Xcode 7
Free Native Development

13 and older
Pre-release Software
Documentation & Resources
Developer Forums
Test on Device
Apple Services
Analytics & Crash Reporting
Team Development
Technical Support
Distribution on all App Stores
Software Update
Free Space Required to Install

<table>
<thead>
<tr>
<th>1GB</th>
<th>iOS 8</th>
<th>iOS 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5GB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Free Space Required to Install

iOS 8: 4.6GB

iOS 9: 2GB
Free Space Required to Install

- **iOS 8**: 4.6 GB
- **iOS 9**: 1.3 GB
Software Update

iOS 9.0.1 is ready to install.

Install Now

Install Later

Details
Installation Complete
Software Update Version 9.0.1 was successfully installed.

More Info   Close
App Slicing
32-bit
64-bit
GPU low
GPU high
Images 1x
Images 2x
Images 3x
32-bit

Images 1x

GPU low

Images 2x

Images 3x

GPU high

64-bit
On Demand Resources
Sliced for device
Hosted by Apple
Downloaded when needed
Reclaimed as appropriate
## Sliced Sizes

<table>
<thead>
<tr>
<th></th>
<th>Regular</th>
<th>ODR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>App Size</strong></td>
<td>4 GB</td>
<td>2 GB</td>
</tr>
<tr>
<td><strong>Max Resources In-Use</strong></td>
<td>N/A</td>
<td>2 GB</td>
</tr>
<tr>
<td><strong>Max Resources Total</strong></td>
<td>N/A</td>
<td>&gt;&gt; 4 GB</td>
</tr>
</tbody>
</table>
Relative GPU Performance

iPad
iPad 2
iPad 3
iPad 4
iPad Air
iPad Air 2
Relative GPU Performance

64-bit Processors

iPad
iPad 2
iPad 3
iPad 4
iPad Air
iPad Air 2
iPhone app

Extension

User Interface

watchOS 1
Speed and responsiveness
Standalone functionality
Access to Watch hardware
User Interface

NSURLConnection

Extension

Web
Complication

Glance

Notification

App
2:00 PM
Marina – Noe Valley
Group B

4:30 PM
SOMA – Cow Hollow
Group A

7:30 PM
Sunset – Dogpatch
Group B
4:30PM
SOMA – Cow Hollow
Group A

7:30PM
Sunset – Dogpatch
Group B

7:10
00:00
Scheduled Updates
Scheduled Updates
Scheduled Updates
2PM Kickoff
Marina  0
Noe Valley  0

Push Updates
2PM Kickoff
Marina  0
Noe Valley  0

23' Goal Marina
Marina  1
Noe Valley  0

Push Updates
2PM Kickoff
Marina  0
Noe Valley  0

23' Goal Marina
Marina  1
Noe Valley  0

47' Goal Noe
Marina  1
Noe Valley  1

Push Updates
Final Score
Marina  1
Noe Valley  2

8:26
57°
Final Score
Marina 1
Noe Valley 2
<table>
<thead>
<tr>
<th>Team</th>
<th>Games Played</th>
<th>Wins</th>
<th>Draws</th>
<th>Losses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castro</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Nob Hill</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cow Hollow</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SOMA</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Group A Standings

- Castro – 2 GP
  - 1W 1D 0L
  - Total: 4

- Nob Hill – 2 GP
  - 1W 0D 1L
  - Total: 3

- Cow Hollow – 3 GP
  - 0W 1D 0L
  - Total: 1

- SOMA – 1 GP
  - 0W 0D 1L
  - Total: 1
<table>
<thead>
<tr>
<th>Team</th>
<th>GP</th>
<th>W</th>
<th>D</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castro</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Nob Hill</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cow Hollow</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SOMA</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
The SOMA–Cow Hollow game is starting at 4:30 PM. This is your last chance to pick the winner!
The SOMA–Cow Hollow game is starting at 4:30PM. This is your last chance to pick the winner!
The SOMA–Cow Hollow game is starting at 4:30PM. This is your last chance to pick the winner!
The SOMA–Cow Hollow game is starting at 4:30 PM. This is your last chance to pick the winner!
The SOMA–Cow Hollow game is starting at 4:30 PM. This is your last chance to pick the winner!
Group B

Sunset
Noe Valley
Dogpatch
Marina

Loss
Win

Thursday

2:00 PM
1-2
2-1

06/11/15
04:15
PassKit integration  Make phone calls

Digital crown  Taptic engine  Animations

Live health sensor data  Audio recording

Start directions  Display alerts  Video playback

Live accelerometer data  Audio playback
Demo

Transitioning to a native app
Foundation Technologies
Compression Algorithms

![Graph showing the comparison of compression algorithmslz4, zlib, and lzma based on speed and compression ratio. The x-axis represents compression ratio, and the y-axis represents speed. lz4 is shown with a higher compression ratio and lower speed, zlib is in the middle, and lzma has a lower compression ratio but higher speed.](image-url)
Compression Algorithms

Speed vs. Compression

- **lz4**
- **zlib**
- **lzma**

The diagram compares the speed and compression of different compression algorithms.
Compression Algorithms

![Graph showing compression and speed comparison for zlib, lz4, and lzma.]

