What’s New in Xcode

Session 104

Ken Orr Developer Tools
Swift 2
Error handling
Availability
Testability
Protocol Extensions
override public func readFromData(data: NSData, ofType typeName: String) throws {
    unarchivedList = NSKeyedUnarchiver.
    unarchiveObjectWithData(data) as? List

    if let unarchivedList = unarchivedList {
        listPresenter?.setList(unarchivedList)
    }
}

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?.archiveableList {
        return NSKeyedArchiver.
        archivedDataWithRootObject (archiveableList)
    }
}
Playgrounds
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground’s Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

```swift
// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create "Seahorse Valley"
m.mandelbrotRect = newMandelbrotRect(points["b"])

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"])
```
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground’s Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

```swift
// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create "Seahorse Valley"
m.mandelbrotRect = newMandelbrotRect(points["b"])

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"])
```
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground’s Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

```swift
// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create “Seahorse Valley”
m.mandelbrotRect = newMandelbrotRect(points["b"]!)

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"]!)
```
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground’s Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

```swift
// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create “Seahorse Valley”
m.mandelbrotRect = newMandelbrotRect(points["b"]!)

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"]!)
```
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground’s Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

```swift
// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create “Seahorse Valley”
m.mandelbrotRect = newMandelbrotRect(points["b"])

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"])
```
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground's Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

```swift
// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create "Seahorse Valley"
let m = MandelbrotRect(newMandelbrotRect(points["b"]!))

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"]!)
```
Mandelbrot Set Rendering

This playground renders the Mandelbrot set by running code embedded within the playground’s Sources folder. You can explore the Sources folder by opening the Project Navigator (CMD+1) and clicking the triangle next to the playground file.

// Create a Mandelbrot set
let m = MandelbrotView(frame: rect)

// Create “Seahorse Valley”
m.mandelbrotRect = newMandelbrotRect(points["b"]!)

// Create Mandelbrot set within the Mandelbrot set
m.mandelbrotRect = newMandelbrotRect(points["e"]!)
case let x where x.hasSuffix("pepper"):
    let vegetableComment = "Is it a spicy \(x)?"
default:
    let vegetableComment = "Everything tastes good in soup."
}

**Experiment:** Try removing the default case. What error do you get?

Notice how `let` can be used in a pattern to assign the value that matched that part of a pattern to a constant.

After executing the code inside the switch case that matched, the program exits from the switch statement. Execution doesn't continue to the next case, so there is no need to explicitly break out the switch at the end of each case's code.

You use `for-in` to iterate over item in a dictionary by providing a pair of names to use for each key-value pair. Dictionaries are an unordered collection, so their keys and values are iterated over in an arbitrary order.

```swift
let interestingNumbers = {
    "Prime": [2, 3, 5, 7, 11, 13],
    "Fibonacci": [1, 1, 2, 3, 5, 8],
    "Square": [1, 4, 9, 16, 25],
}

var largest = 0
for (kind, numbers) in interestingNumbers {
    for number in numbers {
        if number > largest {
            largest = number
        }
    }
}
```

Is it a spicy red pepper?
In ‘Game of Thrones’ who is Tyrion Lannister’s older brother?

Options:
- Eddard "Ned" Stark
- Cersei Lannister
- Jaime Lannister
- Tywin Lannister

Points Total: 650

Trivia Games
Trivia
Sci-fi & Fantasy
In Progress...

AAPL 131.39
Anyone up for a game of Trivia this morning?

That sounds great Amy. I can kick-off the game. Which genre should I choose?

Amy Smith

I am at WWDC watching the “What’s New in Xcode” session.

Johnny Appleseed

Yes!

Angelia O’Neil

Sci-fi & Fantasy sound good?

I am in! Sci-fi & Fantasy is my favorite trivia genre. Let’s play.

Ryan Davidson

Invite me.

Debbie Brown

I’ve got the game running and I am going to invite all of you to play.
App Thinning
Bitcode
Bitcode Slicing
Demo

Jon Hess Developer Tools
### Demo

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Date and Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementing UI Designs in Interface Builder</td>
<td>Pacific Heights</td>
<td>Wednesday 1:30PM</td>
</tr>
<tr>
<td>App Thinning in Xcode</td>
<td>Presidio</td>
<td>Wednesday 9:00AM</td>
</tr>
<tr>
<td>Introducing On Demand Resources</td>
<td>Pacific Heights</td>
<td>Wednesday 4:30PM</td>
</tr>
</tbody>
</table>
Debugging and Profiling Tools
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 locationing uses more energy. Request location and increase precision only when truly necessary.

