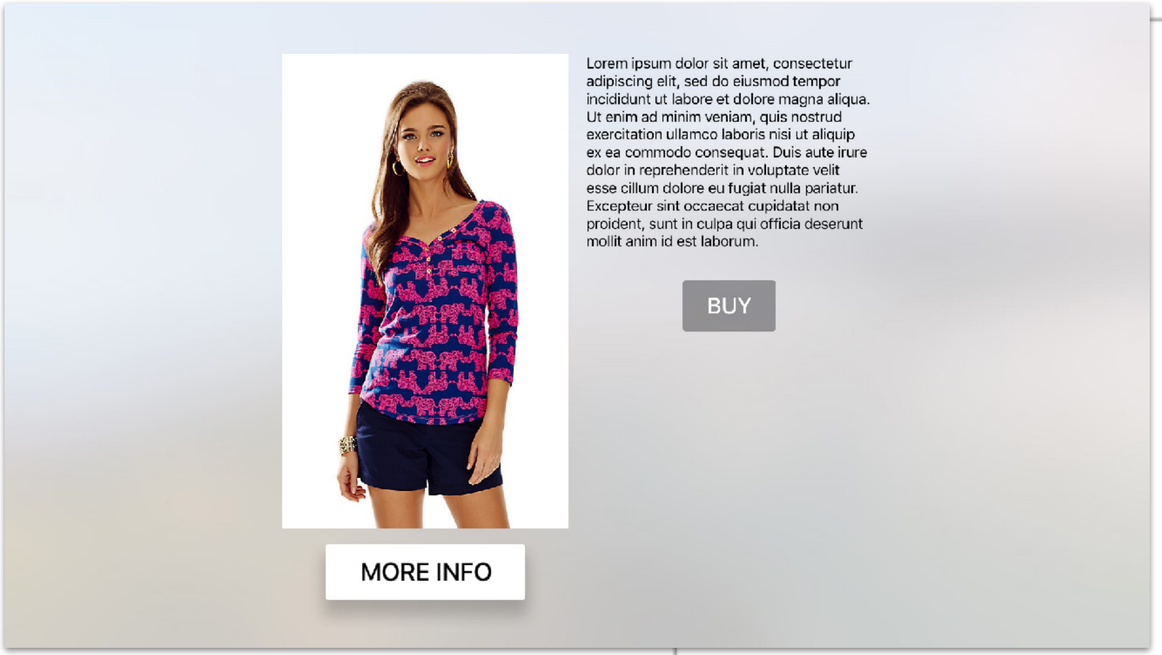


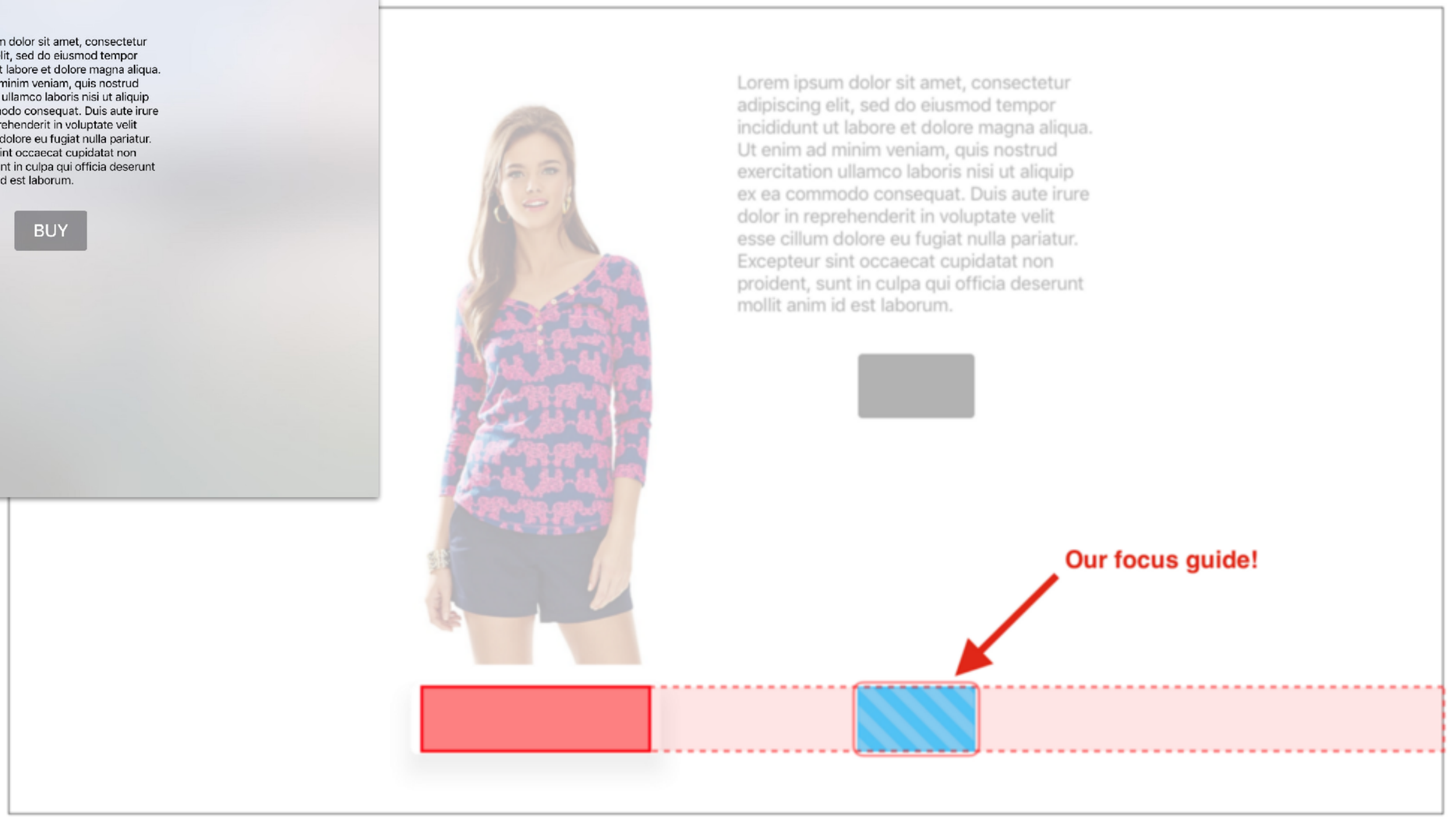
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

BUY

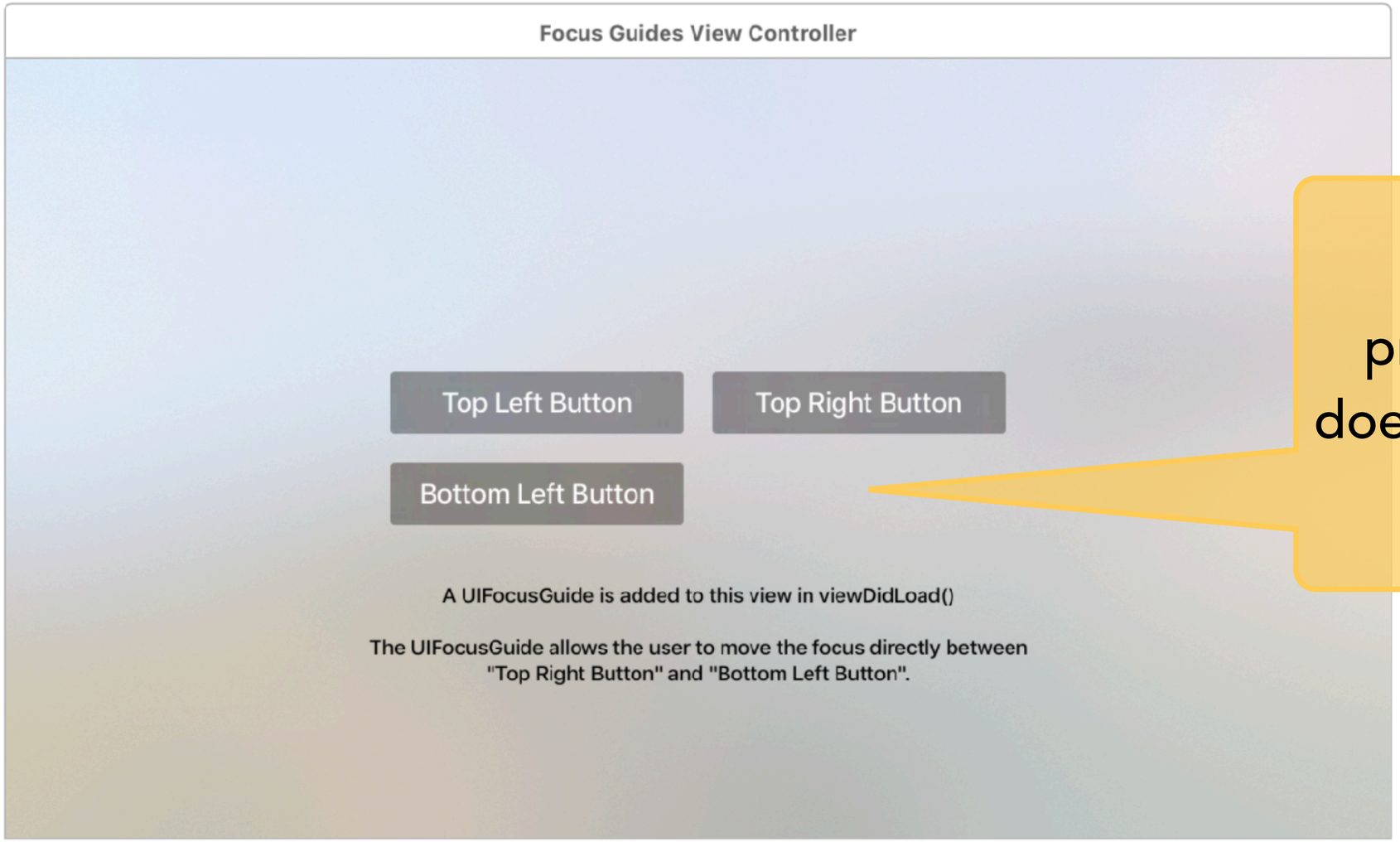
MORE INFO

TVOS APPS


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
BUY
MORE INFO

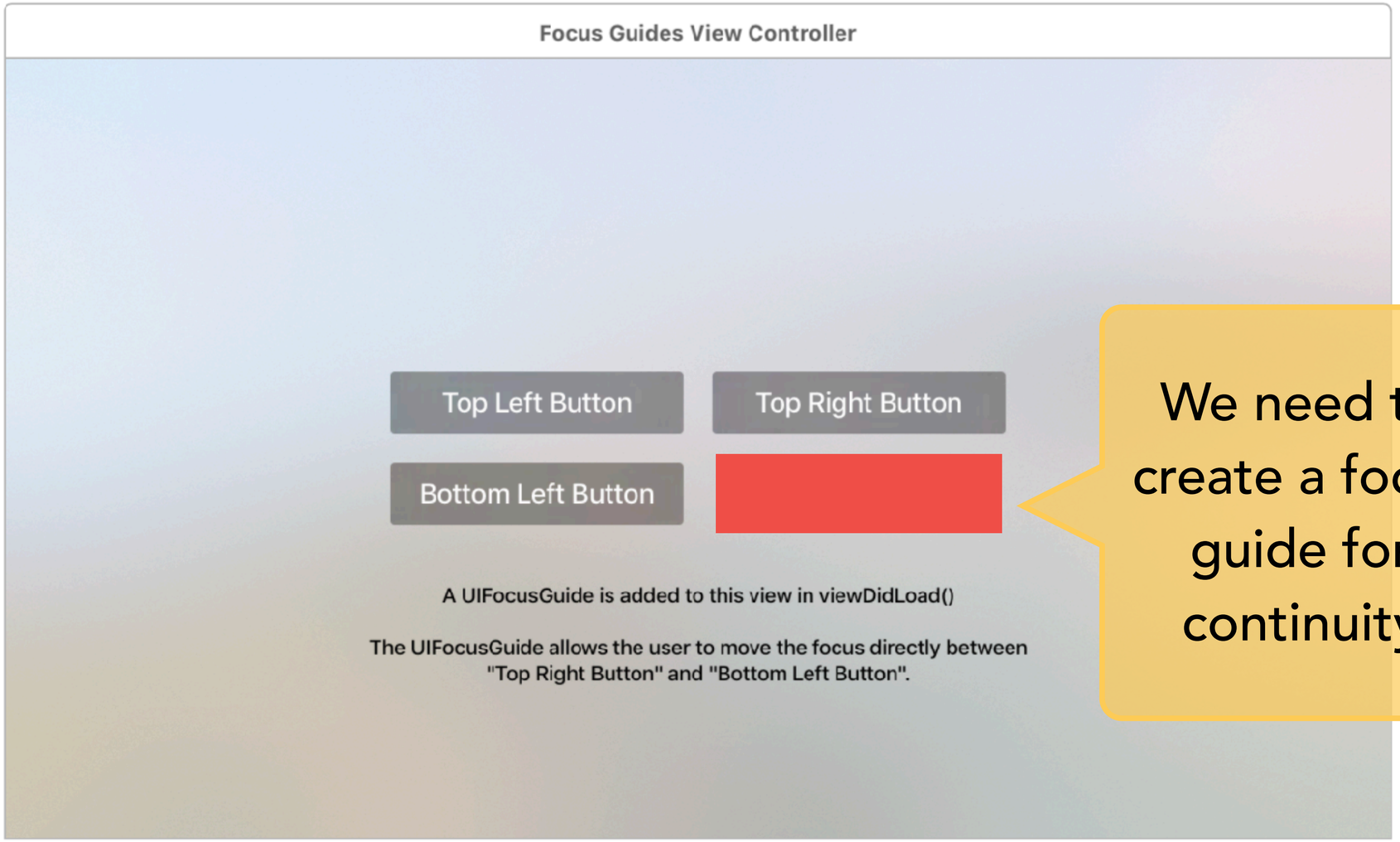

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Our focus guide!

TVOS APPS



Focus preference doesn't let you go here

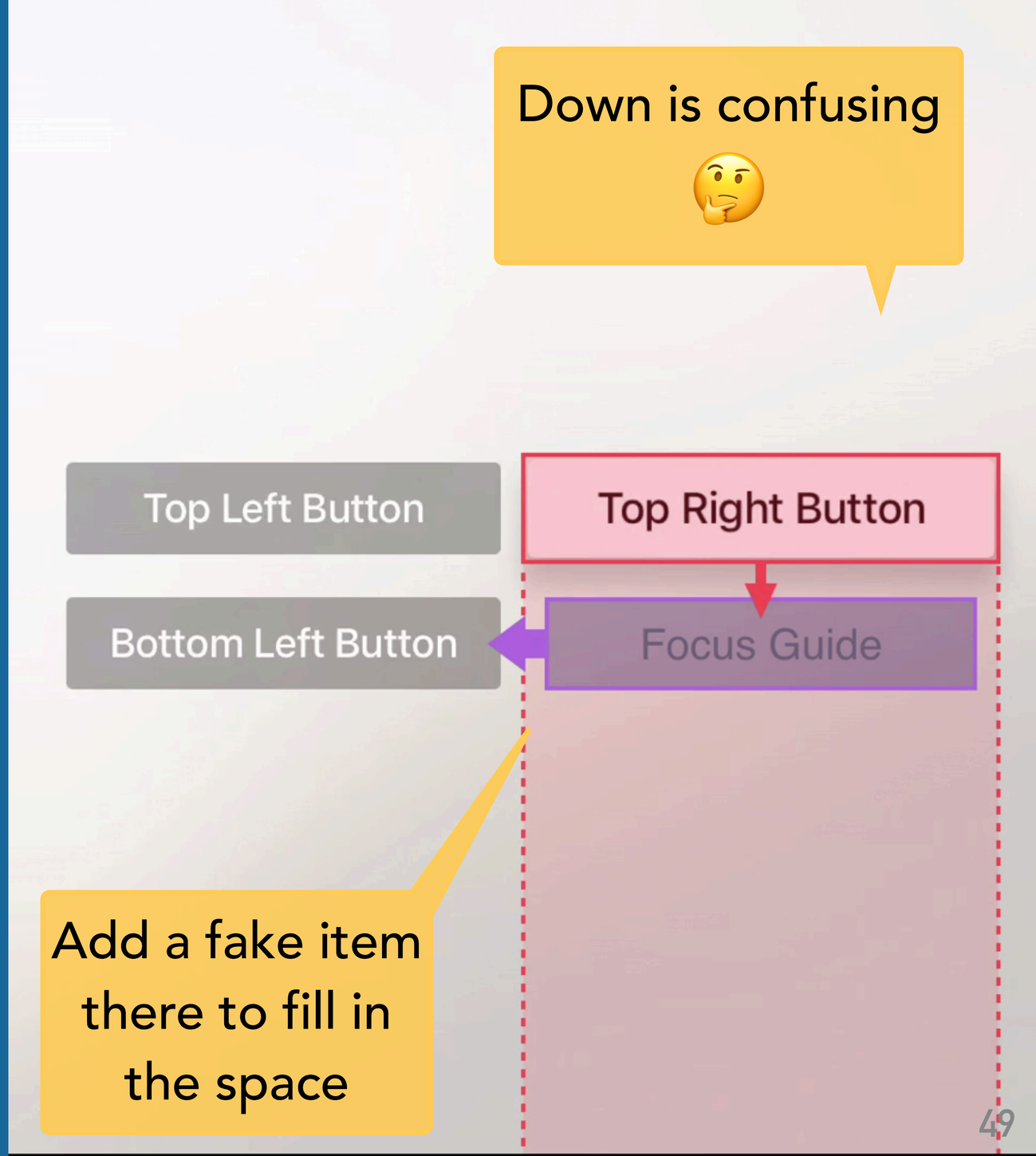
TVOS APPS


Focus Guides View Controller
Top Left Button
Top Right Button
Bottom Left Button
A UIFocusGuide is added to this view in viewDidLoad()
The UIFocusGuide allows the user to move the focus directly between "Top Right Button" and "Bottom Left Button".
We need to create a focus guide for continuity

TVOS APPS

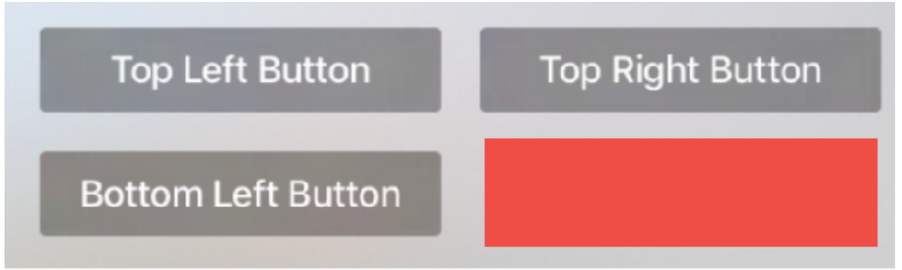
SUBTITLE

- `Focus guide` allows you to create alignments to conform to the focus preferences
  - Typically not visually aligned



FOCUS

```
/*
    Create a focus guide to fill the gap below button 2 and to the right
    of button 3.
*/

focusGuide = UIFocusGuide()
view.addLayoutGuide(focusGuide)

// Anchor the top left of the focus guide.
focusGuide.leftAnchor.constraint(equalTo: topRightButton.leftAnchor).isActive = true
focusGuide.topAnchor.constraint(equalTo: bottomLeftButton.topAnchor).isActive = true

// Anchor the width and height of the focus guide.
focusGuide.widthAnchor.constraint(equalTo: topRightButton.widthAnchor).isActive = true
focusGuide.heightAnchor.constraint(equalTo: bottomLeftButton.heightAnchor).isActive = true
```