- **Competition**
  - zlib
  - lz4
  - lzma

- **Speed**
  - 0.1
  - 1
  - 10

- **Compression**
  - 0.5
  - 1
  - 1.5
Compression Algorithms

- **lz4**: High speed, moderate compression
- **zlib**: Moderate speed, high compression
- **lzma**: Lower speed, very high compression

The diagram compares speed (vertical axis) and compression (horizontal axis) for these algorithms.
Compression Algorithms

![Graph showing compression and speed comparison for various algorithms: lz4, lzfs, zlib, lzma.](image-url)
Decode energy

- zlib
- lzfse
let len = compression_decode_buffer(dst, dlen, 
src, slen, nil, 
COMPRESSION_ZLIB)
Easy Adoption

```swift
let len = compression_decode_buffer(dst, dlen,
   src, slen, nil,
   COMPRESSION_LZFSE)
```
Backlight algorithms
Facedown detection
Adaptive sleep delays
Low power idle
1 hour

Additional time before charging
### Battery Percentage

Battery percentage is always shown in the status bar in Low Power Mode.

#### Battery Usage

<table>
<thead>
<tr>
<th>App</th>
<th>Usage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps</td>
<td>38%</td>
</tr>
<tr>
<td>Home &amp; Lock Screen</td>
<td>27%</td>
</tr>
<tr>
<td>Settings</td>
<td>9%</td>
</tr>
<tr>
<td>Podcasts</td>
<td>6%</td>
</tr>
<tr>
<td>Music</td>
<td>5%</td>
</tr>
<tr>
<td>Notes</td>
<td>4%</td>
</tr>
</tbody>
</table>

*Last 24 Hours*
Low Power Mode

Low Power Mode temporarily reduces power consumption until you can fully charge your iPhone. When this is on, mail fetch, background app refresh, automatic downloads, and some visual effects are reduced or disabled.

Battery Percentage

Battery percentage is always shown in the status bar in

Low Power Mode

Low Power Mode temporarily reduces power consumption until you can fully charge your iPhone. When this is on, mail fetch, background app refresh, automatic downloads, and some visual effects are reduced or disabled.

Battery Percentage

Battery percentage is always shown in the status bar in the status bar in
Apple ID
Two-factor authentication
Apple ID

Sign in with your Apple ID to use iCloud, iTunes, the App Store, and more.

Apple ID  example@appleid.com

Password  Required

Don't have an Apple ID or forgot it?

Your Apple ID is the account you use to access all Apple services.
About Apple ID and Privacy

Use different Apple IDs for iCloud & iTunes?
To finish signing in, enter the verification code sent to your other devices.
Verification Code
Enter this verification code on your iPad to finish signing in.

277 531

Cancel
10,000 combinations
Touch ID or Enter Passcode

10,000 combinations
Touch ID or Enter Passcode

1 million combinations
Downgrade attacks
Weak encryption
Best practices
TLS 1.2
Forward secrecy
Use **NSURLSession**

On by default with iOS 9/OS X 10.11

Can declare exception domains in Info.plist
Admin = Kernel
Admin $\neq$ Kernel
Protects system files
No installing in system locations
Protects system processes
Streamlined developer workflow
Recovery partition utility
IPv6
IPv4 addresses exhausted
Key carriers deploying IPv6-only
Critical to support IPv6
OS X since 10.1
iOS since iOS 4.0
Mature and field-proven
Use the networking frameworks

Avoid use IPv4-specific APIs

Avoid hard-coded addresses
Internationalization
World-Wide App Store Sales
World-Wide App Store Sales

- US: 31%
- Japan: 25%
- China: 16%
- Others: 28%
Localization Guide

https://developer.apple.com/internationalization
Grace Murray Hopper
Grace Murray Hopper
Grace Murray Hopper

王冬龄

NSPersonNameComponentsFormatter
Right to Left Languages
Demo

Right to Left

Sara Radi
UIView.
userInterfaceLayoutDirectionForSemanticContentAttribute()
Universal links
Hi Derek,

I thought you might be interested in these guys, up in the Okanagan.

https://twitter.com/Bellawines/status/60464018547302400

Hope to see you soon

Lauren
These little beauties are all ready for the voyage to #YYJ Monday. Western Canada's only ancestrale coming soon!

Bella Wines
@Bellawines
Hi Derek

I thought you might be interested in these guys, up in the Okanagan.

https://twitter.com/Bellawines/status/604404018547302400

Hope to see you soon

Lauren
These little beauties are all ready for the voyage to #YYJ Monday. Western Canada's only ancestrale coming soon!

5/29/15, 2:48 PM
These little beauties are all ready for the voyage to #YYJ Monday. Western Canada’s only ancestrale coming soon!
These little beauties are all ready for the voyage to YYJ Monday. Western Canada’s only ancestrale coming soon!
These little beauties are all ready for the voyage to #YYJ Monday. Western Canada's only ancestrale coming soon!
Register app links

NSUserActivity

application:continueUserActivity:restoration Handler:
iCloud remote access
Setup accessory with camera
Motorized window shades
Home security systems
Sensors
Programmable switches
Notifications
Notifications
Notifications
User Management

Allow Remote Access
Remote access allows you to control your accessories when you are not at home.

