TimeoutController

Behaviour Composition with Storyboard's

Lets define some behaviour?

Video Player Controls

- When user taps on the video, visibility should be toggled
- While the user is interacting with controls, the view should remain visible
- When user completes an interaction, controls should hide after an interval

What do we need?

— A view to hold our controls
— Touch event handling for our controls
— A tap gesture to toggle visibility
— A timer

Separation of Concerns

Timer vs Interaction

- `TimeoutController`
- `TimeoutInteractionProvider`

Storyboard/XIB support

- `TimeoutControllerHost`

What is a TimeoutController?

— A wrapper around a timer

— An observer for interaction providers

```
final class TimeoutController: NSObject {
    var interactionProviders: [TimeoutInteractionProviding]?
    var timeout: TimeInterval
    var timeoutHandler: () -> Void

    func resume()
    func pause()
}
```

TimeoutController (Internal)

Interaction provider's call one of the following methods to provide feedback.

```
var timer: Timer?

func interactionBegan()
func interactionEnded()
```

What is an interaction provider?

A view, gesture or other object that notifies a `TimeoutController` when an interaction occurs.

```swift
/*
 Automatic conformance is provided for UIView and UIGestureRecognizer subclasses.
 */
protocol TimeoutInteractionProviding {

    /// The TimeoutController associated with this provider
    var timeoutController: TimeoutController? { get }

}
```

VideoViewController

```
func viewDidLoad() {
    let providers = [playPauseButton, volumeSlider, backgroundTapGesture]

    timeoutController = TimeoutController(providers: providers, timeout: 3)

    timeoutController.timeoutHandler = {
        setControls(hidden: true)
    }
}

func handleBackgroundTapGesture() {
    if isControlsViewHidden {
        setControls(hidden: false)
        timeoutController?.resume()
    } else {
        setControls(hidden: true)
        timeoutController?.pause()
    }
}
```

Full implementation is actually over 25 lines of code

TimeoutControllerHost

Lets extract everything from our view controller

```swift
final class TimeoutControllerHost: NSObject {
    var timeoutController: TimeoutController?

    weak var viewToHide: UIView
    weak var tapGesture: UITapGestureRecognizer

    private func tapGestureHandler()
    private func timeoutHandler()
}
```

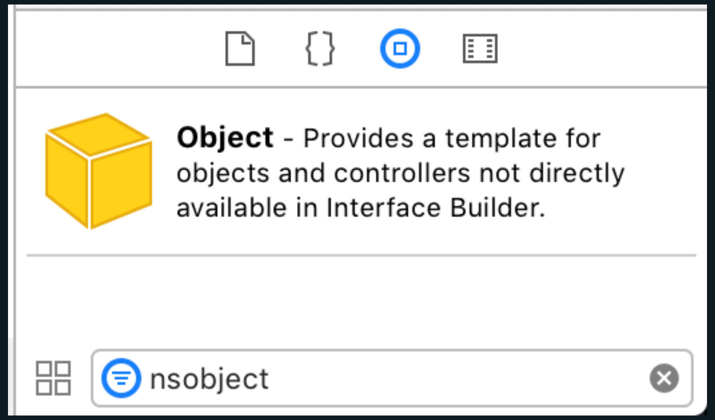
VideoViewController

```
private var timeoutHost: TimeoutControllerHost
private var controlsView: ControlsView
private var tapGesture: UITapGestureRecognizer

override func viewDidLoad() {
    timeoutHost = TimeoutControllerHost(viewToHide: controlsView, tapGesture: tapGesture)

    let providers = [playPauseButton, volumeSlider, backgroundTapGesture]
    timeoutHost.timeoutController = TimeoutController(providers: providers, timeout: 3)
}
```

Down to 8 lines of code! But we can do better.

