Multitasking Essentials for Media-Based Apps on iPad in iOS 9
Adopting Picture in Picture and Mastering Shared Resources
Session 211

Stefan Hafeneger Picture in Picture Architect
Jonathan Bennett Media Systems Product Lead
## Multitasking Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Multitasking Essentials for Media-Based Apps on iPad in iOS 9</td>
<td>Pacific Heights</td>
<td>Wednesday 2:30PM</td>
</tr>
<tr>
<td>Optimizing Your App for Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Wednesday 3:30PM</td>
</tr>
</tbody>
</table>
## Multitasking Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Multitasking Essentials for Media-Based Apps on iPad in iOS 9</td>
<td>Pacific Heights</td>
<td>Wednesday 2:30PM</td>
</tr>
<tr>
<td>Optimizing Your App for Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Wednesday 3:30PM</td>
</tr>
</tbody>
</table>
What You Will Learn

Adopt Picture in Picture in video playback applications
Master shared resources in multitasking environment
Picture in Picture
Picture in Picture = PiP
Picture in Picture

Supported hardware

iPad Air 2
iPad Air
iPad mini 3
iPad mini 2
Should my app adopt Picture in Picture?
Video Playback APIs
iOS 8

MediaPlayer
AVKit
AVFoundation
WebKit
Video Playback APIs
iOS 8

MediaPlayer
Video Playback APIs

Deprecated APIs in iOS 9

MPMoviePlayerController
MPMoviePlayerViewController
Video Playback APIs

Deprecated APIs in iOS 9

MPMoviePlayerController

MPMoviePlayerViewController

Replaced by AVPlayerViewController
  • Introduced in AVKit in iOS 8
Video Playback APIs

Deprecated APIs in iOS 9

MPMoviePlayerController
MPMoviePlayerViewController

Replaced by **AVPlayerViewController**

- Introduced in AVKit in iOS 8

Mastering Modern Media Playback
Video Playback APIs
iOS 9

AVKit
AVFoundation
WebKit
This video is playing in Picture in Picture.
Picture in Picture

Supported video playback APIs

AVKit
AVFoundation
WebKit
Picture in Picture

Supported video playback APIs

- AVKit
- AVFoundation
- WebKit
Supported video playback APIs

AVKit

AVFoundation

WebKit
AVKit
AVFoundation
WebKit
IMG_0353

Creator: John Appleseed
Camera: iPhone 6
Duration: 00:01:00
Dimensions: 1920 x 1080
Format: H.264, AAC
FPS: 30
Data Size: 80.8 MB
Data Rate: 10.78 Mbit/s
Location: Photo of people skateboarding on a road with a clear blue sky in the background.
Demo

Adopting Picture in Picture with AVPlayerViewController

Felix Heidrich AVKit Engineer
AVPlayerViewController

How to support Picture in Picture
AVPlayerViewController

How to support Picture in Picture
AVPlayerViewController
How to support Picture in Picture

```swift
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    // Get shared audio session.
    let audioSession = AVAudioSession.sharedInstance()

    do {
        // Set playback audio session category.
        try audioSession.setCategory(AVAudioSessionCategoryPlayback)
    } catch { ... }

    return true
}
```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]? ) -> Bool {
    // Get shared audio session.
    let audioSession = AVAudioSession.sharedInstance()

do {
    // Set playback audio session category.
    try audioSession.setCategory(AVAudioSessionCategoryPlayback)
    } catch { ... }

    return true
}
AVPlayerViewController

How to support Picture in Picture

```swift
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    // Get shared audio session.
    let audioSession = AVAudioSession.sharedInstance()
    do {
        // Set playback audio session category.
        try audioSession.setCategory(AVAudioSessionCategoryPlayback)
    } catch { ... }
    return true
}
```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    // Get shared audio session.
    let audioSession = AVAudioSession.sharedInstance()

    do {
        // Set playback audio session category.
        try audioSession.setCategory(AVAudioSessionCategoryPlayAndRecord)
    } catch {
        ...
    }

    return true
}
class AVPlayerViewController : UIViewController {
    // Whether or not to allow Picture in Picture playback. Default is YES.
    var allowsPictureInPicturePlayback: Bool
}
AVPlayerViewController
Persistent Video Overlay

