Secure

- Short-lived
- Out-of-process
- Sandboxed
- On demand
- Remote views
App Store
iMessage App Store
iMessage App Store
7:00 PM
Get Open Table >
No coding required
Use familiar image editing tools
Build with Xcode
Submit to iTunes Connect
UIKit
Messages extension point
Build with Xcode
Submit to iTunes Connect
Privacy
Marcos

Wanna go see Finding Dory?

Delivered

Let's See Finding Dory

Add comment or Send

FANDANGO

TAP THE MOVIE YOU WANT TO SEE
SDK Simulator
Siri
Instant
Launch on latest hardware
Speech → Intent → Action → Response

Vocabulary

App Logic

User Interface
Speech

Intent

Vocabulary

App vocabulary: plist
User vocabulary: code
Intent - Action - Response

App Logic

Extension
Resolve parameters
Confirm intent
Handle intent
Response

User Interface

Extension
Custom user interface
UIKit
Example: Hologram
“Send a Hologram to Obi-Wan saying you’re my only hope”
Send a Hologram to Obi-Wan saying you’re my only hope
“Send a Hologram to Obi-Wan saying you’re my only hope”
“Send a Hologram to Obi-Wan saying you’re my only hope”

Domain: Messages
Intent: sendMessage
“Send a Hologram to Obi-Wan saying you’re my only hope”

Domain: Messages

Intent: sendMessage
“Send a Hologram to Obi-Wan saying you’re my only hope”

Domain: Messages
Intent: sendMessage
“Send a Hologram to Obi-Wan saying you’re my only hope”

Domain: Messages
Intent: sendMessage
“Send a Hologram to **Obi-Wan** saying you’re my only hope”

**Domain:** Messages

**Intent:** sendMessage

**Recipient:** Obi-Wan

**Content:** You’re my only hope
Intent
Domain: Messages
Recipient: Obi-Wan
Content: You’re my only hope
Domain: Messages
Intent: sendMessage
Recipient: Obi-Wan → Old Ben Kenobi
Content: You’re my only hope
Domain: Messages
Intent: sendMessage
Recipient: Old Ben Kenobi
Content: You’re my only hope
Domain: Messages
Intent: sendMessage
Recipient: Old Ben Kenobi
Content: You’re my only hope
Domain: Messages
Intent: sendMessage
Recipient: Old Ben Kenobi
Content: You’re my only hope
Response

Send a Hologram to Obi-Wan saying you’re my only hope.

Here’s your Hologram message:

To: Old Ben Kenobi

You’re my only hope
Swift
About Swift

Swift is a general purpose programming language built using a modern approach to safety, performance, and software design patterns.

The goal of the Swift project is to create the best available language for uses ranging from systems programming, to mobile and desktop apps, scaling up to cloud services. Most importantly, Swift is designed to make writing and maintaining correct programs easier for the developer. To achieve this goal, we believe that the most obvious way to write Swift code must also be:

Safe. The most obvious way to write code should also behave in a safe manner. Undefined behavior is the enemy of safety, and developer mistakes should be caught before software is in production. Opting for safety sometimes means Swift will feel strict, but we believe that clarity saves time in the long run.

Fast. Swift is intended as a replacement for C-based languages (C, C++, and Objective-C) for such, Swift must be comparable to those languages in performance for most tasks. Performance must also be predictable and consistent, not just fast in short bursts that require a clean-up later. There are lots of languages with novel features — being fast is rare.
Language project downloads on GitHub
Watched language project on GitHub

#1
#1

Favorited language project on GitHub
>350

Contributors
>3500

Pull Requests
Open evolution
Open swift-evolution
Open roadmap
Evolution proposals

100
Move Fast and Fix Things
Move Fast and Fix Things
Swift 3
content = text.stringByTrimmingCharactersInSet(
    NSCharacterSet.newlineCharacterSet())
content = text. trimmingCharacters(newlines)
content = text.trimmingCharacters( .newlines)
content = text.trimmingCharacters(in: .newlines)
Fundamentals

- Clarity at the point of use is your most important goal. Entities such as methods and properties are declared only once but used repeatedly. Design APIs to make those uses clear and concise. When evaluating a design, reading a declaration is seldom sufficient; always examine a use case to make sure it looks clear in context.

