

Educate | Engage | Evolve

Intro to iOS Development

Jared Alexander

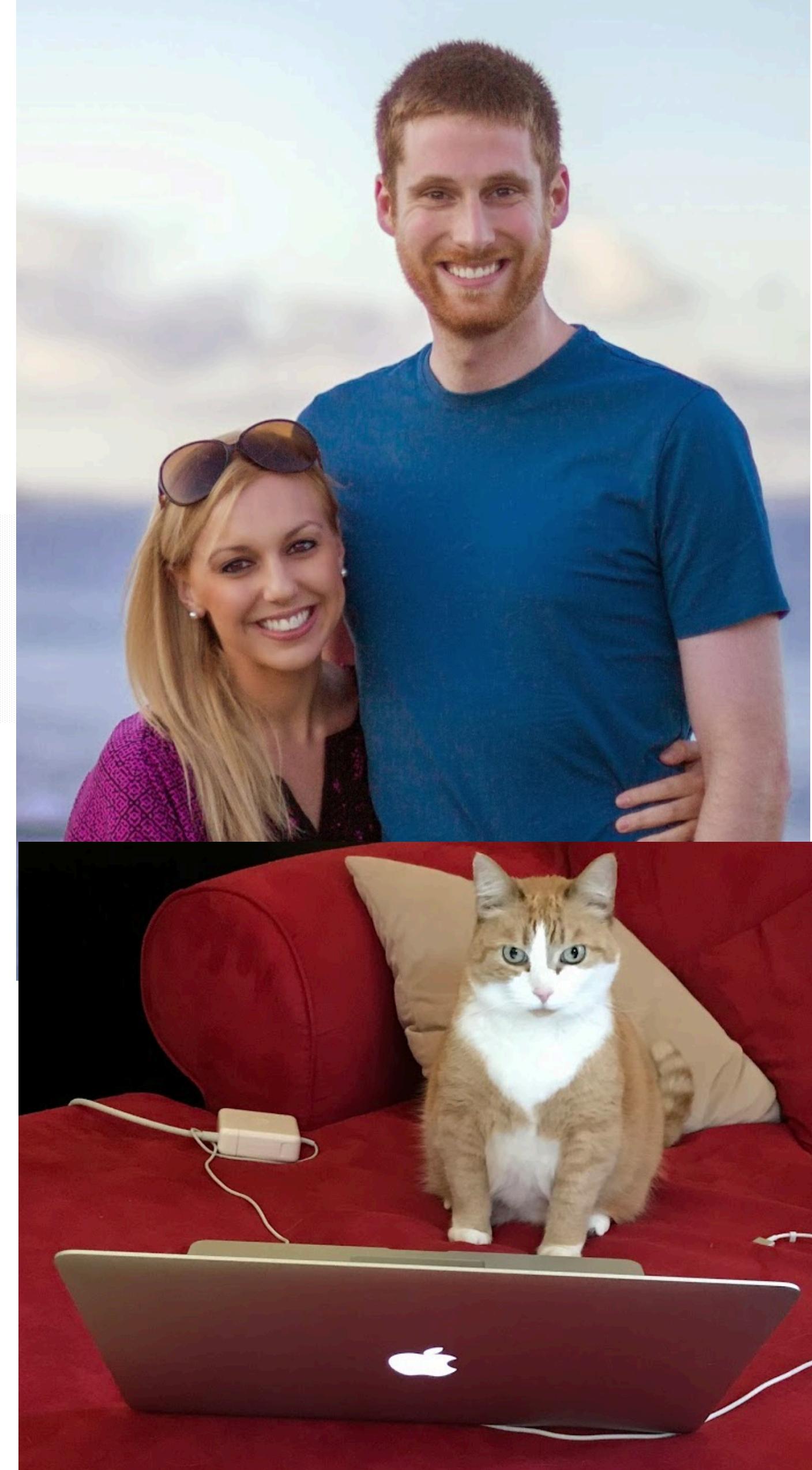
Presenter Bio



**Jared
Alexander**

Lead Software Engineer

- Dean of Mobile, Capital One
- Adjunct Professor, GWU
- Co-Founder, Tysons Apps
- 10+ years software dev experience
- 6+ years of mobile dev experience
- Hobbies:
 - app dev!
 - teaching
 - playing + watching basketball
 - playing video games
 - Netflix binging
 - traveling the world
 - spending time with my wife, Rose, and cat, Wahoo



Let's take a little quiz....

Learning Objectives

1

Understand the
current mobile
landscape

2

Learn about iOS and
Xcode

3

Gain experience
building an app!



1

Understand the current mobile landscape

Cellular technology throughout the years...



“
*"An iPod, a phone and an Internet
communicator... "These are not separate
devices. This is one device. And we are calling it
iPhone.*

— STEVE JOBS, CEO OF APPLE

iPhone

Believe it or not, the original iPhone, launched in June 2007, had no App Store. Steve Jobs originally encouraged developers to focus on building great web experiences. Developers clamored for an iPhone Software Development Kit (SDK), which Apple eventually delivered, and the App Store was launched in July 2008.



“Holy crap, I guess we’re not going to ship that phone.”

— ANDY RUBIN, DIRECTOR OF ANDROID

Speaking of Android....

Check out “Intro to Android” here at WIT,
taught by my colleague Shannon!



