WWDC 2014 Recap + Swift Introduction

Austin iOS Developers
Anthony Blatner

Jackrabbit Mobile

Agenda

- Recap new announcements
- Review new Swift language highlights
- Swift programming demo

Continuity

- **Handoff --**
  move work, apps, documents, easily between iOS and Mac devices

© 2014 Jackrabbit Mobile

Continuity

- **AirDrop**
  Share files between mobile devices and Macs using AirDrop

- Easily send files to contacts

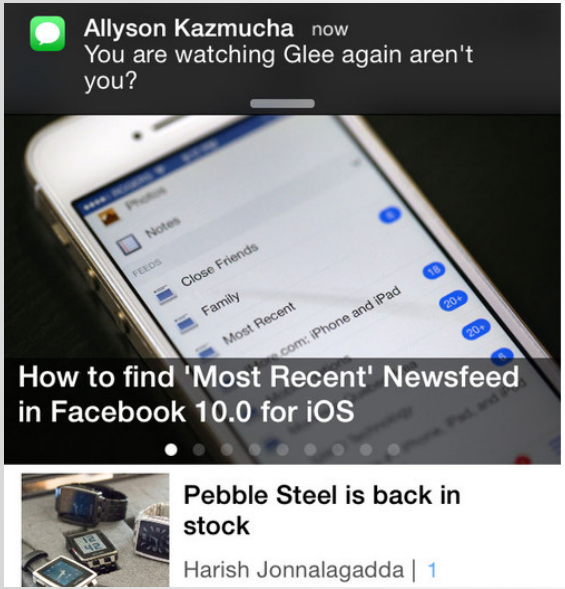
© 2014 Jackrabbit Mobile

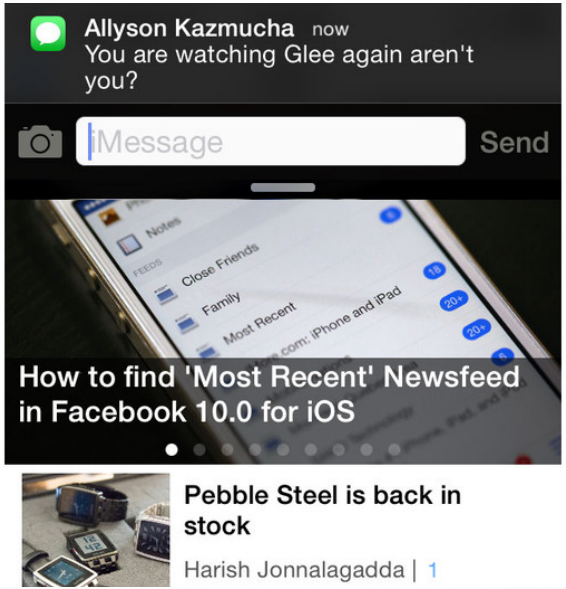
Interactive Notifications

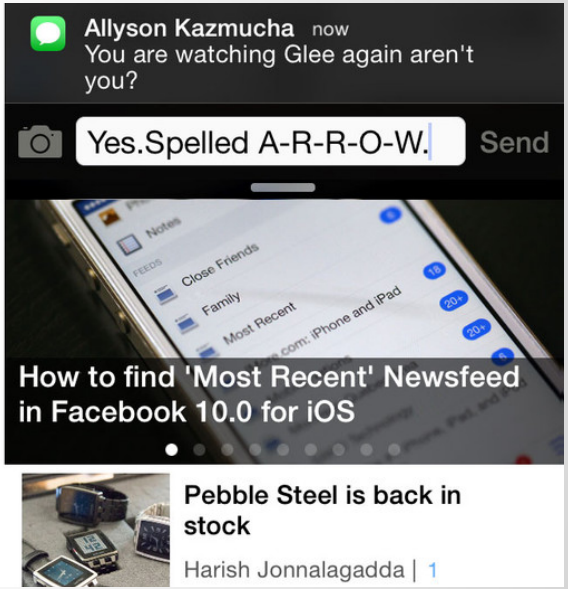
- Respond directly from the Notifications lock screen
- The useful functionality expands beyond Apple's own apps
- Third-party widgets can be added to the Notifications Center now -- interact with widgets without launching the full application

© 2014 Jackrabbit Mobile

Interactive Notifications







© 2014 Jackrabbit Mobile

iCloud Drive

Open documents from one app inside another compatible app. You'll also be able to access files from your Mac that have been saved and synced through iCloud.

You can store and sync files of any type, not just the ones designed specifically for iCloud.