- Clarity is more important than brevity. Although Swift code can be compact, it is a non-goal to enable the smallest possible code with the fewest characters. Brevity in Swift code, where it occurs, is a side-effect of the strong type system and features that naturally reduce boilerplate.

- Write a documentation comment for every declaration. Insights gained by writing documentation can have a profound impact on your design, so don’t put it off.
class NSDate {
    func addingTimeInterval(_ : NSTimeInterval) -> NSDate
}

@interface NSDate
-(NSDate*)dateByAddingTimeInterval:(NSTimeInterval)ti;
@end
@interface NSDate
-(NSDate*)dateByAddingTimeInterval:(NSTimeInterval)ti;
@end

class NSDate {
    func addingTimeInterval(_ : NSTimeInterval) -> NSDate
}

swiftc
Swift 2

let myDate = NSDate()

Swift 3

var myDate = Date()
Swift 2

let myDate = NSDate()
let myDate2 = myDate.dateByAddingTimeInterval(60)

Swift 3

var myDate = Date()
myDate.addTimeInterval(60)
Swift 2

```swift
let myDate = NSDate()
let myDate2 = myDate.dateByAddingTimeInterval(60)
```

Swift 3

```swift
var myDate = Date()
myDate.addTimeInterval(60)
```
Swift 2

let cal = NSCalendar(calendarIdentifier: NSCalendarIdentifierChinese)

Swift 3

let cal = Calendar(identifier: .chinese)
Swift 2

```swift
let cal = NSCalendar(calendarIdentifier: NSCalendarIdentifierChinese)
let components = NSDateComponents()
components.day = 1
```

Swift 3

```swift
let cal = Calendar(identifier: .chinese)
let components = DateComponents(day: 1)
```
Swift 2

```swift
let cal = NSCalendar(calendarIdentifier: NSCalendarIdentifierChinese)
let components = NSDateComponents()
components.day = 1
tomorrow = cal?.dateByAddingComponents(components, toDate: myDate, options: [])
```

Swift 3

```swift
let cal = Calendar(identifier: .chinese)
let components = DateComponents(day: 1)
tomorrow = cal?.date(byAdding: components, to: myDate)
```
let queue = dispatch_queue_create("com.xyzcorp.myqueue", nil)

dispatch_async(queue) {
    print("Hello World")
}
let queue = dispatch_queue_create("com.xyzcorp.myqueue", nil)

queue.async {
    print("Hello World")
}

let queue = DispatchQueue(label: "com.xyzcorp.myqueue")

queue.async {
    print("Hello World")
}
transform = CGAffineTransformTranslate(transform, toCenter.x, toCenter.y)
transform = CGAffineTransformRotate(transform, angle)

CGContextSetGrayStrokeColor(context, 0.5, 1.0)

CGContextDrawPath(context, .Stroke)
transform = translate(transform, toCenter.x, toCenter.y)
rotate(angle)

setgrayStrokeColor(context, 0.5, 1.0)
drawPath(context, stroke)
transform = transform.translateBy(toCenter.x, toCenter.y)
    .rotate(angle)

context.setStrokeColor(gray: 0.5, 1.0)

context.drawPath(.stroke)
transform = transform.translateBy(x: toCenter.x, y: toCenter.y)
    .rotate(angle)

context.setStrokeColor(gray: 0.5, alpha: 1.0)

context.drawPath(using: .stroke)
Omit needless words  Consistent parameter labels  Predictable type system  Better error and warnings

Generic type aliases  Better performance

Stringly typed enums  swift.org

Import ObjC Lightweight Generics

New collection index model  #selector

Playgrounds in downloadable toolchains

Standard Library View in Xcode  #keypath  Tuple comparisons

Swift on Linux  Unused value warnings

Swifty Cocoa  API Design Guidelines

Package manager  WMO on by default

Faster compile times  Improved Numerics
Swift
Newton's Cradle and UIKit Dynamics

This playground uses UIKit Dynamics to create a Newton's Cradle. Commonly seen on desks around the world, a Newton's Cradle is a device that illustrates conservation of momentum and energy.