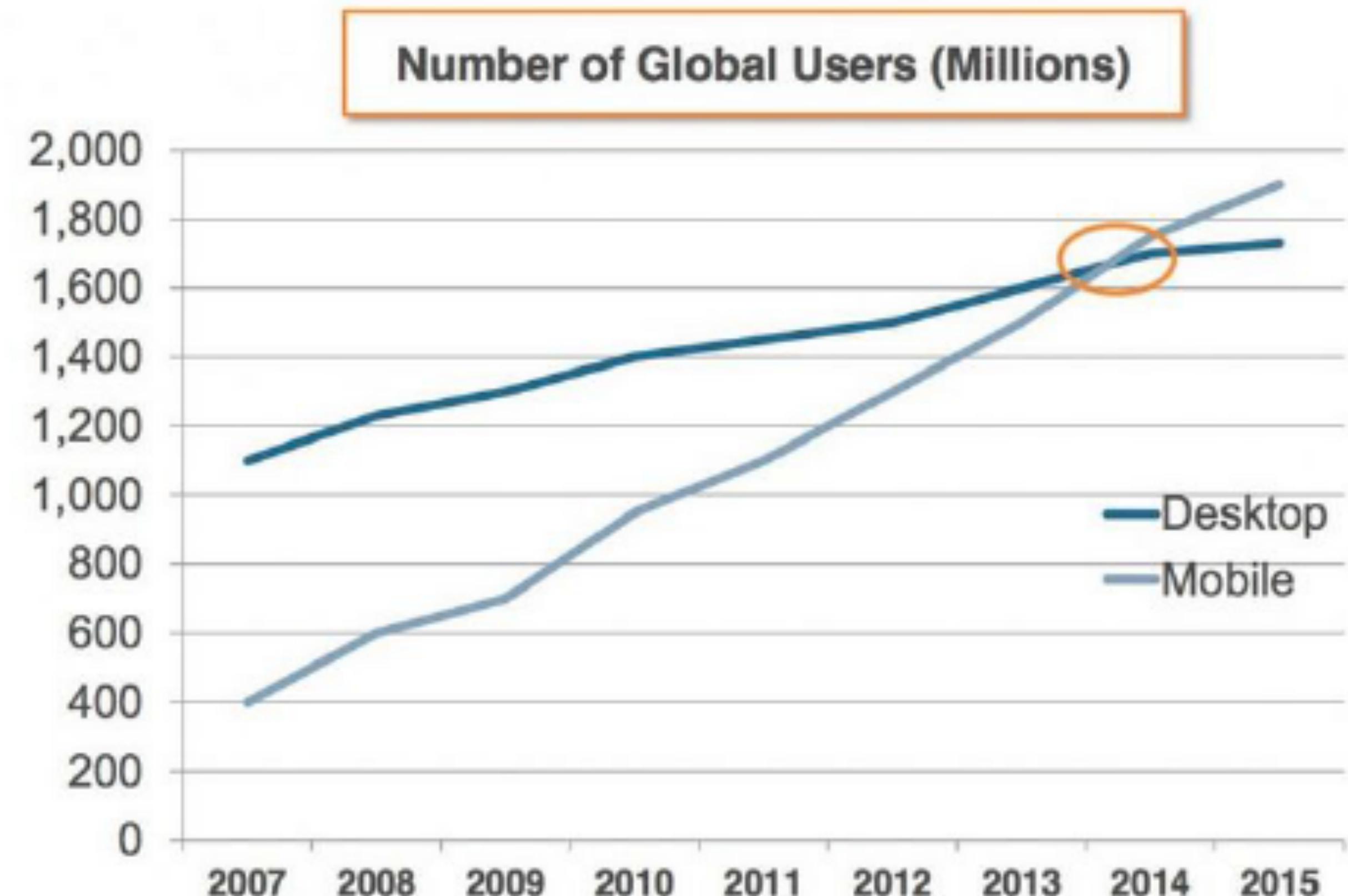
Shannon
Lockie-Krahe

Educate | Engage | Evolve

Why Mobile?

