App Frameworks

#WWDC17

Now Playing and Remote Commands on tvOS

Session 251

Justin Voss, tvOS Engineer

Now-Playing Info

Metadata Appearance

AVPlayerViewController

TV Remote app

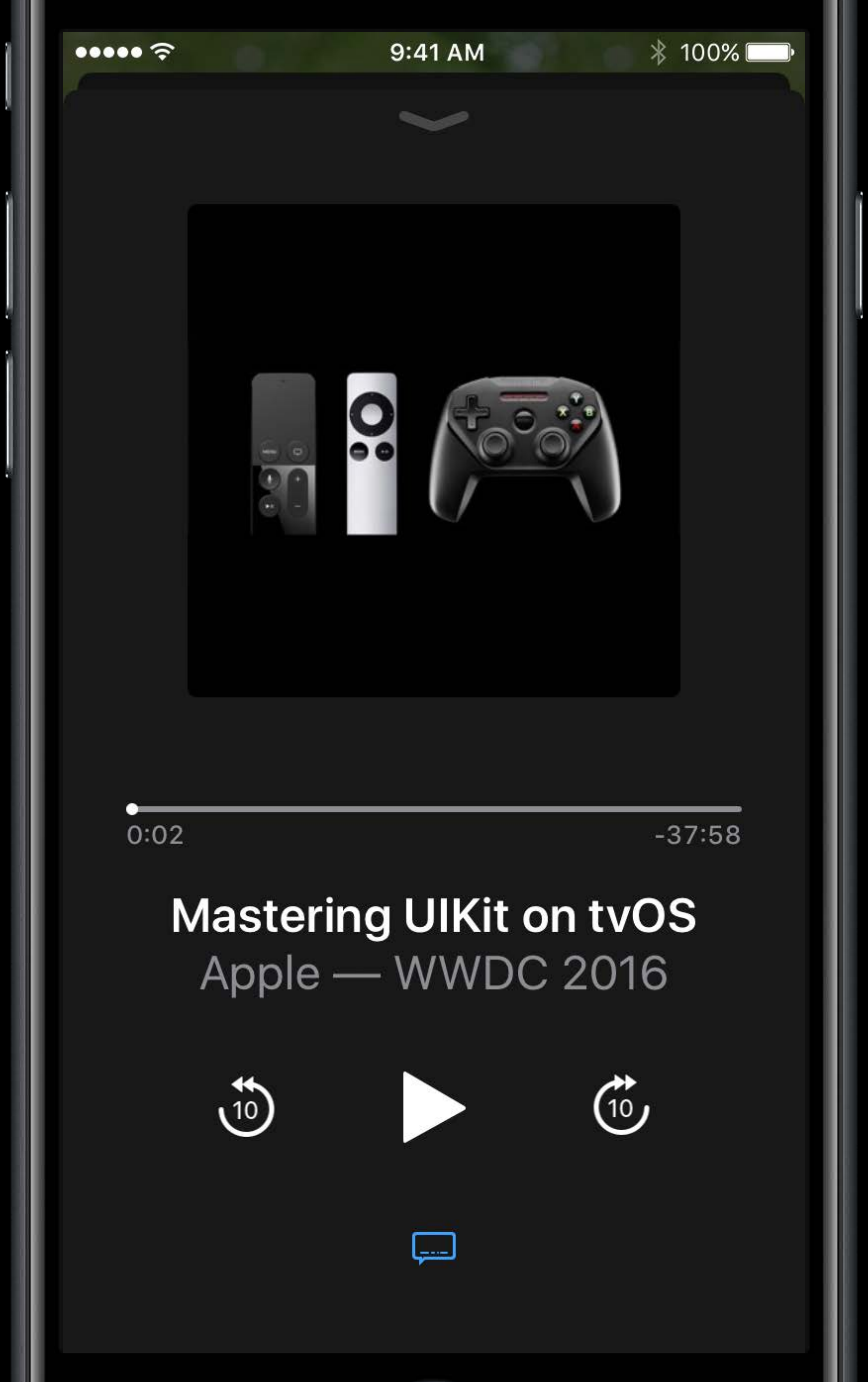
Now-Playing badge

Now-Playing idle screen

Mastering UIKit on tvOS

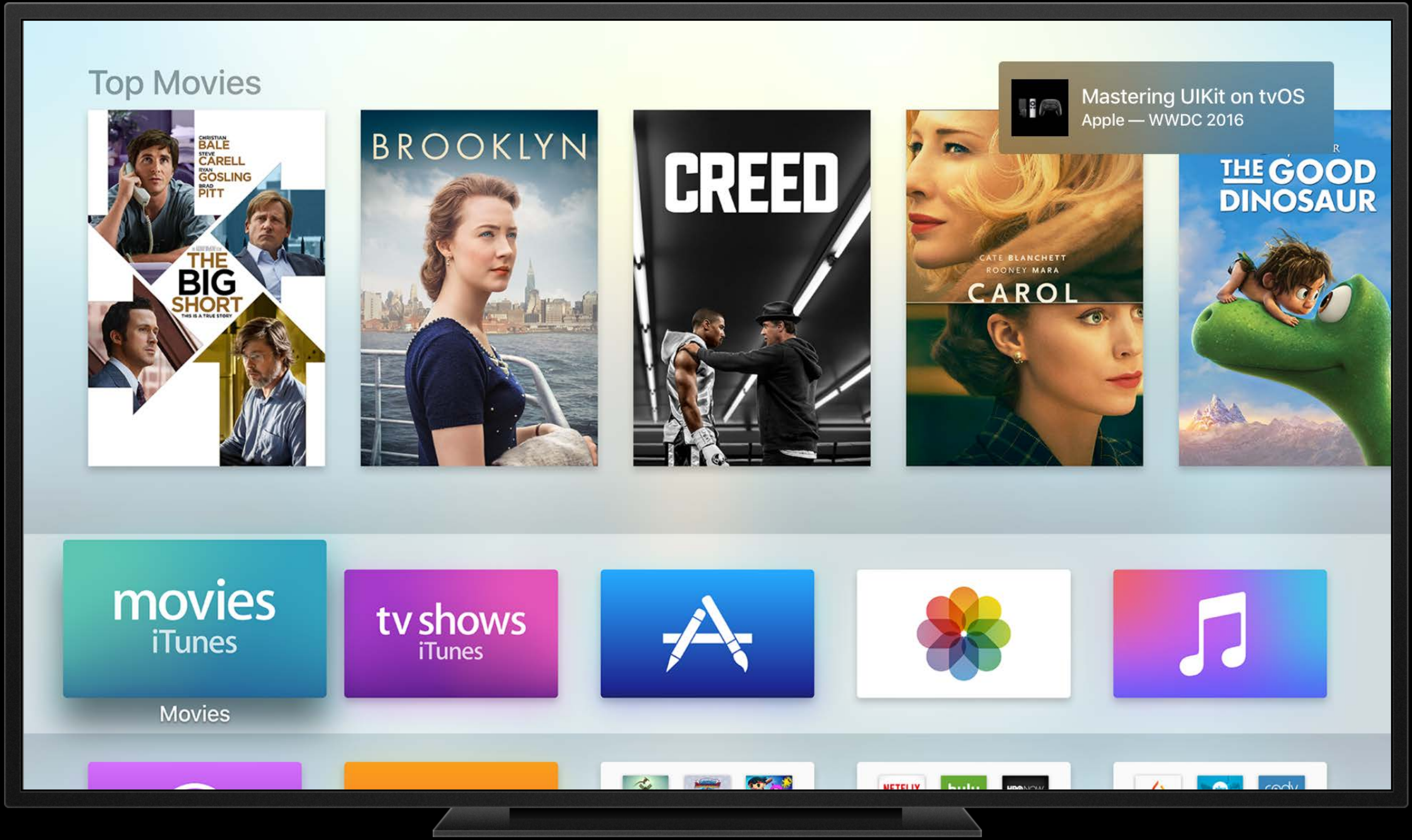
38 min 1987 CC

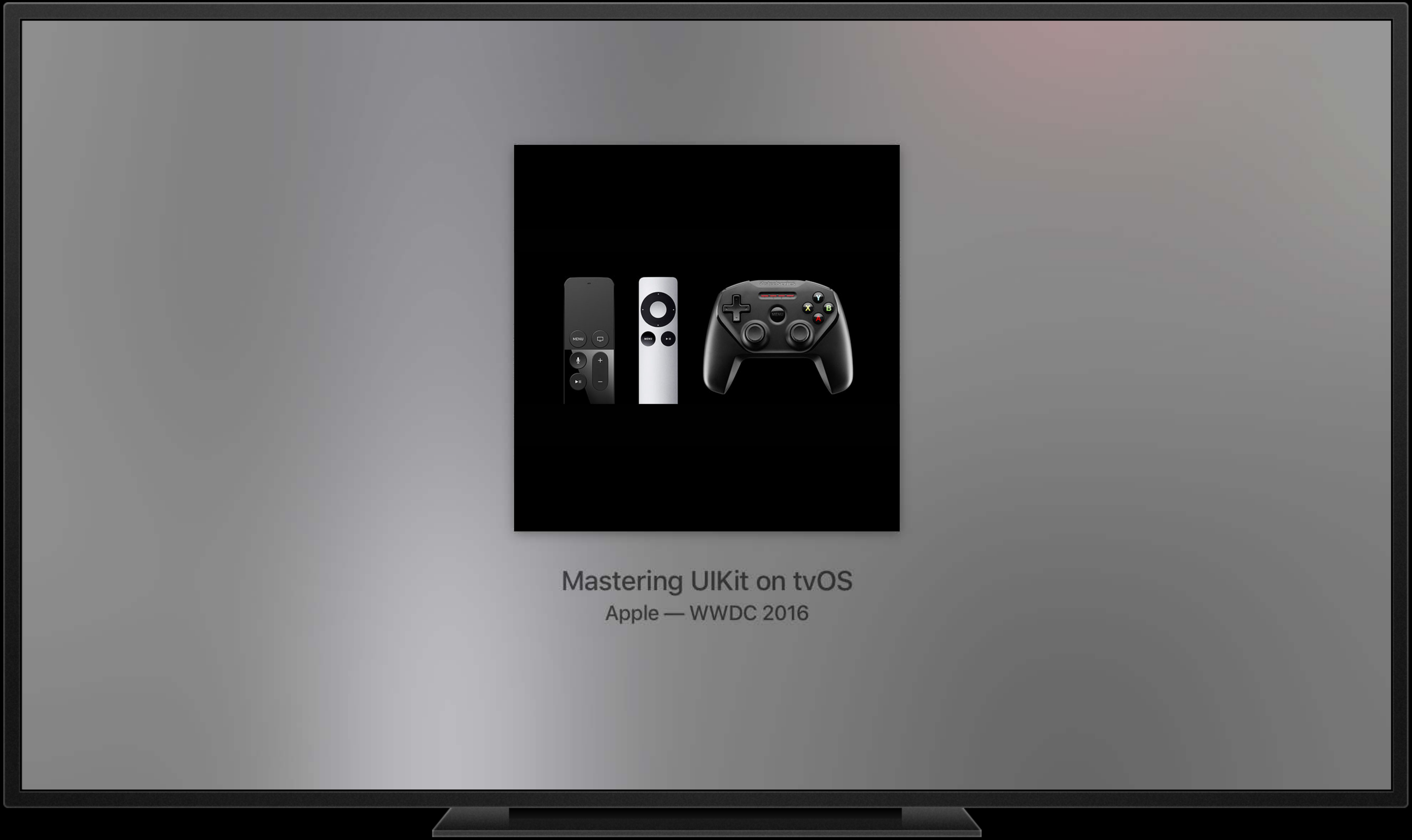
Learn to take your tvOS beyond the basics and create amazing big-screen experiences with these UIKit tips and tricks.