Supported in primary application
When application enters background if
• `AVPlayerViewController` is presented full screen
• Video content is playing
• Picture in Picture is possible
AVPlayerViewController
Persistent Video Overlay

User can turn on/off behavior in Settings

• General > Multitasking > Persistent Video Overlay
AVPlayerViewController

Restore user interface before stop

```swift
func playerViewController(playerViewController: AVPlayerViewController,
restoreUserInterfaceForPictureInPictureStopWithCompletionHandler
completionHandler: (Bool) -> Void) {
    // Present player view controller again.
    navigationController?.presentViewController(playerViewController, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
```
func playerViewController(playerViewController: AVPlayerViewController, restoreUserInterfaceForPictureInPictureStopWithCompletionHandler completionHandler: (Bool) -> Void) {
    // Present player view controller again.
    navigationController?.presentViewController(playerViewController, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
func playerViewController(playerViewController: AVPlayerViewController, restoreUserInterfaceForPictureInPictureStopWithCompletionHandler completionHandler: (Bool) -> Void) {
    // Present player view controller again.
    navigationController?.presentViewController(playerViewController, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
func playerViewController(playerViewController: AVPlayerViewController,
    restoreUserInterfaceForPictureInPictureStopWithCompletionHandler
    completionHandler: (Bool) -> Void) {
    // Present player view controller again.
    navigationController?.presentViewController(playerViewController, 
        animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
This video is playing in Picture in Picture.
AVPictureInPictureController

New AVKit class in iOS 9

AVKit
How to set up

// Check whether Picture in Picture is supported on device.
if AVPictureInPictureController.isPictureInPictureSupported() {
    // Create Picture in Picture controller.
    pipController = AVPictureInPictureController(playerLayer: playerLayer)!

    // Set delegate.
    pipController.delegate = self
}
AVPictureInPictureController
How to set up

// Check whether Picture in Picture is supported on device.
if AVPictureInPictureController.isPictureInPictureSupported() {
    // Create Picture in Picture controller.
    pipController = AVPictureInPictureController(playerLayer: playerLayer)!

    // Set delegate.
    pipController.delegate = self
}
AVPictureInPictureController

How to set up

// Check whether Picture in Picture is supported on device.
if AVPictureInPictureController.isPictureInPictureSupported() {
    // Create Picture in Picture controller.
    pipController = AVPictureInPictureController(playerLayer: playerLayer)!

    // Set delegate.
    pipController.delegate = self
}
AVPictureInPictureController
How to set up

// Check whether Picture in Picture is supported on device.
if AVPictureInPictureController.isPictureInPictureSupported() {
    // Create Picture in Picture controller.
    pipController = AVPictureInPictureController(playerLayer: playerLayer)!

    // Set delegate.
    pipController.delegate = self
}
AVPictureInPictureController
Enable/disable Picture in Picture button

// Find out whether Picture in Picture is possible.
let pipPossible = pipController.pictureInPicturePossible

// Enable/disable Picture in Picture button.
pipButton.enabled = pipPossible
AVPictureInPictureController
Enable/disable Picture in Picture button

// Find out whether Picture in Picture is possible.
let pipPossible = pipController.pictureInPicturePossible

// Enable/disable Picture in Picture button.
pipButton.enabled = pipPossible
AVPictureInPictureController
Enable/disable Picture in Picture button

// Find out whether Picture in Picture is possible.
let pipPossible = pipController.pictureInPicturePossible

// Enable/disable Picture in Picture button.
pipButton.enabled = pipPossible
AVPictureInPictureController
How to start Picture in Picture

func pipButtonTapped(sender: AnyObject?) {
    // Make sure Picture in Picture is not already active.
    if !pipController.pictureInPictureActive {
        // Start Picture in Picture on button tap.
        pipController.startPictureInPicture()
    }
}
AVPictureInPictureController

How to start Picture in Picture

```swift
func pipButtonTapped(sender: AnyObject?) {
    // Make sure Picture in Picture is not already active.
    if !pipController.pictureInPictureActive {
        // Start Picture in Picture on button tap.
        pipController.startPictureInPicture()
    }
}
```
func pipButtonTapped(sender: AnyObject?) {
    // Make sure Picture in Picture is not already active.
    if !pipController.pictureInPictureActive {
        // Start Picture in Picture on button tap.
        pipController.startPictureInPicture()
    }
}
AVPictureInPictureController