Let's create an instance of our UIKit Dynamics-based Newton's Cradle. Try adding more colors to the app to increase the number of balls in the device.

```swift
let newtonCrade = newtonCradeattachments
```

**Note and update**

Try changing the size and spacing of the balls and see that changes the device. What happens if you make ball0's mass a negative number?

```swift
let newtonCradeball0.mass = -10.0
```

**Behavior**

Adjust elasticity and friction to change how the balls react to collisions.

```swift
let newtonCradeball0.ellipseConstraints = .all
```

**Shape and rotation**

How does Newton's Cradle look if you can square instead of circles and allow them to rotate?

```swift
let newtonCradeball0.shaped = .square
```

**Gravity**

Change the angle and amplitude of gravity to see what Newton's Device might look like in another world.

```swift
let newtonCradeball0.gravity = .bounce
```
Your character, Byte, loves to collect gems, but can’t do it alone. In this first puzzle, you need to write Swift commands to move Byte across the world and collect the gem.

1. Look for the gem in the puzzle world.
2. Enter the correct combination of the moveForward() and collectGem() commands.
3. Tap Run Code.

moveForward()
moveForward()
movestraight()
collectGem()

Congratulations!
You've written your first lines of Swift code. Byte performed the commands you wrote and did exactly what you asked, in exactly the order that you specified.
Designed for Touch
Goal: Use a for loop to repeat a sequence of commands.

In this puzzle, Byte must collect four gems that are located in the same relative locations around a square. You’ll create a loop that repeats the code below for each of the sides to solve the entire puzzle.

1. Drag a for loop from the code library, then drop it above the existing code.
2. Tap the bottom curly brace to select the loop.
3. Tap and hold that curly brace, then drag it downward to pull the existing code into the loop.

```plaintext
for (let i = 0; i < 4; i++) {
    moveForward();
    collectGem();
    moveForward();
    moveForward();
    moveForward();
    turnRight();
}
```
**Goal:** Use a for loop to repeat a sequence of commands.

In this puzzle, you'll collect gems that are located around a square. Here is the plan:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

1. Draw the circle with the cursor, then drag it to the left.
2. Tap the circle to collect the gems.
3. Tap the circle again to repeat the process.

```swift
for i in 1...5 {
    moveForward() // move in a straight line
    collectGems() // collect gems
}
```

- moveForward(): Move in a straight line.
- collectGems(): Collect gems.
- moveToRight(): Move to the right.
Goal: Use a for loop to repeat a sequence of commands.

In this puzzle, Byte must collect four gems that are located in the same relative locations around a square. You'll create a loop that repeats the code below for each of the sides to solve the entire puzzle.

1. Drag a For loop from the code library, then drop it above the existing code.
2. Tap the bottom curly brace to select the loop.
3. Tap and hold that curly brace, then drag it downward to pull the existing code into the loop.

```plaintext
for i in 1...4:
    code
    moveForward()
    collectGem()
    moveForward()
    moveForward()
    moveForward()
    turnRight()
```

Run My Code  Hint
Goal: Use a for loop to repeat a sequence of commands.

In this puzzle, Byte must collect four gems that are located in the same relative locations around a square. You'll create a loop that repeats the code below for each of the sides to solve the entire puzzle.

1. Drag a for loop from the code library, then drop it above the existing code.
2. Tap the bottom curly brace to select the loop.
3. Tap and hold that curly brace, then drag it downward to pull the existing code into the loop.

```python
for i in range(4):
    moveForward()
    collectGem()
    moveForward()
    moveForward()
    moveForward()
    turnRight()
```
Goal: Use a for loop to repeat actions.

In this puzzle, Byte is located in the same spot. You'll create a loop that moves Byte to different spots to solve the puzzle.

1. Drag the for loop from above the existing code to the box.
2. Tap the bottom curly bracket to close the loop.
3. Tap and hold the bottom syntax to add an action.

for i in 1...4:
    moveForward()
    collectGem()
    moveForward()
    turnRight()...

Repeat these steps to complete the puzzle.
for i in 1 ... 4 {
    moveForward()
    collectGem()
    moveForward()
    move
    turnRight()
}
Lessons and Templates
LEARN TO CODE 1
Fundamentals of Swift

LEARN TO CODE 2
Beyond the Basics
Answers

show("Enter a number!")

let response = ask("Number")

func isPrime(_ number: Int) -> Bool {
    let root = sqrt(Double(number))
    for i in 2 ... Int(root) {
        if number % i == 0 {
            return false
        }
    }
    return true
}

guard let number = Int(response) else {
    show("You must enter a number!")
    exit(0)
}