Mobile is how people connect

It's become the primary way people use the internet



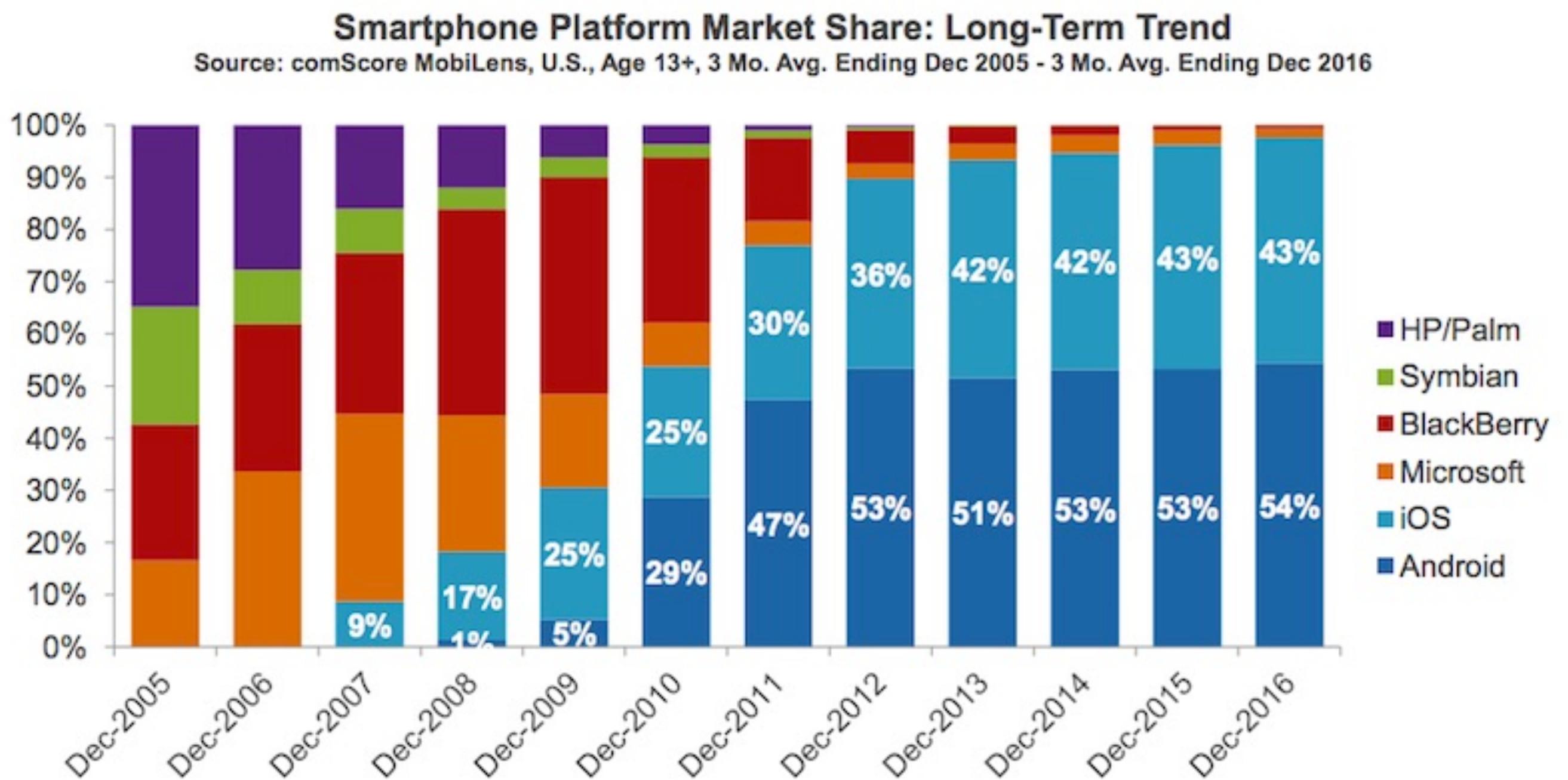
© comScore, Inc. Proprietary and Confidential.

24

Source: Morgan Stanley Research

Mobile app development is a multi-billion dollar industry

- Apple's App Store has **2.2+ million apps**
- Google Play has **2.8+ million apps**
- Over **2.3 billion projected smartphone users** worldwide in 2017



Mobile is special

- It's an interface to everything, available on-demand, always connected, **in your pocket**
- It's hyper-contextual, it can know
 - where you are (GPS / cell tower-based location)
 - what you're doing (e.g. motion sensors to detect walking/biking/driving/etc.)
 - what you're seeing and hearing (camera/microphone)
 - what devices are around you and interact with them (Bluetooth/Wi-Fi/NFC)
- It's the way to consume other important advances in tech
 - AI, Machine Learning – e.g. personal assistants
 - Computer Vision – e.g. landmark identification
 - Augmented Reality – e.g. real-time foreign sign translation
 - Cloud – e.g. browsing the web

2

Learn about iOS and Xcode

A little history

iOS made it's debut in June 2007

See anything missing?



A little history

The iOS SDK and App Store didn't arrive until a year later in July 2008



iOS development in 2008

One screen resolution...until 2010 (iPhone 4)

No multi-tasking...until 2010 (iOS 4)

iOS development == iPhone development...until 2010 (iPad)

Code written in Objective-C...until 2014 (Swift & iOS 8)



iOS development today

9 resolutions (across iPhone, iPad, & Apple Watch)

Multi-tasking

Code written in Swift

iOS development != iPhone development

iPhone, iPad, Apple Watch (watchOS), Apple TV (tvOS), CarPlay





iOS 11 adoption

Sep 16th, 2017

- Oct 19th, 2017

DONE

Hour

Day

Week



mixpanel

THIS REPORT WAS GENERATED FROM 332,191,817,161 RECORDS. • TIME/DATE IN US/PACIFIC

Swift is the language used to build iOS apps

- Released in June 2014 at WWDC
- Programming language created by Apple for iOS & Mac development (also used for watchOS & tvOS)
- Takes language ideas of Objective-C, Ruby, Python, C#, and more!
- Benefits over its precursor, Objective-C
 - simpler syntax
 - more concise
 - safer
 - more fun (Playgrounds!)