How to start Picture in Picture

```swift
func pipButtonTapped(sender: AnyObject?) {
    // Make sure Picture in Picture is not already active.
    if !pipController.pictureInPictureActive {
        // Start Picture in Picture on button tap.
        pipController.startPictureInPicture()
    }
}
```

App Store review team will reject your app if used inappropriately
func pictureInPictureControllerDidStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Dismiss modal video playback view controller.
    dismissViewControllerAnimated(true, completion: nil)
}
func pictureInPictureControllerDidStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Dismiss modal video playback view controller.
    dismissViewControllerAnimated(true, completion: nil)
}
func pictureInPictureControllerDidStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Dismiss modal video playback view controller.
    dismissViewControllerAnimated(true, completion: nil)
}
AVPictureInPictureController

While Picture in Picture is active

pipController
AVPictureInPictureController

While Picture in Picture is active
func pictureInPictureController(pipController: AVPictureInPictureController, restoreUserInterfaceForPictureInPictureStopWithCompletionHandler completionHandler: (Bool) -> Void) {
    // Present video playback view controller again.
    navigationController?.presentViewController(self, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
func `pictureInPictureController` (pipController: AVPictureInPictureController, restoreUserInterfaceForPictureInPictureStopWithCompletionHandler completionHandler: (Bool) -> Void) {
    // Present video playback view controller again.
    navigationController?.presentViewController(self, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
func pictureInPictureController(pipController: AVPictureInPictureController, 
restoreUserInterfaceForPictureInPictureStopWithCompletionHandler 
completionHandler: (Bool) -> Void) {
    // Present video playback view controller again.
    navigationController?.presentViewController(self, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
func pictureInPictureController(pipController: AVPictureInPictureController, restoreUserInterfaceForPictureInPictureStopWithCompletionHandler completionHandler: (Bool) -> Void) {
    // Present video playback view controller again.
    navigationController?.presentViewController(self, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}
func pictureInPictureController(pipController: AVPictureInPictureController, 
restoreUserInterfaceForPictureInPictureStopWithCompletionHandler 
completionHandler: (Bool) -> Void) {
    // Present video playback view controller again.
    navigationController?.presentViewController(self, animated: true) {
        // Don’t forget to call completion handler.
        completionHandler(true)
    }
}

Need to reuse AVPlayerLayer passed to AVPictureInPictureController
This video is playing in Picture in Picture.
AVPictureInPictureController

Hide playback controls and show placeholder artwork

```swift
func pictureInPictureControllerWillStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide playback controls.
    hidePlaybackControls() 

    // Show placeholder artwork.
    showPlaceholderArtwork()
}
```
AVPictureInPictureController
Hide playback controls and show placeholder artwork

```swift
func pictureInPictureControllerWillStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide playback controls.
    hidePlaybackControls()

    // Show placeholder artwork.
    showPlaceholderArtwork()
}
```
func pictureInPictureControllerWillStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide playback controls.
    hidePlaybackControls()

    // Show placeholder artwork.
    showPlaceholderArtwork()
}
func pictureInPictureControllerWillStartPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide playback controls.
    hidePlaybackControls()

    // Show placeholder artwork.
    showPlaceholderArtwork()
}
func pictureInPictureControllerDidStopPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide placeholder artwork.
    hidePlaceholderArtwork()

    // Show playback controls.
    showPlaybackControls()
}
func pictureInPictureControllerDidLoadPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide placeholder artwork.
    hidePlaceholderArtwork()

    // Show playback controls.
    showPlaybackControls()
}

AVPictureInPictureController
Hide playback controls and show placeholder artwork
func pictureInPictureControllerDidStopPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide placeholder artwork.
    hidePlaceholderArtwork()

    // Show playback controls.
    showPlaybackControls()
}
func pictureInPictureControllerDidStopPictureInPicture(pipController: AVPictureInPictureController) {
    // Hide placeholder artwork.
    hidePlaceholderArtwork()

    // Show playback controls.
    showPlaybackControls()
}
AVPictureInPictureController
Persistent Video Overlay

