Agenda

Digital Crown
Agenda

Digital Crown

Media Playback
Agenda

Digital Crown
Media Playback
Audio Recording
Agenda

Digital Crown
Media Playback
Audio Recording
Security
Digital Crown

Nathan de Vries watchOS Engineer
Core Interactions
Core Interactions
Core Interactions
Core Interactions
Digital Crown—Why?

Intuitive
Digital Crown—Why?

Intuitive

Keep your UI visible
Digital Crown–Why?

Intuitive
Keep your UI visible
Precise
Digital Crown—Why?

Intuitive

Keep your UI visible

Precise

Playful
WKInterfacePicker
List style
WKInterfacePicker
List style
WKInterfacePicker
Stack style
WKInterfacePicker
Stack style
WKInterfacePicker
Sequence style
WKInterfacePicker
Sequence style
Focus Styles
Focus Styles
Focus Styles
Focus Styles
Contextual Indicator
Contextual Indicator
Coordinating Images
Coordinating Images
Coordinating Images
Coordinating Images
Creating a Picker
Creating a Picker

- **Switch** - A control for indicating a binary value.
- **Slider** - A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker** - A control for selecting an item from a list.
Creating a Picker
Creating a Picker
Creating a Picker
Creating a Picker

class InterfaceController: WKInterfaceController {
    @IBOutlet weak var picker : WKInterfacePicker!

    override func awakeWithContext(context: AnyObject?) {
        super.awakeWithContext(context)
    }
}
Creating a Picker
Creating a Picker
Configuring Items
Configuring Items
Configuring Items

picker.setItems([
    WKPickerItem(
        title: ITEM 1,
        caption: CAPTION 1,
        accessoryImage: ⬇️
    ),
    WKPickerItem(
        title: ITEM 2,
        caption: CAPTION 2,
        accessoryImage: ⬆️
    ),
    WKPickerItem(
        title: ITEM 3,
        caption: CAPTION 3,
        accessoryImage: ⬇️
    ),
    WKPickerItem(
        title: ITEM 4,
        caption: CAPTION 3,
        accessoryImage: ⬈️
    )
])
Configuring Items

```swift
WKPickerItem
caption: CAPTION 1
contentImage: □
caption: CAPTION 2
contentImage: □
caption: CAPTION 3
contentImage: □
caption: CAPTION 4
contentImage: □
picker.setItems([...])
```
Specifying Item Images

// Image bundled with the app
let itemOne = WKPickerItem()
itemOne.contentImage = WKImage(imageName: "picker-item-content")

// Image data loaded from a file or network
let itemTwo = WKPickerItem()
itemTwo.contentImage = WKImage(imageData: contentImageData)

// UIImage drawn in code
let itemThree = WKPickerItem()
itemThree.contentImage = WKImage(image: contentImage)
Specifying Item Images

// Image bundled with the app
let itemOne = WKPickerItem()
itemOne.contentImage = WKImage(imageName: "picker-item-content")

// Image data loaded from a file or network
let itemTwo = WKPickerItem()
itemTwo.contentImage = WKImage(imageData: contentImageData)

// UIImage drawn in code
let itemThree = WKPickerItem()
itemThree.contentImage = WKImage(image: contentImage)
Specifying Item Images

// Image bundled with the app
let itemOne = WKPickerItem()
itemOne.contentImage = WKImage(imageName: "picker-item-content")

// Image data loaded from a file or network
let itemTwo = WKPickerItem()
itemTwo.contentImage = WKImage(imageData: contentImageData)

// UIImage drawn in code
let itemThree = WKPickerItem()
itemThree.contentImage = WKImage(image: contentImage)
// Image bundled with the app
let itemOne = WKPickerItem()
itemOne.contentImage = WKImage(imageName: "picker-item-content")

// Image data loaded from a file or network
let itemTwo = WKPickerItem()
itemTwo.contentImage = WKImage(imageData: contentImageData)

// UIImage drawn in code
let itemThree = WKPickerItem()
itemThree.contentImage = WKImage(image: contentImage)
Handling Selected Item Changes

```swift
var pickerItems : [WKPickerItem]!

...

@IBAction func pickerAction(selectedIndex: Int) {
    var item = pickerItems[selectedIndex]
    print("The selected item is \(item.title) with index \(selectedIndex).")
}
```
Coordinating Images