Mastering UIKit on tvOS

Apple — WWDC 2016


Top Movies
Mastering UIKit on tvOS
Apple — WWDC 2016
CHRISTIAN BALE
STEVE CARELL
RYAN GOSLING
BRAD PITT
THE BIG SHORT
BROOKLYN
CREED
CATE BLANCHETT
ROONEY MARA
CAROL
THE GOOD DINOSAUR
movies
iTunes
Movies
tv shows
iTunes



Mastering UIKit on tvOS

Apple — WWDC 2016

TVML Metadata

Add custom metadata using properties on the Javascript type MediaItem

- Title, subtitle, description
- Artwork URL
- Content rating, explicit-ness

```
var item = new MediaItem('video', 'https://example.com/video.mp4');

item.title = 'Best Practices on tvOS';
item.description = 'Learn to make your apps great.';
item.artworkImageURL = 'https://example.com/image.jpg';
```

AVMetadataItem

Set these properties:

- identifier, to indicate what kind of metadata item this is
- value, to provide the metadata itself
- extendedLanguageTag, to indicate what language this metadata is for
  - Most of the time you should provide “und” to use the same data everywhere
- dataType, to indicate the format of the value
  - This is only required for artwork image metadata

```swift
// Examples of String Metadata

let playerItem = AVPlayerItem(url: videoURL)

let title = AVMutableMetadataItem()
title.identifier = AVMetadataCommonIdentifierTitle
title.value = "Mastering UIKit on tvOS" as NSString
title.extendedLanguageTag = "und"

let description = AVMutableMetadataItem()
description.identifier = AVMetadataCommonIdentifierDescription
description.value = "Learn to make your apps great." as NSString
description.extendedLanguageTag = "und"

playerItem.externalMetadata = [title, description]
```

```swift
// Examples of String Metadata

let playerItem = AVPlayerItem(url: videoURL)

let title = AVMutableMetadataItem()
title.identifier = AVMetadataCommonIdentifierTitle
title.value = "Mastering UIKit on tvOS" as NSString
title.extendedLanguageTag = "und"

let description = AVMutableMetadataItem()
description.identifier = AVMetadataCommonIdentifierDescription
description.value = "Learn to make your apps great." as NSString
description.extendedLanguageTag = "und"

playerItem.externalMetadata = [title, description]
```

```swift
// Examples of String Metadata

let playerItem = AVPlayerItem(url: videoURL)

let title = AVMutableMetadataItem()
title.identifier = AVMetadataCommonIdentifierTitle
title.value = "Mastering UIKit on tvOS" as NSString
title.extendedLanguageTag = "und"

let description = AVMutableMetadataItem()
description.identifier = AVMetadataCommonIdentifierDescription
description.value = "Learn to make your apps great." as NSString
description.extendedLanguageTag = "und"

playerItem.externalMetadata = [title, description]
```

```swift
// Examples of String Metadata

let playerItem = AVPlayerItem(url: videoURL)

let title = AVMutableMetadataItem()
title.identifier = AVMetadataCommonIdentifierTitle
title.value = "Mastering UIKit on tvOS" as NSString
title.extendedLanguageTag = "und"

let description = AVMutableMetadataItem()
description.identifier = AVMetadataCommonIdentifierDescription
description.value = "Learn to make your apps great." as NSString
description.extendedLanguageTag = "und"

playerItem.externalMetadata = [title, description]
```

```swift
// Examples of String Metadata

let playerItem = AVPlayerItem(url: videoURL)

let title = AVMutableMetadataItem()
title.identifier = AVMetadataCommonIdentifierTitle
title.value = "Mastering UIKit on tvOS" as NSString
title.extendedLanguageTag = "und"

let description = AVMutableMetadataItem()
description.identifier = AVMetadataCommonIdentifierDescription
description.value = "Learn to make your apps great." as NSString
description.extendedLanguageTag = "und"

playerItem.externalMetadata = [title, description]
```

```swift
// Examples of Date Metadata

let date = Date(timeIntervalSince1970: 546782400) // April 30, 1987

let metadataDateFormatter = DateFormatter()
metadataDateFormatter.dateFormat = "yyyy-MM-dd'T'HH:mm:ssZ"

let creationDate = AVMutableMetadataItem()
creationDate.identifier = AVMetadataCommonIdentifierCreationDate
creationDate.value = metadataDateFormatter.string(from: date) as NSString
creationDate.extendedLanguageTag = "und"
```

```
// Examples of Date Metadata

let date = Date(timeIntervalSince1970: 546782400) // April 30, 1987

let metadataDateFormatter = DateFormatter()
metadataDateFormatter.dateFormat = "yyyy-MM-dd'T'HH:mm:ssZ"

let creationDate = AVMutableMetadataItem()
creationDate.identifier = AVMetadataCommonIdentifierCreationDate
creationDate.value = metadataDateFormatter.string(from: date) as NSString
creationDate.extendedLanguageTag = "und"
```

```swift
// Examples of Date Metadata

let date = Date(timeIntervalSince1970: 546782400) // April 30, 1987

let metadataDateFormatter = DateFormatter()
metadataDateFormatter.dateFormat = "yyyy-MM-dd'T'HH:mm:ssZ"

let creationDate = AVMutableMetadataItem()
creationDate.identifier = AVMetadataCommonIdentifierCreationDate
creationDate.value = metadataDateFormatter.string(from: date) as NSString
creationDate.extendedLanguageTag = "und"
```