7 is prime!
let nShapes = 100
for i in 0..<nShapes {
    let di = Double(i)
    let dm = Double(nShapes) - i
    let size = Double(nShapes) - i
    let square = Rectangle(width: size / 2, height: size / 2)
    square.opacity = 0
    square.animate(duration: 3, delay: 3 / di, {
        square.rotation = di / 3.14
        square.scale = 1 - (di / dm)
        square.color = Color(color:ColorHSL(hue: 0.5 / Float(nShapes), saturation: 1.0 - (di / size), blue: 1.0 / Float(nShapes), alpha: 1))
    })
}
Real iOS APIs
Real Swift Code
Challenges: Use the AND, OR, and NOT operators to manipulate data in the world.

Each of these operators influences the way your program's code runs.

- The `AND` operator outputs a `True` value when both inputs are `True`.
- The `OR` operator outputs a `True` value when one or both inputs are `True`.
- The `NOT` operator outputs `False` when the input is `True`.

Use this program to solve the first and second challenges and then the third challenge, as shown.

```c
for i in range(8):
    print(bin(i))
```

Write a program to solve the third challenge, and then use the code of challenge 3 to solve the third challenge.
Challenge: Teleport through the gem.

For your first challenge, there's a lip world. A portal teleports byte from
with byte facing the same direction.

You'll need to use all the commands in the right order, to toggle through the portal, and collect the gems.

Don't worry if you don't get it right your chance to experiment!

moveForward();
moveForward();
turnLeft();
moveForward();
toggleSwitch();
moveForward();
moveForward();
turnLeft();
moveForward();
moveForward();
moveForward();
Challenge: Teleport through the gem.

For your first challenge, there's a real world. A portal teleports Byte from with Byte facing the same direction.

You'll need to use all the commands far, in the right order, to toggle on through the portal, and collect the gem.

Don't worry if you don't get it right your chance to experiment!

```
moveForward()
moveForward()
moveForward()
turnLeft()
moveForward()
moveForward()
```
Source Editor
```swift
// the new coding font
let message = "Hello WWDC!"
let font = ["SF Mono", 12]
let awesomeLevel1 = 11
```
let message = "Hello WWDC!"
let font = ["SF Mono", 12]
let awesomeLevel1 = 11
Swift Color Literals

let backgroundColor = ✿
let highlightColor = 🎈
let tintColor = 💜
Swift Color Literals

```swift
let backgroundColor = *
let highlightColor = *
let tintColor = *
```
Swift Image Literals

let icon = 🌞 sticker_sun
let fav = 🌳 sticker_tree
Swift Image Literals

```swift
let icon = sticker_s
let fav = sticker_tree
```
func createSession(username: String) -> Bool {
    // implementation forthcoming
}
func createSession(username: String) -> Bool {
    // implementation forthcoming
}
Xcode App Extensions
Source Editing

- Content addition and deletion
- Content transformation
- Content selection
- Pasteboard modification
- In-file navigation
App Store
Developer ID
Sierra and El Capitan
Separate process from Xcode
Xcode 8 runtime secured by System Integrity Protection
Source Editing Extension
API Reference
An SCNBox geometry models a six-sided polyhedron whose faces are all rectangles. The edges and corners of a box may be square or rounded.

Overview

Define the shape of the box in the x-, y-, and z-axes dimensions of its local coordinate space by setting its `width`, `height`, and `length` properties. Add rounded edges and corners to a box with its `cornerRadius` property. To position and orient a box in a scene, attach it to the `geometry` property of an `SCNNode` object.
Unified API Reference
Symbol Browsing
Swift and Objective-C
An SCNBox geometry models a six-sided polyhedron whose faces are all rectangles. The edges and corners of a box may be square or rounded.
SDK Availability
Swift and Objective-C Syntax
Full Offline API Reference

Language
Swift, Objective-C

SDKs
iOS 8.0+
OS X 10.8+
tvOS 9.0+
watchOS 2.0+
SDK Availability
Swift and Objective-C Syntax
Full Offline API Reference
Demo
Accessibility + Scriptability
135 New Diagnostics
Three New Analyzers
New Testing Options
Captured Crash Logs