Background Activity
Activity occurring while your app is in the background. Apps in the background should significantly limit activity; most should be completely idle. Reduce activity immediately when placed into a background state and notify the system once activity is complete.

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.
Location Instrument

A screenshot of the Instruments app from macOS, showing a profile of a Location Instrument. The timeline and details view display various events, including changes in location accuracy, such as from Low to High, and when all location managers were released. The duration and energy impact are also noted for each event.
Metal System Trace
Address Sanitizer
// AppDelegate.m
// TestASan

// Copyright © 2015 Apple. All rights reserved.

#import "AppDelegate.h"
#import "DrawManager.h"

@interface AppDelegate ()
@end

@implementation AppDelegate

- (void)applicationDidFinishLaunching:(NSNotification *)aNotification
{
    DrawManager *manager = [[DrawManager alloc] init];
    [manager calculateAxes];
    
    NSInteger xx = manager.xAxis;
    xx = 7;
    [manager invalidateLayout];
}

- (void)applicationWillTerminate:(NSNotification *)aNotification {
    // Insert code here to tear down your application
}
@end
## Demo

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Debugging and the Address Sanitizer</td>
<td>Mission</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td>What’s New in LLDB</td>
<td>Nob Hill</td>
<td>Tuesday 2:30PM</td>
</tr>
</tbody>
</table>
Fully Symbolicated
App Store and Test Flight
Full Xcode Integration
Memory issue when logging

Thread 0
- libsystem_kernel.dylib: _pthread_kill
- libsystem_pthread.dylib: pthread_kill
- libsystem_c.dylib: abort
- libsystem_malloc.dylib: free

UsageStatistics::UsageLogEndLogging

- Trivia: Trivia+GamePlayerPick...2003 devices
- Crashed while filtering player...1833 devices
- Hard to reproduce crash when...1833 devices
- Assertion in image loading...1622 devices
- Crash when mashing answer...7226 devices
- Assertion when multiple play...1122 devices
- Blow the stack when loading...1087 devices
- Trivia: Trivia+GamePlayerPick...815 devices
- Crash when inviting too ma...735 devices

Crash Log Details
- Binary: Trivia 2.0 (56)
- Thread 0
- iOS Version: 8.0 (13A4254n)
- Device: iPhone 6 Plus

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

Last 2 Weeks Operating System

- iOS 8.3: 42%
- iOS 8.4: 36%
- iOS 9.2: 23%
Demo

Itai Rom Developer Tools
Testing
Performance

Asynchronous
User Interface Testing
Code Coverage
Build 3 targets
Run
Debug
Test
Debug
Profile
Release
Analyze
Debug
Archive
Release
Install
Debug

Info
Arguments
Diagnostics

Build Configuration: Debug
Code Coverage: Gather coverage data
Debugger: Debug executable
Debug Process As: Me

Trivia Tests
- None
- None

Trivia Tests UI Tests
- None
- None
<table>
<thead>
<tr>
<th>Tests</th>
<th>Test Application Data</th>
<th>Test Location</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivia Tests</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Trivia Tests UI Tests</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
(void) positionImageViews

CGFloat imageViewWidth = self.bounds.size.width / numberOfColumns;
for (int column = 0; column < numberOfColumns; ++column) {
    CGFloat x = column * imageViewWidth;
    CGPoint center = CGPointMake(x, center.y);
    UIImage *imageView = [[self.imageViews objectAtIndex:column] image);
    imageView.frame = CGRectMake(x, y, imageView.width, imageView.height);
}

(void) layoutSubviews

CGFloat imageViewWidth = self.bounds.size.width / numberOfColumns;
self.scrollView.frame = self.bounds;
self.scrollView.contentSize = CGSizeMake(self.bounds.size.width - imageViewWidth, self.bounds.size.height);
[self positionImageViews];

(void) startAnimating

if ([self displayLink]) {
    self.displayLink = [[DisplayLink alloc] initWithTarget:self selectors:@selector(displayLinkDidFire:)];
    [self.displayLink add.toRunLoop:[NSRunLoop mainRunLoop] forMode:NSDefaultRunLoopMode];
}

(void) displayLinkDidFire:

CGPoint contentOffset = self.scrollView.contentOffset;
contentOffset.x += 1;
self.scrollView.contentOffset = contentOffset;