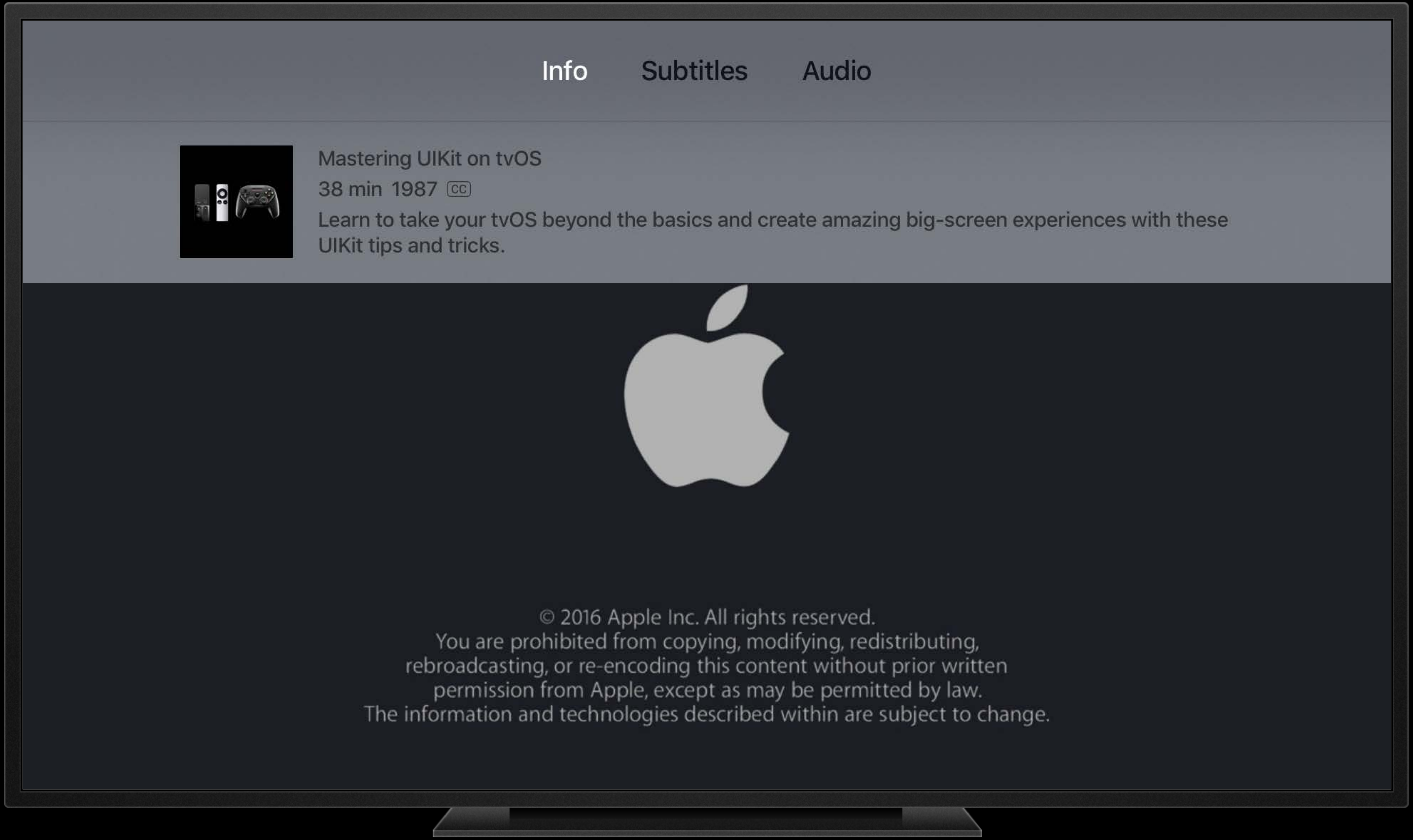
```swift
// Examples of Date Metadata

let date = Date(timeIntervalSince1970: 546782400) // April 30, 1987

let metadataDateFormatter = DateFormatter()
metadataDateFormatter.dateFormat = "yyyy-MM-dd'T'HH:mm:ssZ"

let creationDate = AVMutableMetadataItem()
creationDate.identifier = AVMetadataCommonIdentifierCreationDate
creationDate.value = metadataDateFormatter.string(from: date) as NSString
creationDate.extendedLanguageTag = "und"
```

```
// Example of Image Metadata

let artworkPath = Bundle.main.path(forResource: "MyArtwork", ofType: "jpg")!
let artworkData = FileManager.default.contents(atPath: artworkPath)!

let artwork = AVMutableMetadataItem()
artwork.identifier = AVMetadataCommonIdentifierArtwork
artwork.value = artworkData as NSData
artwork.dataType = kCMMetadataBaseDataType_JPEG as String
artwork.extendedLanguageTag = "und"
```

```swift
// Example of Image Metadata

let artworkPath = Bundle.main.path(forResource: "MyArtwork", ofType: "jpg")!
let artworkData = FileManager.default.contents(atPath: artworkPath)!

let artwork = AVMutableMetadataItem()
artwork.identifier = AVMetadataCommonIdentifierArtwork
artwork.value = artworkData as NSData
artwork.dataType = kCMMetadataBaseDataType_JPEG as String
artwork.extendedLanguageTag = "und"
```

```swift
// Example of Image Metadata

let artworkPath = Bundle.main.path(forResource: "MyArtwork", ofType: "jpg")!
let artworkData = FileManager.default.contents(atPath: artworkPath)!

let artwork = AVMutableMetadataItem()
artwork.identifier = AVMetadataCommonIdentifierArtwork
artwork.value = artworkData as NSData
artwork.dataType = kCMMetadataBaseDataType_JPEG as String
artwork.extendedLanguageTag = "und"
```

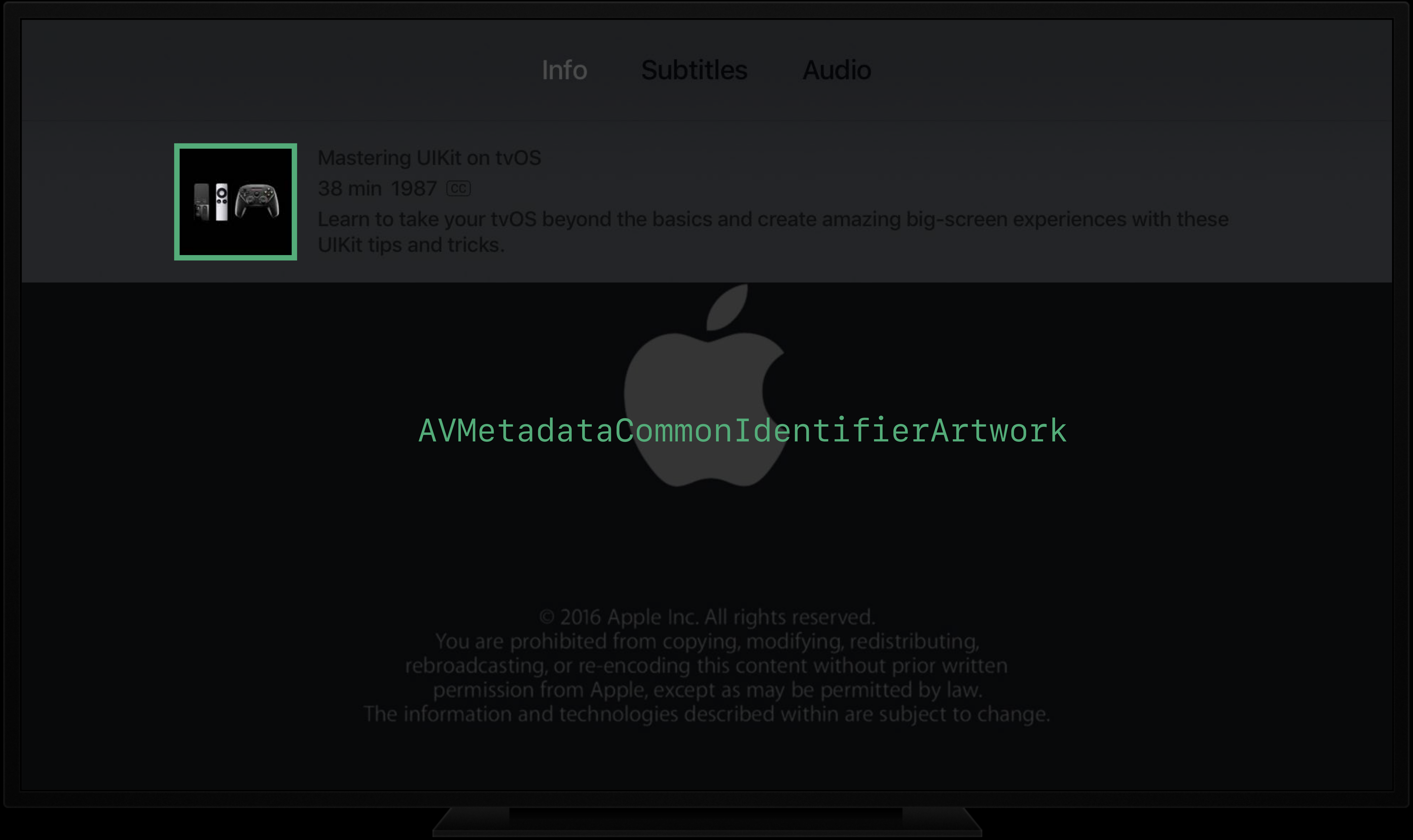
```swift
// Example of Image Metadata

let artworkPath = Bundle.main.path(forResource: "MyArtwork", ofType: "jpg")!
let artworkData = FileManager.default.contents(atPath: artworkPath)!

let artwork = AVMutableMetadataItem()
artwork.identifier = AVMetadataCommonIdentifierArtwork
artwork.value = artworkData as NSData
artwork.dataType = kCMMetadataBaseDataType_JPEG as String
artwork.extendedLanguageTag = "und"
```



Mastering UIKit on tvOS

38 min 1987 CC

Learn to take your tvOS beyond the basics and create amazing big-screen experiences with these UIKit tips and tricks.

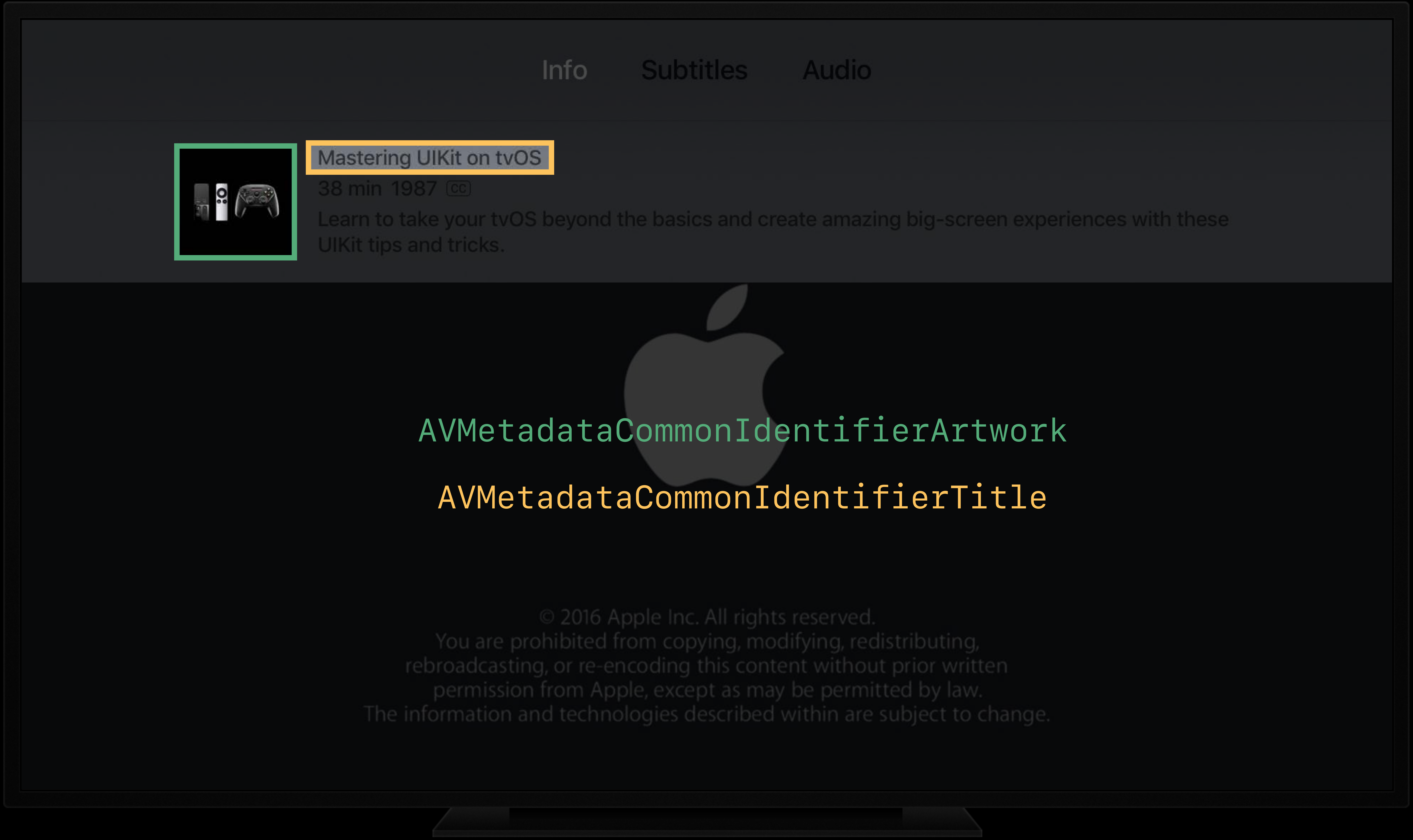


Mastering UIKit on tvOS

38 min 1987 CC

Learn to take your tvOS beyond the basics and create amazing big-screen experiences with these UIKit tips and tricks.