PEOPLE

Emily Parker
Meg Parker

Invite People
Users can control accessories in your home. You can share your home with anybody with an iCloud account.
App Search
App Search
CoreSpotlight
App Search

CoreSpotlight
App indexing extension
App Search

CoreSpotlight
App indexing extension
NSUserActivity
App Search

CoreSpotlight
App indexing extension
NSUserActivity
Web markup
App Search
App Suggestions
App Search
App Suggestions
Multitasking
Striking pictures of London commuters in alternative newspaper Urbis

Mark Sanders’ latest photography project was inspired by L.A. Louver’s paintings of busy street scenes. Seeking a new format for his photographs of London commuters, he teamed up with former art director and designer of Mr. Porter, Patrick Guilfoyle to present the project as a newspaper. Urbis is a nod to the free London papers, but rather than something to be left behind on the tube, its beautiful images and quality paper stock make it feel like something to keep. Also unlike a typical newspaper, Urbis is virtually wordless as the photographs speak for themselves.

Inside, abuzz-tailed ants walk the streets of the London, headphones in, phones in hand, heads down, but Mark’s photographs manage to make the well-trodden territory look fresh. The photographs have found his distint style shooting for the likes of Asystel, Menzies, Margaret

This week’s most viewed articles

Shining

Rumours: Smith’s ‘snub’ reportedangkan for the first time with quotes from Kate Bush (still up)

3D Flyover Tour of London
Adaptive UI

- Size Classes
- Bounds Changes
- Dynamic Type
- Autolayout
Dynamic Type
Auto layout
Size Classes
Pitchfork @pitchfork
Watch @theweeknds full @Coachella performance p4k.co/Ly1t8

Pitchfork @pitchfork
"I just desperately wanted to be happy again in a way that wasn’t forced"—@bassking’s path through the darkness p4k.co/JKBrhQ

Pitchfork @pitchfork
@Drake announces tour with @future p4k.co/Ly1s8P
Striking pictures of London commuters in alternative newspaper *Urbis*
Striking pictures of London commuters in alternative newspaper Urbis
Striking pictures of London commuters in alternative newspaper Urbis

Mark London's latest photography project was inspired by L.S. Lowry's paintings of busy street scenes. Searching for a new format for his photographs of London commuters, he teamed up with writer art director and designer of Mr. Porter, Patrick Griffiths to present the project as a newspaper. Urbis is a nod to the free London papers, but rather than something to be left on the tube, its beautiful images and quality paper stock make it feel like something to keep. More like a typical newspaper, Urbis is virtually weightless so the photographs speak for themselves.

Inside, smart-witted suit and ties the streets of the London, headphones in, phones in hand, heads down, but Mark's photographs manage to make this well-travelled territory look fresh. The photographer has honed his distinct style shooting for the likes of Eypfel, Memoirs, Margaret
Striking pictures of London commuters in alternative newspaper style

Mark Nebro’s ‘Shine’ photography project was inspired by J.R. Cauty’s print copy of ‘The Street Across’. Having been a professional photographer of music concerts, he wanted to get close to a subject that he knew well with London as his base and designer of the layout. Mark is still building the print as a newspaper. It is a personal project to break some taboos, but more than anything it’s the ability to adapt and create that makes him feel as if he’s living a typical newspaper. It is to visually rework the London’s mornings.

Shine, almost an abstract, works on the streets of London, headlines in the air, articles in hand. Londoners, met by photographers on the streets, will see the newspaper in their morning work. The photograph has the same impact as Anthony N Green’s POTUS for the Observer at 7am. Mark’s Howl is a newspaper for the ‘unseen’ – people that work in their office work, following their work, Mark is already working on his next project which will continue the theme of people and their movements in life.
Adopt adaptive UI
Adopt adaptive UI
Use a launch storyboard
Adopt adaptive UI
Use a launch storyboard
Support all orientations
Sarah Castelblanco

To: Eden Sears

Not the same without you

Today at 9:14 AM

I just walked by your old cube and looked to see if you were in there. Not sure when I'll stop doing that. I just know that the guy who's sitting there now is probably starting to get annoyed with me. What can I say? We miss you. All of us. Especially me. Although I'm probably more productive now without you and your seemingly endless supply of office gossip.

Work is pretty much the same. I switched cubes and have gotten closer to an actual window. That's the good part. The bad part is that I'm directly across from Larry. You know, the guy with the terrific hair and the terrible taste in music. I actually caught myself singing along to the music coming from his office the other day. It definitely wasn't my proudest moment.

I've been traveling a bit more, managed to parlay into a week off shows and ate nearly every one of the adventure. You'd have loved it.

I'm eager to hear what's going on. Advancement is worth not being.

Sarah
Background media playback
Background media playback
Enable picture in picture
AVPlayerViewController
WKWebView
Window Management
11 Hot Beach Escapes

This past winter will go down in history as one of the snowiest ever, which means you've more than earned a summer getaway to the beach. So kick back, sink your toes in the sand and open a cold one. At these 11 places, the heat index is so high you'll never want to leave.