(void) setDelegate:(id<SplashScreenPeopleWallViewDelegate> __nullable)delegate

if ([self.delegate respondsToSelector:@selector(nextImageForSplashScreenPeopleWallView:)]) {
    for (UIImage *imageView in self.imageViews) {
        imageView.image = [self.delegate nextImageForSplashScreenPeopleWallView: imageView];
    }
}

(void) reloadData

if ([self.delegate respondsToSelector:@selector(nextImageForSplashScreenPeopleWallView:)]) {
    for (UIImage *imageView in self.imageViews) {
        imageView.image = [self.delegate nextImageForSplashScreenPeopleWallView: imageView];
    }
}

@end
(void)positionImageViews
    
    CGFloat imageViewWidth = self.bounds.size.width / numberOfColumns;
    for (int column = 0; column < numberOfColumns; column++) {
        CGFloat x = column * imageViewWidth;
        for (int row = 0; row < numberOfRows; row++) {
            CGFloat y = row * imageViewWidth;
            UIImage *imageView = imageViews[0].imageForIndex(column + numberOfColumns + row];
            imageView.frame = CGRectMake(x, y, imageViewWidth, imageViewHeight);
        }
    }

(void)layoutSubviews
    
    CGFloat imageViewWidth = self.bounds.size.width / numberOfColumns;
    self.scrollView.frame = self.bounds;
    self.scrollView.contentSize = CGSizeMake(self.bounds.size.width - imageViewWidth, self.bounds.size.height);
    [self positionImageViews];

    - (void)startAnimation
        
        if ([self displayLink]) {
            self.displayLink = [[CALayer displayLinkWithTarget:self selector:@selector(displayLinkDidFire:)];
            self.displayLink.addTargetRunLoop([NSRunLoop mainRunLoop] forMode:NSDefaultRunLoopMode);
        }

    - (void)displayLinkDidFire:(CALayer displayLink)
        
        CGRect contentOffset = self.scrollView.contentOffset;
        contentOffset.x += self.contentViewOffset.contentInsetOffset;
        self.scrollView.contentOffset = contentOffset;

    - (void)setDelegate:(id<SplashScreenPeopleWallViewDelegate> __nullable)delegate
        
        [self.delegate = delegate];

    - (void)reloadData
        
        if ([self.delegate respondsToSelector:@selector(nextImageForSplashScreenPeopleWallView:)]) {
            for (UIImage *imageView in self.imageViews) {
                imageView.image = [self.delegate nextImageForSplashScreenPeopleWallView:self];
            }
        }

@end
(void) positionImagesViews

CGFloat imageViewWidth = self.bounds.size.width / numberOfColumns;

for (int column = 0; column < numberOfColumns; column++) {
    CGFloat x = column * imageViewWidth;

    for (int row = 0; row < numberOfRows; row++) {
        CGFloat y = row * imageViewWidth;

        UIImage *imageView = [self.imageView objectAtIndex:column + numberOfRows * row];
        imageView.frame = CGRectMake(x, y, imageViewWidth, imageViewHeight);
    }
}

(void) layoutSubviews

CGFloat imageViewWidth = self.bounds.size.width / numberOfColumns;

self.scrollView.frame = self.bounds;
self.scrollView.contentSize = CGSizeMake(self.bounds.size.width - imageViewWidth, self.bounds.size.height);

[self positionImagesViews];

(void) startAnimating

if ([self.displayLink linkWithTarget:self selector:@selector(displayLinkDidFire:)]) {
    [self.displayLink addToRunLoop:NSRunLoop.mainRunLoop] forMode:NSDefaultRunLoopMode;
}

(void) displayLinkDidFire:(CGLDisplayLink *)displayLink

CGPoint contentOffset = self.scrollView.contentOffset;
contentOffset.x -= self.delegate.nextImageForSplashScreenPeopleWallView;
contentOffset.y = contentOffset;

(void) setDelegate:(id<SplashScreenPeopleWallViewDelegate> __nullable)delegate

_delegate = delegate;

[self reloadData];

(void) reloadData

if ([self.delegate respondsToSelector:@selector(nextImageForSplashScreenPeopleWallView:)]) {
    for (UIImage *imageView in self.imageView) {
        imageView.image = @[self.delegate nextImageForSplashScreenPeopleWallView];
    }
}

@end
Demo

Joar Wingfors Developer Tools
## Demo

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Testing in Xcode</td>
<td>Nob Hill</td>
<td>Wednesday 11:00AM</td>
</tr>
<tr>
<td>Continuous Integration and Code Coverage in Xcode</td>
<td>Presidio</td>
<td>Thursday 10:00AM</td>
</tr>
</tbody>
</table>