`AVMetadataCommonIdentifierArtwork`



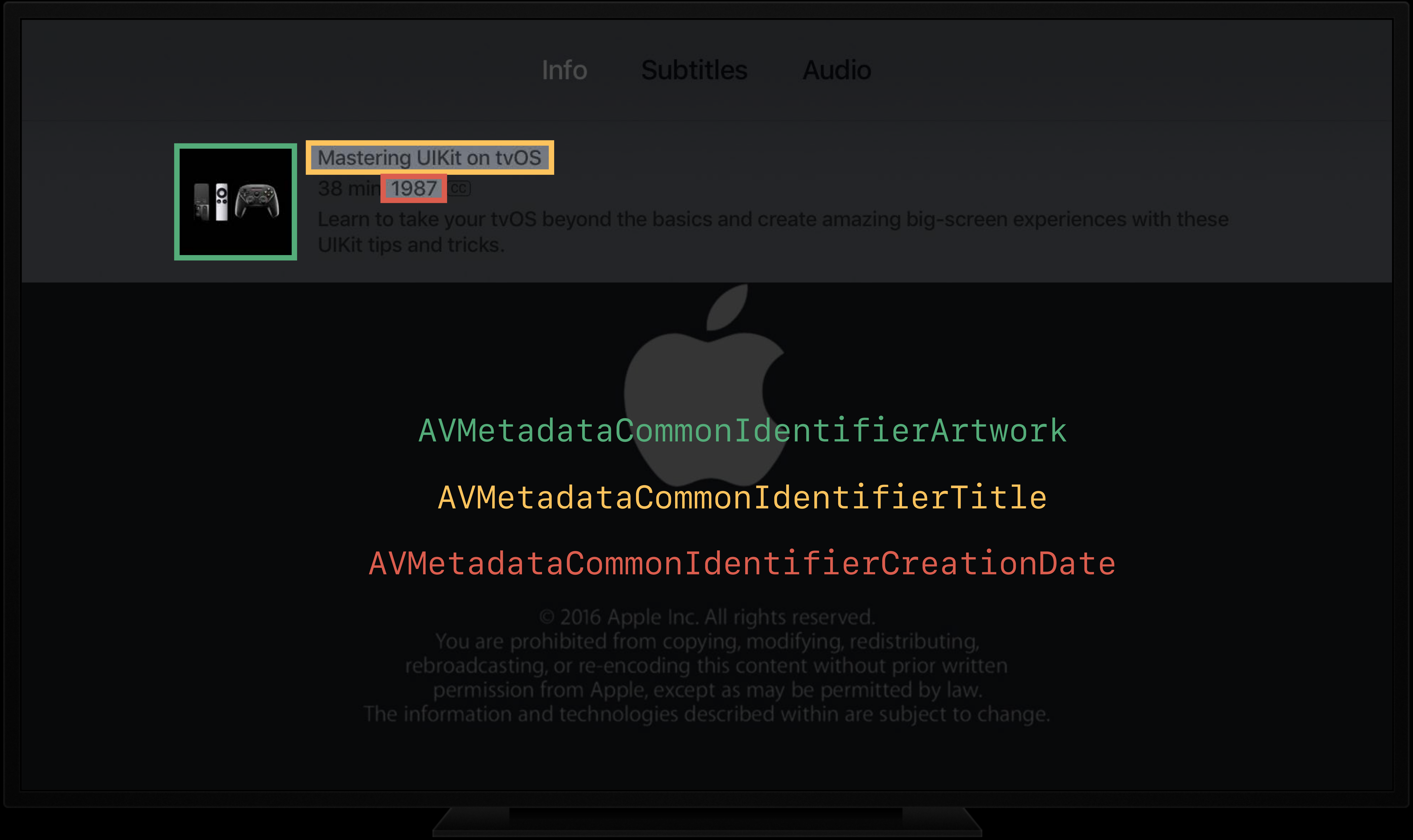
Mastering UIKit on tvOS

38 min 1987 CC

Learn to take your tvOS beyond the basics and create amazing big-screen experiences with these UIKit tips and tricks.

AVMetadataCommonIdentifierArtwork

AVMetadataCommonIdentifierTitle

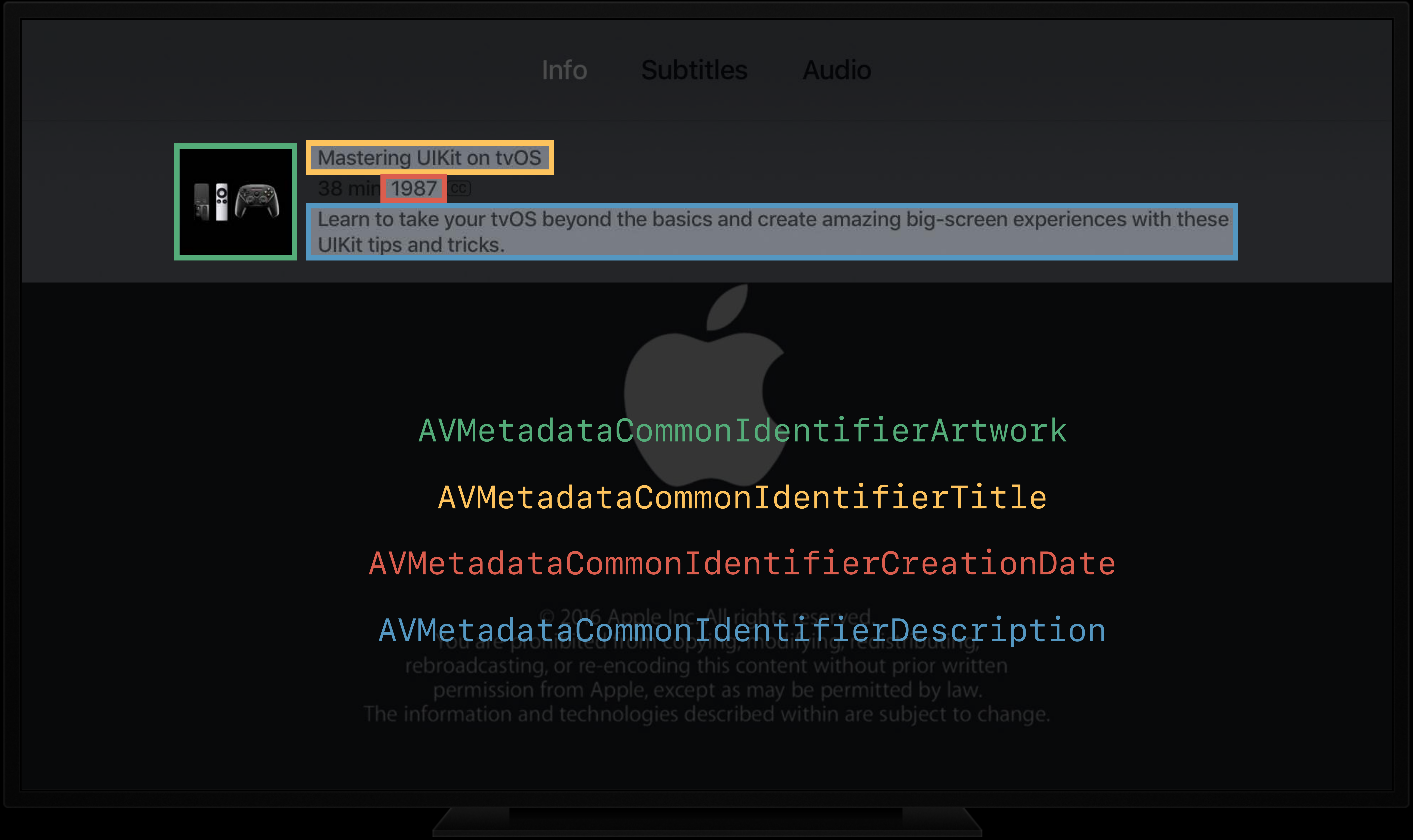


Mastering UIKit on tvOS

38 min 1987 CC

Learn to take your tvOS beyond the basics and create amazing big-screen experiences with these UIKit tips and tricks.

```
AVMetadataCommonIdentifierArtwork
AVMetadataCommonIdentifierTitle
AVMetadataCommonIdentifierCreationDate
```