- Add a custom focus guide

TVOS APPS

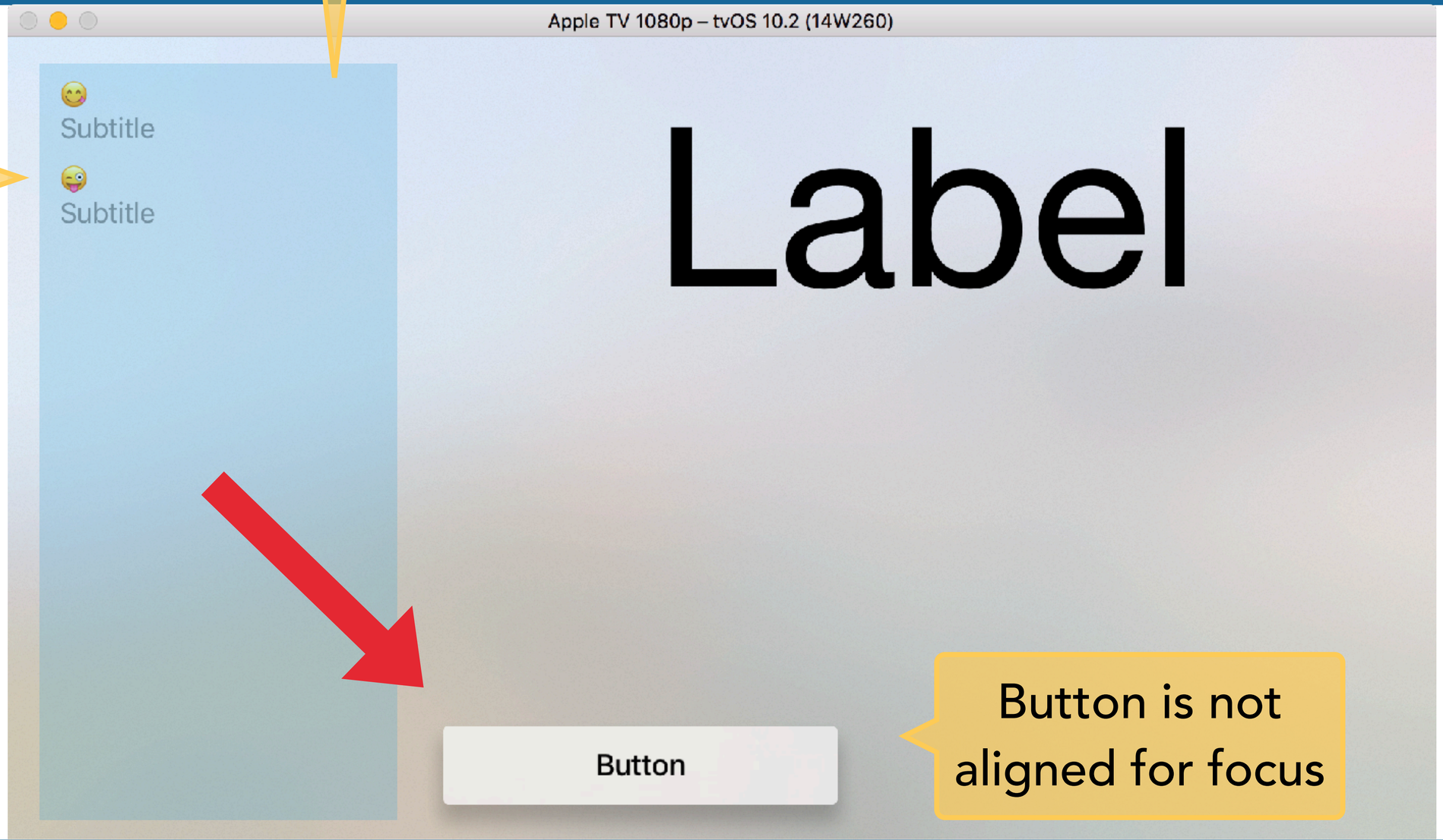
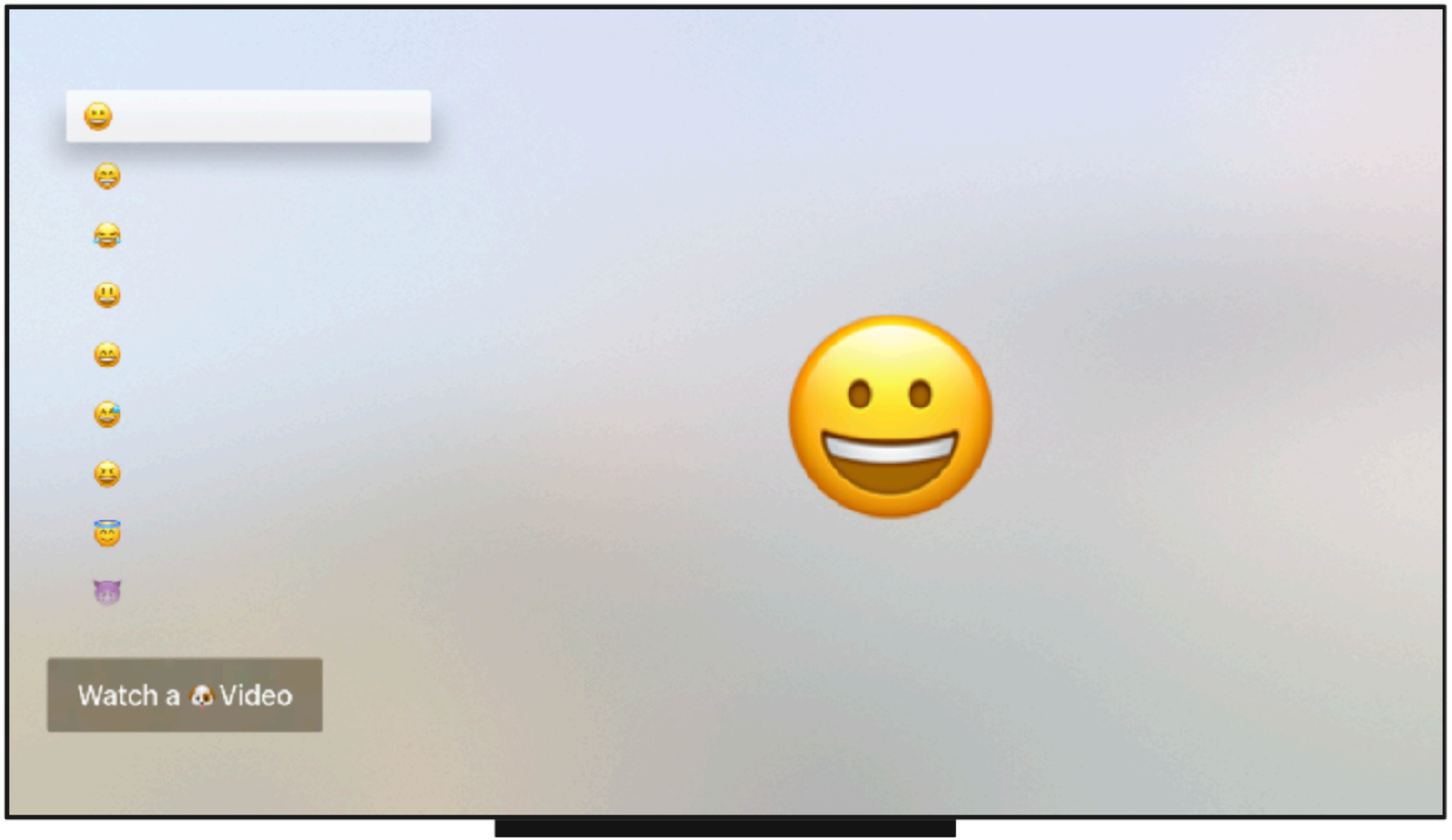
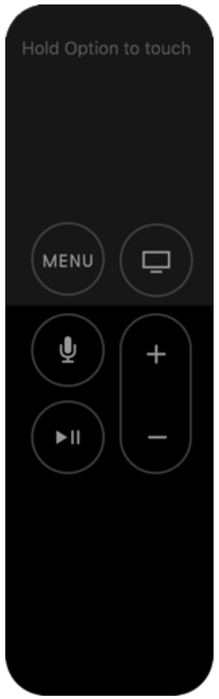


Table
Apple TV 1080p – tvOS 10.2 (14W260)
Subtitle
Subtitle
Label
Aligned for focus
Button
Button is not aligned for focus

