Focus Interaction in tvOS 11

Session 224

Matt Ricketson, UIKit Engineer
Jon Staff, UIKit Engineer
New features and enhancements
Support for SceneKit
New debugging tools
Review
Review
Focus environment hierarchy

- Root Environment
  - Common Parent Environment
    - Environment
      - Item
    - Environment
      - Item
Review

Focus environment hierarchy

- Root Environment
  - Common Parent Environment
    - Environment
    - Environment
      - Item
      - Item

UIFocusItem
Review

Focus environment hierarchy

- Root Environment
- Common Parent Environment
  - Environment
  - Environment
    - Item
    - Item
Review

Focus environment hierarchy
Review
Types of focus updates

User-initiated focus movement

Programmatic focus updates
• Requested via `setNeedsFocusUpdate`
Review

Focus updates

- Root Environment
- Common Parent Environment
- Environment
- Previous Item
- Environment
- Next Item

Focus Update
Review

Focus updates

UIFocusEnvironment

func didUpdateFocus(in:with:)

Focus Update
Focus Update Notifications
Focus Updates

- Root Environment
- Common Parent Environment

Environment
- Previous Item
- Next Item

UIFocusEnvironment

func `didUpdateFocus(in:with:)`
Focus Updates

UIFocusEnvironment

func didUpdateFocus(in:with:)

Other Object

?
Focus Update Notifications

static let UIFocusDidUpdate: Notification.Name

Sent on every focus update
Sent after calling `didUpdateFocus(in:with:)` on all relevant focus environments
User info dictionary includes both the context and animation coordinator
## Focus Update Notifications

### User info keys

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIFocusUpdateContextKey</td>
<td>Provides the UIFocusUpdateContext instance that describes the update.</td>
</tr>
<tr>
<td>UIFocusUpdateAnimationCoordinatorKey</td>
<td>Provides the UIFocusAnimationCoordinator instance associated with the update.</td>
</tr>
</tbody>
</table>
Focus Update Notifications

static let UIFocusMovementDidFail: Notification.Name

Sent when the user tries, but fails, to move focus in a certain direction

Not sent for programmatic focus updates that fail

Also provides a UIFocusUpdateContext instance

Use for providing helpful feedback to the user
Protocol Extensions

UIFocusEnvironment and UIFocusItem
Protocol Extensions

UIFocusItem

Check if an item is currently focused:

```swift
let isFocused = item.isFocused
```
Protocol Extensions
Focus environment containment

Check if a focus environment contains another environment:

```java
if (environment1.contains(environment2)) {
    // do something
}
```
override func didUpdateFocus(in context: UIFocusUpdateContext, with coordinator: UIFocusAnimationCoordinator) {
    super.didUpdateFocus(in: context, with: coordinator)

    if let nextFocusedItem = context.nextFocusedItem, self.contains(nextFocusedItem) {
        // focus is inside the view controller
    }
}
Focus Animation Enhancements
Focus Animations
Review

Use the coordination API for matching system animations
Provided whenever a focus update occurs

open class UIFocusAnimationCoordinator : NSObject {

    open func addCoordinatedAnimations(_ animations: (() -> Void)?,
                                        completion: (() -> Void)? = nil)

    ...

}
Focus Animations

Review

UIKit manages focus animations

Animation duration is influenced by the speed of focus movement

• Quicker movements have quicker animations
• Slower movements have slower animations
• Animations for off-screen items are delayed
Focus Animations

Review

Focusing animations are quicker and more prominent.

Un-focusing animations are slower and subtler.
Focus Animations
Review

```
func didUpdateFocus(in: with:)
```

UIFocusEnvironment

Root Environment

Common Parent Environment

Environment

Environment

Previous Item

Next Item

Focus Moves
Focus Animations

func addCoordinatedAnimations(_:completion:)

Un-focusing Animations

Review
Focus Animations

Review

func addCoordinatedAnimations(_:completion:)

Focusing Animations
Targeted Coordinated Animations

Review
open class UIFocusAnimationCoordinator : NSObject {

    open func addCoordinatedAnimations(_ animations: (() -> Void)?,
        completion: (() -> Void)? = nil)

    open func addCoordinatedFocusingAnimations(_ animations: ((UIFocusAnimationContext) -> Void)?,
        completion: (() -> Void)? = nil)

    open func addCoordinatedUnfocusingAnimations(_ animations: ((UIFocusAnimationContext) -> Void)?,
        completion: (() -> Void)? = nil)

}
**Focus Animations**

New API

```swift
func addCoordinatedFocusingAnimations(_:completion:)
```

Focusing Animations

```swift
func addCoordinatedUnfocusingAnimations(_:completion:)
```