Test Without Building

> xcodebuild test
Runtime Issues
Ambiguous Layouts
Ambiguous Layouts
Data Races
Unlock from Wrong Thread
Thread Leaks
Uninitialized Mutexes
Unsafe Calls in Signal Handlers
Thread Sanitizer
fillDescriptionTextView()
updateUIForTraitCollection(self.navigationController)

loadInitialData()

routeEstimator?.requestUpdate(completion: {
    self.updateDidFinish(withSuccess: success)
})

func resetUpdateState() {
    updateFinished = true
}

override func viewDidLoad(_ animated: Bool) {
    super.viewDidLoad(animated)

    // Scroll to middle image
    scrollView.scrollRectToVisible(CGRect(x: scrollview.bounds.size.width, height: 20))
}
Memory Debugger
Demo
Fix Issue
New Signing Actions
Configuration and Issue Details
Actionable Messages
Provisioning Report
Automatic Code Signing
Customized Code Signing
Per Build Configuration

App Store

PROVISIONING
Per Build Configuration

- development
- App Store
- integration Server
Development Certificates
Provisioning
50x Faster indexing of tests

Xcode Launch: 2x
Release Builds with LTO: 1.3x
Debug Stepping: 2.1x
View Debugger Rendering: 1.7x
Playgrounds Execution: 2.5x
Signing Actions: 4x
Source Control with Git: 5x
Indexing Tests: 50x
3x Speed Improvement

zlib

lzfse
<table>
<thead>
<tr>
<th>High-Priority Apps</th>
<th>Low-Priority Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Traditional Logging

Fragmented approaches
Inflexible
Inefficient
Limited system-wide visibility
Traditional Logging

- Fragmented approaches
- Inflexible
- Inefficient
- Limited system-wide visibility
Traditional Logging

- Fragmented approaches
- Inflexible
- Inefficient
- Limited system-wide visibility

New Logging

- Unified and highly efficient
- Fine-grained logging levels
- In-memory tracing
- Support privacy requirement
Live Streaming
Advanced Filtering
Activity View
Over a Billion Devices
Over a Billion Devices
Apple File System
Modern
Flash/SSD
Resilient
64 bit
Encryption
Cloning

Fast
Zero space
Files and directories
Snapshots

- Full volume
- Mountable
- Support reverting
Developer preview at WWDC on macOS
Coming to all Apple Devices
\[ \forall D, D', S \in \text{Range}(A), d_H(D, D') = 1 : \]
\[ \ln \frac{\Pr[A(D) \in S]}{\Pr[A(D') \in S]} \leq \epsilon \]

\[ f(x; \lambda) = \frac{1}{2\lambda} e^{-|x|/\lambda} \]

\[ \sup_{S \in \sigma(Z)} \sup_{x, x' \in X} \frac{\Pr[Z_i \in S | X_i = x]}{\Pr[Z_i \in S | X_i = x']} \leq e^{\epsilon} \]

\[ H_p = \begin{bmatrix} H_{p/2} & H_{p/2} \\ H_{p/2} & -H_{p/2} \end{bmatrix} \]

\[ \forall i \in [n], d \in S, \left| \ln \frac{\Pr[T_i \in T | d_i = d]}{\Pr[T_i \in T | d_i = \text{NULL}]} \right| \leq \epsilon \]
What proportion of our developers prefer Tabs over Spaces?
Tabs vs Spaces

Add noise

Add noise

Add noise

Add noise
Tabs vs Spaces

Estimated Proportion and Confidence Interval

Prefer Tab

Prefer Spaces 0

Sample #

~60%
Deep Link Suggestions
Deep Link Suggestions

Yosemite National Park
Privacy Budget

- Limit submissions per period
- Anonymous pipeline
- Periodically delete donations from server
Engaging With Your App
Handoff
Universal Links
Deep Links
Magical Moments
App Connections
NSUserActivity
NSUserActivity
People and Places
Places
Places

1 Infinite Loop
Cupertino, CA
95014 USA

7, rue Crozatier
75012 PARIS
France

Apple M E FZCO
دبي فرع دبي
مكتب 302، الطابق 3، مبنى رقم 4، إعمار سكوير
الإمارات العربية المتحدة

Russian National Library
Островского пл. 1/3, см карту

Apple (India) Pvt Ltd
19th Floor, Concord Tower
Yogi Bii Ji
No. 24 Vidhul Mala Marg
Bangalore 560-001