Mastering UIKit on tvOS

38 min 1987 CC

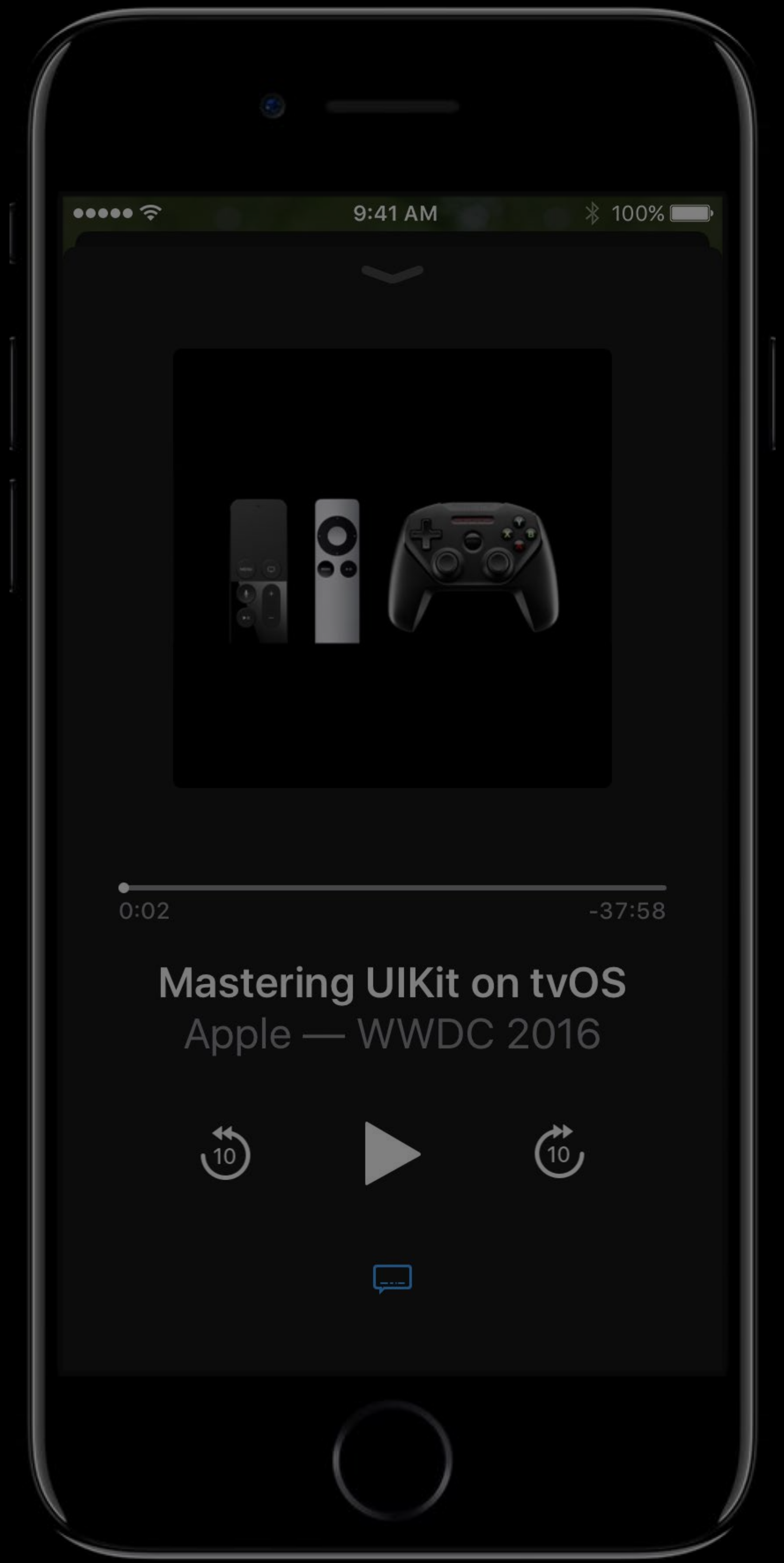
Learn to take your tvOS beyond the basics and create amazing big-screen experiences with these UIKit tips and tricks.