Un-focusing Animations
Demo

Targeted coordinated animations
Targeted Coordinated Animations
New API

Timing information provided by `UIFocusAnimationContext`

```swift
public protocol UIFocusAnimationContext : NSObjectProtocol {

    /// The duration of the main animations in seconds.
    public var duration: TimeInterval { get }
}
```
override func didUpdateFocus(in context: UIFocusUpdateContext, with coordinator: UIFocusAnimationCoordinator) {

    coordinator.addCoordinatedFocusingAnimations({ animationContext in
        let duration =.animationContext.duration / 2.0

        UIView.animate(withDuration: duration,
                        delay: duration, // start at the halfway point
                        options: [.overrideInheritedDuration],
                        animations: {
                            // some animation
                        })
    })
}
override func didUpdateFocus(in context: UIFocusUpdateContext,
    with coordinator: UIFocusAnimationCoordinator) {

    coordinator.addCoordinatedFocusingAnimations({ animationContext in
        let duration = animationContext.duration / 2.0

        UIView.animate(withDuration: duration,
            delay: duration, // start at the halfway point
            options: [.overrideInheritedDuration],
            animations: {
                // some animation
            })
    })
}
override func didUpdateFocus(in context: UIFocusUpdateContext, 
  with coordinator: UIFocusAnimationCoordinator) {

  coordinator.addCoordinatedFocusingAnimations({ animationContext in
    let duration = animationContext.duration / 2.0

    UIView.animate(withDuration: duration, 
      delay: duration, // start at the halfway point
      options: [.overrideInheritedDuration],
      animations: {
        // some animation
      })
  })
}
override func didUpdateFocus(in context: UIFocusUpdateContext, with coordinator: UIFocusAnimationCoordinator) {

    coordinator.addCoordinatedFocusingAnimations({ animationContext in
        let duration = animationContext.duration / 2.0

        UIView.animate(withDuration: duration,
                        delay: duration, // start at the halfway point
                        options: [.overrideInheritedDuration],
                        animations: {
                            // some animation
                        })
    })
}
UIKitFocusAnimationContext

Better non-UIKit support

Consistent feel for focus interaction across frameworks

```swift
override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {
    let node = myNode
    coordinator.addCoordinatedFocusingAnimations({ animationContext in
        let fadeAction = SKAction.fadeAlpha(to: 0.5, duration: animationContext.duration)
        node.run(fadeAction)
    })
}
```
Summary

Use the existing API when targeting isn’t needed

Use the new coordination APIs for targeting animations

Use the animation context for timing information
Custom Focus Sounds
Focus Sounds

Volume adjustments based on speed
- Faster movements have quieter sounds
- Slower movements have louder sounds

Panning adjustments based on location
- Items on the right side of the screen will pan right
- Items on the left side of the screen will pan left
Custom Focus Sounds

New API

Play custom sounds when focus movement occurs

Opt out of UIKit-played sounds
Custom Focus Sounds

New API

Register a sound with an identifier

Provide a sound identifier during a focus update

Sound matching the identifier is played
Class method on `UIFocusSystem`

```swift
open class UIFocusSystem: NSObject {

    /// Registers a sound file for a given identifier.
    open class func register(_ soundFileURL: URL,
                              forSoundIdentifier identifier: UIFocusSoundIdentifier)
}
```
Custom Focus Sounds

New API

Global registration

Register only once per sound identifier

Register early

Accepts standard iOS sound file formats
Custom Focus Sounds
New API

Optional method on `UIFocusEnvironment`

```swift
public protocol UIFocusEnvironment : NSObjectProtocol {
    ...

    optional public func soundIdentifierForFocusUpdate(in context: UIFocusUpdateContext) -> UIFocusSoundIdentifier?

    ...
}
```
Custom Focus Sounds

New API

Root Environment

Common Parent Environment

Environment

Previous Item

Environment

Next Item

UIFocusEnvironment

func soundIdentifierForFocusUpdate(in:)

Focus Update
Custom Focus Sounds
New API

```swift
func soundIdentifierForFocusUpdate(in: UIFocusEnvironment)
```
Custom Focus Sounds
New API

```
func soundIdentifierForFocusUpdate(in: UIFocusEnvironment)
```
Custom Focus Sounds

New API

```
func soundIdentifierForFocusUpdate(in: UIFocusEnvironment)
```
Custom Focus Sounds
New API

- Root Environment
- Common Parent Environment
- Environment
  - Previous Item
- Environment
  - Next Item

func soundIdentifierForFocusUpdate(in:)