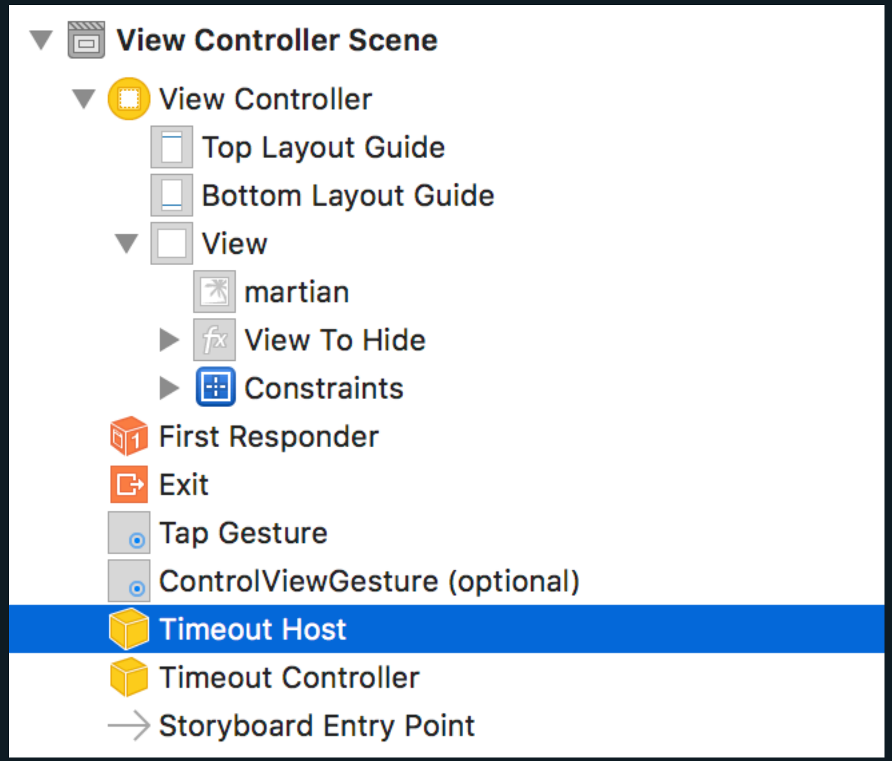
Storyboard Support

Lets add some Storyboard support



Object - Provides a template for objects and controllers not directly available in Interface Builder.

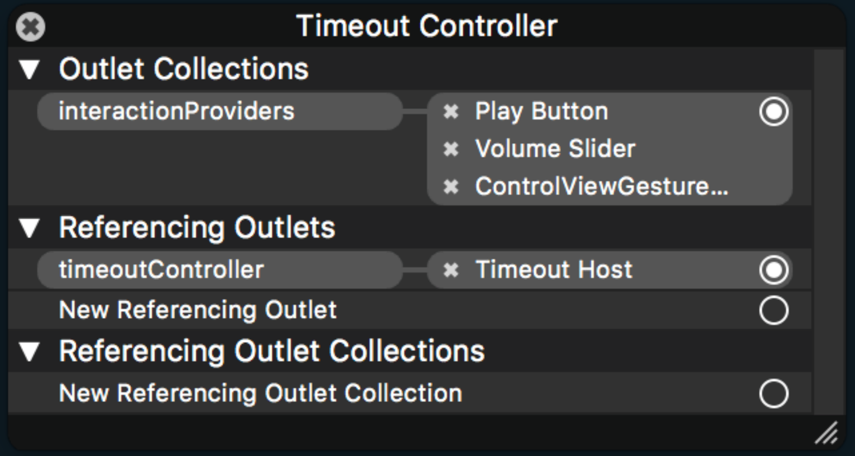
nsobject



- View Controller Scene
  - View Controller
    - Top Layout Guide
    - Bottom Layout Guide
    - View
      - martian
      - View To Hide
      - Constraints
  - First Responder
  - Exit
  - Tap Gesture
  - ControlViewGesture (optional)
  - Timeout Host
  - Timeout Controller
  - Storyboard Entry Point

We can use NSObject's to setup our Scene

Connect IBOutlet's


Timeout Controller

▼ Outlet Collections

interactionProviders — Play Button, Volume Slider, ControlViewGesture...