AVMetadataCommonIdentifierArtwork

AVMetadataCommonIdentifierTitle

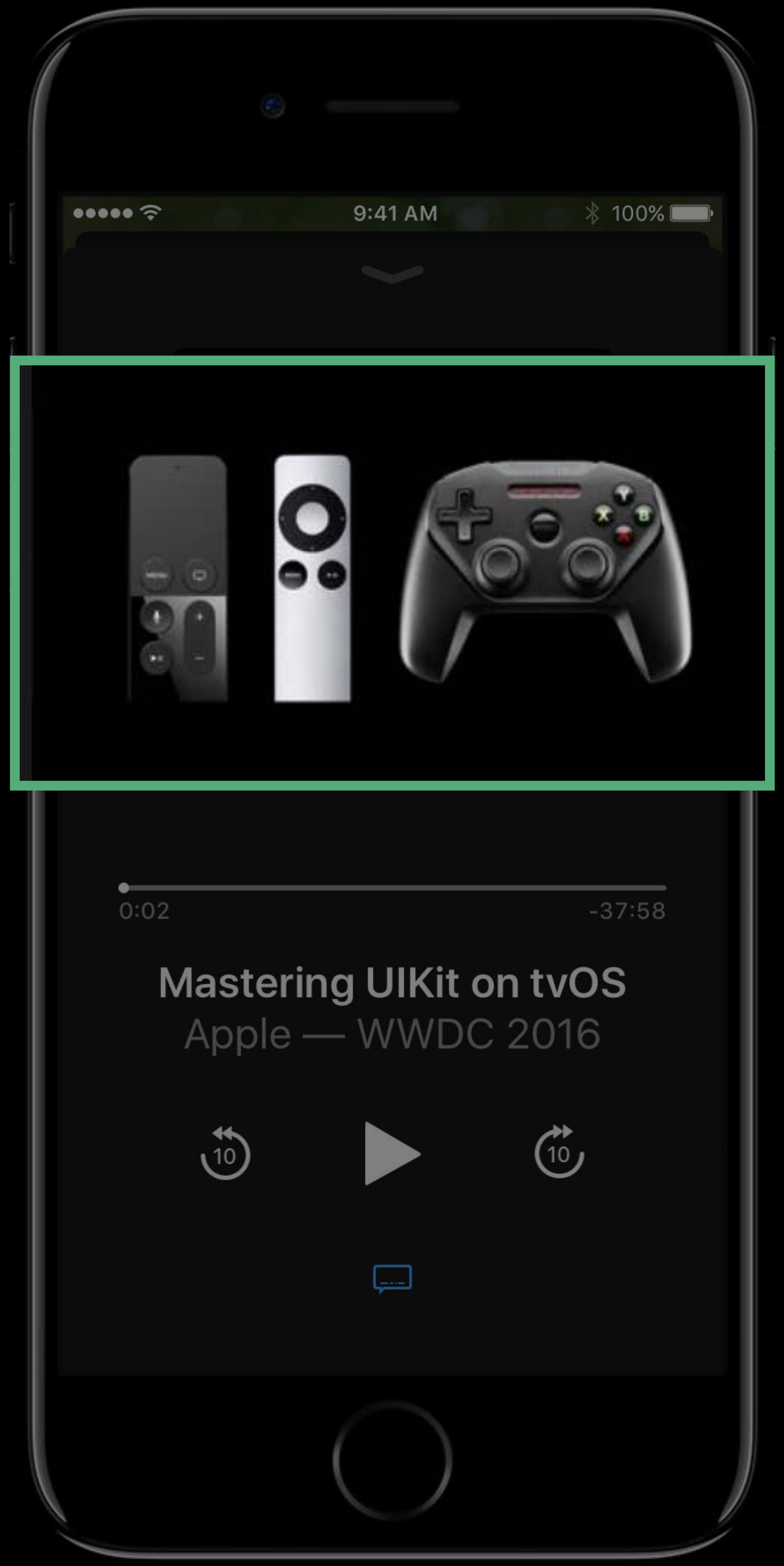
AVMetadataCommonIdentifierCreationDate

AVMetadataCommonIdentifierDescription



Mastering UIKit on tvOS

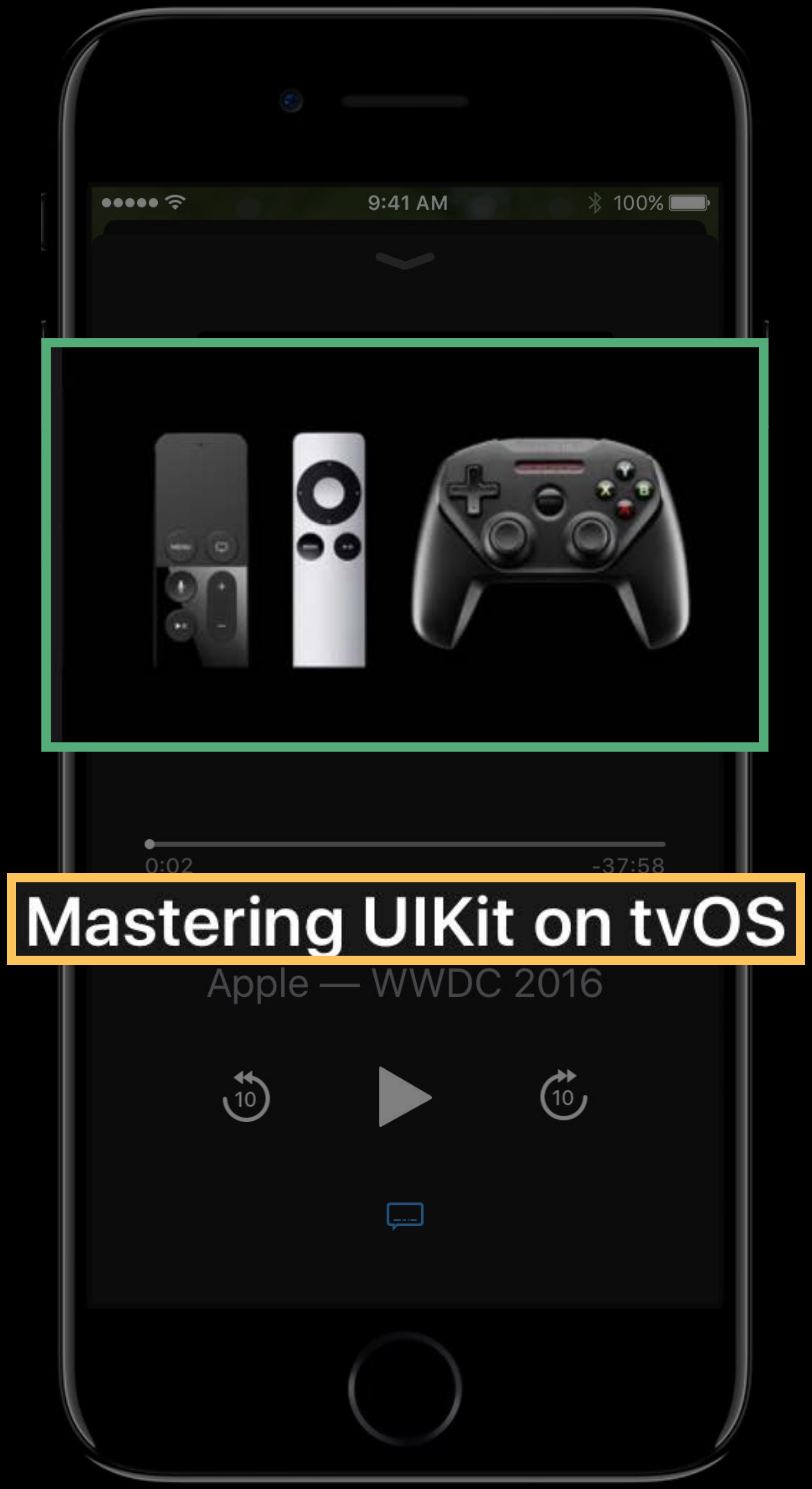
Apple — WWDC 2016



Mastering UIKit on tvOS

Apple — WWDC 2016

`AVMetadataCommonIdentifierArtwork`

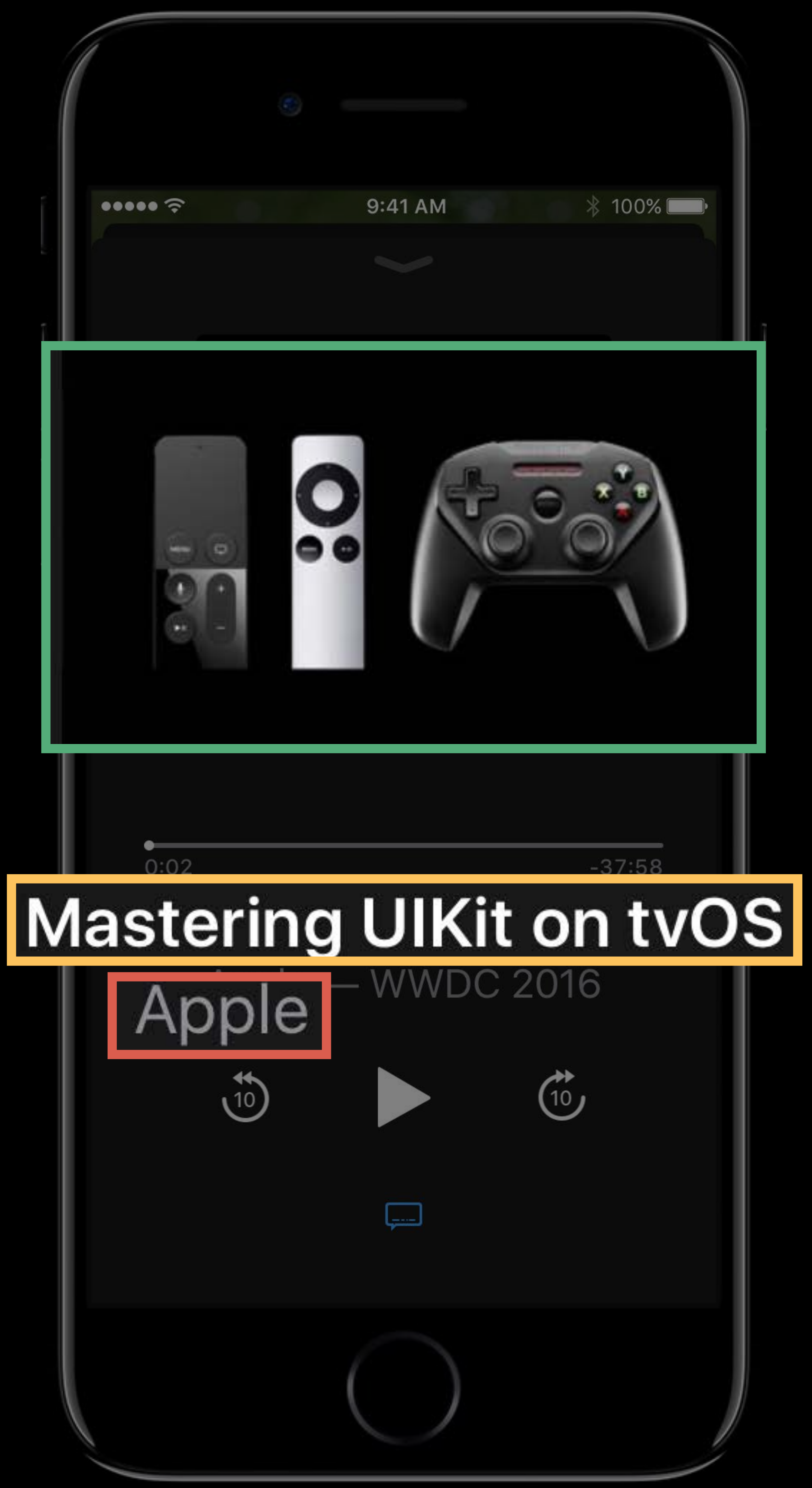


Mastering UIKit on tvOS

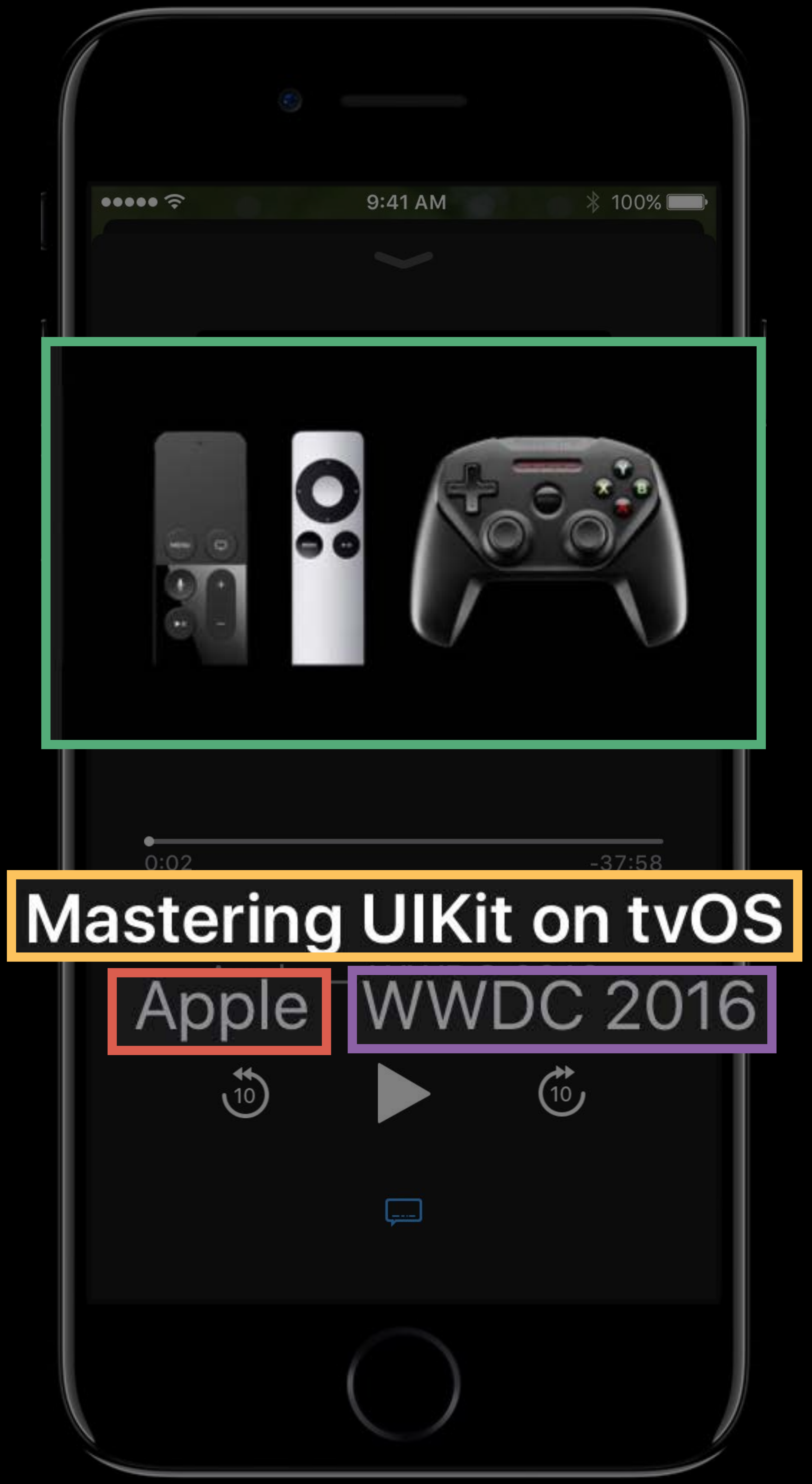
Apple — WWDC 2016

AVMetadataCommonIdentifierArtwork

AVMetadataCommonIdentifierTitle



```
AVMetadataCommonIdentifierArtwork

AVMetadataCommonIdentifierTitle

AVMetadataCommonIdentifierArtist
```



AVMetadataCommonIdentifierArtwork

AVMetadataCommonIdentifierTitle

AVMetadataCommonIdentifierArtist

AVMetadataCommonIdentifierAlbumName

MPNowPlayingInfoCenter

Accepts a dictionary of metadata items

You can specify a media type (audio vs. video)