Апель ВИП
БКСИ НО 24
Конкорд Тауэр
Бенглuru 560-001

Apple ME FZCO
دبي فرع دبي
مكتب 302، الطابق 3، مبنى رقم 4، إعمار سكوير
الإمارات العربية المتحدة

方正
中国北京市朝阳区建国门外大街2号
国贸大厦3座12层
邮编：100004
"Hey Siri take me there"

Getting directions...
Person Details
Service Type
Handle
John Appleseed

 iPhone

(408) 555-0621

work

j.appleseed@icloud.com

social profile (found in WhatsApp)
j.appleseed
John Appleseed

iPhone:
(408) 555-0621

Work:
j.appleseed@icloud.com

WhatsApp:
j.appleseed

Notes:

Send Message
Share Contact
Add to Favorites
Integrating with iOS
Notifications

Welcome to WWDC!
Slide for more

Press home to open
Embedded Attachments
End-to-End Encryption
End-to-End Encryption
Content Extensions
Content Extensions
Widgets
WEATHER
San Francisco
Mostly Cloudy
Chance of Rain: 0%

Now 6PM 7PM 8PM 9PM
16 15 14 13 Sunset 13

NEWS
TOP STORIES
FOX NEWS
Obama endorses Clinton for president, on heels of Sanders meeting
1h
@nytimes
Bernie Sanders meets President Obama and pledges to work to defeat Donald Tr...
1h

FOR YOU

San Francisco
Mostly Cloudy
Len of Rain: 0%
Fox News

Obama endorses Clinton for president, on heels of Sanders meeting

1h

For You

- Serious Eats
- The New York Times
- Yachting World
Update your look
Update your look
Build with iOS 10 SDK
Update your look
Build with iOS 10 SDK
There is no step 3
Right-to-Left
Localization Guide
https://developer.apple.com/internationalization
GROVE
From wood. By hand.

CUSTOM SURFBOARD LINEUP
No adoption needed
Paired by type
NSDocument
iCloud Documents and Desktop
NSDocument
UIDocument
CloudKit
iCloud Drive
iCloud Key-Value Storage
MapKit
Push
CloudKit
Public Data
Public Data
Private Data
CloudKit Sharing
CloudKit Sharing
Things to do for State of the Union

- think up new slides
- rewrite talk
- practice speech

Invite People
Invite People
Manage People
watchOS
Glanceable
Glanceable  Actionable  Responsive
Background runtime
Frontmost during workout
Continuous sensor access
Crown Events
Gesture Recognizers
Gyroscope
SpriteKit and SceneKit
Speaker access
Inline Video
Glanceable  Actionable  Responsive
Demo
The future of TV is apps.
AVKit  GameKit  QuartzCore  SpriteKit  GameplayKit  OpenGL ES  CoreMedia
ViewController  Audio Toolbox  StoreKit  AVFoundation  MapKit
Dark Appearance  ReplayKit  PhotoKit  App Badging  HomeKit
Up Next  External Accessory  Universal Links  SpriteKit Focus
Metal  SceneKit  CloudKit  UIKit  Foundation  MediaToolbox
JavascriptCore  Accessibility  CoreAudio  MediaPlayer  CoreGraphics  TVMLKit
The Americans is a period drama about the complex marriage of two KGB spies posing as Americans in suburban Washington D.C.
The Americans is a period drama about the complex marriage of two KGB spies posing as Americans in suburban Washington D.C.
Multipeer Connectivity
Four simultaneous controllers
Updated controller policy
“Color accuracy that is visually indistinguishable from perfect.”

– DisplayMate
Limited Color
sRGB
Wide Color (P3)
APIs
Sharing
PDF and printing
System Apps
Capture Deep and Wide Color Images
API to Access DNG (RAW) camera images
API to capture LivePhotos
Metal Tessellation
Metal Function Specialization
Metal Function Specialization
Memoryless Render Targets

Resource Heaps
Hundreds of millions of devices
Every iOS device and Mac we ship
All our key graphics frameworks
Major game engines
Games
ReplayKit Streaming
ReplayKit Streaming
Pathfinding

Rule Systems

State Machines

Random Sources

Agents

Procedural Generation

Spatial Partitioning

Artificial Intelligence

Entities & Components

Entities & Components
Pathfinding
Rule Systems
Artificial Intelligence
State Machines
Random Sources
Artificial Intelligence
Demo
Over 100 sessions
Over 150 labs
Over 1000 Apple engineers