EMOJI TV

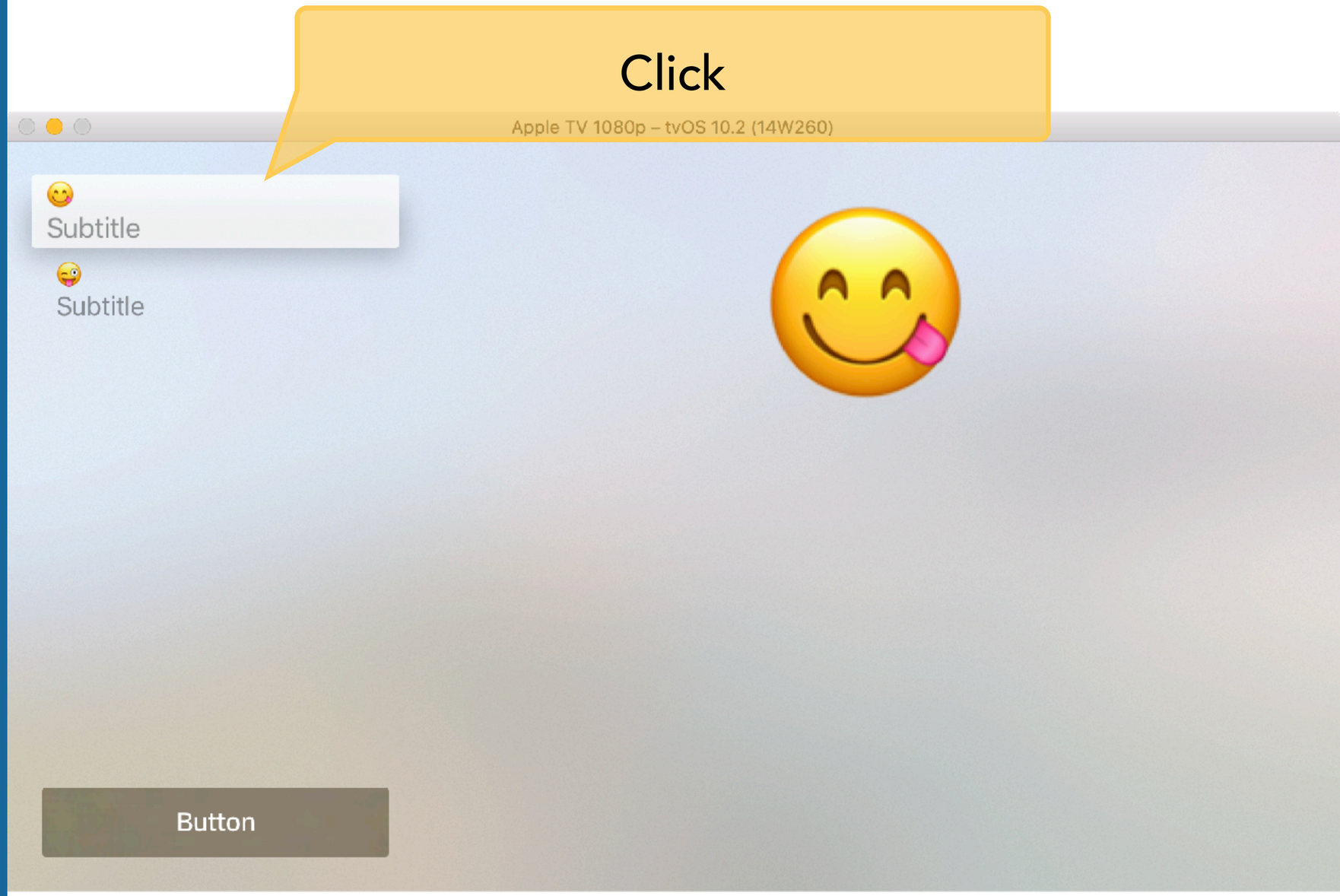
EMOJI TV





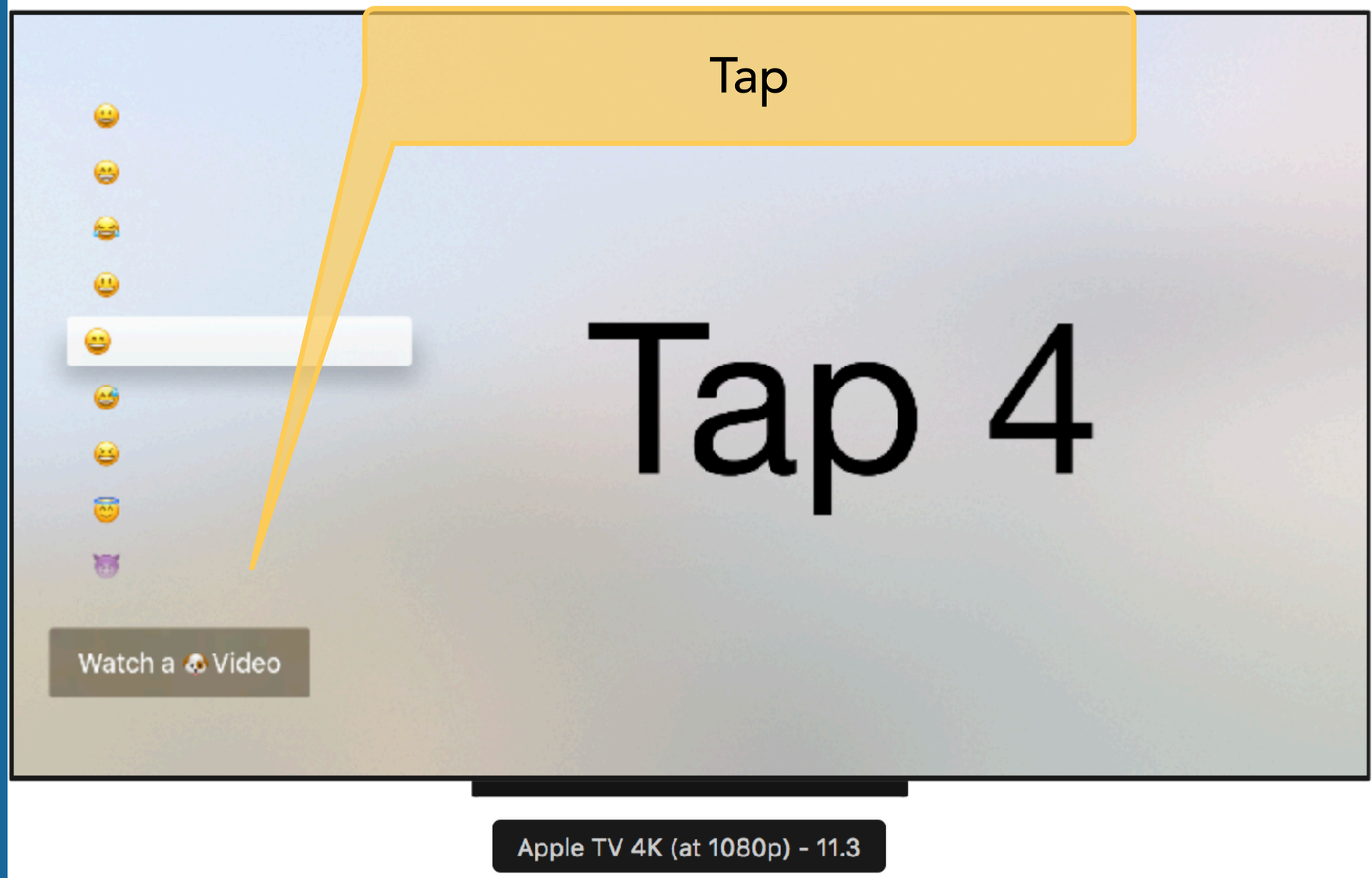
TVOS APPS

- Focus changes on a cell, update the label



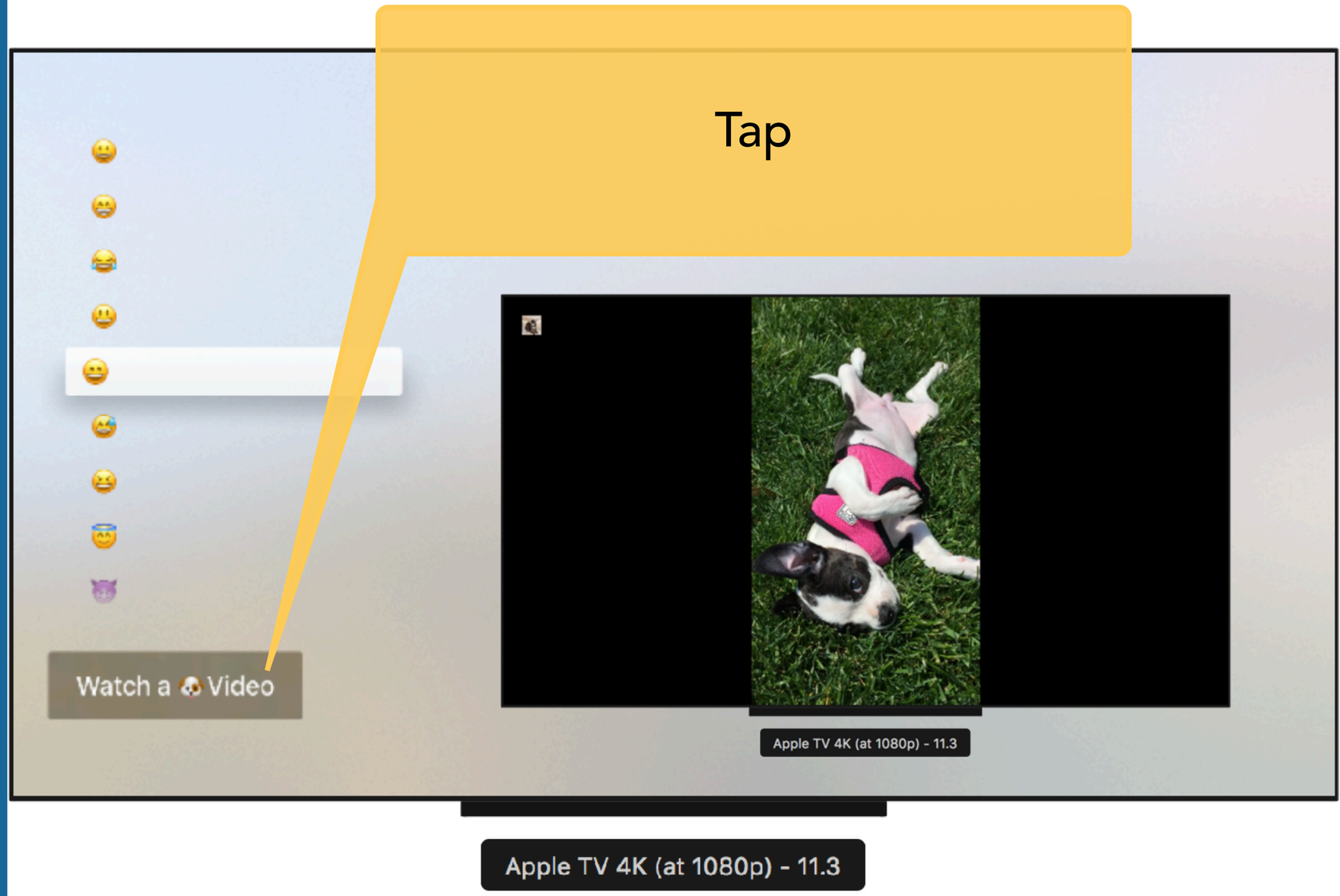
TVOS APPS

- Click on table cell show the row

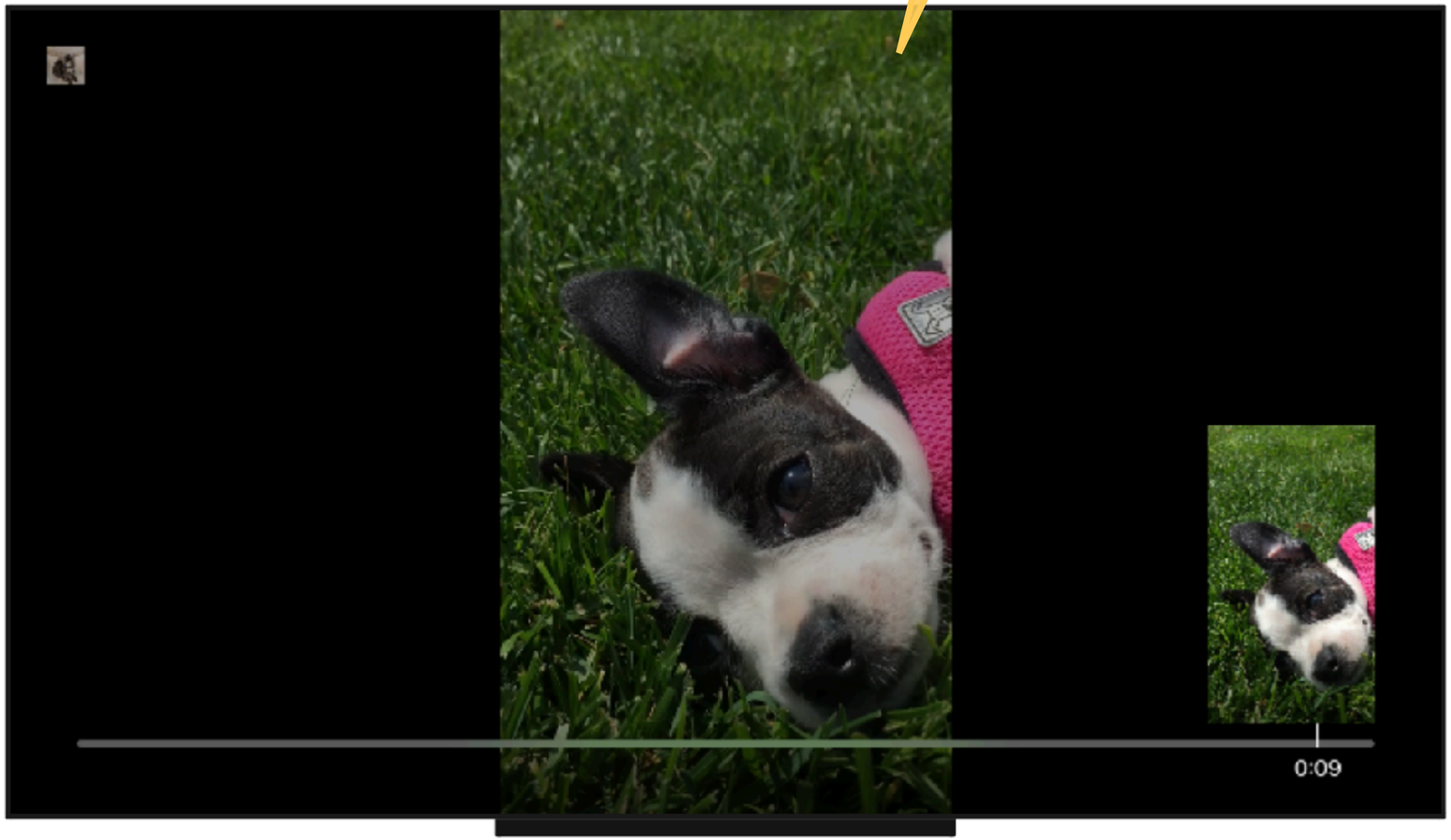


TVOS APPS

- Segue to View Controller to show a video



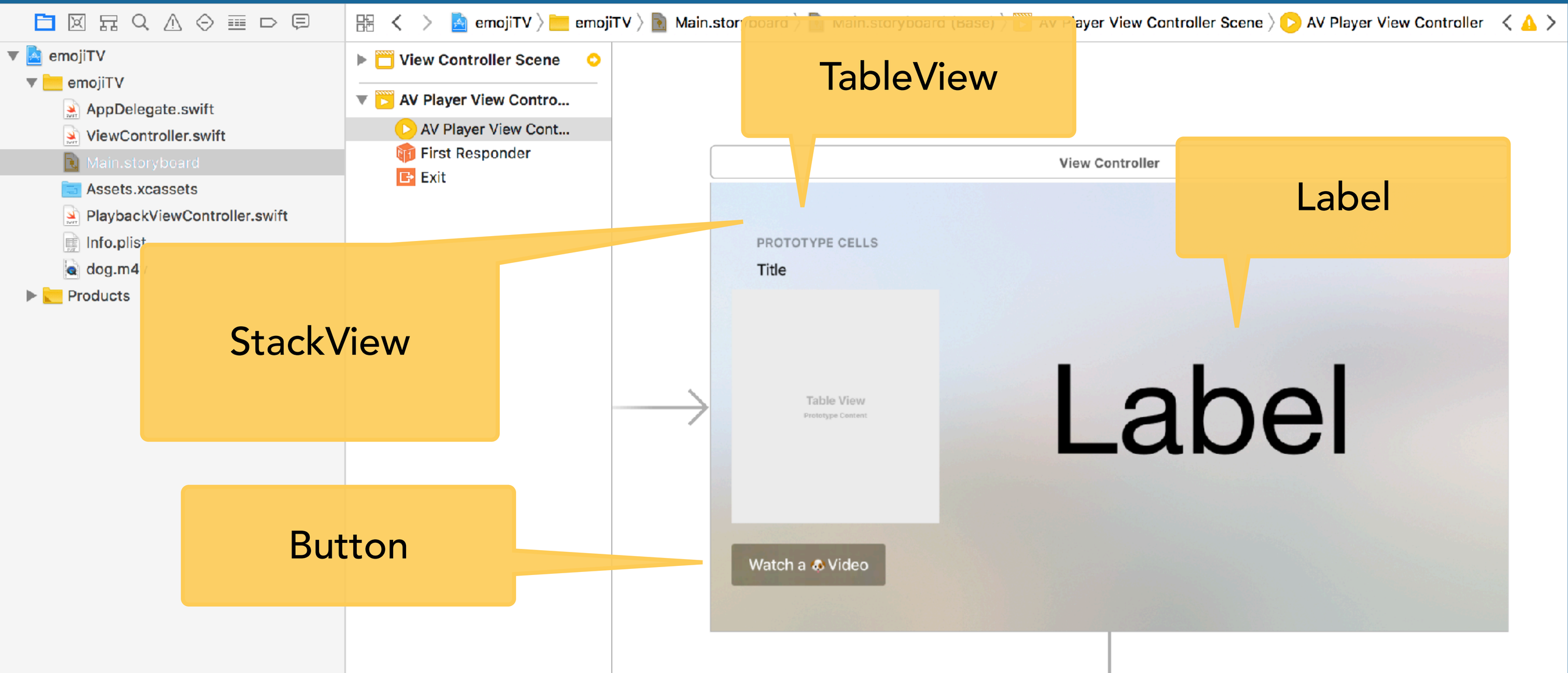
TVOS APPS



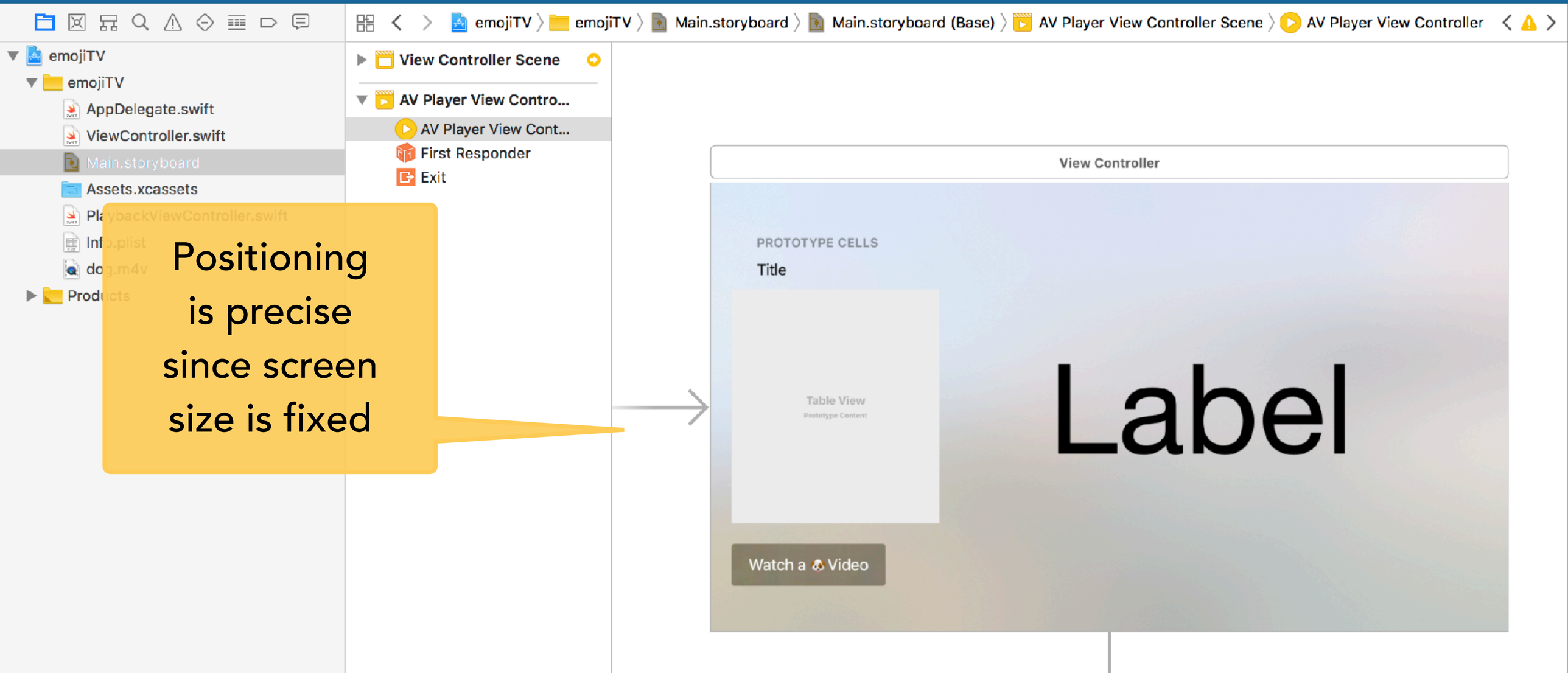
- AVPlayerViewController

STORYBOARD

TVOS APPS


emojiTV
emojiTV
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
PlaybackViewController.swift
Info.plist
dog.m4
Products
View Controller Scene
AV Player View Contro...
AV Player View Cont...
First Responder
Exit
TableView
Label
StackView
Button
View Controller
PROTOTYPE CELLS
Title
Table View
Prototype Content
Label
Watch a Video

TVOS APPS


emojiTV
emojiTV
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
Positioning is precise since screen size is fixed
emojiTV
emojiTV
Main.storyboard
Main.storyboard (Base)
AV Player View Controller Scene
AV Player View Controller
View Controller Scene
AV Player View Contro...
AV Player View Cont...
First Responder
Exit
View Controller
PROTOTYPE CELLS
Title
Table View
Prototype Content
Label
Watch a Video

TVOS APPS

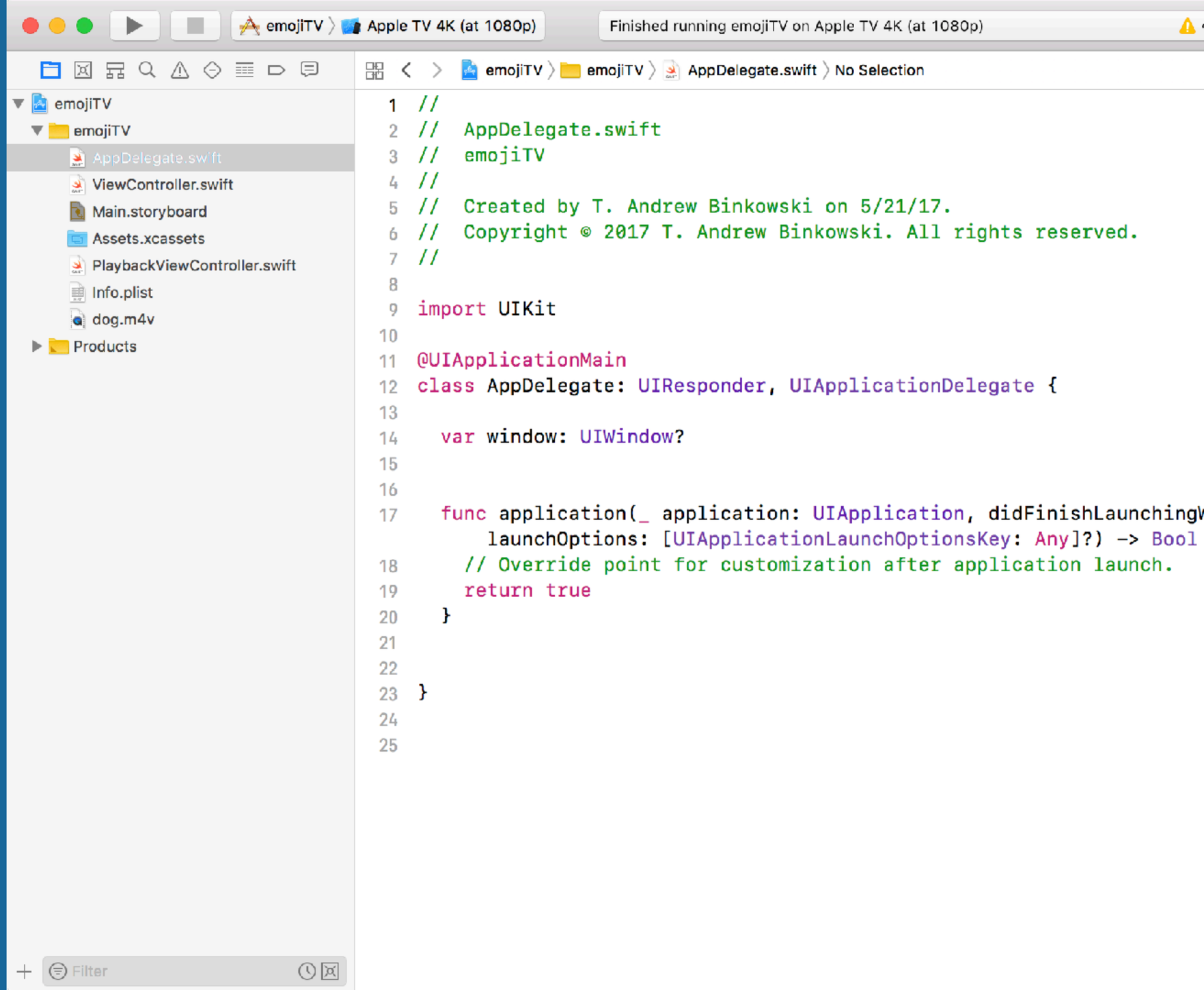
Nothing new

```
// MARK: - Actions
@IBAction func tapButton(_ sender: UIButton) {
  print("Tapped: \(String(describing: sender.titleLabel?.description))")
}

// MARK: - Outlets
@IBOutlet weak var bigLabel:UILabel!
@IBOutlet weak var button: UIButton!
@IBOutlet weak var tableView: UITableView! {
  didSet {
    self.tableView.delegate = self
    self.tableView.dataSource = self
  }
}
```

APP DELEGATE

TVOS APPS

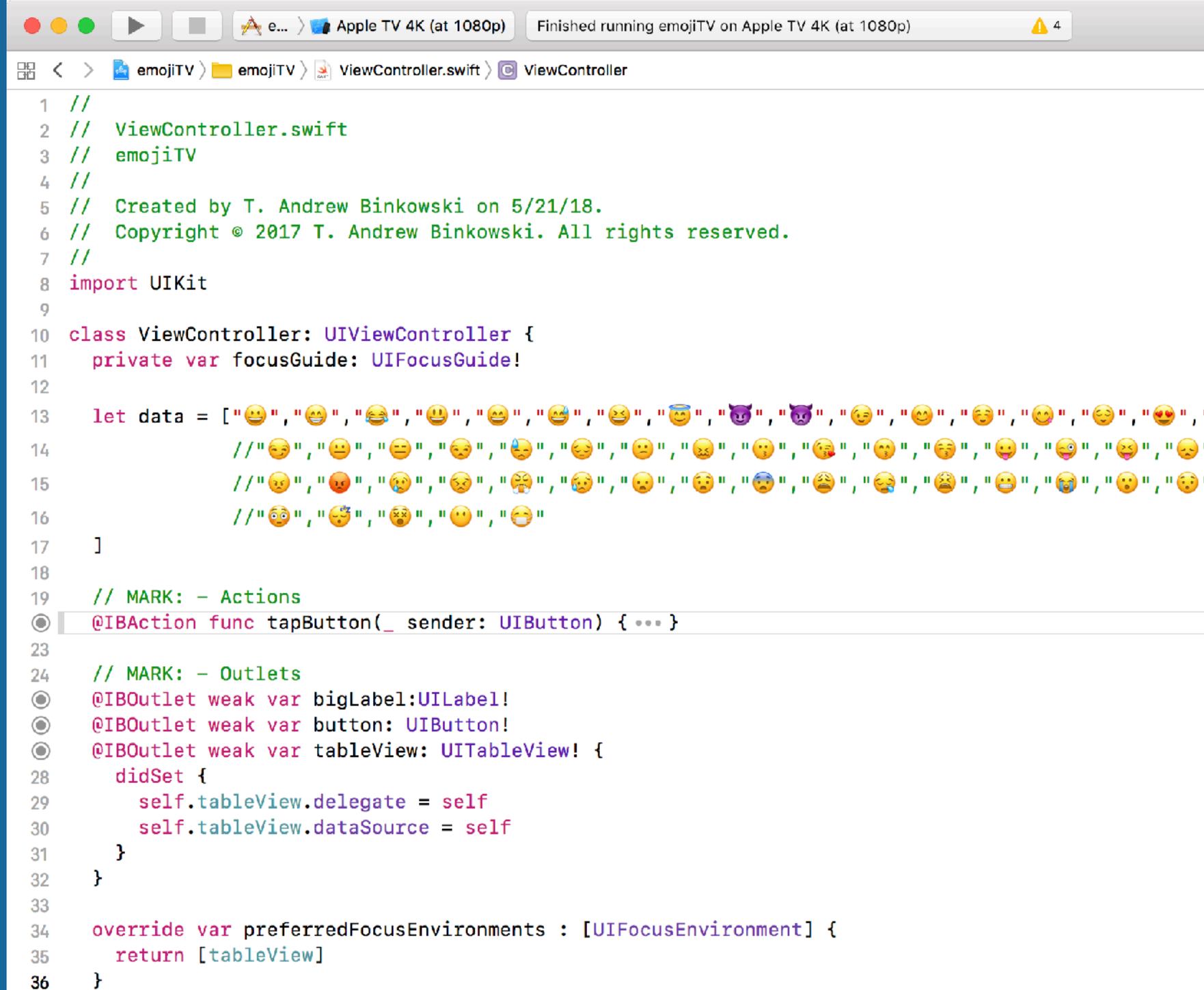
- AppDelegate



VIEW CONTROLLER

TVOS APPS

- ViewController implements a UITableView



TVOS APPS

```
  func numberOfSections(in tableView: UITableView) -> Int {
    return 1
  }

  func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return self.data.count
  }
}

extension ViewController: UITableViewDelegate {

  func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
    cell.textLabel?.text = data[indexPath.row]

    return cell
  }

  func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    print("tap: \(indexPath.row)")
    self.bigLabel?.text = "Tap \(indexPath.row)"
```

Nothing new

TVOS APPS

Tap on cell triggers update

```
  func numberOfSections(in tableView: UITableView) -> Int {
    return 1
  }

  func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return self.data.count