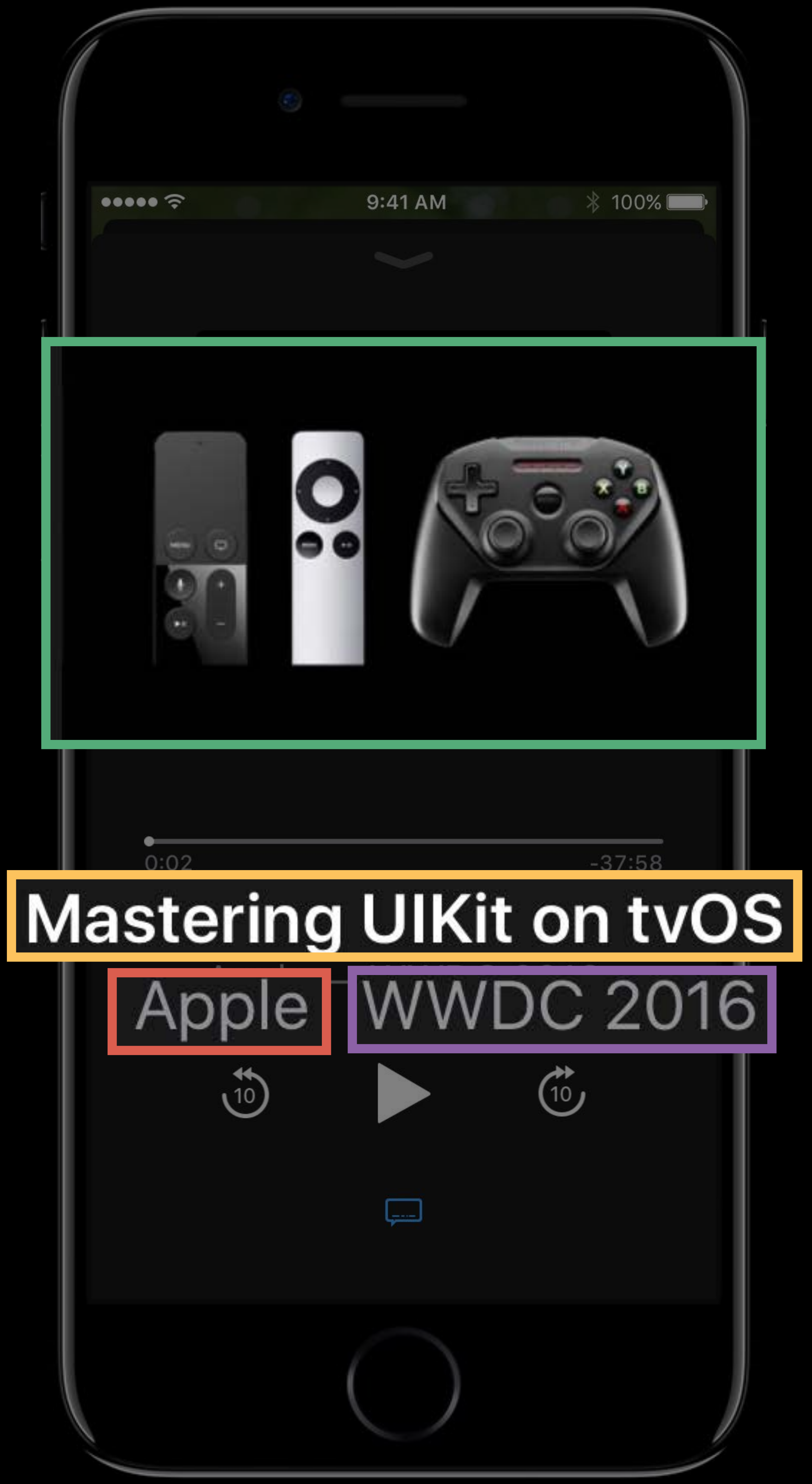
Artwork must be provided as a MPMediaItemArtwork object

Update the info after every change to playback

```
// MPNowPlayingInfoCenter Example

let artwork = MPMediaItemArtwork(boundsSize: nativeImageSize) { (size) -> UIImage in
    return findImageThatFits(size)
}

let center = MPNowPlayingInfoCenter.default()
center.nowPlayingInfo = [
    MPMediaItemPropertyTitle: "Mastering UIKit on tvOS",
    MPMediaItemPropertyArtist: "Apple",
    MPMediaItemPropertyAlbumTitle: “WWDC 2016",
    MPMediaItemPropertyMediaType: MPNowPlayingInfoMediaType.audio.rawValue,
    MPMediaItemPropertyArtwork: artwork,
    MPNowPlayingInfoPropertyElapsedPlaybackTime: 0.0,
    MPMediaItemPropertyPlaybackDuration: 3300.0,
]
```

```swift
// MPNowPlayingInfoCenter Example

let artwork = MPMediaItemArtwork(boundsSize: nativeImageSize) { (size) -> UIImage in
    return findImageThatFits(size)
}

let center = MPNowPlayingInfoCenter.default()
center.nowPlayingInfo = [
    MPMediaItemPropertyTitle: "Mastering UIKit on tvOS",
    MPMediaItemPropertyArtist: "Apple",
    MPMediaItemPropertyAlbumTitle: “WWDC 2016",
    MPMediaItemPropertyMediaType: MPNowPlayingInfoMediaType.audio.rawValue,
    MPMediaItemPropertyArtwork: artwork,
    MPNowPlayingInfoPropertyElapsedPlaybackTime: 0.0,
    MPMediaItemPropertyPlaybackDuration: 3300.0,
]
```

```swift
// MPNowPlayingInfoCenter Example

let artwork = MPMediaItemArtwork(boundsSize: nativeImageSize) { (size) -> UIImage in
    return findImageThatFits(size)
}

let center = MPNowPlayingInfoCenter.default()
center.nowPlayingInfo = [
    MPMediaItemPropertyTitle: "Mastering UIKit on tvOS",
    MPMediaItemPropertyArtist: "Apple",
    MPMediaItemPropertyAlbumTitle: “WWDC 2016",
    MPMediaItemPropertyMediaType: MPNowPlayingInfoMediaType.audio.rawValue,
    MPMediaItemPropertyArtwork: artwork,
    MPNowPlayingInfoPropertyElapsedPlaybackTime: 0.0,
    MPMediaItemPropertyPlaybackDuration: 3300.0,
]
```