70%
Coordinating Images

Group

Picker
let progressImages = UIImage.animatedImageWithImages([/* images */], duration: 0.0)

progressInterfaceGroup.setBackgroundImage(progressImages)

picker.setCoordinatedAnimations([progressInterfaceGroup, ...])
let progressImages = UIImage.animatedImageWithImages([...], duration: 0.0)

progressInterfaceGroup.setBackgroundImage(progressImages)

picker.setCoordinatedAnimations([ progressInterfaceGroup, ... ])
let progressImages = UIImage(animatedImageWithImages(
    // Images here
), duration: 0.0)

progressInterfaceGroup.setBackgroundImage(progressImages)

picker.setCoordinatedAnimations([ progressInterfaceGroup, ... ])

Coordinating Images

WKInterfacePicker

selected index

WKInterfaceGroup or WKInterfaceImage

Image

Item

Image

Item

Image

Item

Image

Item
Coordinating Images

WKInterfacePicker

WKInterfaceGroup or WKInterfaceImage

selected index
Demo
Digital Crown

Recap

Three customizable styles
Focus and indicator support
Image animation coordination
Media Playback

Chloe Chang watchOS Engineer
Media Playback
Audiovisual
Media Playback
Audiovisual
Media Playback
Audio-only
if let url = NSBundle.mainBundle().URLForResource("sample", withExtension: "mov") {

    let options: [NSObject: AnyObject] = [
        WKMediaPlayerControllerOptionsAutoplayKey: true,
        WKMediaPlayerControllerOptionsStartTimeKey: 3.0,
        WKMediaPlayerControllerOptionsVideoGravityKey: WKVideoGravity.ResizeAspect.rawValue
    ]

    presentMediaPlayerControllerWithURL(url, options: options) {
        didPlayToEnd, endTime, error in
        /* check playback result */
    }
}
Media Playback

```swift
if let url = NSBundle.mainBundle().URLForResource("sample", withExtension: "mov") {

    let options: [NSObject: AnyObject] = [
        WKMediaPlayerControllerOptionsAutoplayKey: true,
        WKMediaPlayerControllerOptionsStartTimeKey: 3.0,
        WKMediaPlayerControllerOptionsVideoGravityKey: 
            WKVideoGravity.ResizeAspect.rawValue
    ]

    presentMediaPlayerControllerWithURL(url, options: options) {
        didPlayToEnd, endTime, error in
        /* check playback result */
    }
}
```
if let url = NSBundle.mainBundle().URLForResource("sample", withExtension: "mov") {

    let options: [NSObject: AnyObject] = [
        WKMediaPlayerControllerOptionsAutoplayKey: true,
        WKMediaPlayerControllerOptionsStartTimeKey: 3.0,
        WKMediaPlayerControllerOptionsVideoGravityKey: WKVideoGravity.ResizeAspect.rawValue
    ]

    presentMediaPlayerControllerWithURL(url, options: options) {
        didPlayToEnd, endTime, error in
        /* check playback result */
    }
}
Media Playback

if let url = NSBundle.mainBundle().URLForResource("sample", withExtension: "mov") {

    let options: [NSObject: AnyObject] = [
        WKMediaPlayerControllerOptionsAutoplayKey: true,
        WKMediaPlayerControllerOptionsStartTimeKey: 3.0,
        WKMediaPlayerControllerOptionsVideoGravityKey: WKVideoGravity.ResizeAspect.rawValue
    ]

    presentMediaPlayerControllerWithURL(url, options: options) {
        didSetPlayToEnd, endTime, error in
        /* check playback result */
    }
}
Media Playback
WKInterfaceMovie

Snow fun in Lake Tahoe last weekend!
Movie - Displays a play button and poster image for audiovisual content.
Movie

- Video Gravity: Resize Aspect
- Poster Image: No Image

Any Screen Size
var url: NSURL
var poster: WKImage
@IBOutlet weak var movie: WKInterfaceMovie!

func setupMovie() {
    movie.setMovieURL(url)
    movie.setVideoGravity(.ResizeAspectFill)
    movie.setPosterImage(poster)
}
var url: NSURL
var poster: WKImage
@IBOutlet weak var movie: WKInterfaceMovie!

func setupMovie() {
    movie.setMovieURL(url)
    movie.setVideoGravity(.ResizeAspectFill)
    movie.setPosterImage(poster)
}
var url: NSURL
var poster: WKImage
@IBOutlet weak var movie: WKInterfaceMovie!