© 2014 Jackrabbit Mobile

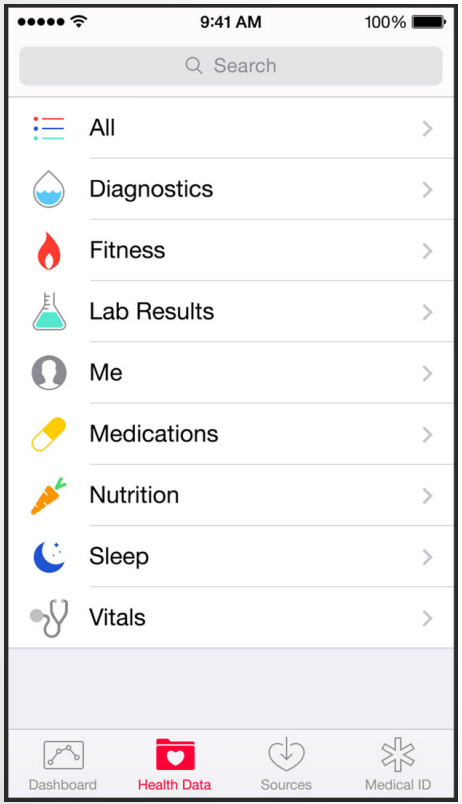
HealthKit

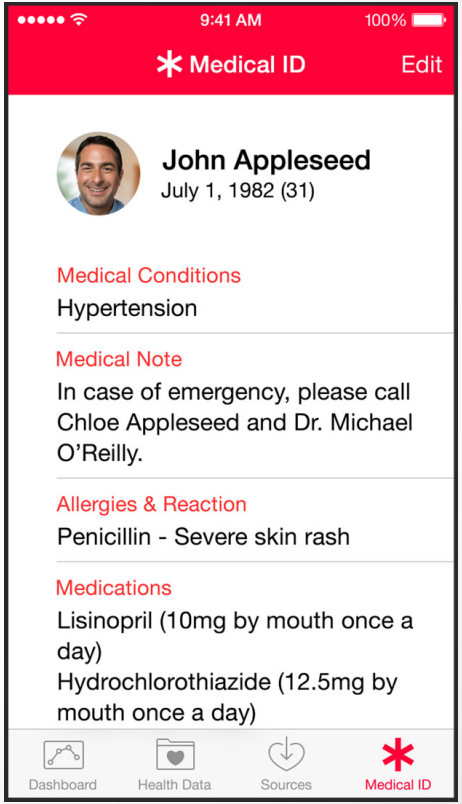
- App for tracking personal health and fitness data
- Monitor important health metrics on a daily basis
- Track vital signs, diet and sleeping habits

© 2014 Jackrabbit Mobile

http://www.jackrabbitmobile.com

HealthKit





© 2014 Jackrabbit Mobile

HomeKit

- Allow iPhones to start controlling smart devices, such as garage-door openers, lights, and security cameras
- Apple will run a certification program for HomeKit
- Apple’s “Internet of Things”