Swift is simpler and more concise



Objective-C

```
NSString *str = @"hello,";  
str = [str stringByAppendingString:@" world"];
```

Swift

```
var str = "hello,"  
str += " world"
```

Swift is simpler and more concise



Objective-C

```
NSMutableArray * array = [NSMutableArray new];  
[array addObject:@1];
```

Swift

```
var array = [Int]()  
array.append(1)
```

Swift is simpler and more concise



Swift is safer



Objective-C

```
int i = 5;
if(i < 10)
    NSLog(@"i is less than 10");
    [self doSomething];
```

Swift

```
var i = 5
if (i < 10) {
    print("i is less than 10")
    doSomething()
}
```

```
if(i < 10)          ① Expected '{' after 'if' condition
    print("i is less than 10")
    doSomething()
```

Swift is safer - Optionals

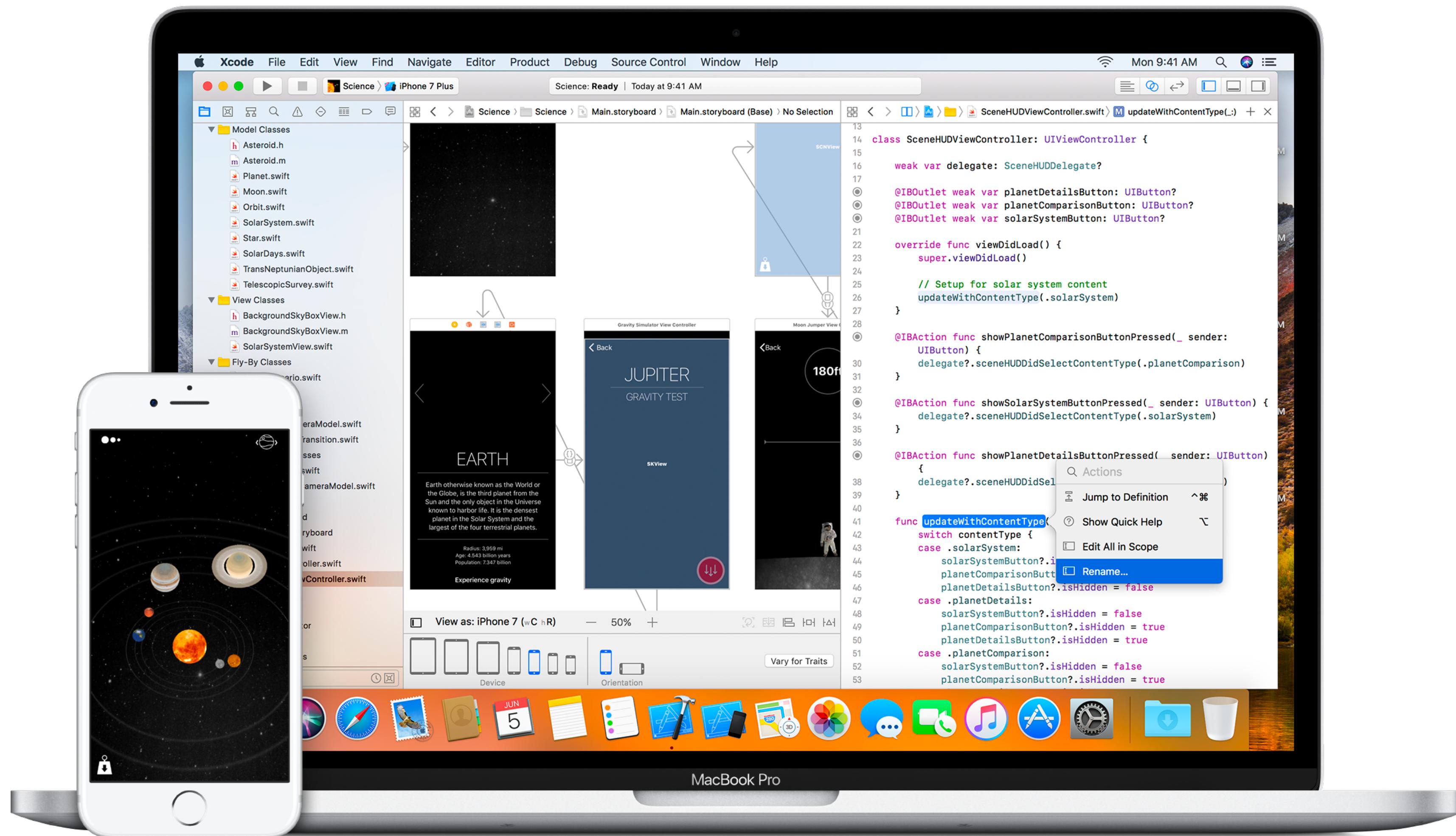
- Optionals handle the absence of a value
- An optional can represent two possibilities