  }
}

extension ViewController: UITableViewDelegate {

  func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
    cell.textLabel?.text = data[indexPath.row]

    return cell
  }

  func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    print("tap: \(indexPath.row)")
    self.bigLabel?.text = "Tap \(indexPath.row)"
```

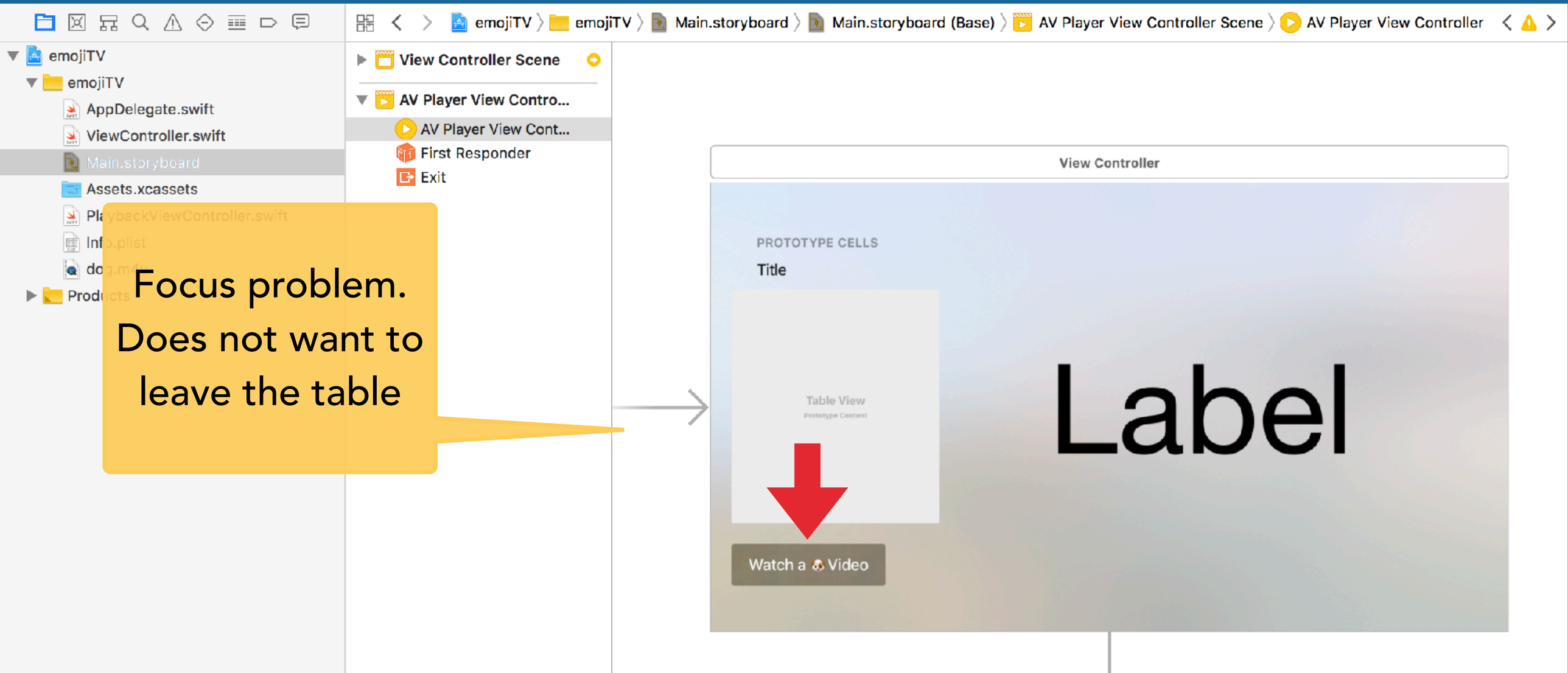
TVOS APPS

Focus triggers update

```
extension ViewController: UITableViewDelegate {

  func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell { ••• }

  func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) { ••• }

  /// Update the big emoji label when we change focus on the table (no click required)
  func tableView(_ tableView: UITableView,
                 didUpdateFocusIn context: UITableViewFocusUpdateContext,
                 with coordinator: UIFocusAnimationCoordinator) {
    // this gives you the indexpath of the focused cell
    if let indexPath = context.nextFocusedIndexPath {
      self.bigLabel?.text = data[indexPath.row]
    }
  }

}
```

FOCUS

TVOS APPS


emojiTV
emojiTV
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
Products
emojiTV
emojiTV
Main.storyboard
Main.storyboard (Base)
AV Player View Controller Scene
AV Player View Controller
View Controller Scene
AV Player View Contro...
AV Player View Cont...
First Responder
Exit
Focus problem.
Does not want to
leave the table
View Controller
PROTOTYPE CELLS
Title
Table View
Prototype Content
Label
Watch a Video

TVOS APPS

```
// MARK: - UIFocusEnvironment

override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {
  super.didUpdateFocus(in: context, with: coordinator)

  // Update the focus guide's `preferredFocusedView` depending on which
  // button has the focus
  guard let nextFocusedView = context.nextFocusedView else { return }

  switch nextFocusedView {
  case tableView:
    focusGuide.preferredFocusEnvironments = [button]

  case button:
    focusGuide.preferredFocusEnvironments = [tableView]

  default:
    focusGuide.preferredFocusEnvironments = []
  }
}
```

Allow table to button focus switch

TVOS APPS

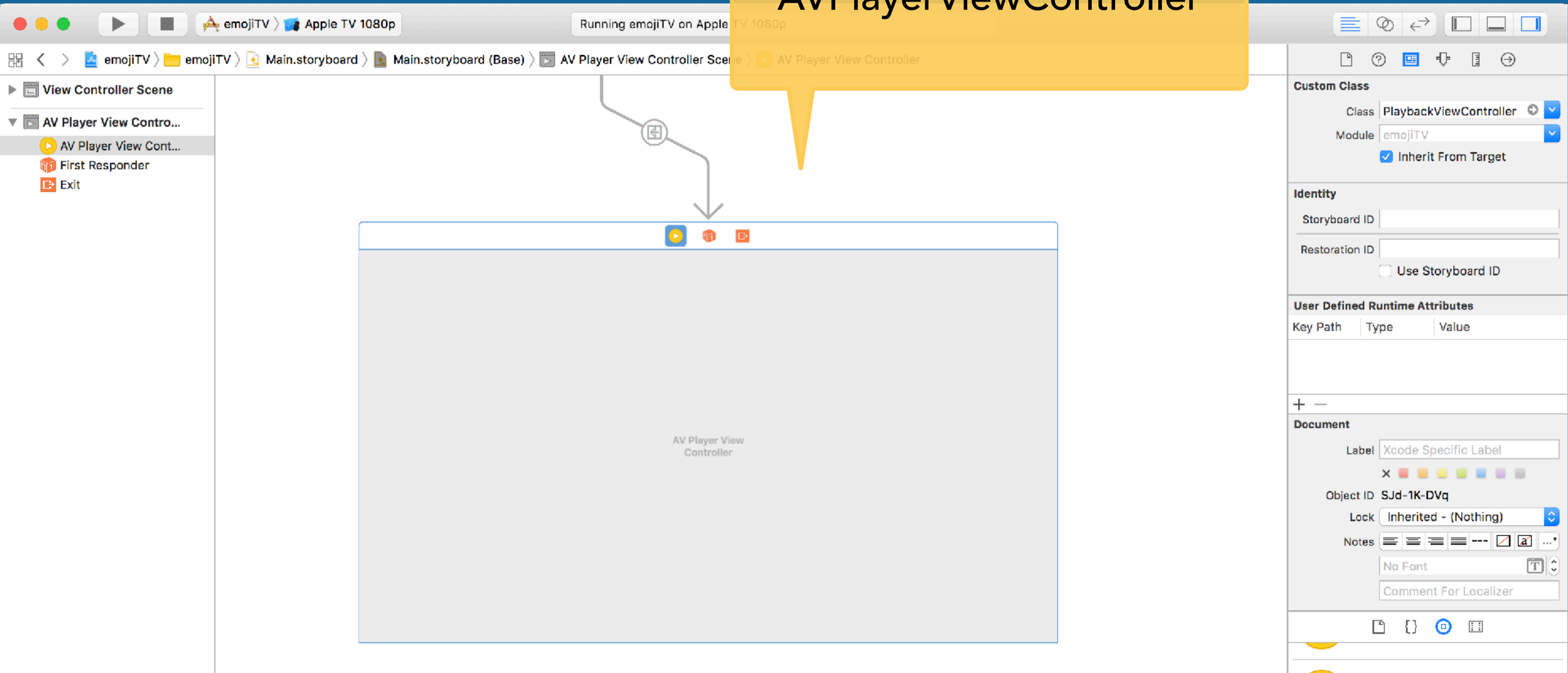
Set the table to have initial focus

```
/// MARK: - Focus
private var focusGuide: UIFocusGuide!

override var preferredFocusEnvironments : [UIFocusEnvironment] {
  return [tableView]
}
```

PLAYING VIDEO

PLAYING VIDEO

AVPlayerViewController



PLAYING VIDEO

- All-in-one for video playback
  - Airplay
  - Picture-in-Picture
  - Video controls
  - Chapter markers
  - Interstitial (ad) content
  - Overlays
  - ....and more...