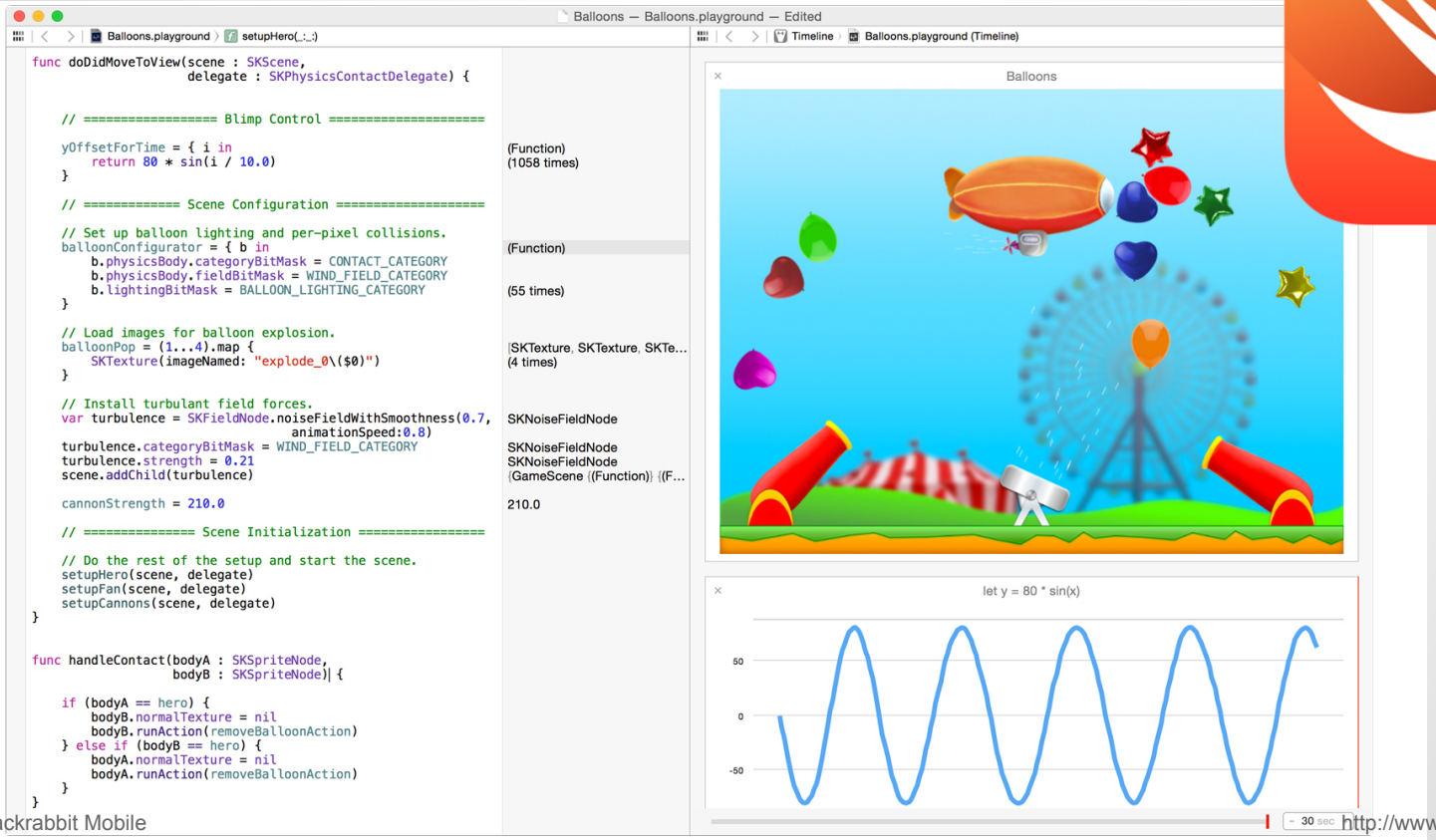
© 2014 Jackrabbit Mobile

HomeKit

- **August**: This Bluetooth-enabled smart lock can be installed around your existing single-cylinder deadbolt setup
- **Philips Hue**: This line of connected bulbs has an app that lets you change colors and control your lights remotely, and set schedules
- **Chamberlain MQ Garage**: Includes Wi Fi hub and door sensor to monitor and control your garage doors.
- **Netatmo Weather Station**: Monitors air quality, CO2 alerts
- **Kwikswet Kevo**: This Bluetooth 4.0-compatible smart lock
- **Withings Smart Baby Monitor**: listen and talk to your baby, play lullabies, or turn on the night light. You can connect via Wi-Fi, Bluetooth, or Ethernet, and monitor noises, motion, temperature, and humidity.
- **Haier**: Smart air conditioner
- **Honeywell**: Like Nest Thermostat.
- **Schlage**: Camelot Touchscreen Deadbolt
- **Cree**: SmartCast system geared toward businesses
- **Osram Sylvania**: Lighting products
- **iDevices**: iGrill Mini Bluetooth-enabled thermometer probe to monitor the temperature of your food.
- **iHome**: Speakers, radios, and alarm clocks
- **SkyBell**: Smart Wi-Fi doorbell. Camera, motion sensor, see who’s at the door and notifications on bell ring.
- **Texas Instruments**, Marvell, and Broadcom

© 2014 Jackrabbit Mobile

Swift



© 2014 Jackrabbit Mobile

Swift

“Swift is an innovative new programming language for Cocoa and Cocoa Touch. Writing code is interactive and fun, the syntax is concise yet expressive, and apps run lightning-fast. Swift is ready for your next iOS and OS X project — or for addition into your current app — because Swift code works side-by-side with Objective-C.”

© 2014 Jackrabbit Mobile

Swift

Swift is designed to provide seamless compatibility with Cocoa and Objective-C. You can use Objective-C APIs (ranging from system frameworks to your own custom code) in Swift, and you can use Swift APIs in Objective-C

Interoperability lets you interface between Swift and Objective-C code, allowing you to use Swift classes in Objective-C and to take advantage of familiar Cocoa classes, patterns, and practices when writing Swift code.

Mix and match allows you to create mixed-language apps containing both Swift and Objective-C files that can communicate with each other.

© 2014 Jackrabbit Mobile

Swift

So, don’t freak out…

Our apps won’t change drastically just yet -- we can keep using Objective-C. Swift will likely be used for shorthand, optimizing performance, and supporting future frameworks.