func setupMovie() {
    movie.setMovieURL(url)
    movie.setVideoGravity(.ResizeAspectFill)
    movie.setPosterImage(poster)
}
# Media Playback

## Recommended formats

<table>
<thead>
<tr>
<th>Video</th>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264 High Profile</td>
<td>Bit Rate: 32 kbps</td>
</tr>
<tr>
<td>Bit Rate: 160 kbps</td>
<td>AAC Stereo</td>
</tr>
<tr>
<td>Frame Rate: 30 fps</td>
<td></td>
</tr>
<tr>
<td>Full screen: 208 x 260</td>
<td></td>
</tr>
<tr>
<td>16:9 aspect ratio: 320 x 180</td>
<td></td>
</tr>
</tbody>
</table>
Media Playback

Long-form audio
Media Playback

Long-form audio

Play in the background
Media Playback

Long-form audio

Play in the background
Podcasts, music
Media Playback

Long-form audio

Play in the background
Podcasts, music
Routed to Bluetooth headphones
Media Playback
Long-form audio

Play in the background
Podcasts, music
Routed to Bluetooth headphones
Now Playing glance integration
let asset = WKAudioFileAsset.assetWithURL(URL: url)
let playerItem = WKAudioFilePlayerItem.playerItemWithAsset(asset: asset)

let player = WKAudioFilePlayer.playerWithPlayerItem(playerItem: playerItem)
player.play()
//player.pause()

let queuePlayer = WKAudioFileQueuePlayer.queuePlayerWithItems(items: [...])
queuePlayer.play()
//queuePlayer.pause()
let asset = WKAudioFileAsset.assetWithURL(URL: url)
let playerItem = WKAudioFilePlayerItem.playerItemWithAsset(asset: asset)

let player = WKAudioFilePlayer.playerWithPlayerItem(playerItem: playerItem)
player.play()
//player.pause()

let queuePlayer = WKAudioFileQueuePlayer.queuePlayerWithItems(items: [...])
queuePlayer.play()
//queuePlayer.pause()
let asset = WKAudioFileAsset.assetWithURL(URL: url)
let playerItem = WKAudioFilePlayerItem.playerItemWithAsset(asset: asset)

let player = WKAudioFilePlayer.playerWithPlayerItem(playerItem: playerItem)
player.play()

//player.pause()

let queuePlayer = WKAudioFileQueuePlayer.queuePlayerWithItems(items: [...])
queuePlayer.play()
//queuePlayer.pause()
let asset = WKAudioFileAsset.assetWithURL(URL: url)
let playerItem = WKAudioFilePlayerItem.playerItemWithAsset(asset: asset)

let player = WKAudioFilePlayer.playerWithPlayerItem(playerItem: playerItem)
player.play()
//player.pause()

let queuePlayer = WKAudioFileQueuePlayer.queuePlayerWithItems(items: [...])
queuePlayer.play()
//queuePlayer.pause()
Media Playback
Long-form audio

Enable background mode in the app's Info.plist.
Media Playback

Long-form audio

Enable background mode in the app's Info.plist
Media Playback

Long-form audio

Enable background mode in the app's Info.plist

- Required background modes:
  - Item 0: App plays audio or streams audio/video using AirPlay
Media Playback

Resource locations

Watch App

Watch Extension
Media Playback

Resource locations

Watch App

Watch Extension

App's Container

Extension's Container
Media Playback

Resource locations

- Watch App
- Watch Extension
- App's Container
- Extension's Container
Media Playback

Resource locations

Watch App

Watch Extension

App's Container

Extension's Container
Media Playback

Resource locations

Watch App

Watch Extension

App’s Container

Extension’s Container

VIDEO

AUDIO

IMAGE
Media Playback

Resource locations

Watch App

Watch Extension

App’s Container

Extension’s Container
Media Playback

Resource locations

- Watch App
- Watch Extension
- App's Container
- Extension's Container
Media Playback

Resource locations

Watch App

Watch Extension

App's Container

Shared Container for App Group

Extension's Container
Media Playback

App groups

Sharing Data with Your Containing App
App Extension Programming Guide
Audio Recording
Audio Recording
Audio Recording
Audio Recording

```swift
presentAudioRecordingControllerWithURL(url, 
    preset: .NarrowBandSpeech, 
    maximumDuration: 60.0, 
    actionTitle:"Send") { didSave, error in 
    /* check recording result */
}
```
presentAudioRecordingControllerWithURL(url, preset: .NarrowBandSpeech, maximumDuration: 60.0, actionTitle: "Send") { didSave, error in
    /* check recording result */
}
Audio Recording