AVPlayerViewController

An object that displays the video content from a player object along with system-supplied playback controls.

SDKs

iOS 8.0+

tvOS 9.0+

Framework

AVKit

On This Page

Overview
Topics
Relationships
See Also

Overview

Using `AVPlayerViewController` makes it easy for you to add media playback capabilities to your application matching the styling and features of the native system players. Since `AVPlayerViewController` is a system framework class, your playback applications automatically adopt the new aesthetics and features of future operating system updates without any additional work from you.

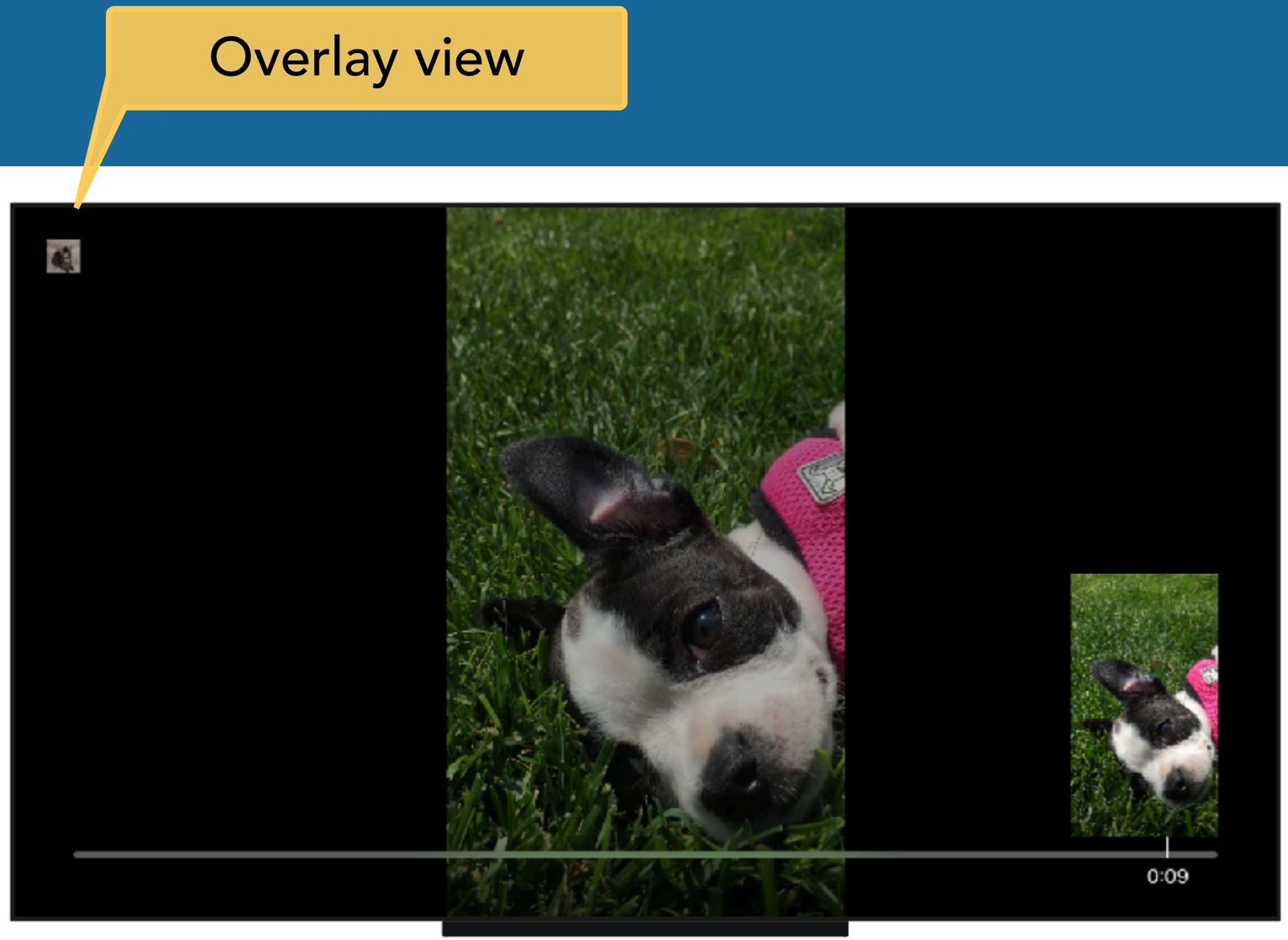
> **Important**
>
> Do not subclass `AVPlayerViewController`. Overriding this class's methods is unsupported and results in undefined behavior.

Supporting AirPlay

AirPlay lets users play your video content on an Apple TV and stream audio content to third-party AirPlay speakers and receivers. `AVPlayerViewController` automatically supports AirPlay, but you need to perform some project and audio session configuration before it can be enabled in your application. See Configuring Audio Settings for iOS and tvOS in Media Playback Programming Guide for more on how to perform this configuration.

See AirPlay Overview for more information about the features and capabilities of AirPlay.

TVOS APPS



- AVPlayerViewController

PLAYING VIDEO

emojiTV 〉 emojiTV 〉 PlaybackViewController.swift 〉 viewDidLoad()

```
//  Copyright © 2017 T. Andrew Binkowski. All rights reserved.
//

import UIKit
import AVKit

class PlaybackViewController: AVPlayerViewController {

  override func viewDidLoad() {
    super.viewDidLoad()

    /*•••
    guard let path = Bundle.main.path(forResource: "dog", ofType:"m4v") else {
      return
    }

    // Set up the player
    self.player = AVPlayer(url: URL(fileURLWithPath: path))
    self.player?.play()
    NotificationCenter.default.addObserver(self,
                                           selector: #selector(PlaybackViewController.itemDidFinishPlaying(notification:)),
                                           name: NSNotification.Name.AVPlayerItemDidPlayToEndTime,
                                           object: player?.currentItem)

  }
```

Load up and play

PLAYING VIDEO

```
    // Set up the player
    self.player = AVPlayer(url: URL(fileURLWithPath: path))
    self.player?.play()
    NotificationCenter.default.addObserver(self,
                                           selector: #selector(PlaybackViewController.itemDidFinishPlaying(notification:)),
                                           name: NSNotification.Name.AVPlayerItemDidPlayToEndTime,
                                           object: player?.currentItem)

  }

  deinit {
    NotificationCenter.default.removeObserver(self)
  }

  /// Automatically dismiss the player when done
  @objc func itemDidFinishPlaying(notification: NSNotification) {
    print ("Notification sent with done playing")
    self.dismiss(animated: true) {
      print("Dismissed player view controller")
    }
  }

}
```

Notification to dismiss viewer when done

THE UNIVERSITY OF CHICAGO

ADVANCED tvOS APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 9

THE UNIVERSITY OF CHICAGO

ADVANCED iOS APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 9

LOCALIZATION

LOCALIZATION

- 155 App Stores worldwide
  - Adding new ones all the time
- Your apps can be for sale globally with minimal effort
  - Toggle switch in iTunes Connect
- Making your app make sense in International App Stores is a completely different thing

LOCALIZATION

- Top 5 countries
  - China
  - US
  - Japan
  - UK
  - Australia

LOCALIZATION

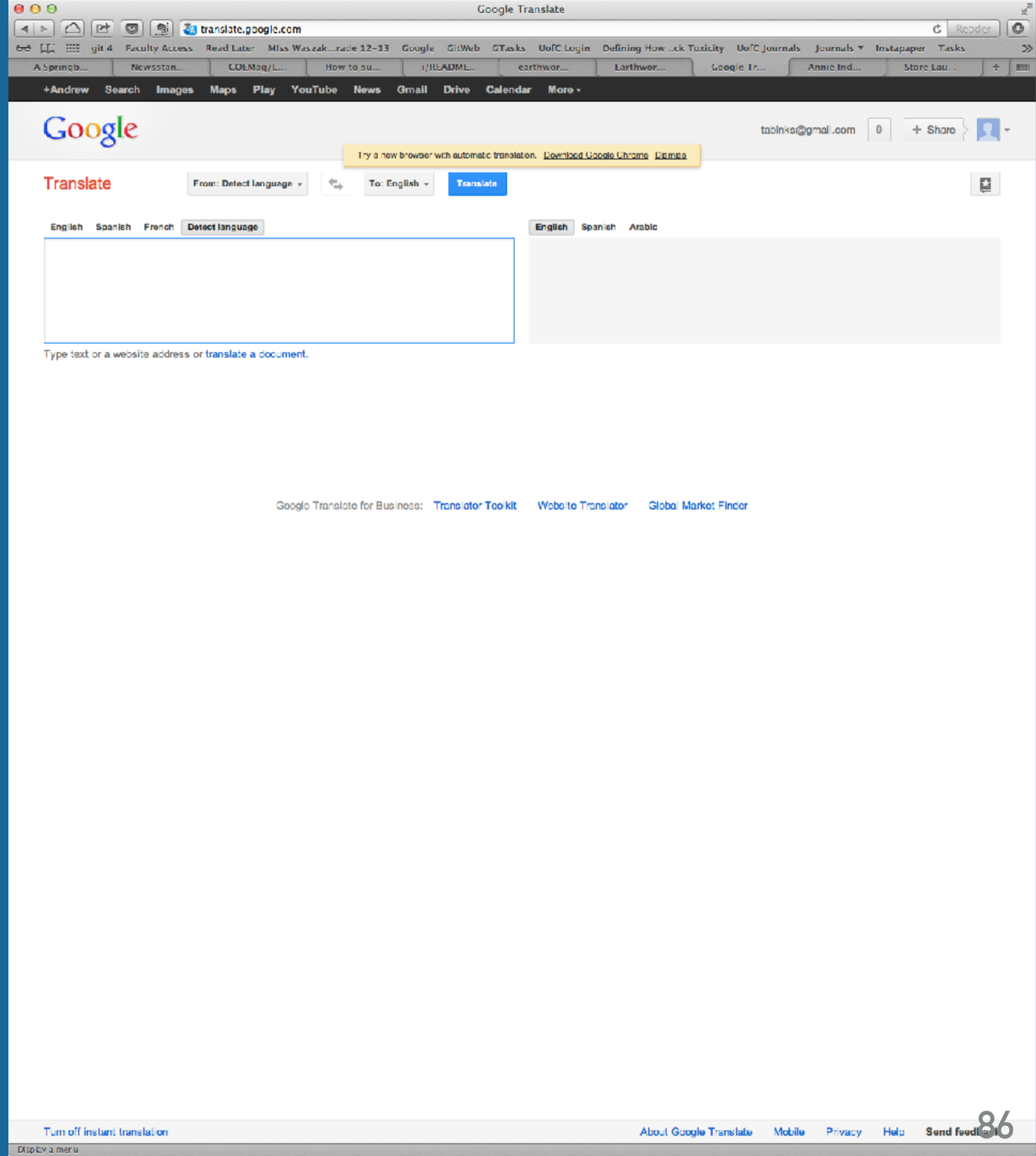
Translation is not as simple as using Google Translate

LOCALIZATION

Bad for in-app purchase

LOCALIZATION

- Apple makes localization straightforward through APIs and built-in Xcode support
    - Localize strings, numbers, images
- Managing the localization (and accuracy) can be tedious
    - Preparing assets for each language support

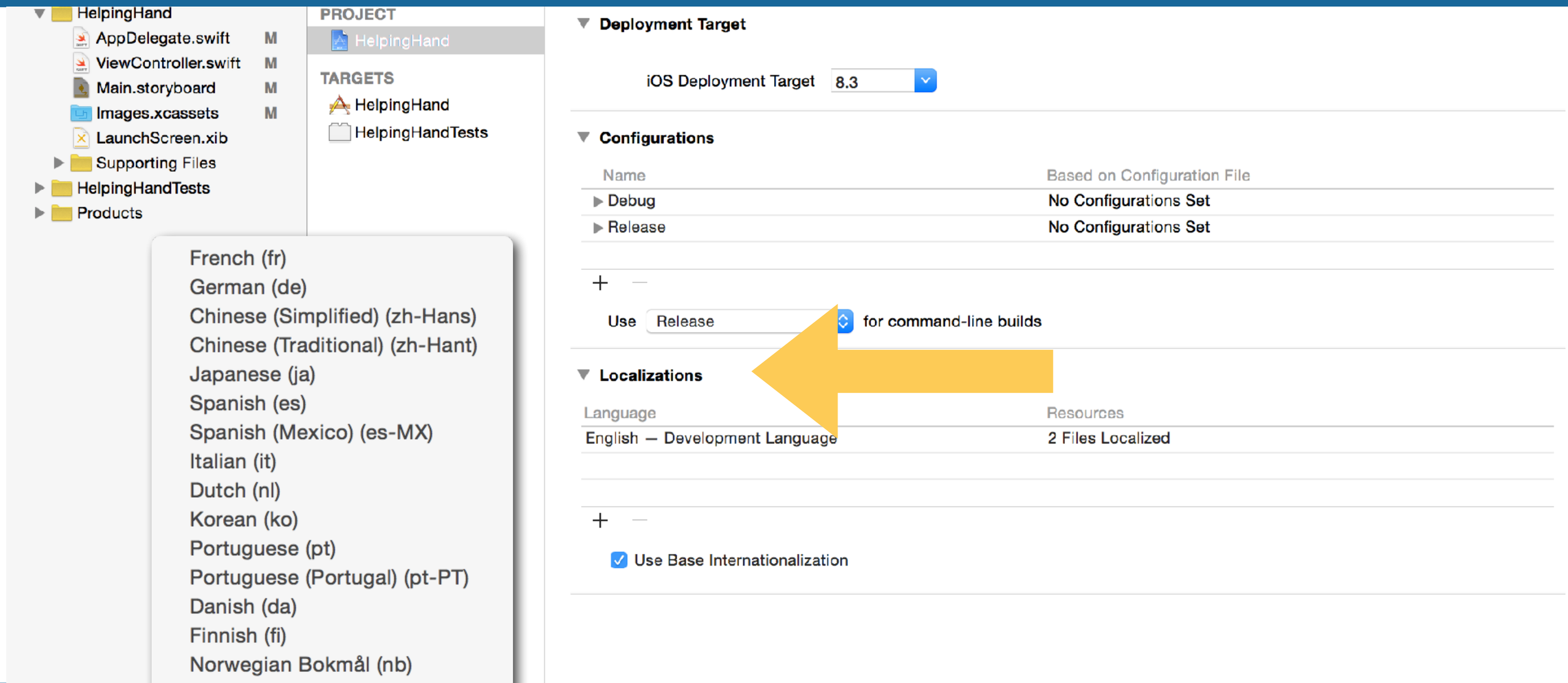


LOCALIZATION

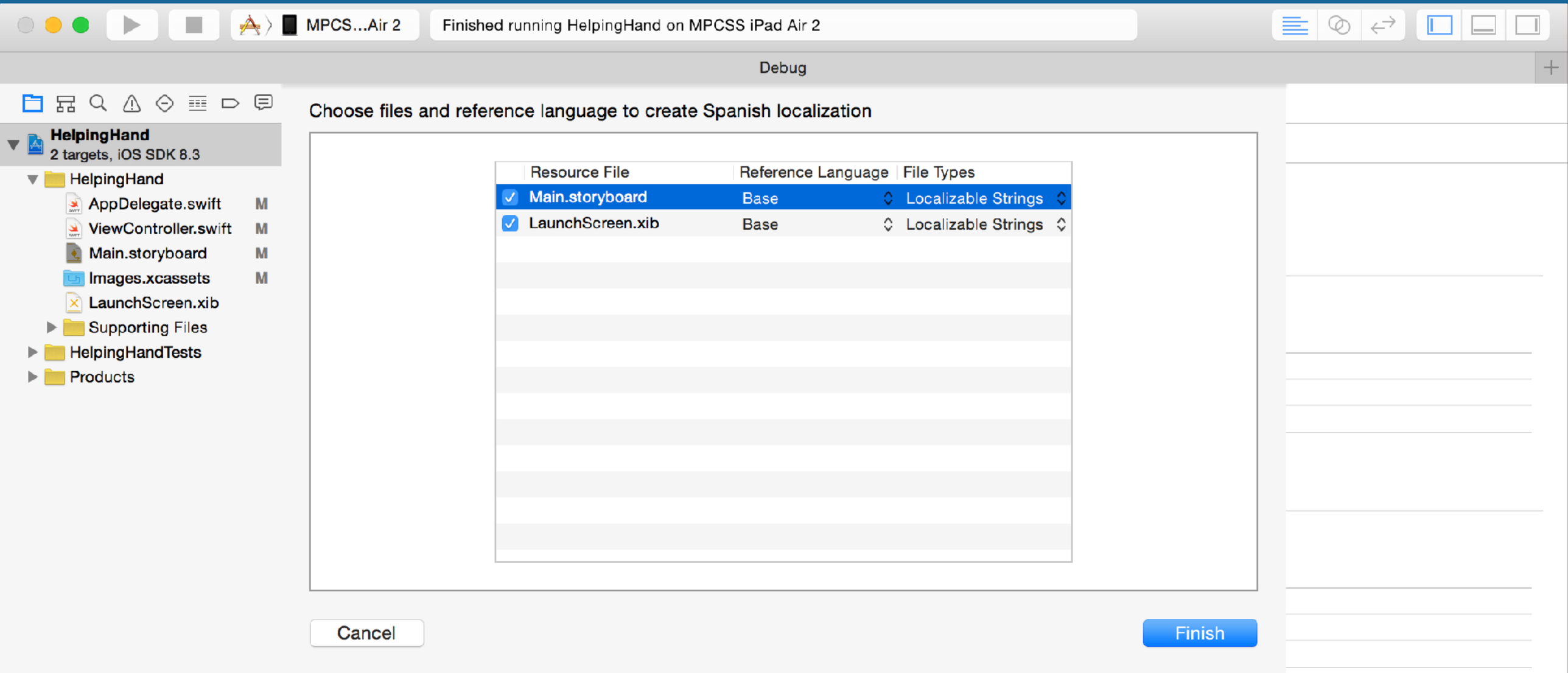
- Google translation service (for Google Play only)
  - Service available directly to app developers in the Google Play Developer Console
  - Third-party translation services managed by Google Play
  - Pay per word