1. Santa Barbara

Where to Stay

Macakizi Hotel
Mugla, Turkey

One&Only Hayman Island
Queensland, Australia

Tropical vacation ideas

Na Pali Properties Kauai—talk to Jane
3-bedroom Airbnb house on Boipeba Island, Bahia
Look into Lamai Beach, Thailand
Ask Christy about Italian villas
Check with Grandma about time-share
Any resizable window
Narrow and wide geometries
New APIs to optimize experience
NSPressureConfiguration
NSGestureRecognizer
NSView.pressureChangeWithEvent()
NSEventType.EventTypePressure
SCHOOL NEWS

The New Library Opens

After nearly a year long wait, the new library with a much larger collection of science, history, and art books finally opened up to all students and faculty to take advantage of during the school day.

This marks the completion of our six-million dollar development project, which started last January. The library can be entered from the west side of the school and hours are from 7:00 am – 4:00 pm M-F.

“We wanted to create a space for students across grade levels to learn and enjoy”, said Beth Crittin, school librarian. Crittin hand-selected the rich collection of books and resources in multiple reading for boys across wide set of subject matters.

The library also has 15 learning labs, where students and teachers can work together on projects. Each learning lab is equipped with an Apple Mac, and Apple Merced helps students take advantage of the new technology.

It's been a long time coming but it's finally ready for everybody! – William S.
SCHOOL NEWS

The New Library Opens

After much construction and planning, the School District proudly opens its newest addition to the campus.

The new library is a space designed to support learning and collaboration. With its modern design and ample resources, it is a hub for students and faculty alike.

It's been a long time coming, but it's finally here for everyone.

---

The Library offers resources for success.

---

School Board and Staff: "We are thrilled to open our doors to this new space.

The library is a place where students can come together to learn, grow, and develop their interests."
File Coordination

LSSupportsOpeningDocumentsInPlace

application(app: UIApplication, openURL url: NSURL, options: [String: AnyObject])
Free (with limits)
Free

Storage

10 GB Assets
100 MB Database

Transfer

2 GB/day Assets
40 requests/second
Free

---

**Storage**

- 1 PB Assets
- 10 TB Database

**Transfer**

- 200 TB/day Assets
- 400 requests/second
CloudKit Dashboard
CloudKit Web Services
Full access via JSON
JS library
Secure sign in with Apple ID
Fast
Safe
Modern
Interactive
OSI-approved license
Later this year
Code contributions accepted
“Developers are still going nuts for Apple’s new Swift Programming Language”

Klint Finley, Wired
Swift 2

- Rectangles: 3.6x
- DeltaBlue: 4.9x
- Dictionary: 5.3x
- Histogram: 5.6x
- NopDeInit: 6.1x
- String Builder: 8.2x

Swift performance improvement
Synthesized “headers” in Xcode  
Availability checking  
Multi-payload enums

Swift 1.2 to 2.0 Migrator  
Shorter compile time  
New playgrounds  
@testable

Improved option sets

Markdown in comments

Faster debug mode

Objective-C generics

Error Handling Model

C function pointer APIs

Availability checking

Shorter compile time

New playgrounds

@testable

do {}

Parallelized WMO

guard

Faster runtime
defer

Protocol extensions

Nullability in Objective-C

Faster runtime

defer

Mutability warnings

Recursive nested functions

Pattern matching in "if"

SIMD support

Availability checking
Error Handling Model
Boilerplate
Duplicated cleanup code
Implicit control flow bugs
Accidentally ignored errors
func loadMetadata(error: NSErrorPointer) -> Bool
func loadMetadata(error: NSErrorPointer) -> Bool

func loadMetadata() throws
do {
    ...
    ...
    ...
} catch {
    // Handle all errors
}
do {

... 

... 

} catch NSCocoaError.UserCancelledError {
    // Handle special case

} catch {
    // Handle all errors
}
do {
    try loadDocument()
    calculateResult()
} catch NSCocoaError.UserCancelledError {
    // Handle special case
} catch {
    // Handle all errors
}
throw FileSystemError.OutOfSpace
throw FileSystemError.OutOfSpace

enum FileSystemError : ErrorType {
    case MalformedPath(String)
    case InsufficientPermissions
    case OutOfSpace
    case LostInSpace
}
Availability Checking
Availability Checking
Availability Checking
Availability Checking

doSomething()
doAnotherThing()
doWithAnimation()
setParentView()
layoutGuides
setWidth()
class ViewController: UIViewController {

    func rearrangeView() {

        guides = view.layoutGuides
    }
class ViewController: UIViewController {

    func rearrangeView() {

        // 'layoutGuides' is only available on iOS 9.0 or newer
        guides = view.layoutGuides
    }
}
class ViewController: UIViewController {

    func rearrangeView() {

        guides = view.layoutGuides
    }
}
class ViewController: UIViewController {

    func rearrangeView() {
        if #available(iOS 9.0, *) {
            guides = view.layoutGuides
        }
    }

    // Fix-it: Add if #available version check
    // Fix-it: Add @available attribute to enclosing instance method
    // Fix-it: Add @available attribute to enclosing class
class ViewController: UIViewController {

    func rearrangeView() {
        if #available(iOS 9.0, *) {
            guides = view.layoutGuides
        }
    }

    // Fix-it
    // Add if #available version check
    // Add @available attribute to enclosing instance method
    // Add @available attribute to enclosing class
class ViewController: UIViewController {

    @available(iOS 9.0, *)

    func rearrangeView() {

        guides = view.layoutGuides

        // Fix-it: Add if #available version check
        // Fix-it: Add @available attribute to enclosing instance method
        // Fix-it: Add @available attribute to enclosing class
    }
}
class ViewController: UIViewController {