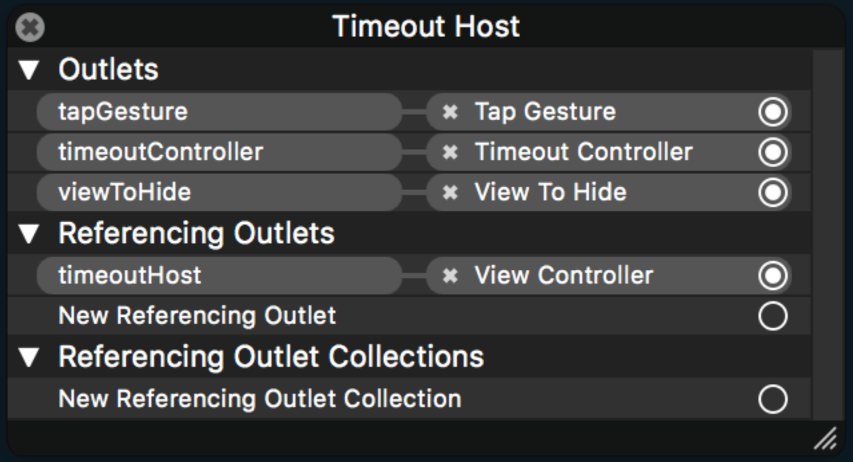
▼ Referencing Outlets

timeoutController — Timeout Host

New Referencing Outlet

▼ Referencing Outlet Collections

New Referencing Outlet Collection


Timeout Host

▼ Outlets

tapGesture — Tap Gesture

timeoutController — Timeout Controller

viewToHide — View To Hide

▼ Referencing Outlets

timeoutHost — View Controller

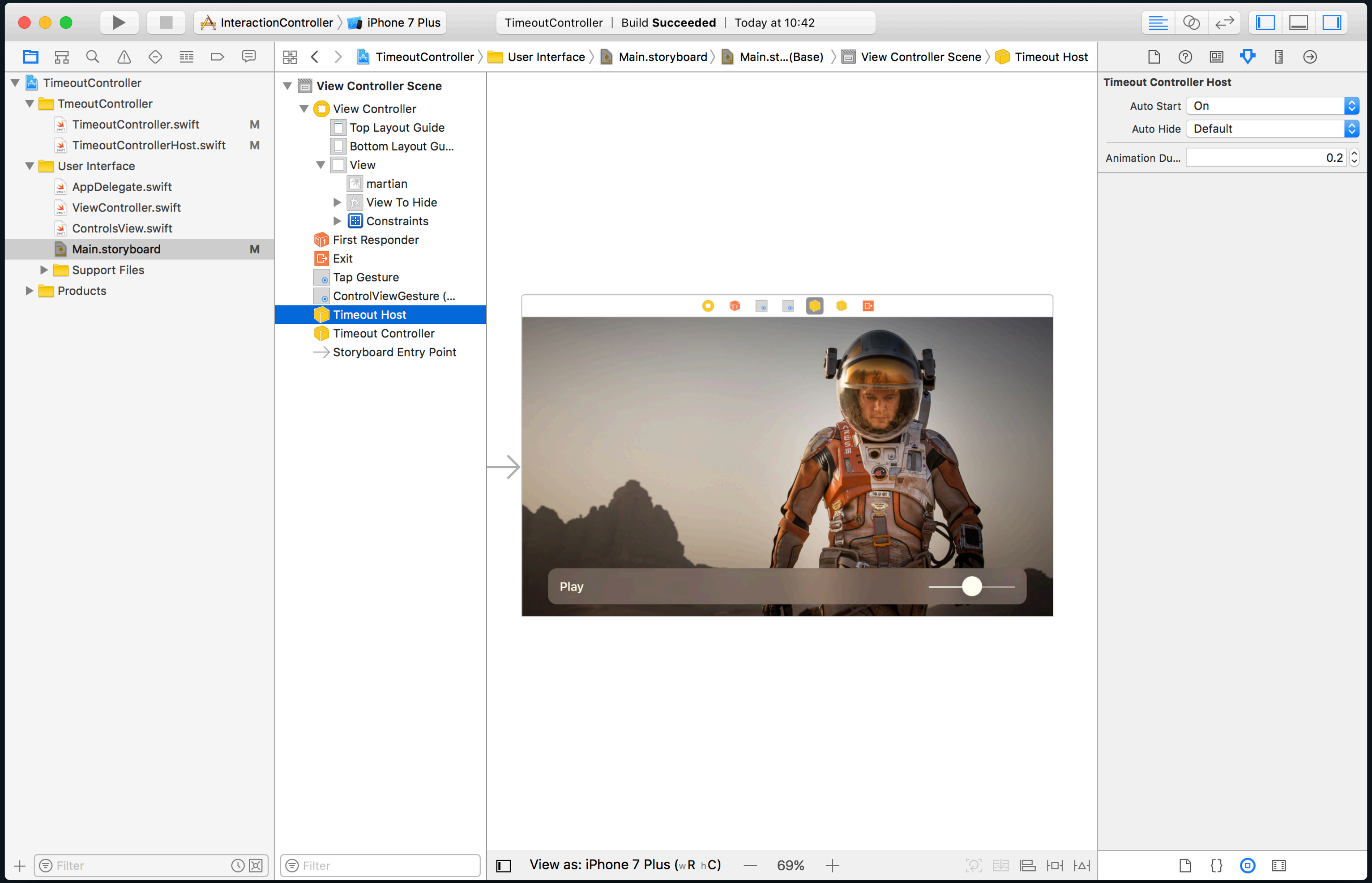
New Referencing Outlet

▼ Referencing Outlet Collections

New Referencing Outlet Collection

- Our player controls will be our interaction providers
- We have to connect our host to our timeout controller

Completed Scene



Timeout Controller Host

Auto Start: On

Auto Hide: Default

Animation Du...: 0.2

VideoViewController

```
@IBOutlet private var timeoutHost: TimeoutControllerHost!
```

1 line of code!!

Now build and run.