Supported file types

AAC Codec: .mp4, .m4a
LPCM Codec: .wav
## Audio Recording

### Presets

<table>
<thead>
<tr>
<th>Presets</th>
<th>Sample Rate</th>
<th>Bit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAC</td>
<td>24 kbps</td>
<td>128 kbps</td>
</tr>
<tr>
<td>Narrow band speech</td>
<td>8 kHz</td>
<td>24 kbps</td>
</tr>
<tr>
<td>Wide band speech</td>
<td>16 kHz</td>
<td>32 kbps</td>
</tr>
<tr>
<td>High quality audio</td>
<td>44.1 kHz</td>
<td>96 kbps</td>
</tr>
<tr>
<td>LPCM</td>
<td>16 kHz</td>
<td>256 kbps</td>
</tr>
<tr>
<td></td>
<td>44.1 kHz</td>
<td>705.6 kbps</td>
</tr>
</tbody>
</table>
Security
Security

Keychain
Security

Keychain

Locks/unlocks with Apple Watch
Security

Keychain

Locks/unlocks with Apple Watch

No iCloud Keychain
let secret = "YouCanNeverGuess"
if let secretData = secret.dataUsingEncoding(NSUnicodeStringEncoding) {
    let attributes: [NSString: NSObject] = [
        kSecClass: kSecClassGenericPassword,
        kSecAttrAccessible: kSecAttrAccessibleWhenUnlocked,
        kSecAttrService: "myservice",
        kSecAttrAccount: "account name",
        kSecValueData: secretData
    ]
    SecItemAdd(attributes, nil)
}
Summary

Digital Crown
Media playback
Audio recording
Security
More Information

Documentation
watchOS 2 Transition Guide
WatchKit Programming Guide
http://developer.apple.com/watchOS

Sample Code
Lister
WatchKitMoviePlayer
WatchKit Catalog
http://developer.apple.com/watchOS

Technical Support
Apple Developer Forums
Developer Technical Support
http://developer.apple.com/forums

General Inquiries
Jake Behrens, watchOS Frameworks Evangelist
behrens@apple.com
<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing WatchKit for watchOS 2</td>
<td>Presidio</td>
<td>Tuesday 10:00AM</td>
</tr>
<tr>
<td>Privacy and Your App</td>
<td>Pacific Heights</td>
<td>Tuesday 2:30PM</td>
</tr>
<tr>
<td>Building Watch Apps</td>
<td>Pacific Heights</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Security and Your Apps</td>
<td>Mission</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>WatchKit In-Depth, Part 1</td>
<td>Pacific Heights</td>
<td>Wednesday 9:00AM</td>
</tr>
<tr>
<td>Introducing Watch Connectivity</td>
<td>Pacific Heights</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Designing for Apple Watch</td>
<td>Presidio</td>
<td>Wednesday 4:30PM</td>
</tr>
</tbody>
</table>
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout and Animation Techniques for WatchKit</td>
<td>Pacific Heights</td>
<td>Thursday 10:00AM</td>
</tr>
<tr>
<td>Networking with NSURLSession</td>
<td>Pacific Heights</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>WatchKit Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>Apple Watch Design Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 3:30PM</td>
</tr>
<tr>
<td>Lab</td>
<td>Location</td>
<td>Time</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>AVKit and AV Foundation Lab</td>
<td>Graphics, Games, and Media Lab A</td>
<td>Wednesday 1:30PM</td>
</tr>
<tr>
<td>WatchKit Lab</td>
<td>Frameworks Lab D</td>
<td>Wednesday 3:30PM</td>
</tr>
<tr>
<td>AVKit and AV Foundation Lab</td>
<td>Graphics, Games, and Media Lab B</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Watch Connectivity Lab</td>
<td>Frameworks Lab B</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td>WatchKit and ClockKit Complications Lab</td>
<td>Frameworks Lab A</td>
<td>Friday 1:30PM</td>
</tr>
<tr>
<td>AVKit and AV Foundation Lab</td>
<td>Graphics, Games, and Media Lab B</td>
<td>Friday 1:30PM</td>
</tr>
</tbody>
</table>