    @available(iOS 9.0, *)

    func rearrangeView() {

        guides = view.layoutGuides

        // Fix-it
        // Add if #available version check
        // Add @available attribute to enclosing instance method
        // Add @available attribute to enclosing class
    }
}
@available(iOS 9.0, *)

class ViewController: UIViewController {

    func rearrangeView() {

        guides = view.layoutGuides

        // Fix-it: Add if #available version check
        // Fix-it: Add @available attribute to enclosing instance method
        // Fix-it: Add @available attribute to enclosing class
    }
}
let x = filter(map(numbers) { $0 * 3 }, isOdd)
let x = numbers

Fluent Interfaces
let x = numbers.map { $0 * 3 }.filter(isOdd)

Fluent Interfaces
func myMethod(x : Thing?) {

}

Early Exits
func myMethod(x : Thing?) {
    if let x = x {
        ...
        ...
        ...
    }
}
func myMethod(x : Thing?) {
    guard let x = x else { return }
    ... 
    ... 
    ... 
}

Early Exits
func myMethod(x : Thing?) {
    guard let x = x else { return }
    ...
    ...
    ...
}

Early Exits
class func componentsFromLocaleIdentifier(String)
class func componentsFromLocaleIdentifier(String) -> [NSObject : AnyObject]
class func componentsFromLocaleIdentifier(String) -> [String : String]
Swift in Xcode
override public func readFromData(data: NSData, ofTypeName: String) throws {
    unarchivedList = NSKeyedUnarchiver. 
    unarchiveObjectWithData(data) as? List
    if let unarchivedList = unarchivedList {
        listPresenter?.setList(unarchivedList)
        return true
    }
    throw NSError(domain: NSCocoaErrorDomain, 
    code: NSFileReadCorruptFileError, 
    userInfo: [ 
        NSLocalizedDescriptionKey: 
        NSLocalizedString("Could not read file.", comment: "Read error description"), 
        NSLocalizedFailureReasonErrorKey: 
        NSLocalizedString("File was in an invalid format.", comment: "Read failure reason")
    ]
}

override public func dataOfType(typeName: String) throws -> NSData {
    let outError: NSError! = NSError(domain: 
    "Migrator", code: 0, userInfo: nil)
    if let archiveableList = listPresenter?.
    archivedListWithRootObject (archiveableList) {
        return NSKeyedArchiver.
    }
    return false
}

override public func readFromData(data: NSData, ofTypeName: String, error outError: NSErrorPointer) -> Bool {
    unarchivedList = NSKeyedUnarchiver. 
    unarchiveObjectWithData(data) as? List
    if let unarchivedList = unarchivedList {
        listPresenter?.setList(unarchivedList)
        return true
    }
    if outError != nil {
        outError.message = NSError(domain: NSCocoaErrorDomain, 
        code: NSFileReadCorruptFileError, userInfo: [ 
        NSLocalizedDescriptionKey: 
        NSLocalizedString("Could not read file.", comment: "Read error description"), 
        NSLocalizedFailureReasonErrorKey: 
        NSLocalizedString("File was in an invalid format.", comment: "Read failure reason")
    ]
}
    return false
}
// SplashScreenPeopleWallView.swift

import UIKit

protocol SplashScreenPeopleWallViewDelegate {
    func nextImageForSplashScreenPeopleWallView(wallView: SplashScreenPeopleWallView) -> UIImage?
}

/// Custom view that renders in IB with photos of my group of friends
public class SplashScreenPeopleWallView: UIView, UIScrollViewDelegate {

    @IBOutlet var delegate: SplashScreenPeopleWallViewDelegate? {
        didSet {
            reloadData()
        }
    }

    private static let kNumberOfColumns = 5
    private static let kNumberOfRows = 3

    private var scrollView: UIScrollView!
    private var imageViews: [UIImageView] = []
    private var displayLink: CADisplayLink!

    required public init(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
        commonInit()
    }

    public override init(frame: CGRect) {
        super.init(frame: frame)
        commonInit()
    }

    private func commonInit() {
        scrollView = UIScrollView(frame: CGRectZero)
        scrollView.showsHorizontalScrollIndicator = false
        scrollView.scrollEnabled = false
        for _ in 0 ..< SplashScreenPeopleWallView.kNumberOfRows * (SplashScreenPeopleWallView.kNumberOfColumns + 1) {

        }
    }

    public override func layoutSubviews() {
        super.layoutSubviews()
        reloadData()
    }

    public override func touchBegan(_ touch: UITouch, with event: UIEvent?) {
        super.touchBegan(touch, with: event)
        reloadData()
    }

    private func reloadData() {
        displayLink?.invalidate()
        displayLink = CADisplayLink(target: self, selector: #selector(self.reloadData))
        displayLink?.add(to: CADisplayLink.mainDisplayLink)
        displayLink?.fire()
    }