© 2014 Jackrabbit Mobile

Swift - Features

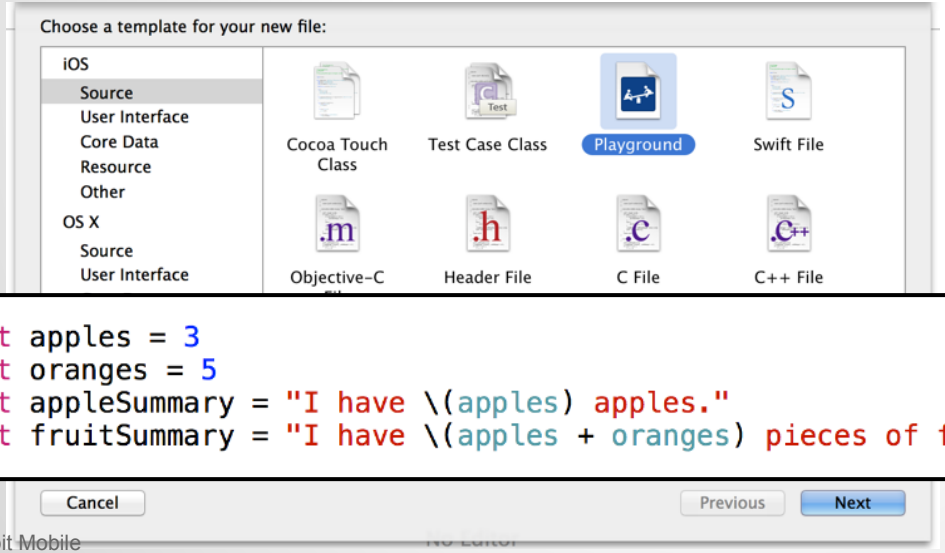
- Closures unified with function pointers
- Tuples and multiple return values
- Generics
- Fast and concise iteration over a range or collection
- Structs that support methods, extensions, protocols.
- Functional programming patterns, e.g.: map and filter
- Inferred types make code cleaner and less prone to mistakes
- Memory is managed automatically (ARC)
- You don’t need semi-colons
- All the information about a particular class resides in a single .swift file -- no longer separate header & implementation files.

© 2014 Jackrabbit Mobile

Swift - Interactive Playgrounds

“Playgrounds make writing Swift code incredibly simple and fun”

Type a line of code and the result appears immediately.



```
11 let apples = 3
12 let oranges = 5
13 let appleSummary = "I have \(apples) apples."
14 let fruitSummary = "I have \(apples + oranges) pieces of fruit."
```

```
3
5
"I have 3 apples."
"I have 8 pieces of fruit."
```

© 2014 Jackrabbit Mobile

Swift - Values

Three-character keywords define a variable (**var**) or constant (**let**).

```
var myVariable = 42
myVariable = 50
let myConstant = 42
```

Compiler inferred types

```
let implicitInteger = 70
let implicitDouble = 70.0
let explicitDouble: Double = 70
```

Simple ways to include values in strings

```
let apples = 3
let appleSummary = "I have \(apples) apples."
```

Optionals -- a value either contains a value or contains nil -- question marks (?) mark optionals.

```
var optionalName: String? = "John Appleseed"
```

© 2014 Jackrabbit Mobile

Swift - Functions

Declaration and invocation

```
func greet(name: String, day: String) -> String {
   return "Hello \(name), today is \(day)."
}
greet("Bob", "Tuesday")
```

Multiple Return Values

```
func getGasPrices() -> (Double, Double, Double) {
   return (3.59, 3.69, 3.79)
}
getGasPrices()
```

© 2014 Jackrabbit Mobile

Swift vs Objective-C

Objective-C

```
UITableView *myTableView = [[UITableView alloc] initWithFrame:CGRectZero style:
UITableViewStyleGrouped];
```

Swift

```
let myTableView: UITableView = UITableView(frame: CGRectZero, style: .Grouped)
```

Objective C

```
UIColor *color = [UIColor colorWithRed:0.5 green:0.0 blue:0.5 alpha:1.0];
```

Swift

```
let color = UIColor(red: 0.5, green: 0.0, blue: 0.5, alpha: 1.0)
```

© 2014 Jackrabbit Mobile

Swift vs Objective-C

Objective-C

```
[myTableView insertSubview:mySubview atIndex:2];
```

Swift

```
myTableView.insertSubview(mySubview, atIndex: 2)
```

© 2014 Jackrabbit Mobile

Swift - Code Demo

Jackrabbit Mobile