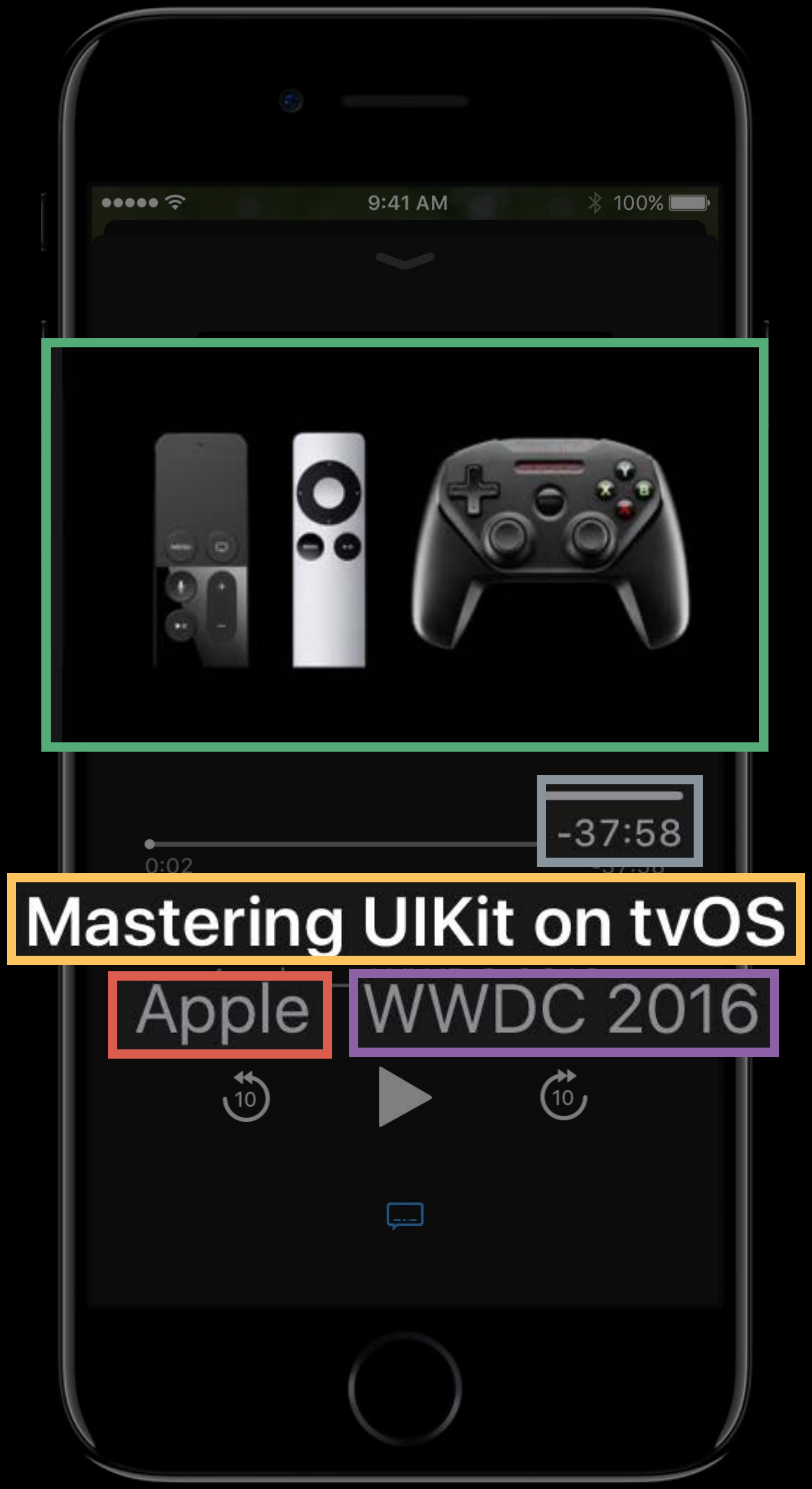


MPMediaItemPropertyArtwork

MPMediaItemPropertyTitle

MPMediaItemPropertyArtist

MPMediaItemPropertyAlbumTitle



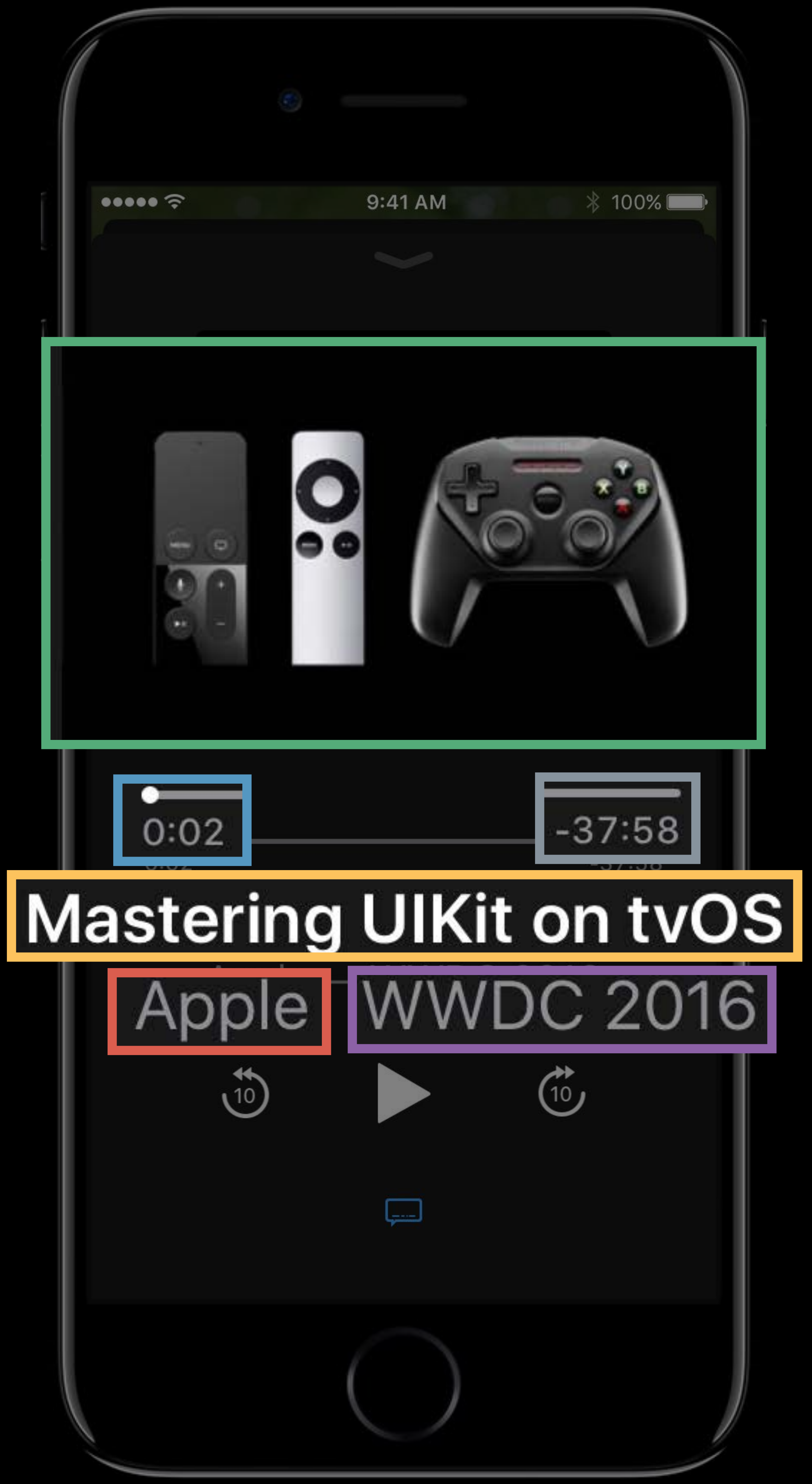
MPMediaItemPropertyArtwork

MPMediaItemPropertyTitle

MPMediaItemPropertyArtist

MPMediaItemPropertyAlbumTitle

MPMediaItemPropertyPlaybackDuration



MPMediaItemPropertyArtwork

MPMediaItemPropertyTitle

MPMediaItemPropertyArtist

MPMediaItemPropertyAlbumTitle

MPMediaItemPropertyPlaybackDuration

MPNowPlayingInfoPropertyElapsedPlaybackTime

Remote Commands

MPRemoteCommandCenter

Register target-actions or callback blocks for each command your app supports

Mark commands as disabled if you can’t support them, even temporarily

Use the return value to indicate success or failure to perform the command

```swift
let commandCenter = MPRemoteCommandCenter.shared()

commandCenter.pauseCommand.addTarget { (event) -> MPRemoteCommandHandlerStatus in
    // Pause playback here.
    return .success
}
```

Command Parameters

Some commands support extra information

- MPSkipIntervalCommand takes an array of preferred skip intervals
- MPRatingCommand takes the minimum and maximum rating values

The command handler receives a subclass of MPRemoteCommandEvent, which contains specifics about the command

- MPSkipIntervalCommandEvent.interval specifies how far to skip
- MPRatingCommandEvent.rating specifies the user’s selected rating

```
// Handling the Skip-Backward Command

commandCenter.skipBackwardCommand.preferredIntervals = [10.0]
commandCenter.skipBackwardCommand.addTarget { (baseEvent) -> MPRemoteCommandHandlerStatus in
    let skipEvent = baseEvent as! MPSkipIntervalCommandEvent

    let currentTime = player.currentTime()
    let interval = CMTime(seconds: skipEvent.interval, preferredTimescale: 1000)
    let skipTime = CMTimeSubtract(currentTime, interval)

    player.seek(to: skipTime, completionHandler: { (success) in
        self.updateNowPlaying()
    })

    return .success
}
```

```
// Handling the Skip-Backward Command

commandCenter.skipBackwardCommand.preferredIntervals = [10.0]
commandCenter.skipBackwardCommand.addTarget { (baseEvent) -> MPRemoteCommandHandlerStatus in
    let skipEvent = baseEvent as! MPSkipIntervalCommandEvent

    let currentTime = player.currentTime()
    let interval = CMTime(seconds: skipEvent.interval, preferredTimescale: 1000)
    let skipTime = CMTimeSubtract(currentTime, interval)

    player.seek(to: skipTime, completionHandler: { (success) in
        self.updateNowPlaying()
    })

    return .success
}
```

```swift
// Handling the Skip-Backward Command

commandCenter.skipBackwardCommand.preferredIntervals = [10.0]
commandCenter.skipBackwardCommand.addTarget { (baseEvent) -> MPRemoteCommandHandlerStatus in
    let skipEvent = baseEvent as! MPSkipIntervalCommandEvent

    let currentTime = player.currentTime()
    let interval = CMTime(seconds: skipEvent.interval, preferredTimescale: 1000)
    let skipTime = CMTimeSubtract(currentTime, interval)

    player.seek(to: skipTime, completionHandler: { (success) in
        self.updateNowPlaying()
    })

    return .success
}
```

```swift
// Handling the Skip-Backward Command

commandCenter.skipBackwardCommand.preferredIntervals = [10.0]
commandCenter.skipBackwardCommand.addTarget { (baseEvent) -> MPRemoteCommandHandlerStatus in
    let skipEvent = baseEvent as! MPSkipIntervalCommandEvent

    let currentTime = player.currentTime()
    let interval = CMTime(seconds: skipEvent.interval, preferredTimescale: 1000)
    let skipTime = CMTimeSubtract(currentTime, interval)

    player.seek(to: skipTime, completionHandler: { (success) in
        self.updateNowPlaying()
    })

    return .success
}
```

```
// Handling the Skip-Backward Command

commandCenter.skipBackwardCommand.preferredIntervals = [10.0]
commandCenter.skipBackwardCommand.addTarget { (baseEvent) -> MPRemoteCommandHandlerStatus in
    let skipEvent = baseEvent as! MPSkipIntervalCommandEvent

    let currentTime = player.currentTime()
    let interval = CMTime(seconds: skipEvent.interval, preferredTimescale: 1000)
    let skipTime = CMTimeSubtract(currentTime, interval)

    player.seek(to: skipTime, completionHandler: { (success) in
        self.updateNowPlaying()
    })

    return .success
}
```

```
// Updating the Info Center

func updateNowPlaying() {

    let center = MPNowPlayingInfoCenter.default()

    center.nowPlayingInfo = [
        MPMediaItemPropertyTitle: "Metadata Title",
        MPMediaItemPropertyArtist: "Metadata Artist",
        MPMediaItemPropertyAlbumTitle: "Metadata Album Title",
        MPNowPlayingInfoPropertyElapsedPlaybackTime: CMTimeGetSeconds(player.currentTime()),
        MPMediaItemPropertyPlaybackDuration: 3300.0,
        MPMediaItemPropertyMediaType: MPNowPlayingInfoMediaType.audio.rawValue,
        MPMediaItemPropertyArtwork: artwork
    ]
}
```

```
// Updating the Info Center

func updateNowPlaying() {

    let center = MPNowPlayingInfoCenter.default()

    center.nowPlayingInfo = [
        MPMediaItemPropertyTitle: "Metadata Title",
        MPMediaItemPropertyArtist: "Metadata Artist",
        MPMediaItemPropertyAlbumTitle: "Metadata Album Title",
        MPNowPlayingInfoPropertyElapsedPlaybackTime: CMTimeGetSeconds(player.currentTime()),
        MPMediaItemPropertyPlaybackDuration: 3300.0,
        MPMediaItemPropertyMediaType: MPNowPlayingInfoMediaType.audio.rawValue,
        MPMediaItemPropertyArtwork: artwork
    ]
}
```

More Information

https://developer.apple.com/wwdc17/251

WWDC17