    @objc private func reloadData() {
        for i in 0..<imageViews.count {
            imageViews[i].image = nil
        }
        for i in 0..<LayoutComponent.kNumberOfRows {
            for j in 0..<LayoutComponent.kNumberOfColumns {
                let index = i * (LayoutComponent.kNumberOfColumns + 1) + j
                if let delegate = delegate, let image = delegate.nextImageForSplashScreenPeopleWallView(self) {
                    if image != nil {
                        imageViews[index].image = image
                    }
                }
            }
        }
        displayLink?.invalidate()
    }
}

protocol LayoutComponentDelegate {
    func nextImageForLayoutComponent(componen: LayoutComponent) -> UIImage?
}

public class LayoutComponent: UIView, UIScrollViewDelegate {

    @IBOutlet var delegate: LayoutComponentDelegate? {
        didSet {
            reloadData()
        }
    }

    private static let kNumberOfColumns = 5
    private static let kNumberOfRows = 3

    private var scrollView: UIScrollView!
    private var imageViews: [UIImageView] = []
    private var displayLink: CADisplayLink!

    required public init(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
        commonInit()
    }

    public override init(frame: CGRect) {
        super.init(frame: frame)
        commonInit()
    }

    private func commonInit() {
        scrollView = UIScrollView(frame: CGRectZero)
        scrollView.showsHorizontalScrollIndicator = false
        scrollView.scrollEnabled = false
        for _ in 0 ..< LayoutComponent.kNumberOfRows * (LayoutComponent.kNumberOfColumns + 1) {

        }
    }

    public override func layoutSubviews() {
        super.layoutSubviews()
        reloadData()
    }

    public override func touchBegan(_ touch: UITouch, with event: UIEvent?) {
        super.touchBegan(touch, with: event)
        reloadData()
    }

    private func reloadData() {
        displayLink?.invalidate()
        displayLink = CADisplayLink(target: self, selector: #selector(self.reloadData))
        displayLink?.add(to: CADisplayLink.mainDisplayLink)
        displayLink?.fire()
    }

    @objc private func reloadData() {
        for i in 0..<imageViews.count {
            imageViews[i].image = nil
        }
        for i in 0..<LayoutComponent.kNumberOfRows {
            for j in 0..<LayoutComponent.kNumberOfColumns {
                let index = i * (LayoutComponent.kNumberOfColumns + 1) + j
                if let delegate = delegate, let image = delegate.nextImageForLayoutComponent(self) {
                    if image != nil {
                        imageViews[index].image = image
                    }
                }
            }
        }
        displayLink?.invalidate()
    }
}
import UIKit

protocol SplashScreenPeopleWallViewDelegate {
    func nextImageForSplashScreenPeopleWallView(wallView: SplashScreenPeopleWallView) -> UIImage?
}

/// Custom view that renders in IB with photos of my group of friends
public class SplashScreenPeopleWallView: UIView, UIScrollViewDelegate {

    var delegate: SplashScreenPeopleWallViewDelegate? = nil

    @IBOutlet var delegate: SplashScreenPeopleWallViewDelegate? {
        didSet {
            reloadData()
        }
    }

    private static let kNumberOfColumns = 5
    private static let kNumberOfRows = 3

    private var scrollView: UIScrollView!
    private var imageViews: [UIImageView] = []
    private var displayLink: CADisplayLink!

    required public init(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
        commonInit()
    }

    public override init(frame: CGRect) {
        super.init(frame: frame)
        commonInit()
    }

    private func commonInit() {
        scrollView = UIScrollView(frame: CGRectZero)
        scrollView.showsHorizontalScrollIndicator = false
        scrollView.scrollEnabled = false

        for _ in 0..<SplashScreenPeopleWallView.kNumberOfRows * (SplashScreenPeopleWallView.kNumberOfColumns + 1) { }
// SplashScreenPeopleWallView.swift

import UIKit

protocol SplashScreenPeopleWallViewDelegate {
    func nextImageForSplashScreenPeopleWallView(wallView: SplashScreenPeopleWallView) -> UIImage?
}

/// Custom view that renders in IB with photos of my group of friends
public class SplashScreenPeopleWallView: UIView, UIScrollViewDelegate {

    var delegate: SplashScreenPeopleWallViewDelegate?

    init(coder aDecoder: NSCoder) {
        super.init(coder: aDecoder)
    }

    init(frame: CGRect) {
        super.init(frame: frame)
    }

    /// This helps the custom view show up properly at design time
    func prepareForInterfaceBuilder() {
    }

    func scrollViewDidScroll(scrollView: UIScrollView) {
    }

    /// The main game view consists a wall of images as well as a scroll area
    func layoutSubviews() {
    }

    /// Drives a pretty simply right to left scroll of the friend's photos
    func startAnimating() {
    }

    func displayLinkDidFire(displayLink: CADisplayLink) {
    }

    func reloadData() {
    }
}
Rich Comments in Playgrounds

Map

The map function takes an array and returns a new array by applying a transform to each element in the array. For example, the following code multiplies the quantity of the ingredient by the cost of the ingredient to find the total cost for each ingredient.

```swift
let totalPrices = shoppingList.map { ingredient in
    return ingredient.quantity * ingredient.price
}
```

```swift
visualize(shoppingList, totalPrices)
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x Tomato</td>
<td>2</td>
<td>$2</td>
</tr>
<tr>
<td>1x Salt</td>
<td>0.5</td>
<td>$0.5</td>
</tr>
<tr>
<td>1x Rice</td>
<td>0.5</td>
<td>$0.2</td>
</tr>
<tr>
<td>4x Garlic</td>
<td>0.5</td>
<td>$0.4</td>
</tr>
<tr>
<td>2x Pepper</td>
<td>0.5</td>
<td>$0.5</td>
</tr>
<tr>
<td>1x Salt</td>
<td>0.5</td>
<td>$0.1</td>
</tr>
</tbody>
</table>

Reduce

You can use the reduce function to combine elements of an array into a single value. The reduce function takes an initial value to start with and then a function to combine each element in the array with the previous value. The following code takes the total price list and adds them together to compute a final remaining cost:

```swift
let sum = totalPrices.reduce(0) { currentPrice, priceToAdd in
    return currentPrice + priceToAdd
}
```
Rich Comments in Playgrounds
/**
 * Produces the destination point for the node, based on the provided translation.
 *
 * The performance of this method is based on a graded algorithm depending on the `node` object's
 * grading and proximity.
 *
 * TODO: This function isn't working quite right -- yet
 *
 ![Sample graph](performance.png)
 *
 - parameter node: The node used to compute the point to be translated to
 - parameter translation: The translational movement to apply to the node
 - parameter duration: The time interval over which to apply the translation

 - returns: The point to which the `node` should move over the specified `duration` time interval.
   Returns `nil` if there is no displacement.
 */

func pointForTranslatingNode(node: SKNode, withTranslationalMovement translation: MovementKind, duration: NSTimeInterval) -> CGPoint? {
    // No translation if the vector is a zeroVector.
    if translation.displacement == float2() { return nil }

    var displacement = translation.displacement

    // If the translation is relative, the displacement vector needs to be
    // rotated to account for the node's current orientation.
    if translation.isRelativeToOrientation {
        displacement = calculateAbsoluteDisplacementFromRelativeDisplacement(displacement)
    }
}
func pointForTranslatingNode(node: SKNode, withTranslationalMovement translation: MovementKind, duration: NSTimeInterval) -> CGPoint {
    // No translation if the vector is a zeroVector.
    if translation.displacement == float2() { return nil }

    var displacement = translation.displacement
    // If the translation is relative, the displacement vector needs to be
    // rotated to account for the node's current orientation.
    if translation.isRelativeToOrientation {
        displacement = calculateAbsoluteDisplacementFromRelativeDisplacement(displacement)
    }
    
    return displacement
}
Multiple pages
Rich text comments
Inline results
Supporting source files
Embedded resources
public class LobbyViewController: UIViewController, UICollectionViewDataSource,
UICollectionViewDelegate {

@IBOutlet var collectionView: UICollectionView!
@IBOutlet var collectionViewImageCell: UITableViewCell!
@IBOutlet var genreLabel: UILabel! 
@IBOutlet var infoTextLabel: UILabel! 
@IBOutlet var readyToPlaySwitch: UISwitch! 
@IBOutlet var startGameButton: UIButton! 
@IBOutlet weak var genreImageWrapView: UIView!

var players = [Player] []
var readyToPlay = false
var viewDidAppearWasCalled = false

// MARK: - View lifecycle
override public func viewDidLoad() {
    super.viewDidLoad()

collectionView.collectionViewLayout = CircleLayout(itemSize: CGSizeMake(88, 113))
genreImageView.image = game?.genre?.image
genreImageView.layer.borderColor = UIColor.redColor().CGColor

if let g = game {
    players += g.players
}
}

public override func viewWillAppear(animated: Bool) {
    super.viewWillApper(animated)

genreLabel.text = game?.genre?.name
readyToPlaySwitch.on = readyToPlay
updateUI()
}

public override func viewDidApper(animated: Bool) {
    super.viewDidAppear(animated)

    viewDidAppearWasCalled = true
    animateInPlayers()
}

// MARK: - Collection view data source
public func collectionView(collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
    return viewDidAppearWasCalled ? players.count : 0
}
In ‘Game of Thrones’ Daenerys chains all but one of her dragons – which one?

Drogon  Viserion
Rhaegal  Tyrion
Round 1
Sci-fi & Fantasy
Round 1
Sci-fi & Fantasy
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?
Round 1
Sci-fi & Fantasy

QUESTION

In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?
Round 1
Sci-fi & Fantasy

QUESTION

In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

Drogon  Viserion  Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

- Drogon
- Viserion
- Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

- Drogon
- Viserion
- Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

**ANSWERS**

- Drogon
- Viserion
- Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

- Drogon
- Viserion
- Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

Drogon
Viserion
Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

Drogon
Viserion
Rhaegal
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?
In ‘Game of Thrones’ Daenerys chains all but one of her dragons — which one?

ANSWERS

- Drogon
- Viserion
- Rhaegal
Storyboard References
Storyboard References
What You See ...
What You See …
One of the five cities of Cinque Terre, this amazing historical town offers stunning views of the Mediterranean. Try out local wine, visit Doria Castle, and walk between the five towns of Cinque Terre.
VERNAZZA

One of the five cities of Cinque Terre, this amazing historical town offers stunning views of the Mediterranean. Try out local wine, visit Doria Castle, and walk between the five towns of Cinque Terre.
One of the five cities of Cinque Terre, this amazing historical town offers stunning views of the Mediterranean. Try out local wine, visit Doria Castle, and walk between the five towns of Cinque Terre.
VERNAZZA

One of the five cities of Cinque Terre, this amazing historical town offers stunning views of the Mediterranean. Try out local wine, visit Doria Castle, and walk between the five towns of Cinque Terre.
One of the five cities of Cinque Terre, Vernazza offers stunning views of the Mediterranean. Try out local wine, visit Doria Castle, and walk between the five towns of Cinque Terre.
On Demand Resource Tags
On Demand Resource Tags

Level-1
On Demand Resource Tags

Level-1  IntroMovie
## Reading and Writing Rates

![Reading and Writing Rates Chart]

## On Demand Resources

<table>
<thead>
<tr>
<th>Tag</th>
<th>Size</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level-1</td>
<td>35.4 MB</td>
<td>Downloaded</td>
</tr>
<tr>
<td>Level-2</td>
<td>76.4 MB</td>
<td>In Use</td>
</tr>
<tr>
<td>Level-3</td>
<td>101 MB</td>
<td>Downloading</td>
</tr>
<tr>
<td>Level-4</td>
<td>84.2 MB</td>
<td>Not Downloaded</td>
</tr>
<tr>
<td>Audio-1</td>
<td>81.6 MB</td>
<td>Not Downloaded</td>
</tr>
<tr>
<td>MakingOfMovie</td>
<td>135.9 MB</td>
<td>Not Downloaded</td>
</tr>
</tbody>
</table>
Debugging and Profiling Tools
Energy

Utilization

Zero
Energy Impact

High
Energy Impact

45%
Overhead

Energy Impact

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Background</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend
Cost represents energy use resulting from the work your app performs. Overhead represents energy use as a result of bringing up radios and other system resources required to perform that work.

High CPU Utilization
CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Network
Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Location
Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.
Address Sanitizer
<p>| 38 10 80 02 | 9F 19 00 00 | 2B A4 00 01 |
| FF 08 00 00 | 70 00 00 A4 | 00 00 E0 A0 |
| 25 A4 00 01 | 28 A0 00 01 | 01 00 00 00 |
| 70 00 00 FF | 03 1E 00 00 | 55 F1 9F 00 |</p>
<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9F 19 00 00</td>
</tr>
<tr>
<td>1</td>
<td>70 00 00 A4</td>
</tr>
<tr>
<td>2</td>
<td>28 A0 00 01</td>
</tr>
<tr>
<td>3</td>
<td>03 1E 00 00</td>
</tr>
</tbody>
</table>

Backtrace of allocation

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2B A4 00 01</td>
</tr>
<tr>
<td>1</td>
<td>00 00 E0 A0</td>
</tr>
<tr>
<td>2</td>
<td>01 00 00 00</td>
</tr>
<tr>
<td>3</td>
<td>55 F1 9F 00</td>
</tr>
</tbody>
</table>

Backtrace of occurrence

The "robot" variable

Heap overflow here
Memory issue when logging
2301 devices

Crash Log Details

- Binary: Trivia 2.0 (55)
- Thread 0
- iOS Version: 9.0 (13A4254n)
- Device: iPhone 6 Plus

Notes

We’ve been unable to reproduce this crash, but several users have reported it.

Last 2 Weeks

- 2301 devices
- Operating System:
  - iOS 8.3: 42%
  - iOS 8.4: 36%
  - iOS 9.0: 23%
```c
#import "UsageLog.h"