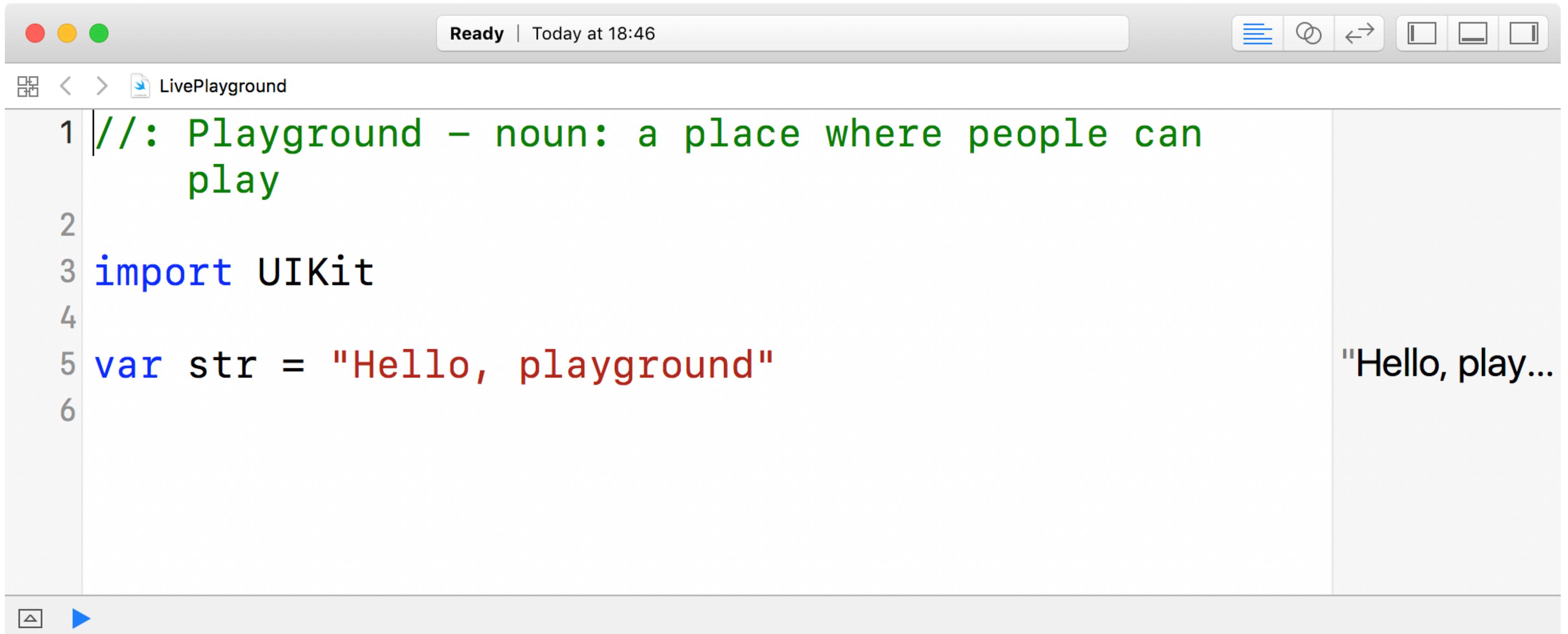
```
func indexOfNumber(numbers: [Int], numberToFind: Int) -> Int? {  
    for (index, number) in numbers.enumerated() {  
        if number == numberToFind {  
            return index  
        }  
    }  
    return nil  
}
```

```
func indexOfNumber(numbers: [Int], numberToFind: Int) -> Int {  
    for (index, number) in numbers.enumerated() {  
        if number == numberToFind {  
            return index  
        }  
    }  
❶    return nil  Compile-time error! Function returns an Int (non-optional)  
}
```

Xcode is the tool used to build iOS apps



Playgrounds are a feature of Xcode



The screenshot shows the Xcode playground interface. The title bar indicates "Ready | Today at 18:46". The file tab shows "LivePlayground". The code editor contains the following Swift code:

```
1 //: Playground - noun: a place where people can
  play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
```

The variable declaration "var str = "Hello, playground"" is followed by its value "Hello, play..." on the right side of the editor.

Let's get some practice writing code in Swift



Let's meet our iOS TA's from Capital One!