ADDING LOCALIZATION TO YOUR APP

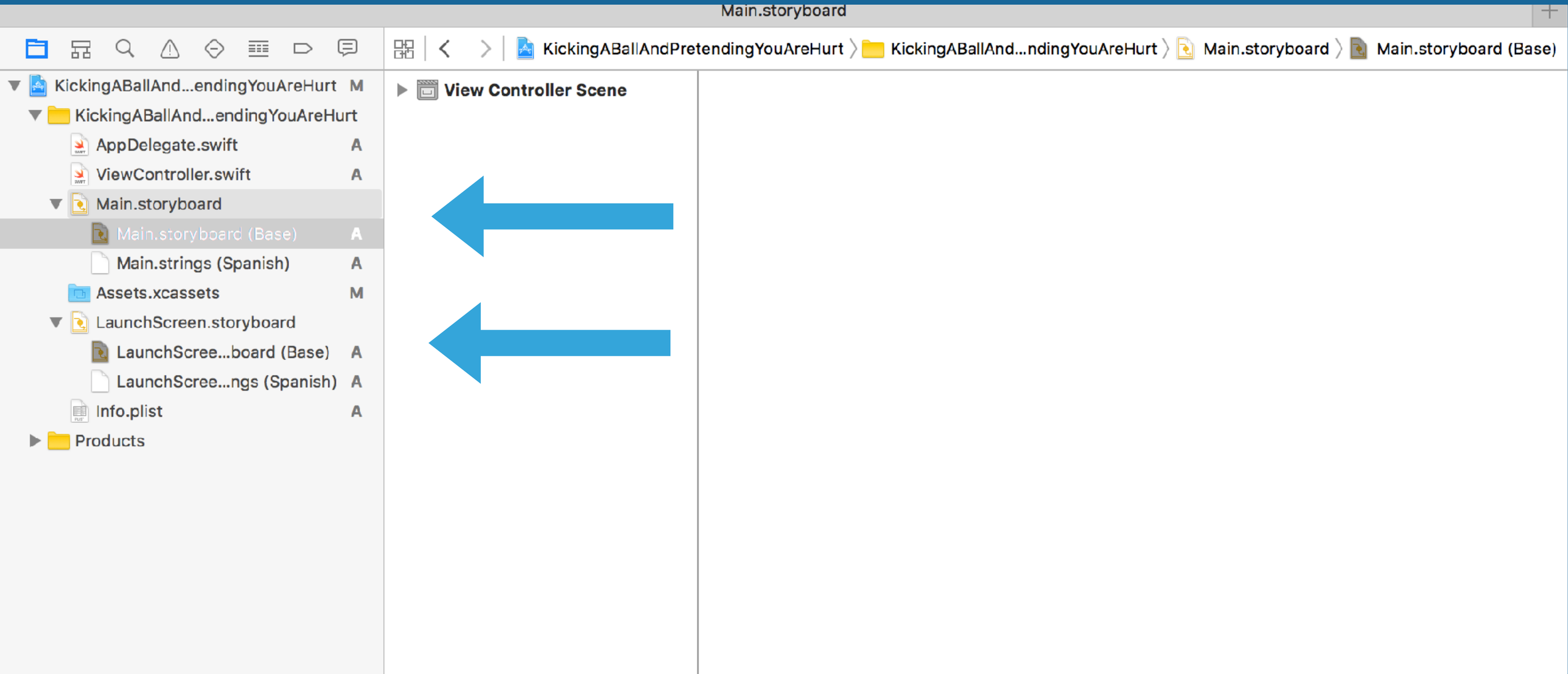
LOCALIZATION



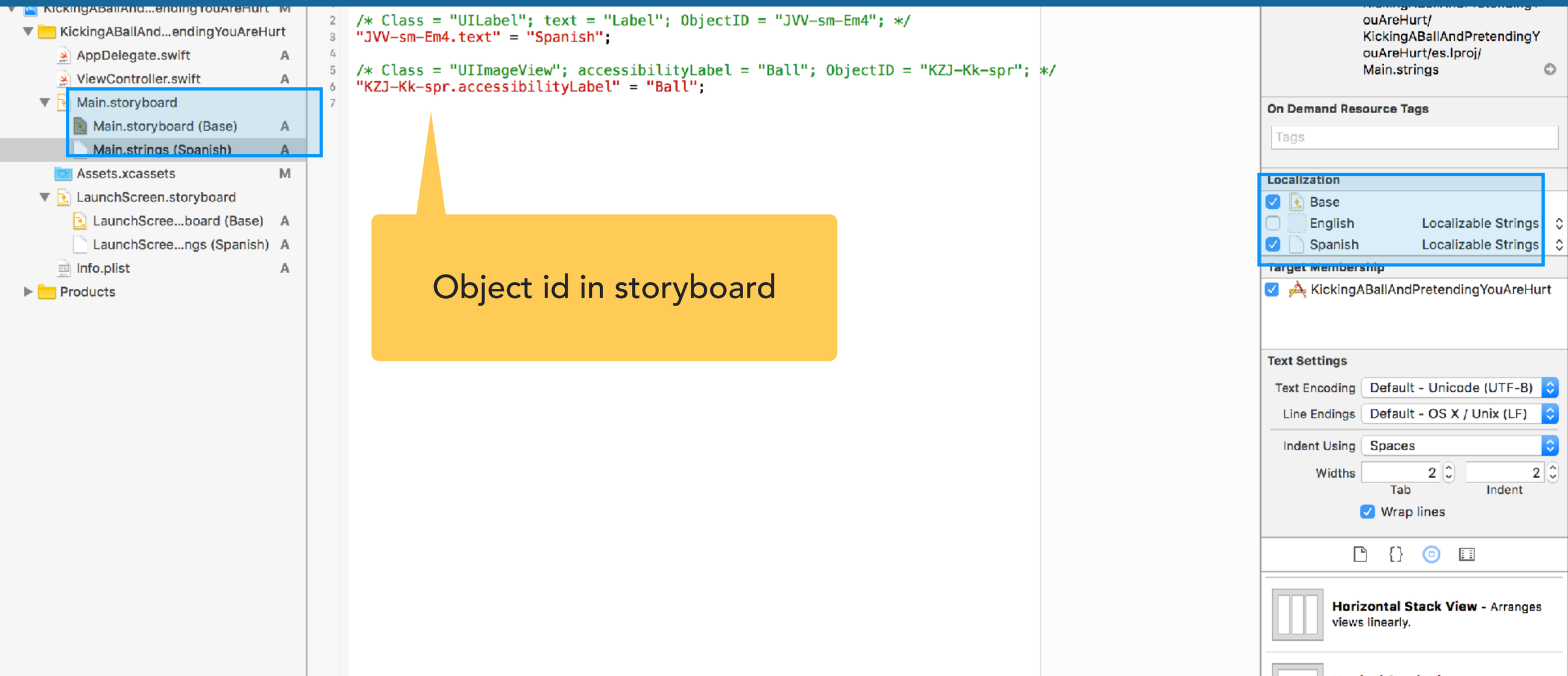
LOCALIZATION



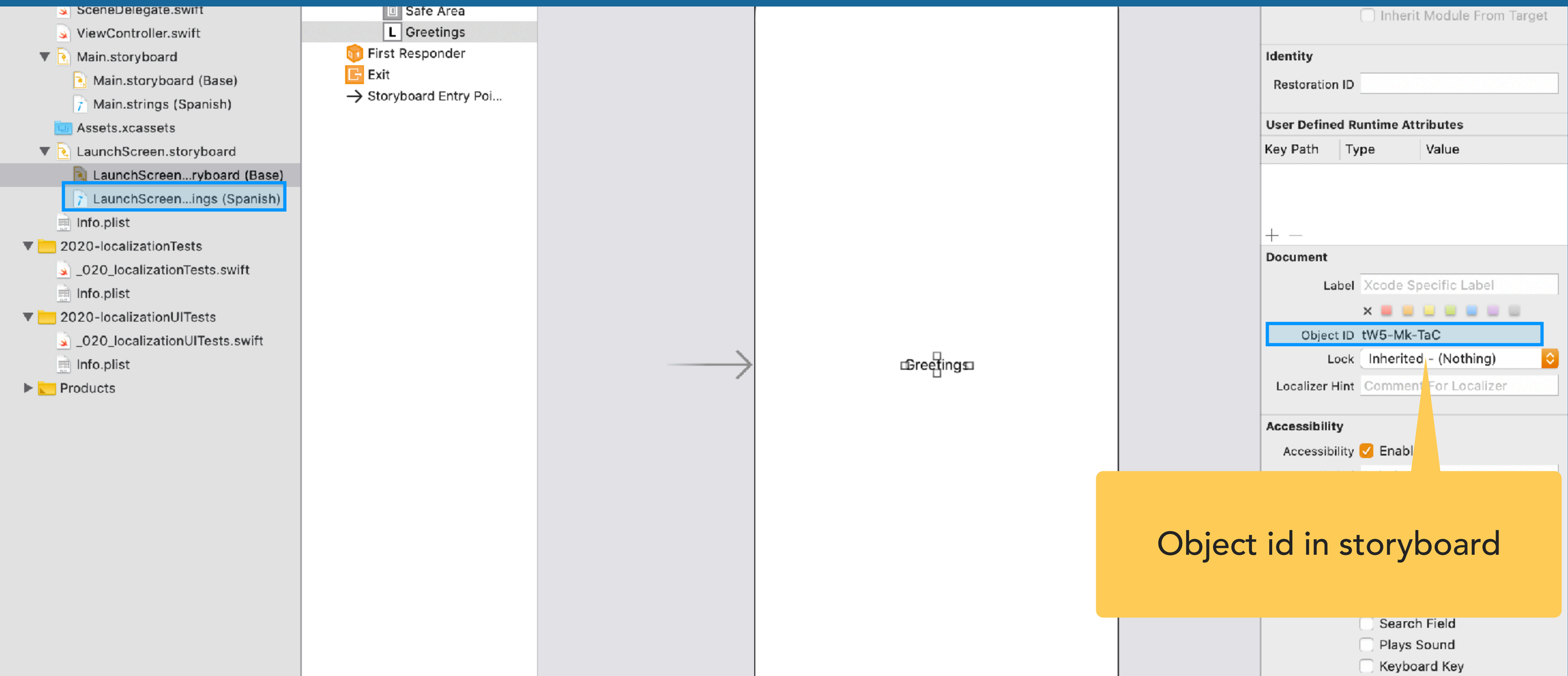
LOCALIZATION



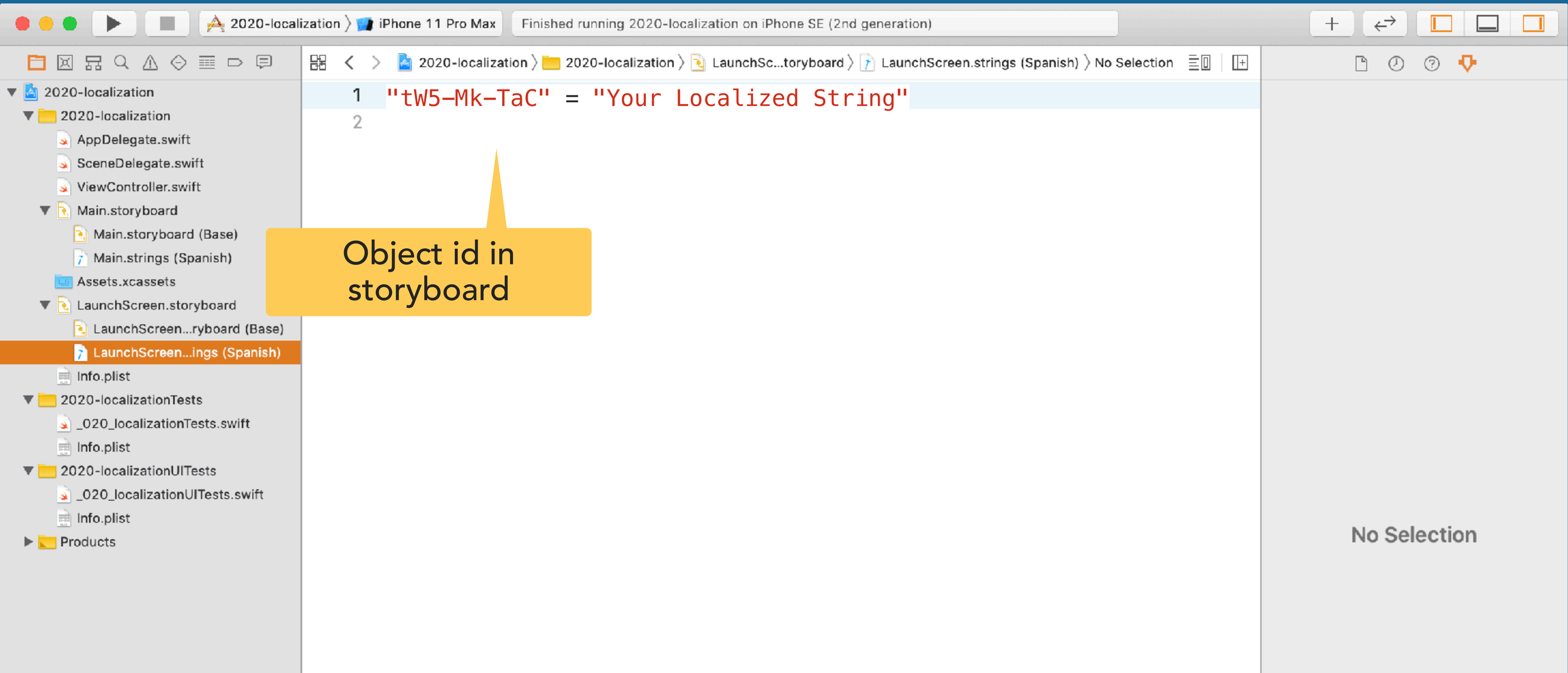
LOCALIZATION



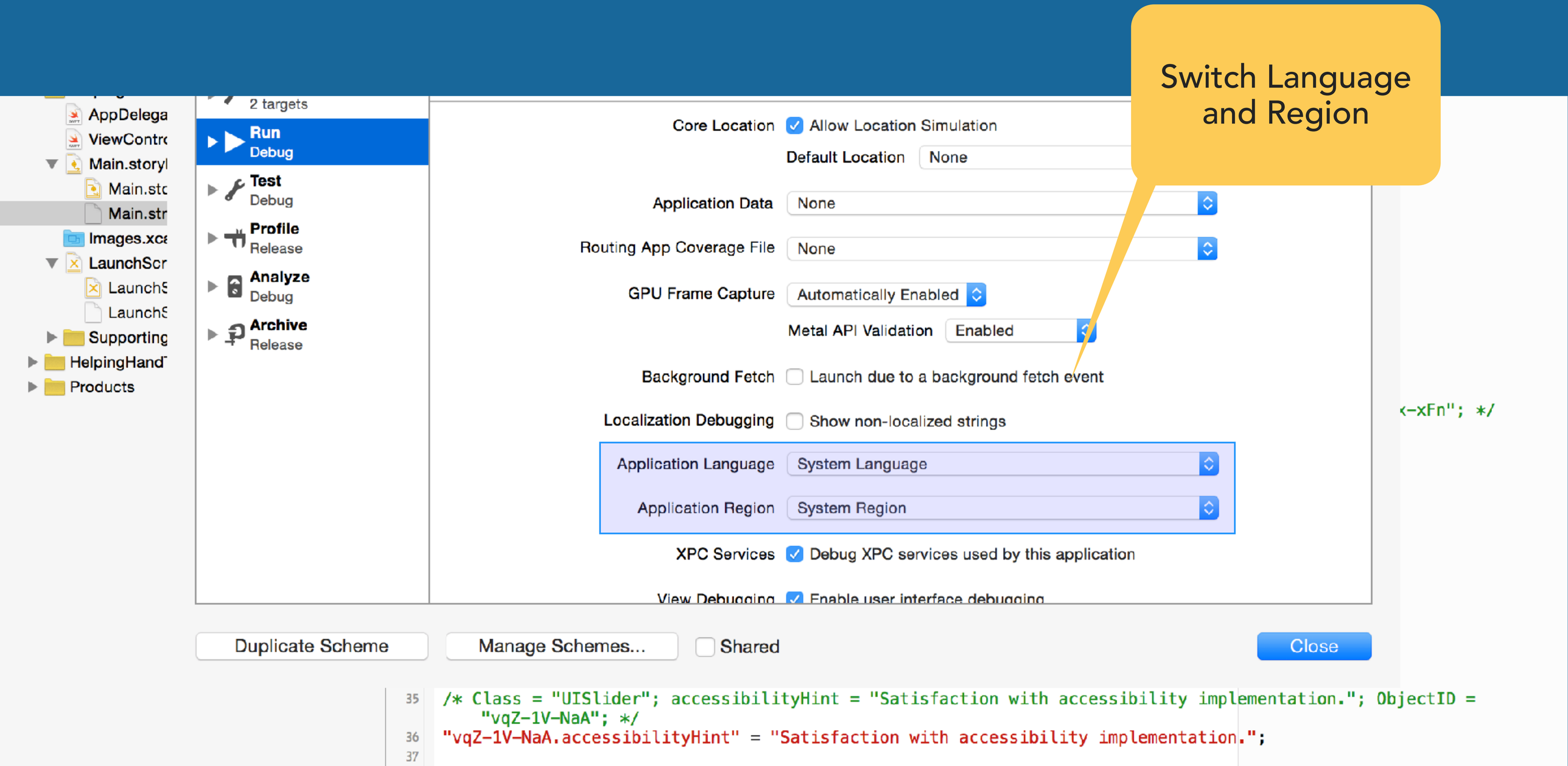
LOCALIZATION



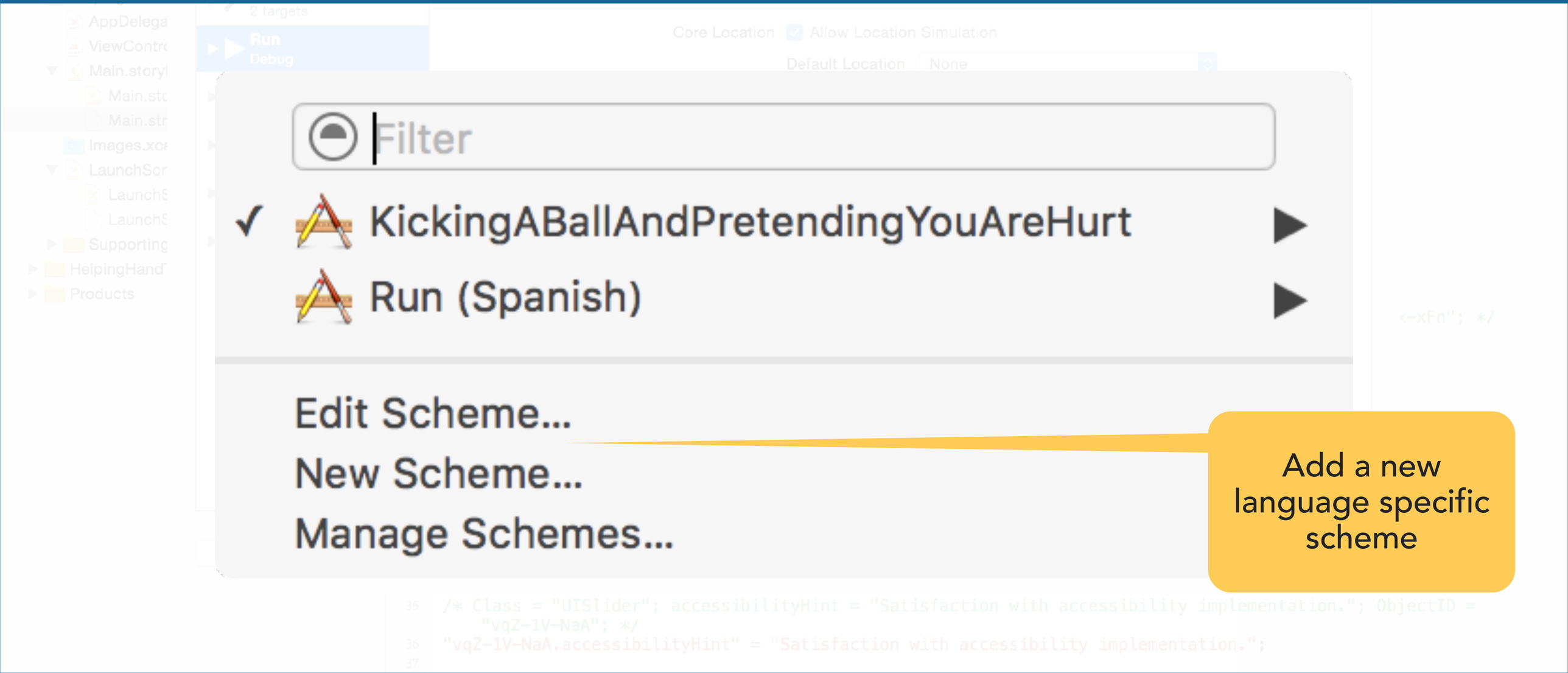
LOCALIZATION



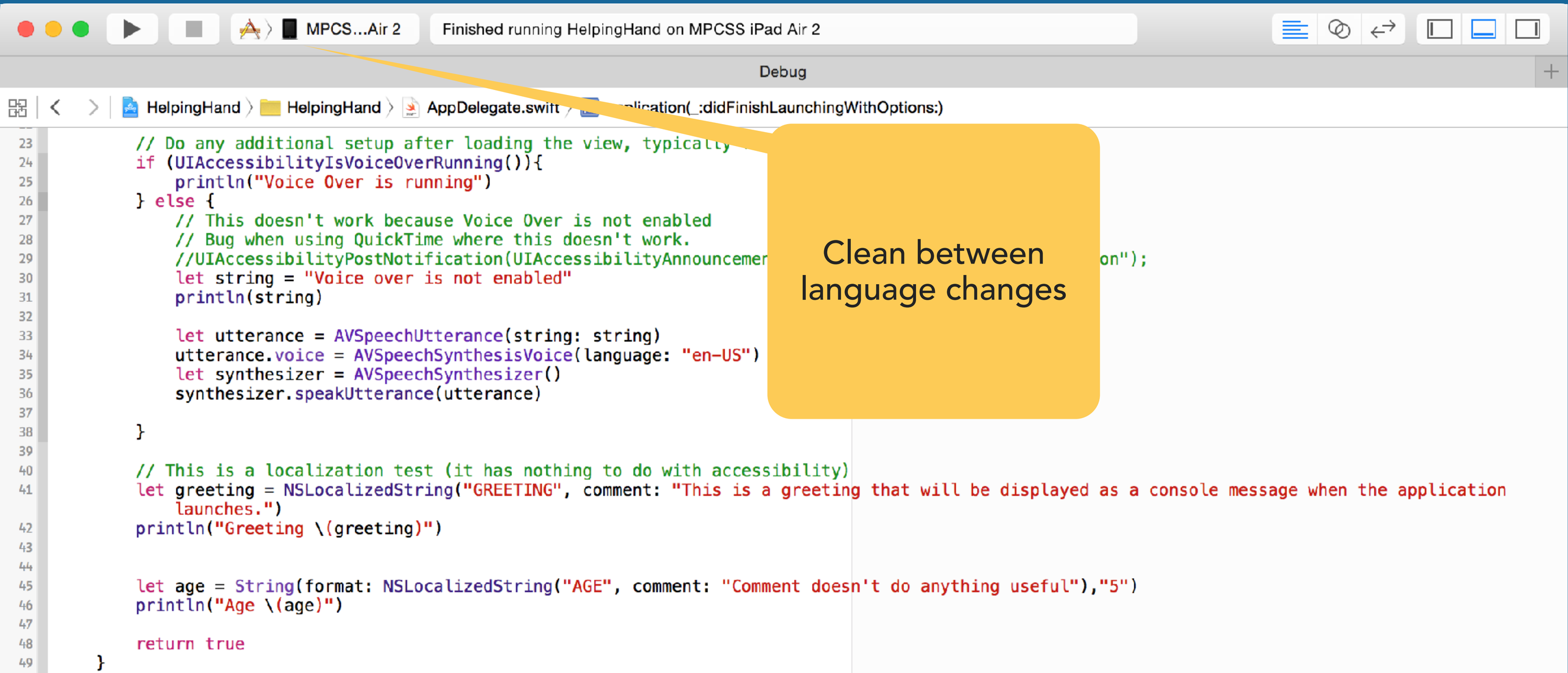
LOCALIZATION



LOCALIZATION

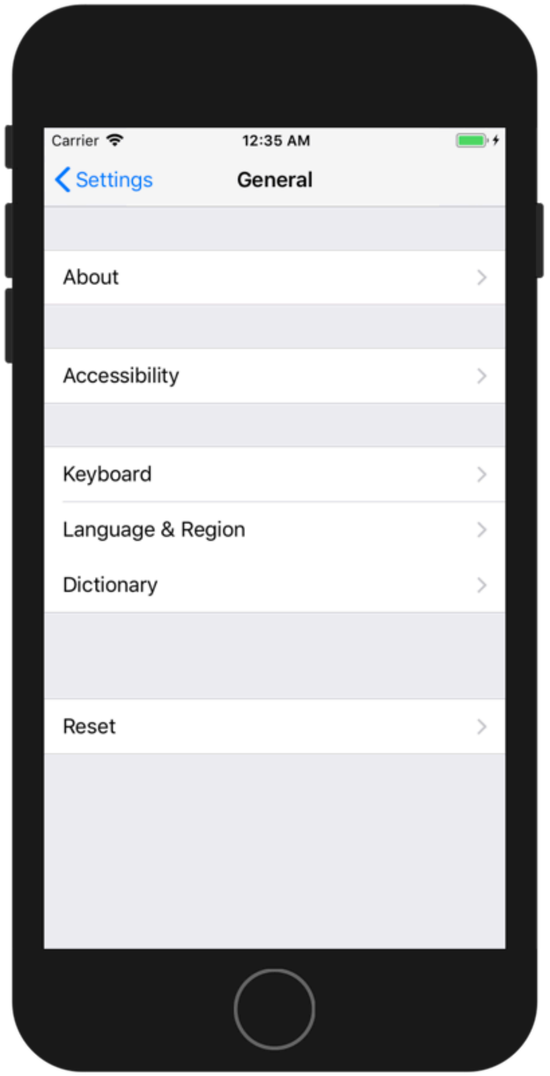


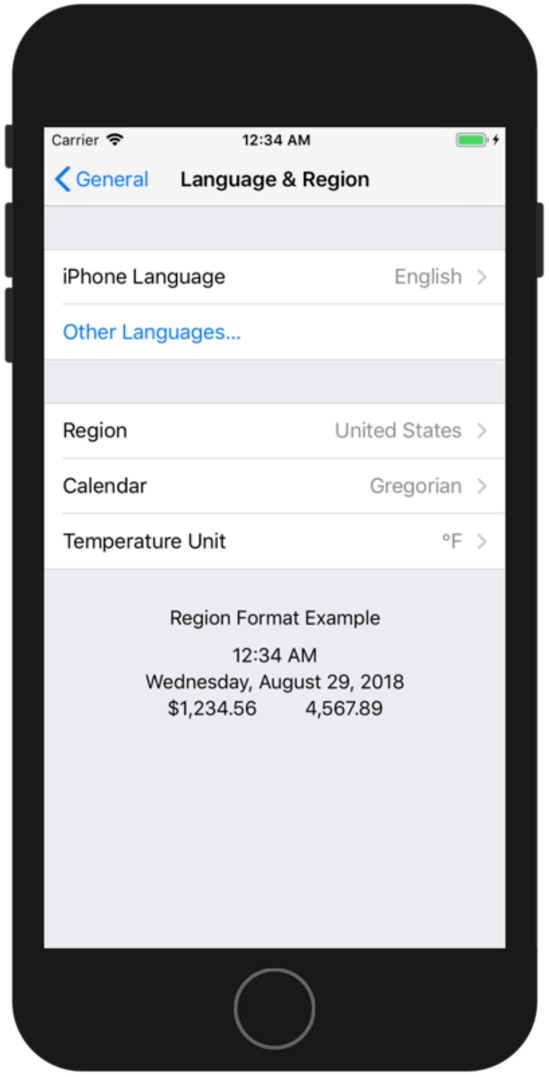
LOCALIZATION




```
        // Do any additional setup after loading the view, typically
        if (UIAccessibilityIsVoiceOverRunning()){
            println("Voice Over is running")
        } else {
            // This doesn't work because Voice Over is not enabled
            // Bug when using QuickTime where this doesn't work.
            //UIAccessibilityPostNotification(UIAccessibilityAnnouncemer                    on");
            let string = "Voice over is not enabled"
            println(string)

            let utterance = AVSpeechUtterance(string: string)
            utterance.voice = AVSpeechSynthesisVoice(language: "en-US")
            let synthesizer = AVSpeechSynthesizer()
            synthesizer.speakUtterance(utterance)

        }

        // This is a localization test (it has nothing to do with accessibility)
        let greeting = NSLocalizedString("GREETING", comment: "This is a greeting that will be displayed as a console message when the application
            launches.")
        println("Greeting \(greeting)")


        let age = String(format: NSLocalizedString("AGE", comment: "Comment doesn't do anything useful"),"5")
        println("Age \(age)")

        return true
    }
```