UIWindow
AVPictureInPictureController
Persistent Video Overlay
WKWebView

HTML5 video Picture in Picture support

Application has background audio mode

AVAudioSession configured with

- AVAudioSessionCategoryPlayback
- AVAudioSessionCategoryPlayAndRecord

Web content does not render custom controls
WKWebView

HTML5 video Picture in Picture support

Application has background audio mode

AVAudioSession configured with

- AVAudioSessionCategoryPlayback
- AVAudioSessionCategoryPlayAndRecord

Web content does not render custom controls
WKWebView

Disable HTML5 video Picture in Picture support

class WKWebViewConfiguration : NSObject {
    // Whether or not to allow Picture in Picture playback. Default is YES.
    var allowsPictureInPictureMediaPlayback: Bool
}
Picture in Picture
Is a lot like background audio

Same rules as background audio and AirPlay apply
App Store review team will reject app if in violation
Shared Resources

Jonathan Bennett Media Systems Product Lead
Shared Resources

Audio

Video

Camera
Multitasking for iPad
Primary and secondary apps

Fullscreen
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture
Multitasking for iPad
Primary and secondary apps

Fullscreen
Slide Over
Split View
Picture in Picture

Background Media
Shared Resources

Audio

Video

Camera
Audio

Best practices

Properly configure AVAudioSession
Audio

Best practices

Properly configure AVAudioSession

• One configuration to rule them all...
Audio

Best practices

Properly configure `AVAudioSession`

• *One configuration to rule them all…*

Only activate session when audio is first needed
Audio

Best practices

Properly configure **AVAudioSession**

- *One configuration to rule them all…*

Only activate session when audio is first needed

Games or sound effects?

- Use **Ambient** category
Audio
Best practices

Properly configure `AVAudioSession`

- *One configuration to rule them all…*

Only activate session when audio is first needed

Games or sound effects?

- Use `Ambient` category

Secondary audio tracks?

- Use `secondaryAudioShouldBeSilencedHint` (iOS 8)
Audio

Best practices

Properly configure AVAudioSession

• One configuration to rule them all…

Only activate session when audio is first needed

Games or sound effects?

• Use Ambient category

Secondary audio tracks?

• Use secondaryAudioShouldBeSilencedHint (iOS 8)

What’s New in Core Audio

Audio Session Programming Guide

WWDC 2014

http://developer.apple.com/iOS
Shared Resources

Audio

Video

Camera
Shared Resources

Audio
Video
Camera
Video

Video has audio?

- Properly configure `AVAudioSession`
Video

Video has audio?

• Properly configure AVAudioSession

Background media

• Discard unnecessary data
Video

HTTP Live Streaming

Multiple stream variants

• Include a 1-3 Mbps variant

Annotate variants with RESOLUTION in master playlist
Cape Town Vacation

Family vacation research for our trip!

Research to-do's:
Cape Town Vacation

Family vacation research for our trip!

Research to-do's:
Shared Resources

Audio

Video

Camera
Shared Resources

Audio

Video

Camera
Camera

Only one app can use the camera
Camera

Only one app can use the camera

• Availability can change at any time
Consider `UIRequiresFullscreen = YES`
Family Vacations

Cape Town Vacation

Family vacation research for our trip!

Research to-do's:
- Weekend in Muizenberg
- Cable car to Table Mountain!
- Day trip to Cape Winelands?
- Xhosa restaurants
Cape Town Vacation

Family vacation research for our trip!

Research to-do's:
- Weekend in Muizenberg
- Cable car to Table Mountain!
- Day trip to Cape Winelands?
- Xhosa restaurants
Cape Town Vacation

Family vacation research for our trip!

Research to-do’s:
- Weekend in Muizenberg
- Cable car to Table Mountain!
- Day trip to Cape Winelands?
- Xhosa restaurants
Camera API

UIImagePickerController
AVCaptureSession
UINavigationController

Single App

<table>
<thead>
<tr>
<th>Feature</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Preview</td>
<td>✔️</td>
</tr>
<tr>
<td>Photo Capture</td>
<td>✔️</td>
</tr>
<tr>
<td>Video Capture</td>
<td>✔️</td>
</tr>
</tbody>
</table>
## UIApplicationPickerController

<table>
<thead>
<tr>
<th></th>
<th>Single App</th>
<th>Multiple Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Camera Preview</strong></td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td><strong>Photo Capture</strong></td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td><strong>Video Capture</strong></td>
<td>![Checkmark]</td>
<td>![X]</td>
</tr>
</tbody>
</table>
## UIImagePickerController

<table>
<thead>
<tr>
<th></th>
<th>Single App</th>
<th>Multiple Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Camera Preview</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Photo Capture</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Video Capture</strong></td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>
YourApp must be full screen to record video.
UIImagePickerController

Active captures may be interrupted
Cape Town Vacation

Family vacation research for our trip!