Christian Flores



Kaitlyn Landmesser



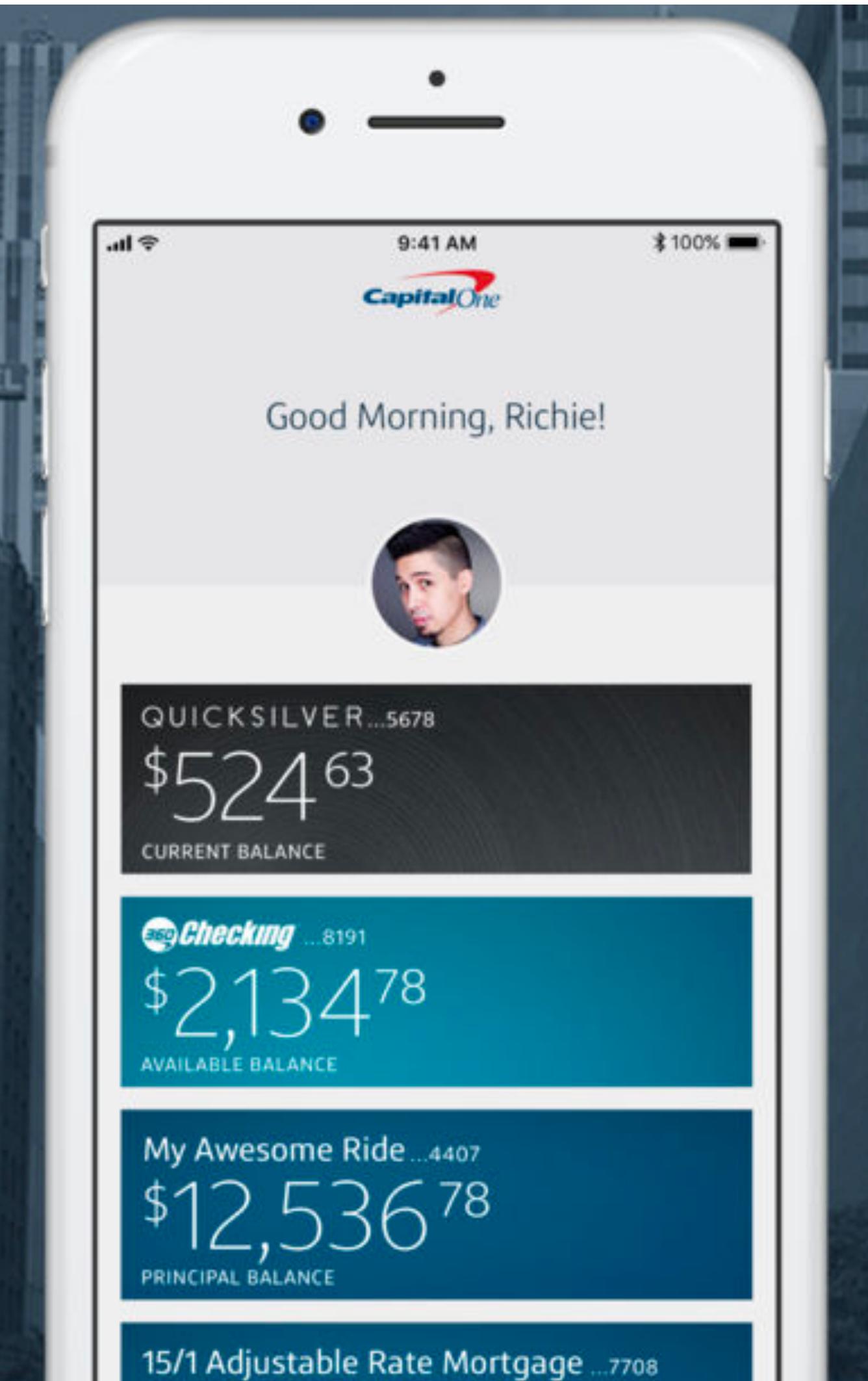
Jason Wang

<Hands-on coding activity begins>

iOS uses Model-View-Controller

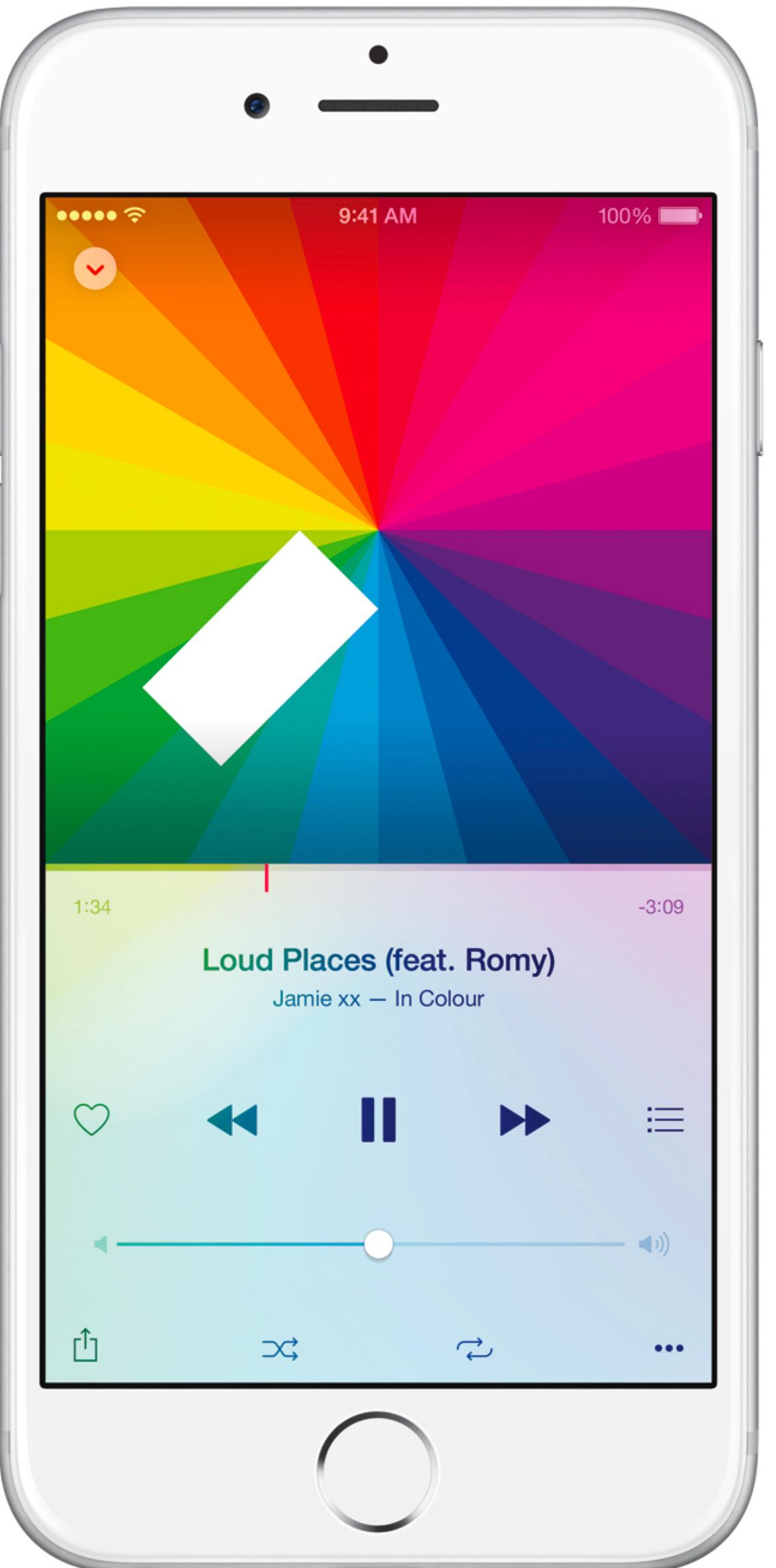
- **Models** encapsulate the data specific to an app
- **Views** are parts of the app that users can see
- **Controllers** contain the “glue” code that defines the business logic for an app