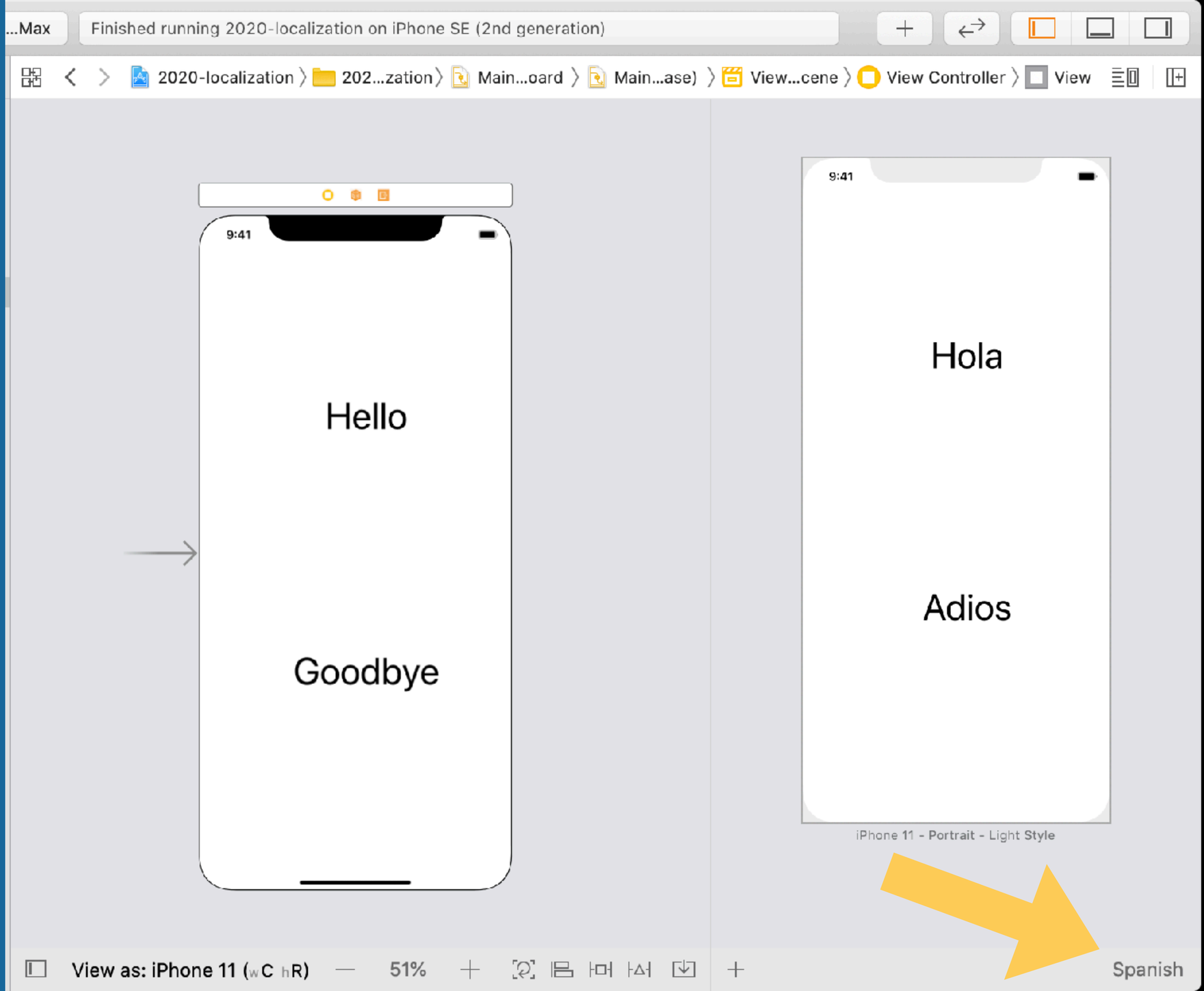
LOCALIZATION





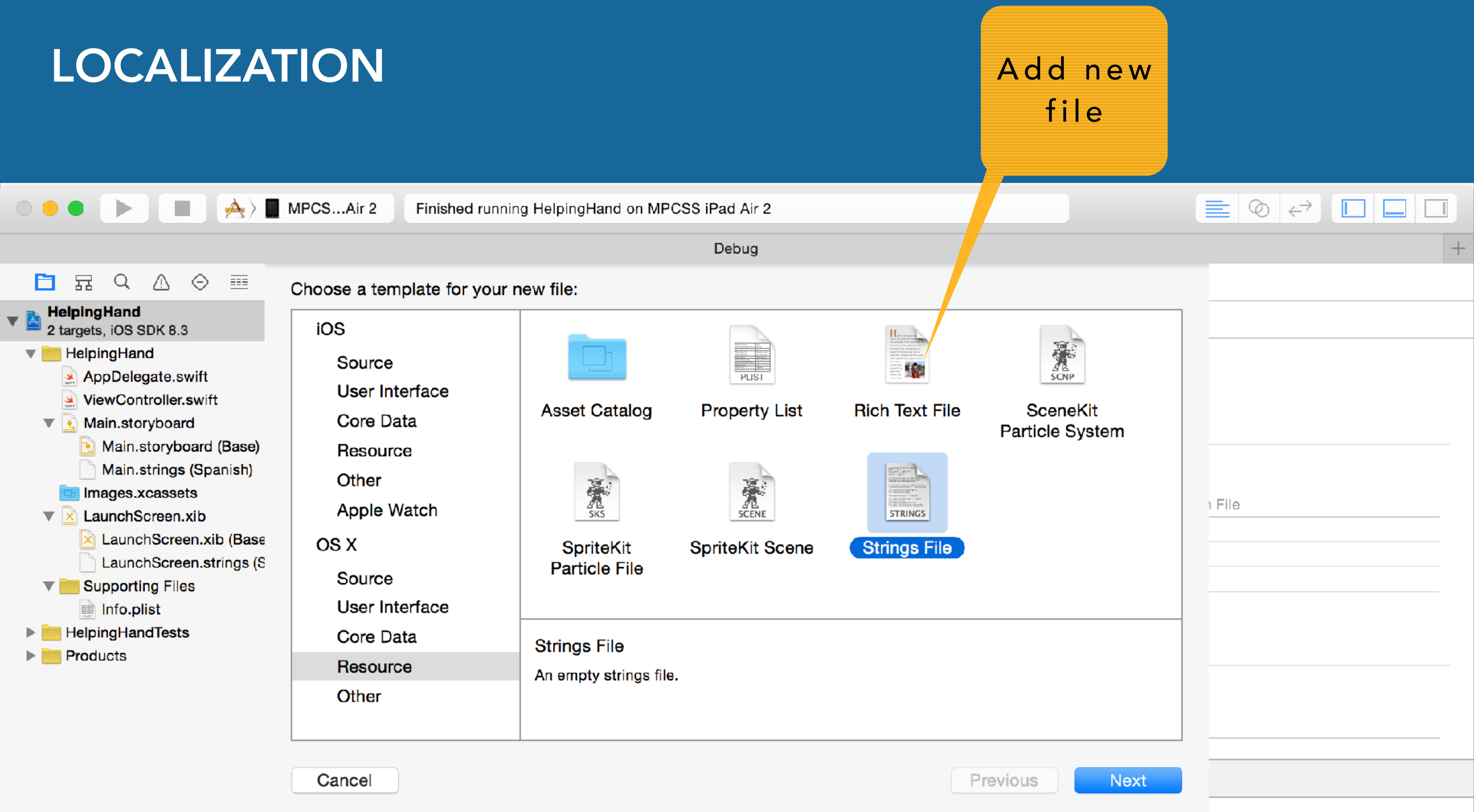
LOCALIZATION

- Preview mode

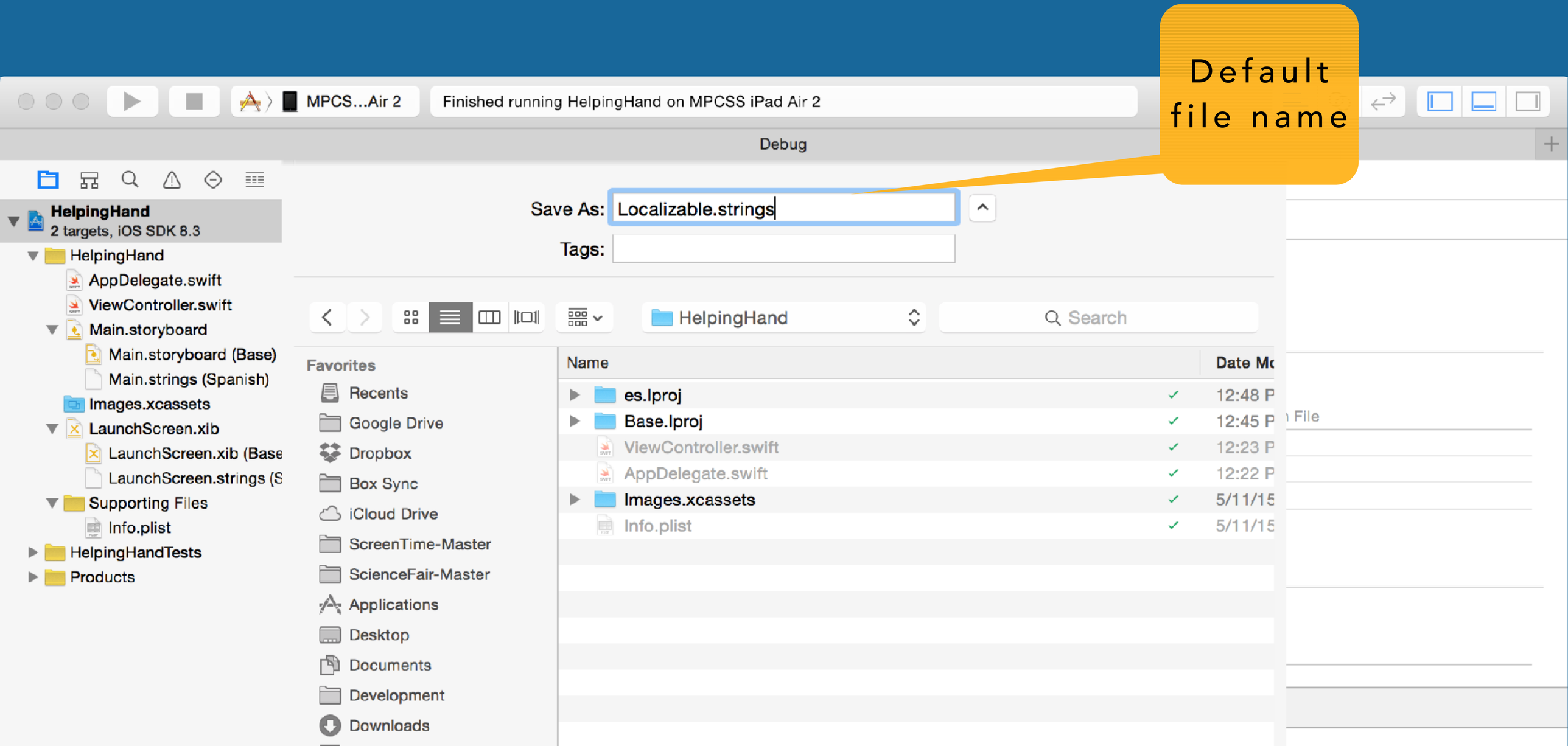


LOCALIZING TEXT STRINGS

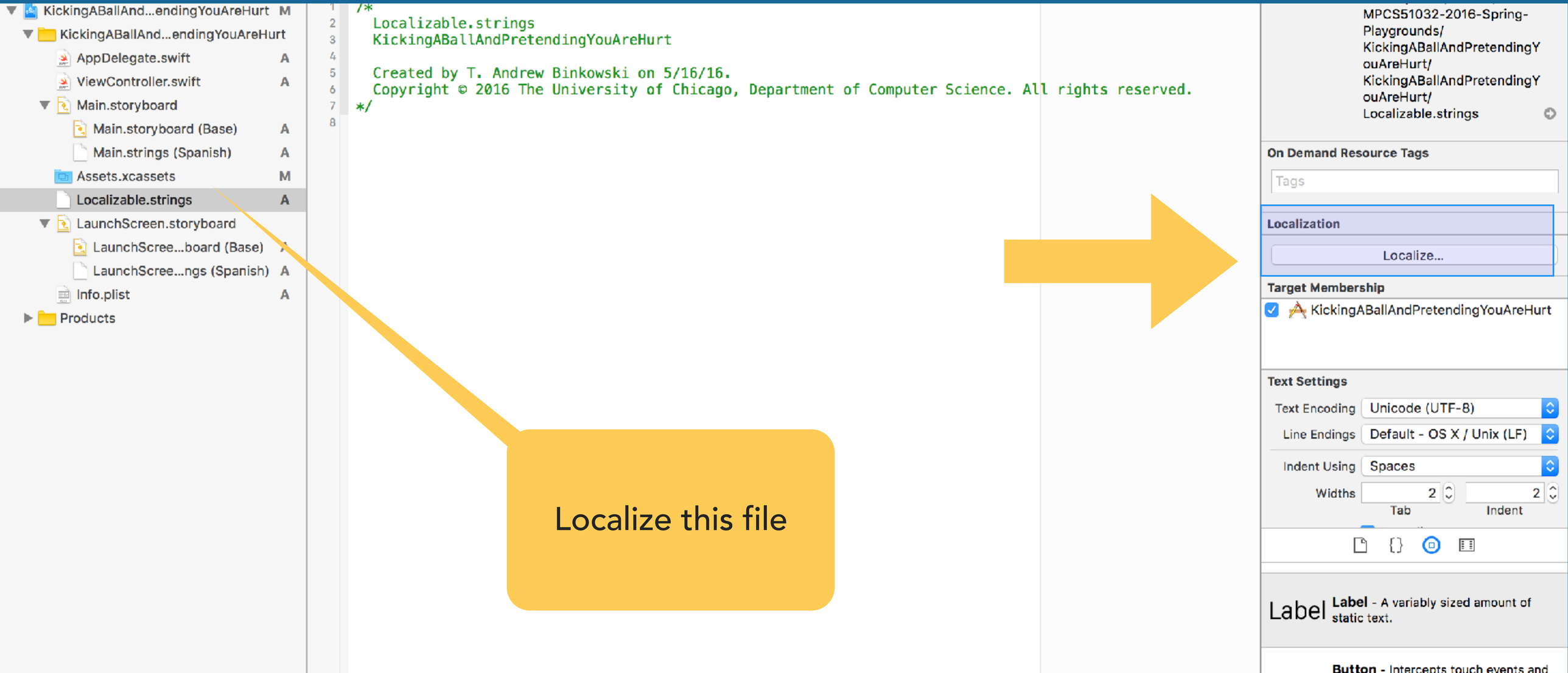
LOCALIZATION



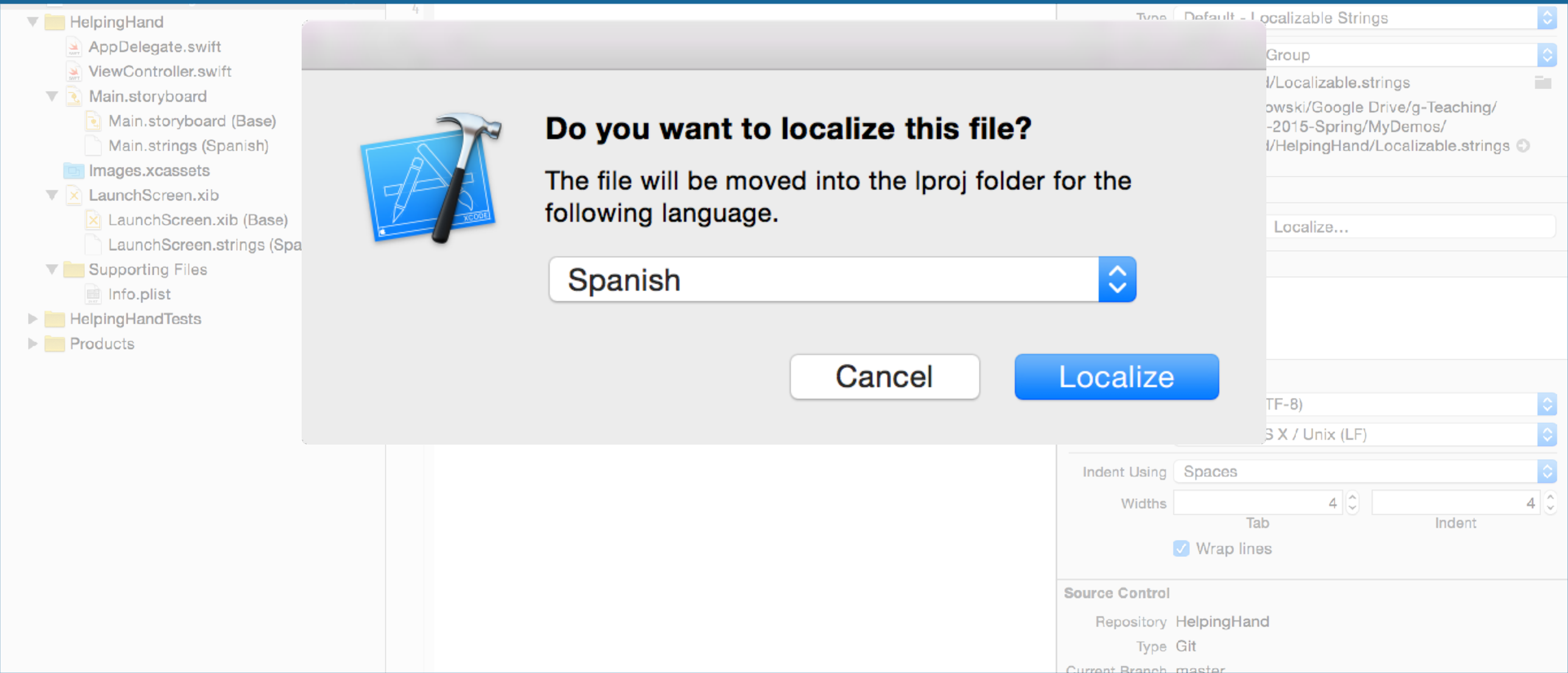
LOCALIZATION



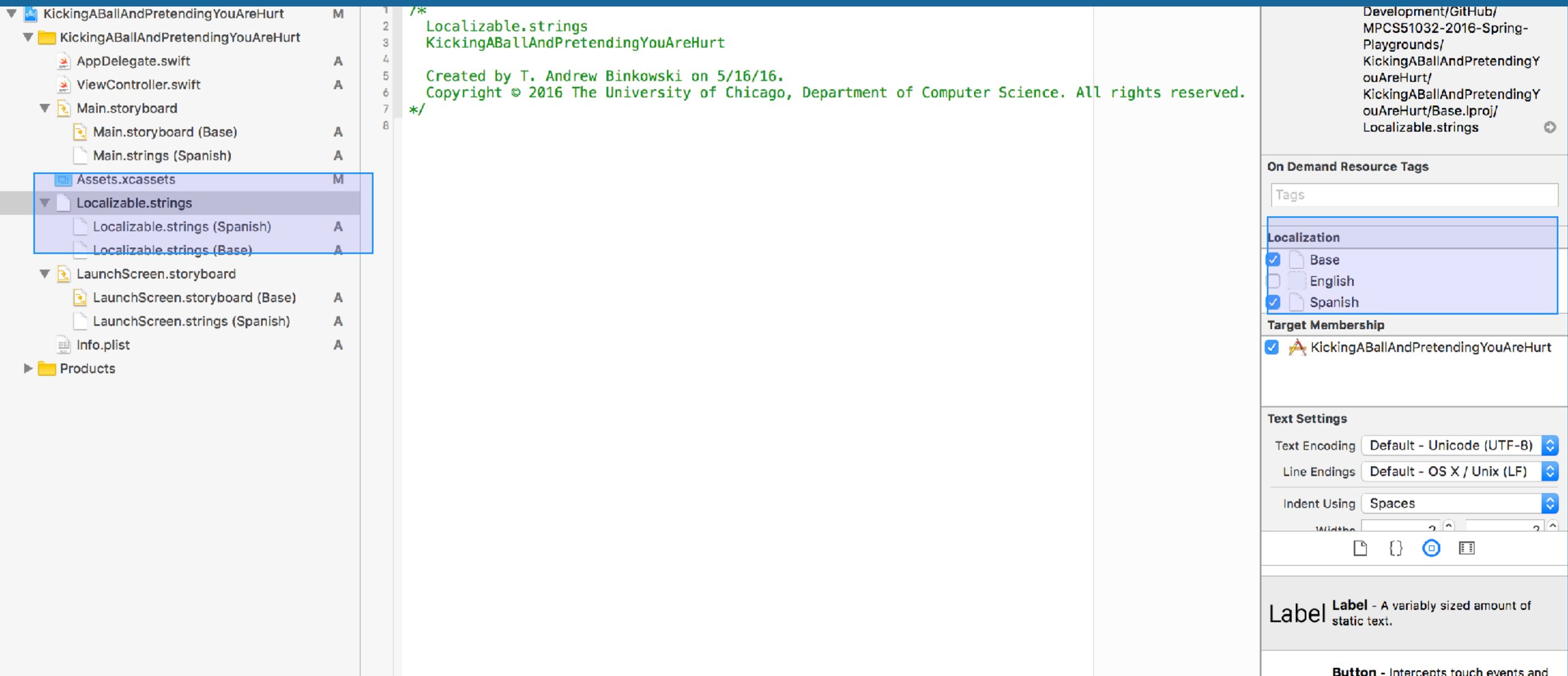
LOCALIZATION



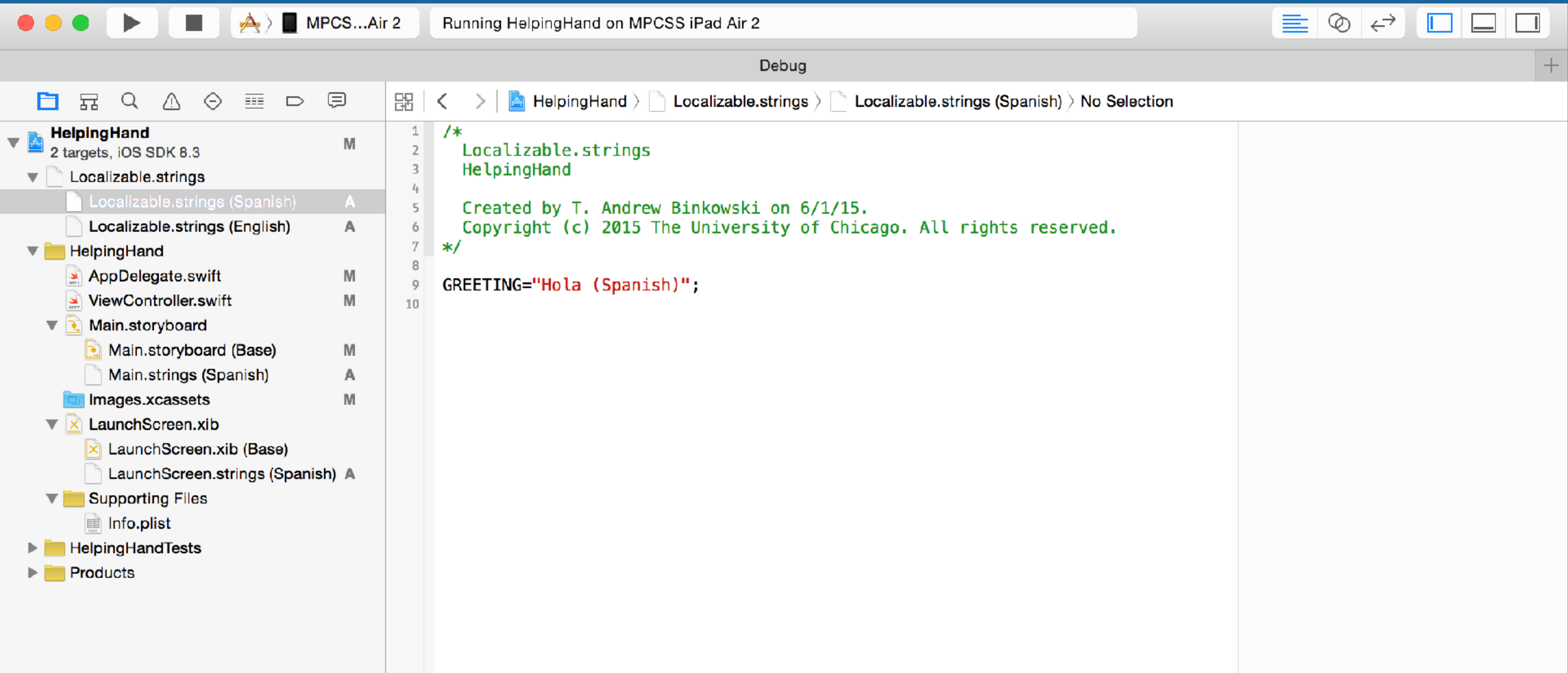
LOCALIZATION



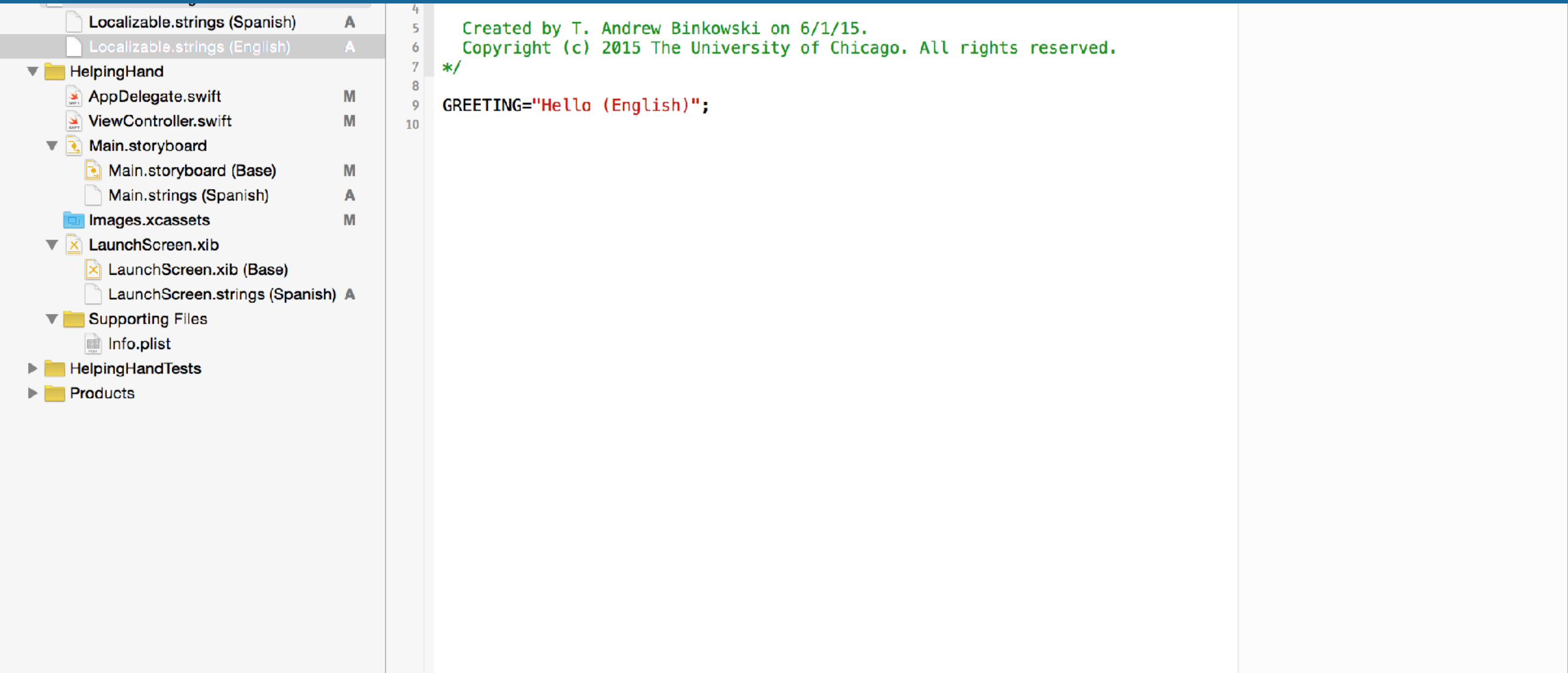
LOCALIZATION



LOCALIZATION



LOCALIZATION



LOCALIZATION

```swift
// This is a localization test (it has nothing to do with accessibility)
let greeting = NSLocalizedString("GREETING", comment: "This is a greeting.")
print("Greeting \(greeting)")
```

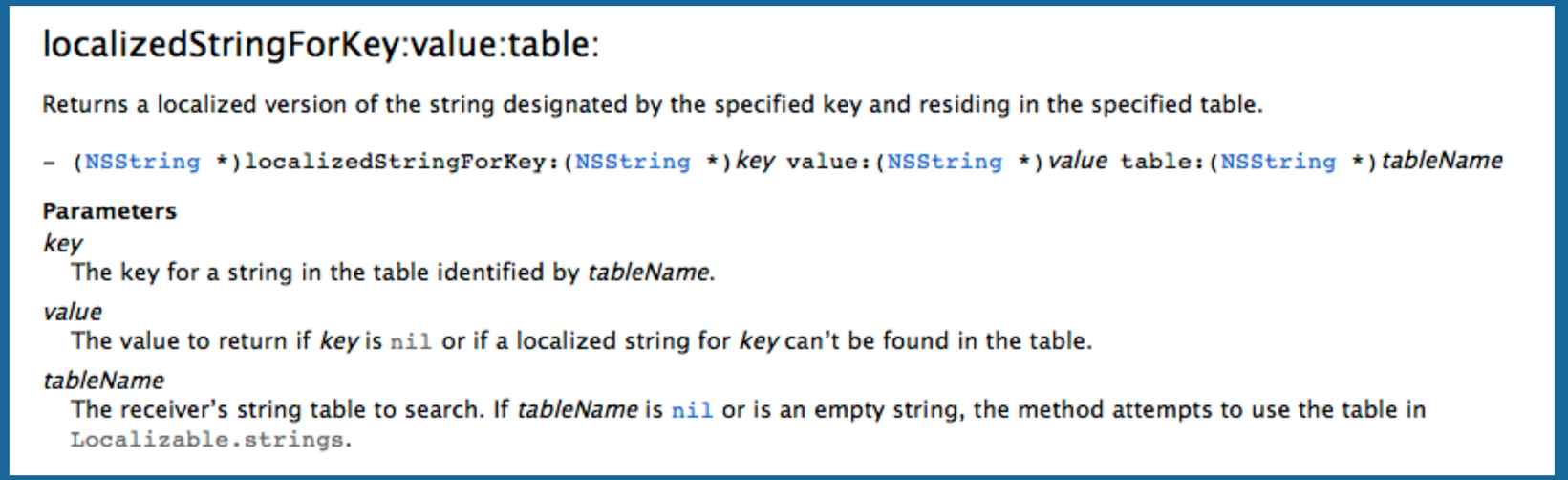
```
/*
  Localizable.strings
  KickingABallAndPretendingYouAreHurt

  Created by T. Andrew Binkowski on 5/16/16.
  Copyright © 2016 The University of Chicago, Department of
Computer Science. All rights reserved.
*/

GREETING = "Hello";
```

```
/*
  Localizable.strings
  KickingABallAndPretendingYouAreHurt

  Created by T. Andrew Binkowski on 5/16/16.
  Copyright © 2016 The University of Chicago, Department o
reserved.
*/

GREETING = "Hola";
```

LOCALIZATION

```
#define NSLocalizedString(key, comment)
[[NSBundle mainBundle] localizedStringForKey:(key) value:@""
                                       table:nil]
```

- Uses the localizedStringForKey: method to look up the string for a given key
  - nil is the default table name and uses “Localized.strings
  -



localizedStringForKey:value:table:

Returns a localized version of the string designated by the specified key and residing in the specified table.

```
- (NSString *)localizedStringForKey:(NSString *)key value:(NSString *)value table:(NSString *)tableName
```

Parameters

key
The key for a string in the table identified by *tableName*.

value
The value to return if *key* is `nil` or if a localized string for *key* can't be found in the table.

tableName
The receiver's string table to search. If *tableName* is `nil` or is an empty string, the method attempts to use the table in `Localizable.strings`.

LOCALIZATION

```
// This is a localization test (it has nothing to do with accessibility)
let greeting = NSLocalizedString("GREETING", comment: "This is a greeting.")
print("Greeting \(greeting)")
```

- Comment is used for “genstrings” tool
- Appears next to default string as aid for translator
  - Useful for large projects to separate the code from the strings

LOCALIZATION

genstrings(1) BSD General Commands Manual genstrings(1)

NAME

genstrings -- generate string table from source code

SYNOPSIS

genstrings [**-a**] [**-s** <routine>] [**-skipTable** <Table>] [**-noPositionalParameters**]
[**-bigEndian** | **-littleEndian**] [**-u**] [**-macRoman**] [**-q**] [**-o** <outputDir>] file
...

DESCRIPTION

The **genstrings** utility generates a **.strings** file(s) from the C or Objective-C (.c or .m) source code file(s) given as the argument(s). A .strings file is used for localizing an application for different languages, as described under "Internationalization" in the Cocoa Developer Documentation.

* C and Objective-C:

Source lines containing text of the form NSLocalizedString("key", comment) or CFCopyLocalized-String("key", comment) will generate an appropriate string table entry to a file named Localiz-able.strings.

Source lines containing NSLocalizedStringFromTable("key", Table, comment) or CFCopyLocalized-StringFromTable("key", Table, comment) will generate an appropriate string table entry in a file named Table.strings.

Source lines with NSLocalizedStringFromTableInBundle("key", Table, bundle, comment) or CFCopyLocalized-

LOCALIZATION

```
        } else {
            // This doesn't work because Voice O
            // Bug when using QuickTime where th
            //UIAccessibilityPostNotification(UI
            let string = "Voice over is not enab
            println(string)

            let utterance = AVSpeechUtterance(st
            utterance.voice = AVSpeechSynthesisV
            let synthesizer = AVSpeechSynthesize
            synthesizer.speakUtterance(utterance

        }

        // This is a localization test (it has nothing to do with accessibility)
        let greeting = NSLocalizedString("GREETING", comment: "This is a greeting that will be displayed as a console message when the application
            launches.")
        println("Greeting \(greeting)")


        let age = String(format: NSLocalizedString("AGE", comment: "Comment doesn't do anything useful"),"5")
        println("Age \(age)")

        return true
    }

    func applicationWillResignActive(application: UIApplication) {
        // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such
```

HelpingHand 〉 Localizable.strings 〉 Localizable.strings (English) 〉 No Selection

```
/*
  Localizable.strings
  HelpingHand

  Created by T. Andrew Binkowski on 6/1/15.
  Copyright (c) 2015 The University of Chicago. All rights reserved.
*/

GREETING="Hello (English)";
AGE="I am %@ years old";
```

Parameters

Parameters

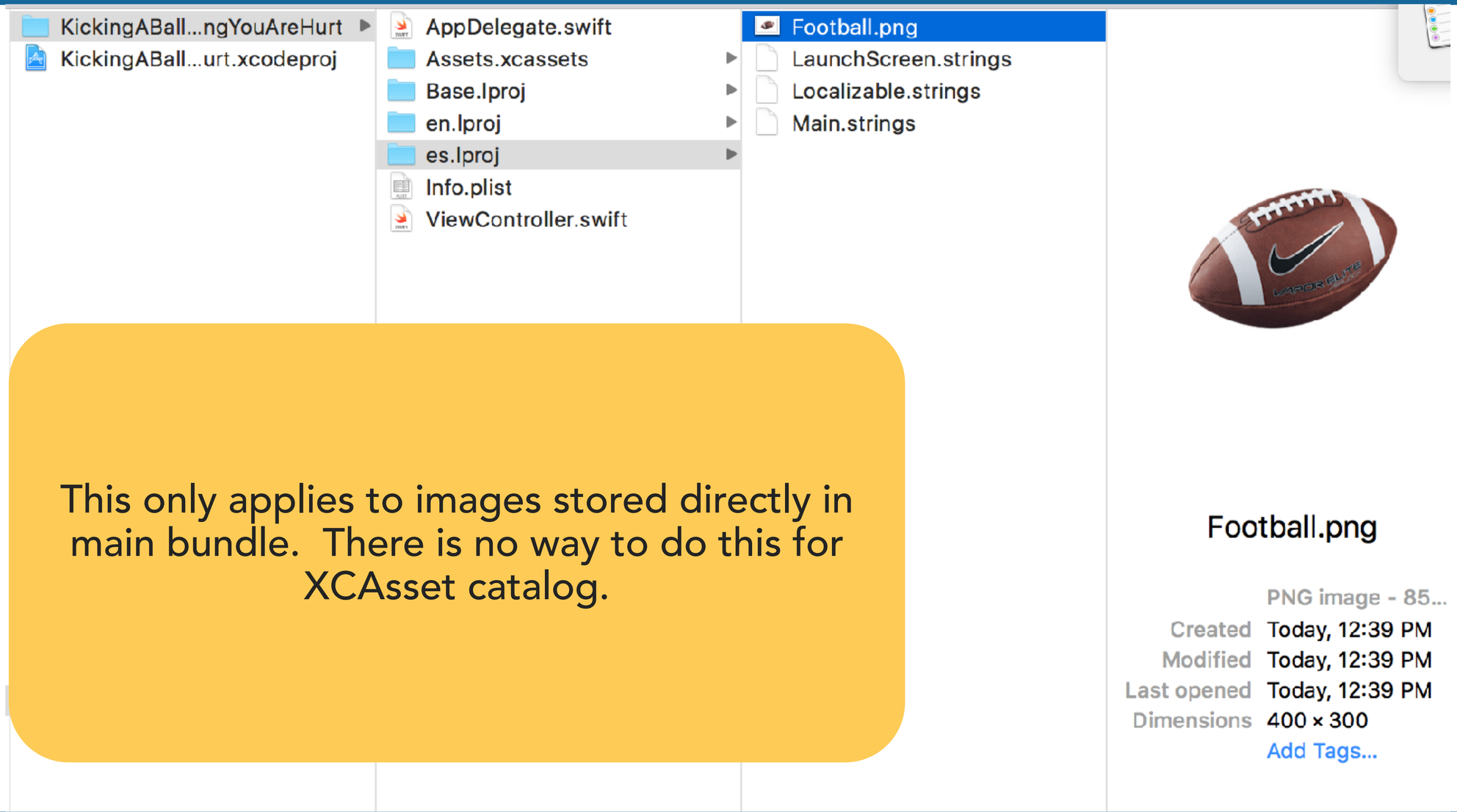
No Selection

```
Voice over is not enabled
Greeting Hello (English)
Age I am 5 years old
```