Research to-do's:
- Weekend in Muizenberg
- Cable car to Table Mountain!
- Day trip to Cape Winelands?
- Xhosa restaurants
UIImagePickerController

Active captures may be interrupted
UIImagePickerController

Active captures may be interrupted

`startVideoCapture()` may return `false`
Camera API

UIImagePickerController
AVCaptureSession
AVCaptureSession

<table>
<thead>
<tr>
<th>Single App</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Preview</td>
</tr>
<tr>
<td>Photo Capture</td>
</tr>
<tr>
<td>Video Capture</td>
</tr>
</tbody>
</table>
## AVCaptureSession

<table>
<thead>
<tr>
<th></th>
<th>Single App</th>
<th>Multiple Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Preview</td>
<td>![Checkmark]</td>
<td>![X]</td>
</tr>
<tr>
<td>Photo Capture</td>
<td>![Checkmark]</td>
<td>![X]</td>
</tr>
<tr>
<td>Video Capture</td>
<td>![Checkmark]</td>
<td>![X]</td>
</tr>
</tbody>
</table>
AVCaptureSession

Interruptions

Observe session interruptions

- AVCaptureSessionWasInterruptedNotification
AVCaptureSession

Interruptions

Observe session interruptions

- AVCaptureSessionWasInterruptedNotification

Check the reason

- AVCaptureSessionInterruptionReasonKey
  - .VideoDeviceNotAvailableWithMultipleForegroundApps
AVCaptureSession

Interruptions

Observe session interruptions

- AVCaptureSessionWasInterruptedNotification

Check the reason

- AVCaptureSessionInterruptionReasonKey
  - VideoDeviceNotAvailableWithMultipleForegroundApps

Adjust UI
AVCaptureSession

Interruptions

Completed Interruptions

• AVCaptureSession automatically resumes
AVCaptureSession

Interruptions

Completed Interruptions

- AVCaptureSession automatically resumes

Observe

- AVCaptureSession InterruptionEndedNotification
AVCaptureSession

Interruptions

Completed Interruptions

• AVCaptureSession **automatically** resumes

Observe

• AVCaptureSession **InterruptionEndedNotification**

Restore camera UI
AVCaptureSession
Sample code

AVCam
Summary

**Adopt Picture in Picture**
- AVKit, AVFoundation, WebKit
- Follow best practices for consistent experience

**Master Shared Resources**
- Configure audio session
- Provide video streaming variants
- Handle camera interruptions
More Information

Documentation
Adopting Multitasking Enhancements on iPad
What’s New in iOS
Start Developing iOS Apps
iOS App Programming Guide

http://developer.apple.com/iOS

Technical Support
Apple Developer Forums
Developer Technical Support

General Inquiries
Curt Rothert, App Frameworks Evangelist
rothert@apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Optimizing Your App for Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Wednesday 3:30PM</td>
</tr>
<tr>
<td>Performance on iOS and watchOS</td>
<td>Presidio</td>
<td>Friday 11:00AM</td>
</tr>
</tbody>
</table>
# Related Labs

<table>
<thead>
<tr>
<th></th>
<th>Location</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVKit and AVFoundation</td>
<td>Media Lab</td>
<td>Wednesday, Thursday, and Friday</td>
</tr>
<tr>
<td>Camera Capture</td>
<td>Media Lab</td>
<td>Wednesday and Thursday</td>
</tr>
<tr>
<td>HTTP Live Streaming</td>
<td>Media Lab</td>
<td>Thursday</td>
</tr>
<tr>
<td>Audio</td>
<td>Media Lab</td>
<td>Thursday</td>
</tr>
<tr>
<td>Cocoa Touch and Multitasking</td>
<td>Frameworks Lab</td>
<td>Wednesday</td>
</tr>
<tr>
<td>Cocoa Touch</td>
<td>Frameworks Lab</td>
<td>Friday</td>
</tr>
</tbody>
</table>
Apple WWDC 15