@implementation UsageLog

char * log_buffer = NULL;
int disconnect_attempts = 0;

-(id)init {
    if (self = [super init]) {
        log_buffer = malloc(50);
        return self;
    }
}

-(void)dealloc {
    free(log_buffer);
}

-(void)logAction: (const char *) action {
    sprintf(log_buffer, "USAGE-LOG: %@", action);
    NSLog(@"%@", [[NSString alloc] initWithUTF8String: log_buffer]);
}

-(void)endLogging {
    if (++disconnect_attempts > 2) free(log_buffer);
}
@end```
Testing
DemoRobots-iOS 167 tests

Hero Tests
- testSaveThePrincess
- testSaveTheWorld
- testZapTheBadGuys
- testFlexTheMuscles

Villain Tests
- testSecretWeapon
- testDoubleSecretWeapon
- testEscapeToLair
Synchronous  Asynchronous
Unit Testing
Unit Testing
Unit Testing

User Interface Testing
Unit Testing

User Interface Testing

Code Coverage
Demo
Testing using Xcode
Games and Graphics
Metal
Game Logic

CPU

OpenGL

Game Logic

GPU

OpenCL
Demo
David McGavran, Adobe Systems
Views
Textures
Models/Meshes
Metal Performance Shaders
Optimized shader library
Compute and graphics
Individually tuned for each GPU
Models and materials
3D meshes
Offline raytraced lighting
State of the art lighting
Physically based materials
Offline raytracing
GameplayKit
Brains of your game
Pathfinding and flocking
Game logic
Entities and Components

State Machines

Agents

Pathfinding

MinMax AI

Random Sources

Rule Systems
Game play recording
Hardware accelerated
Built-in sharing and saving
Voiceover
Trimming
Demo

Xcode game tools
iOS 9
watchOS 2
OS X El Capitan
OS X Server 5
Xcode 7
Sample code

developer.apple.com/wwdc/resources/
Over 115 sessions
Over 115 labs
Over 1000 Apple engineers