LOCALIZING ASSETS

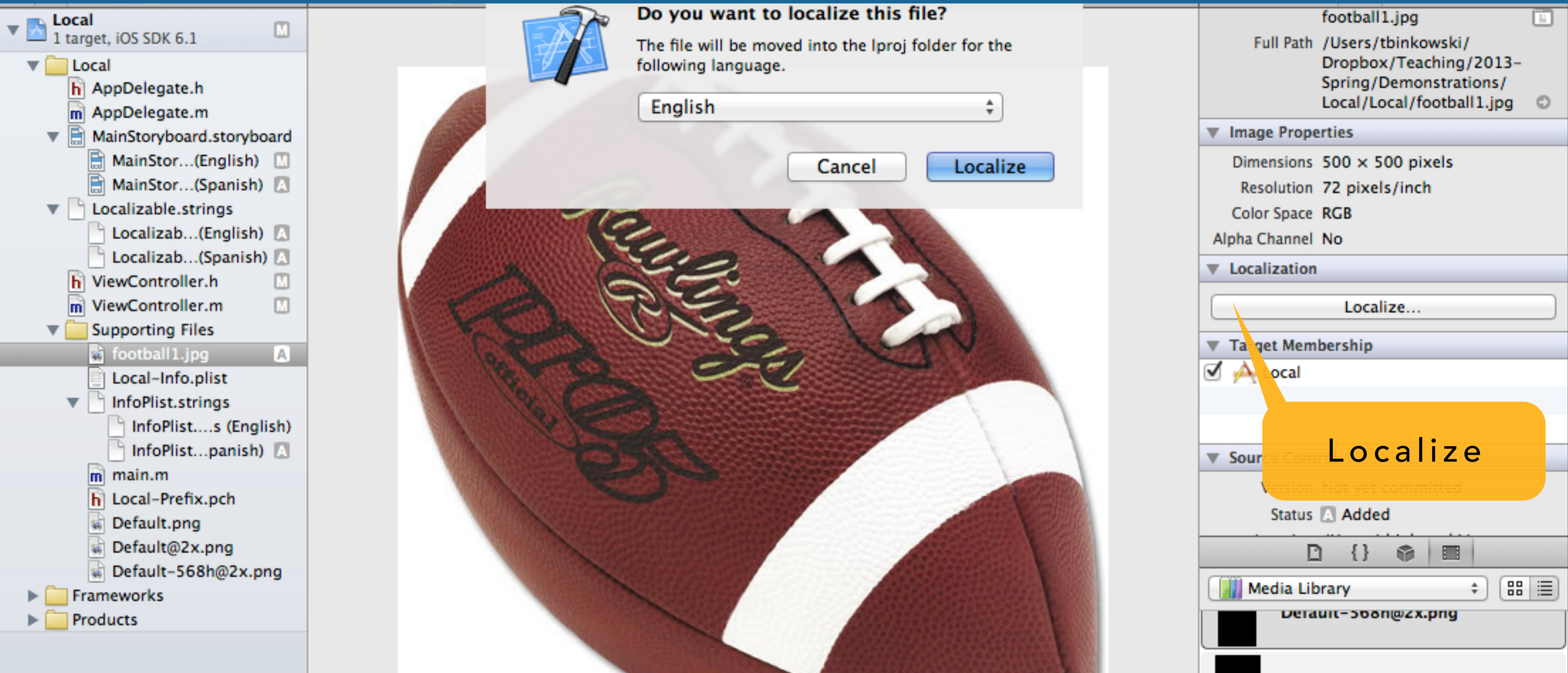
LOCALIZING ASSETS

LOCALIZING ASSETS

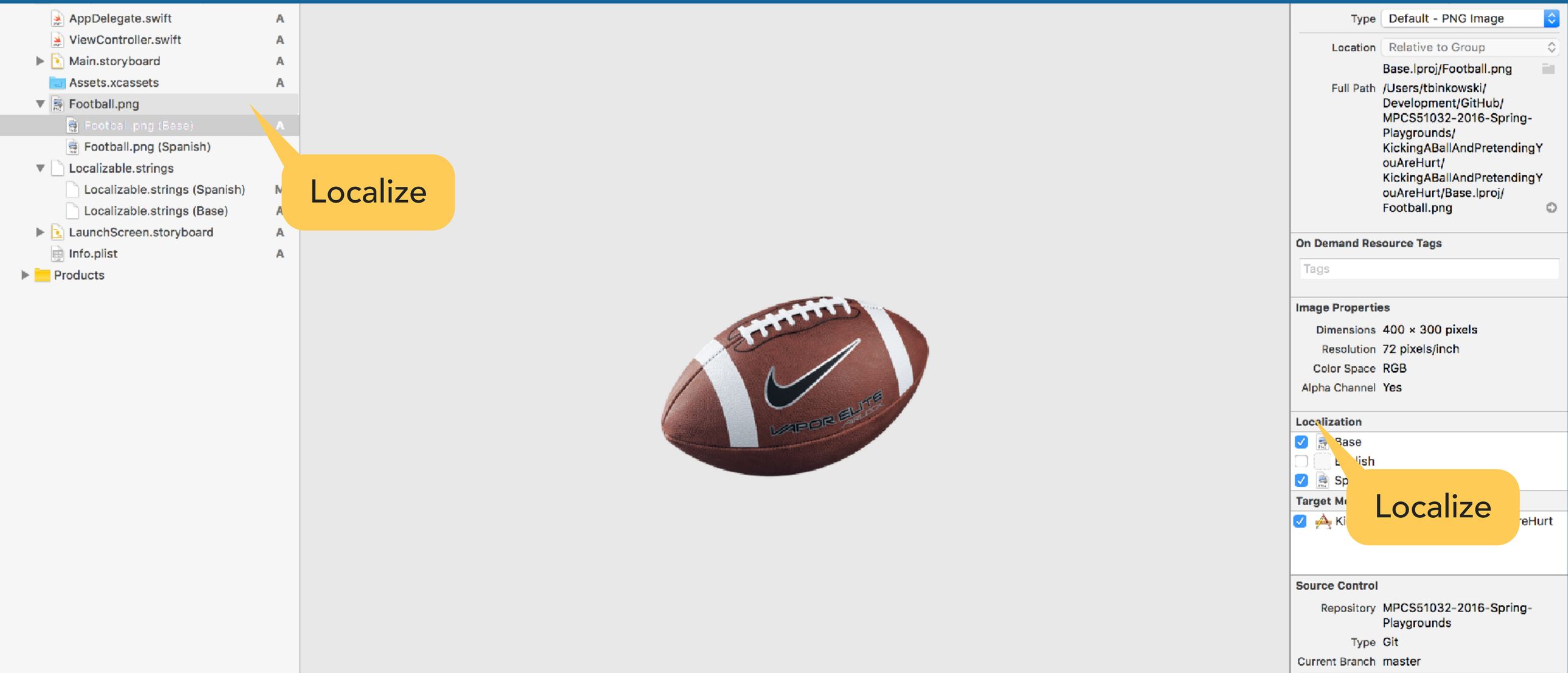


This only applies to images stored directly in main bundle. There is no way to do this for XCAsset catalog.

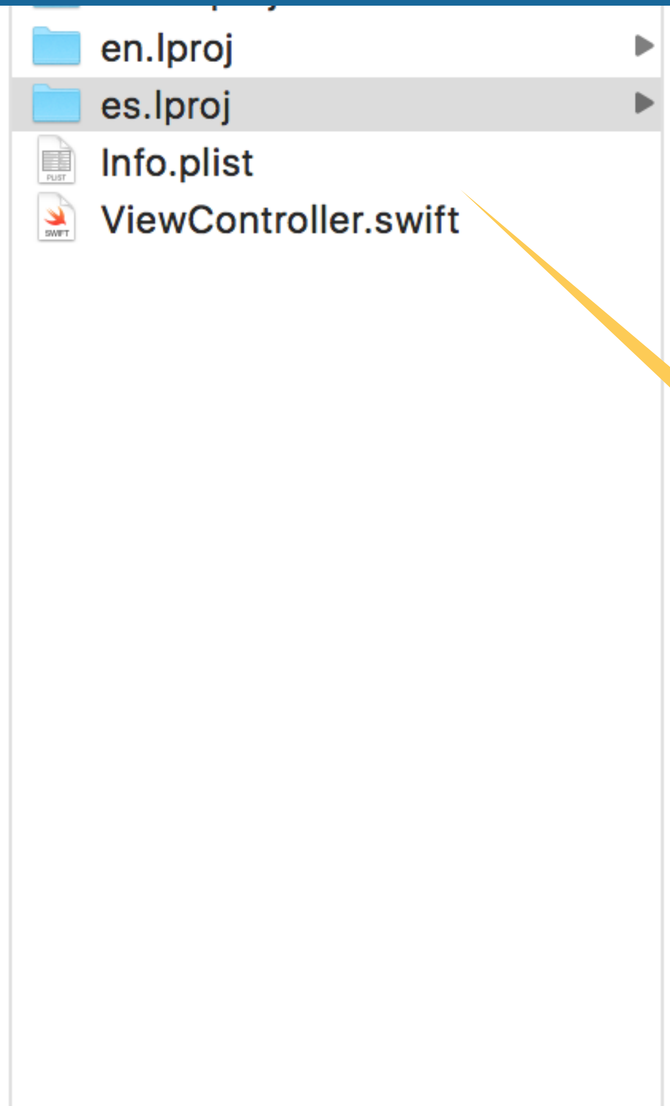
LOCALIZING ASSETS



LOCALIZING ASSETS



LOCALIZING ASSETS



Directory structure for localized application

Football.png

PNG image - 85...
Created Today, 12:39 PM
Modified Today, 12:39 PM
Last opened Today, 12:39 PM

LOCALIZING ASSETS

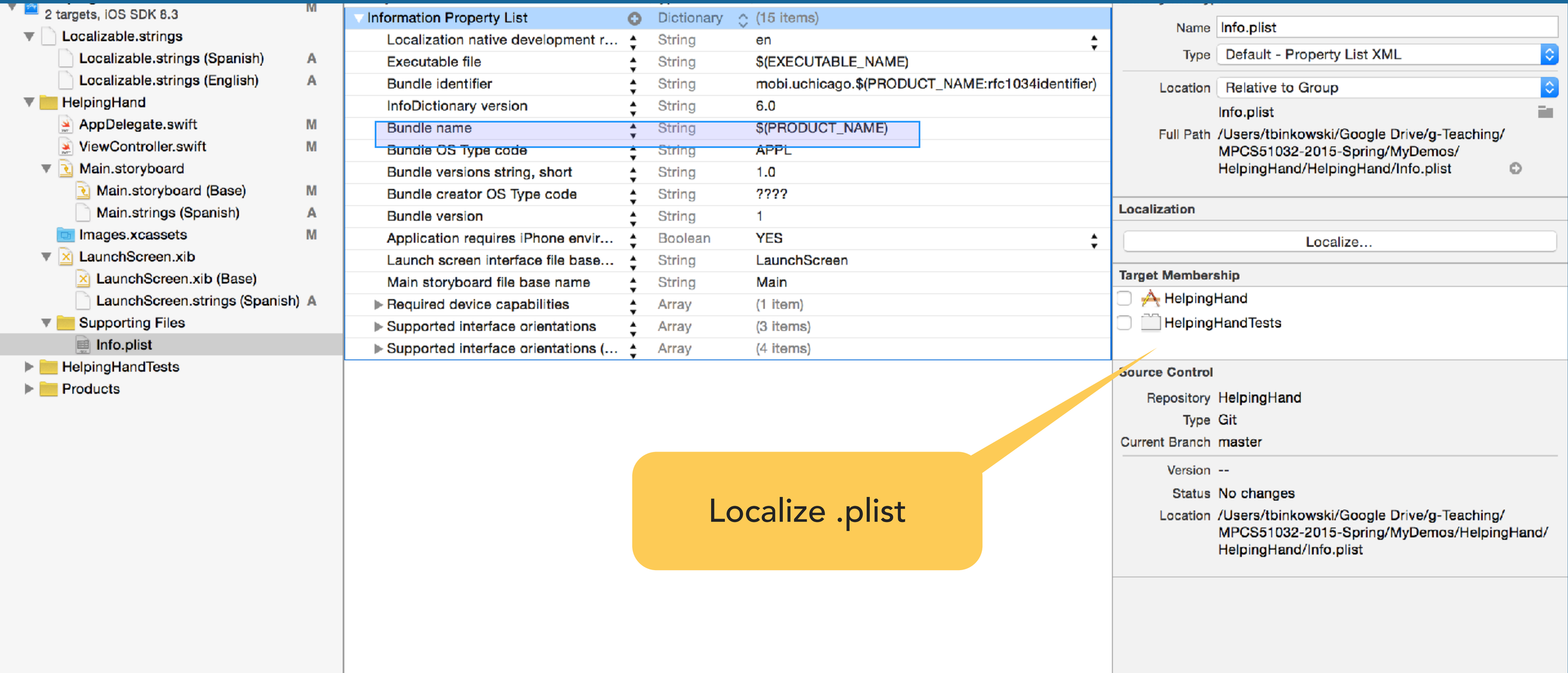
LOCALIZING ASSETS

- Alternative ways
  - Load all images into a XCAsset catalog with unique names
    - Football-en
    - Footabll-es
  - At runtime extract the regionalized name
    - func imageNameWithRegion(name: String) ->String

Or Apple will improve support in XCAsset Catalog

LOCALIZING APP NAME

LOCALIZING APP NAME



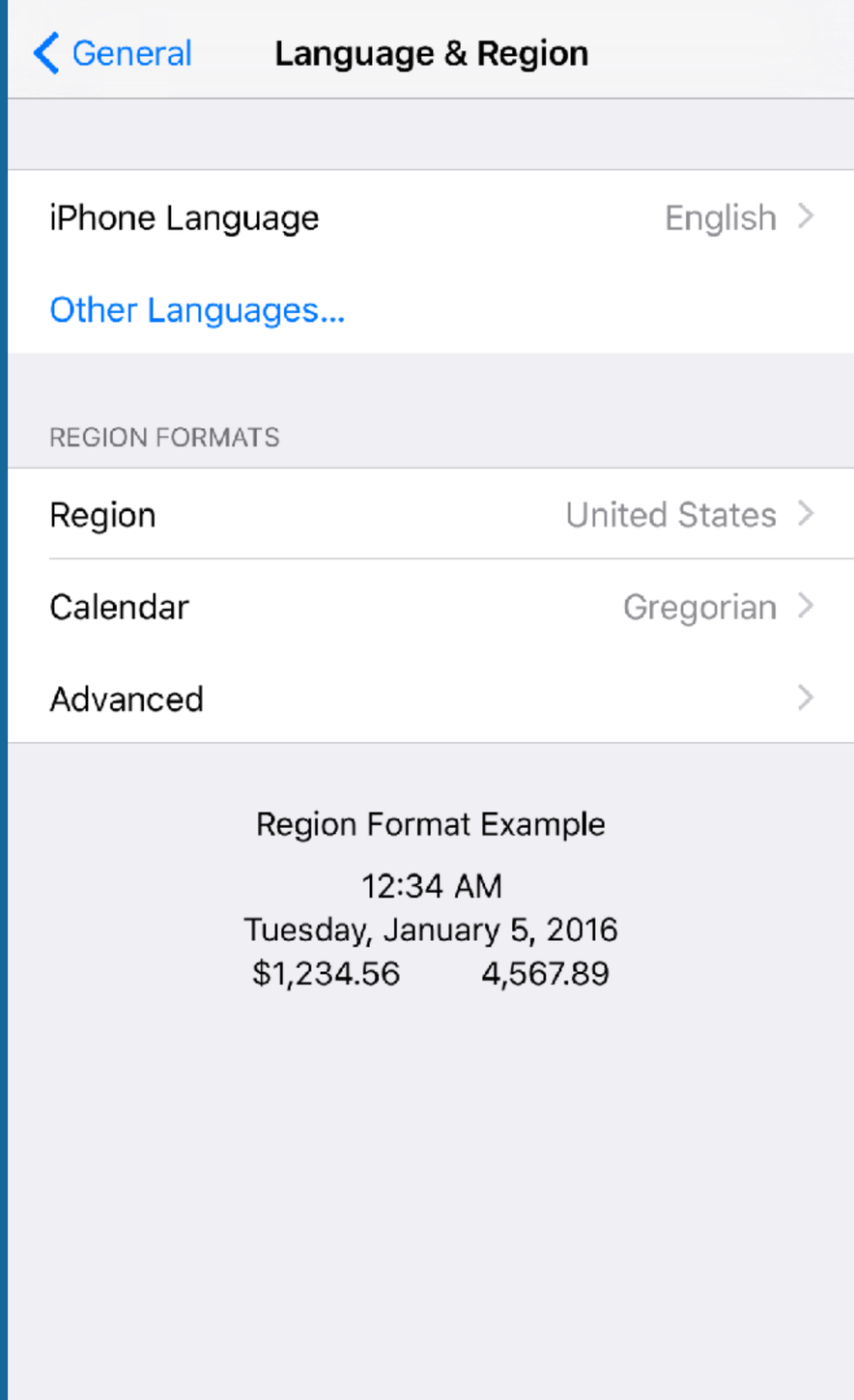
LOCALIZING APP NAME

- There are many different opportunities to localize by tweaking values in the .plist
  - Different launch options
  - Different servers to hit
  - Different Storyboard files to load
  - …
- You can put any values in .plist and read them at runtime

REGIONALIZE YOUR APP

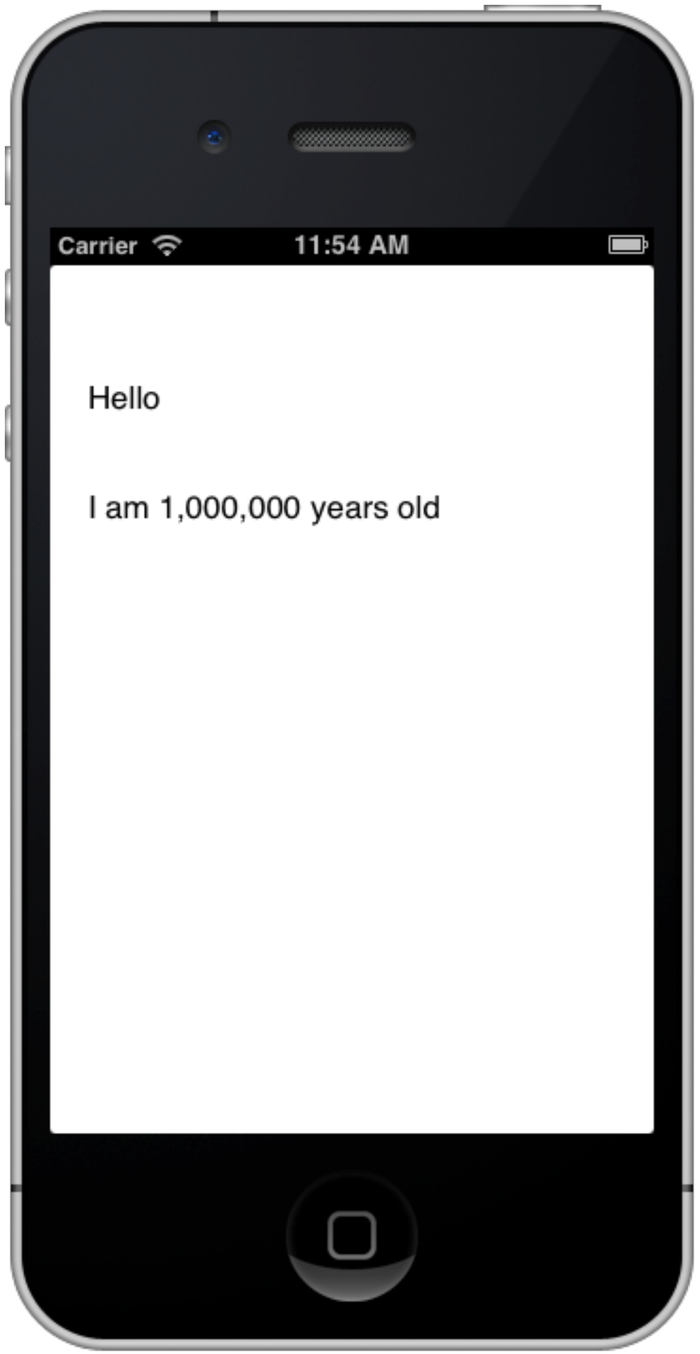
REGIONALIZE

- The region is not the same as the language
  - Different formatting based on the region
- Examples
  - Color vs. colour



REGIONALIZE

- Example
  - 1,000,000 US
  - 1.000.000 Spain



REGIONALIZE

- Simply use 'formatting' APIs and you get all the behavior for free
- NSNumberFormatter will format based on the region

```
NSNumberFormatter *numberFormatter = [[NSNumberFormatter alloc] init];
[numberFormatter setNumberStyle:NSNumberFormatterDecimalStyle];
NSString *numberString = [numberFormatter stringFromNumber:@(1000000)];

self.ageLabel.text = [NSString stringWithFormat:NSLocalizedString(@"I am %@ years old", nil),numberString];
```

THE UNIVERSITY OF CHICAGO

ADVANCED iOS APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 9

FINAL PROJECTS

FINAL PROJECTS

FINAL PROJECT REQUIREMENTS

MPCS51032 - SPRING 2020

Application

All applications will be graded on implementing and complying with the following specifications, as well as the implementation of their functionality. The completeness

FINAL PROJECTS

- Due date
  - Graduating students need projects submitted by June 5th at 5:00pm
  - Non-graduating by June 9th at 5:29pm

CLASS IN REVIEW

CLASS IN REVIEW

- Children's Book App
  - PageViewController
  - UIKitDynamics, Property Animators
  - AVSpeechSynthesis, AVAudioSessions
  - Objective-C
  - NIB base view controllers
  - Shaq and the Beanstalk...

CLASS IN REVIEW

- Journal
  - CloudKit
  - Push Notifications
  - Sync engine with Codable Protocols
  - App Groups
  - Extensions (Today, Share)
  - MLKit Vision Framework
  - MapKit (clustering)
  - Local Authentication
  - Notification Center
  - UISearchController

CLASS IN REVIEW

- Game Playground
  - SpriteKit

CLASS IN REVIEW

- Pop Quiz
  - watchOS using UIKit
  - watchOS using SwiftUI
  - CoreData
  - Cross platform development

CLASS IN REVIEW

- Pop Quiz
  - watchOS using UIKit
  - watchOS using SwiftUI
  - CoreData
  - Cross platform development
  - Notifications
  - Complications

THE UNIVERSITY OF CHICAGO

THANK YOU

SPRING

Binkowski