UIFocusEnvironment
# Custom Focus Sounds

New API

<table>
<thead>
<tr>
<th>Sound Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Sound Identifier</td>
<td>Plays the sound registered with this sound identifier.</td>
</tr>
<tr>
<td>(e.g. <code>myCustomIdentifier</code>)</td>
<td></td>
</tr>
<tr>
<td><code>.default</code></td>
<td>Explicitly force the default UIKit sound to play.</td>
</tr>
<tr>
<td><code>.none</code></td>
<td>Opt out of UIKit-played sounds for this focus update.</td>
</tr>
</tbody>
</table>
override func soundIdentifierForFocusUpdate(in context: UIFocusUpdateContext) -> UIFocusSoundIdentifier? {
    if let nextFocusedItem = context.nextFocusedItem, 
    soundEnabledEnvironment.contains(nextFocusedItem) {
        return myCustomSoundIdentifier
    }
    return nil
}
Custom Focus Sounds
Opting out of sounds

Users expect focus movements to play sounds

Only opt out of focus sounds if your users expect it
• i.e. user-toggled sound settings in a game

Respect user device settings for navigation sounds
Custom Focus Sounds
Best practices

Custom sounds should enhance the user experience

Differentiate between different elements
• i.e. Size or element type
Support for SceneKit

Matt Ricketson, UIKit Frameworks Engineer
Focus Item Types
tvOS 9

UIView

UIFocusEnvironment
Focus Item Types

tvOS 10
Focus Item Types

tvOS 11

UIFocusEnvironment

UIFocusItem

UIView
SKNode
SCNNode
Support for SceneKit
Making nodes focusable

Opt-in to focus support for `SCNNode`

```swift
node.focusBehavior = .focusable
```

Also available on `SKNode`
Demo
Focus support in SceneKit
Support for SceneKit

Summary

Focus movement is two-dimensional, based on screen visibility

Focusable SpriteKit scenes on SceneKit surfaces are allowed

Supports moving focus between UIKit, SpriteKit, and SceneKit

All `UIKitFocusEnvironment` APIs available
New Debugging Tools
Focus Update Logging
Focus Update Logging

Automatically logs focus updates while an application is running

Outlines preferred focus logic

Not for use in production apps (impacts performance)

<table>
<thead>
<tr>
<th>Info</th>
<th>Arguments</th>
<th>Options</th>
<th>Diagnostics</th>
</tr>
</thead>
</table>

Arguments Passed On Launch

- -UIFocusLoggingEnabled YES
UIFocusDebugger

Used for diagnosing focus-related issues

Structured as a command line tool built into UIKit

Use from LLDB while app is paused, using `po`

Not for use in application logic
(lldb) po UIFocusDebugger.status()

Provides the currently focused item
Simulates focus update requests from a specific environment

• i.e. a successful call to `environment.setNeedsFocusUpdate()`

Outlines preferred focus logic used to determine new focus item
UIFocusDebugger
Checking focusability

(lldb) po UIFocusDebugger.checkFocusability(for: item)

Diagnoses potential issues affecting item focusability

Works on SKNode and SCNNode in addition to UIView

Replaces _whyIsThisViewNotFocusable
<table>
<thead>
<tr>
<th>Issue</th>
<th>Ancestor Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIView is hidden</td>
<td>Ancestor UIView is hidden</td>
</tr>
<tr>
<td>UIView alpha is zero</td>
<td>Ancestor UIView alpha is zero</td>
</tr>
<tr>
<td>UIView user interaction is disabled</td>
<td>Ancestor UIView user interaction is disabled</td>
</tr>
<tr>
<td>UIView is not in the a window</td>
<td>Ancestor UIView is not in a window</td>
</tr>
<tr>
<td>UIView returns false from canBecomeFocused</td>
<td>Ancestor UIView returns true from canBecomeFocused</td>
</tr>
<tr>
<td>UIControl is disabled</td>
<td>SpriteKit node not focusable</td>
</tr>
<tr>
<td>UITextView is not selectable</td>
<td>SpriteKit node user interaction disabled</td>
</tr>
<tr>
<td>UISegmentedControl has no enabled segments</td>
<td>SceneKit node not focusable</td>
</tr>
<tr>
<td>Focus items occluding other focus items</td>
<td></td>
</tr>
</tbody>
</table>
(lldb) po UIFocusDebugger.help()

Prints detailed instructions for all debugging commands
Demo
Focus Debugging
Summary

New focus update notifications, `isFocused`, and `contains()`

Provide consistent animations by targeting coordinated animations

Provide custom focus sounds for more immersive experiences

Use focus interaction in SceneKit for more consistent user experiences

Use `UIFocusDebugger` and focus update logging to diagnose focus issues
More Information

https://developer.apple.com/wwdc17/224
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date and Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advances in TVMLKit</td>
<td></td>
<td>WWDC 2017</td>
</tr>
<tr>
<td>What's New in tvOS</td>
<td></td>
<td>WWDC 2017</td>
</tr>
<tr>
<td>SceneKit: What's New</td>
<td></td>
<td>WWDC 2017</td>
</tr>
<tr>
<td><strong>Going Beyond 2D with SpriteKit</strong></td>
<td>Executive Ballroom</td>
<td>Friday 10:00AM</td>
</tr>
</tbody>
</table>