iOS uses Model-View-Controller



Views

- iOS provides many built-in views
- Views can be customized to fit the look and feel of your app
- Custom views can be created, but it's not always a good idea to reinvent the wheel



Label - A variably sized amount of static text.

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text - Displays editable text and sends an action message to a target object when Return is tapped.

Slider - Displays a continuous range of values and allows the selection of a single value.

Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle...

Activity Indicator View - Provides feedback on the progress of a task or process of unknown duration.

Progress View - Depicts the progress of a task over time.

Page Control - Displays a dot for each open page in an application and supports sequential navigation thro...

Stepper - Provides a user interface for incrementing or decrementing a

iOS uses Model-View-Controller

```
struct CreditCard {  
    var acctNumber:Int  
    var transactions:[Transaction]  
}
```

```
struct Transaction {  
    var name:String  
    var amount:Float  
    var date:NSDate  
}
```

```
func calculateBalance() -> Float {  
    var balance:Float = 0.0  
    for transaction in transactions{  
        balance = balance + transaction.amount  
    }  
  
    return balance  
}  
  
func payBillButtonPressed() {  
    //TODO: business logic for paying a bill  
}
```

View Controllers

- Each screen is actually an instance of a **UIViewController**
- Each view controller consists of 2 components

- User interface
- Business logic written in Swift

```
import UIKit

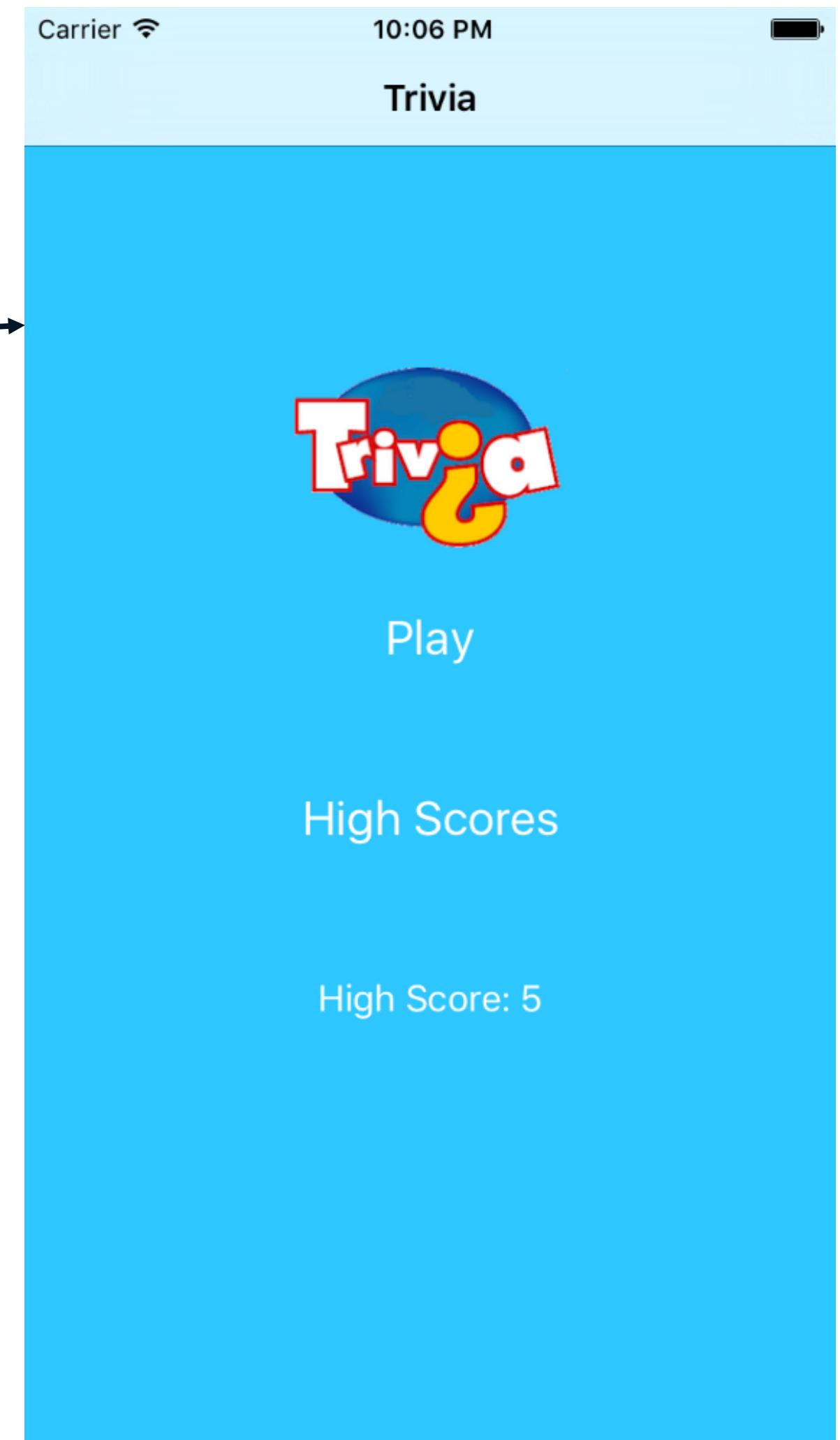
class MenuViewController: UIViewController {

    @IBOutlet weak var highScoreLabel: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
    }

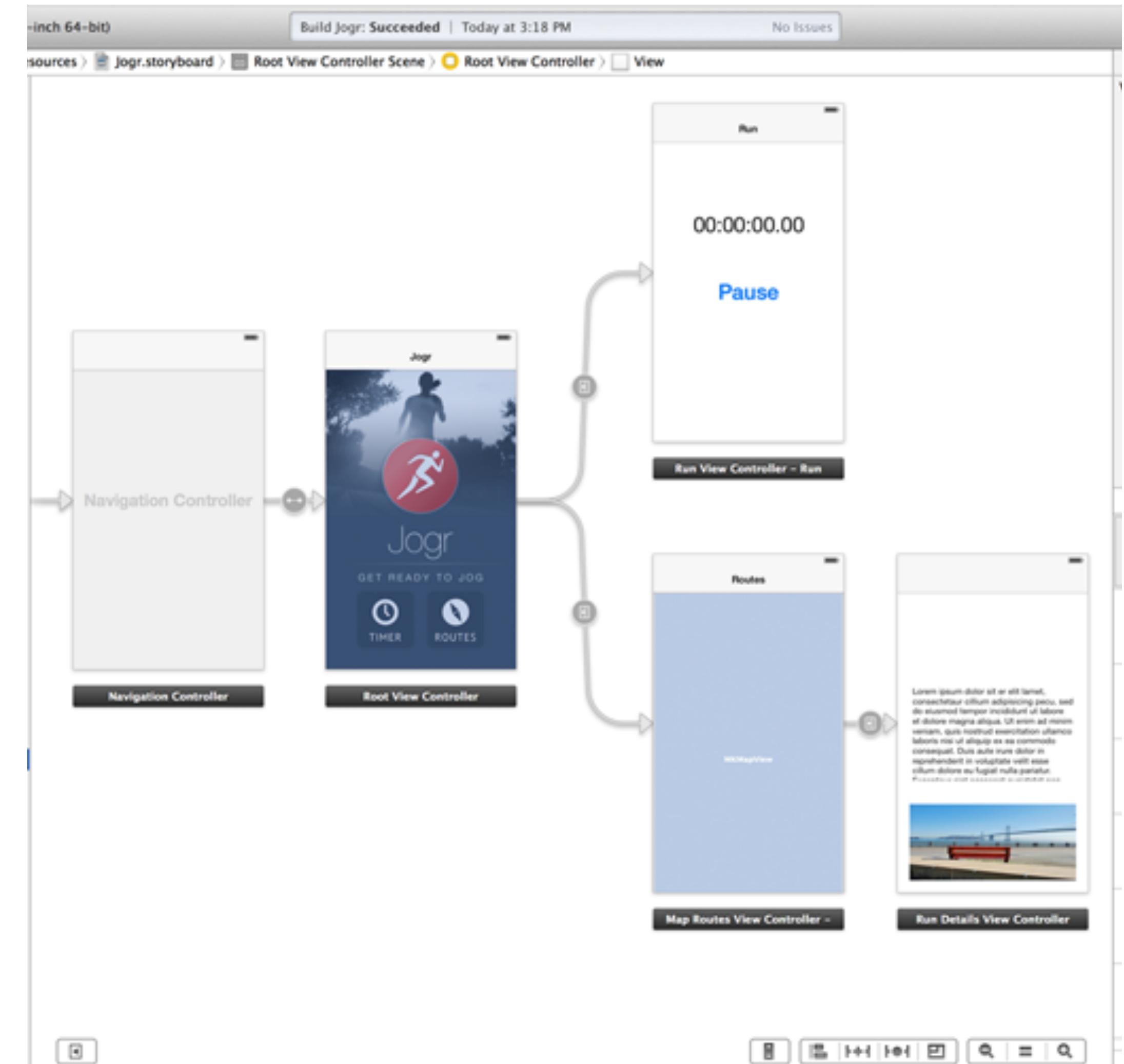
    override func viewWillAppear(animated: Bool) {
        super.viewWillAppear(animated)

        highScoreLabel.text = "High Score: \(Utilities.fetchHighScore())"
    }
}
```



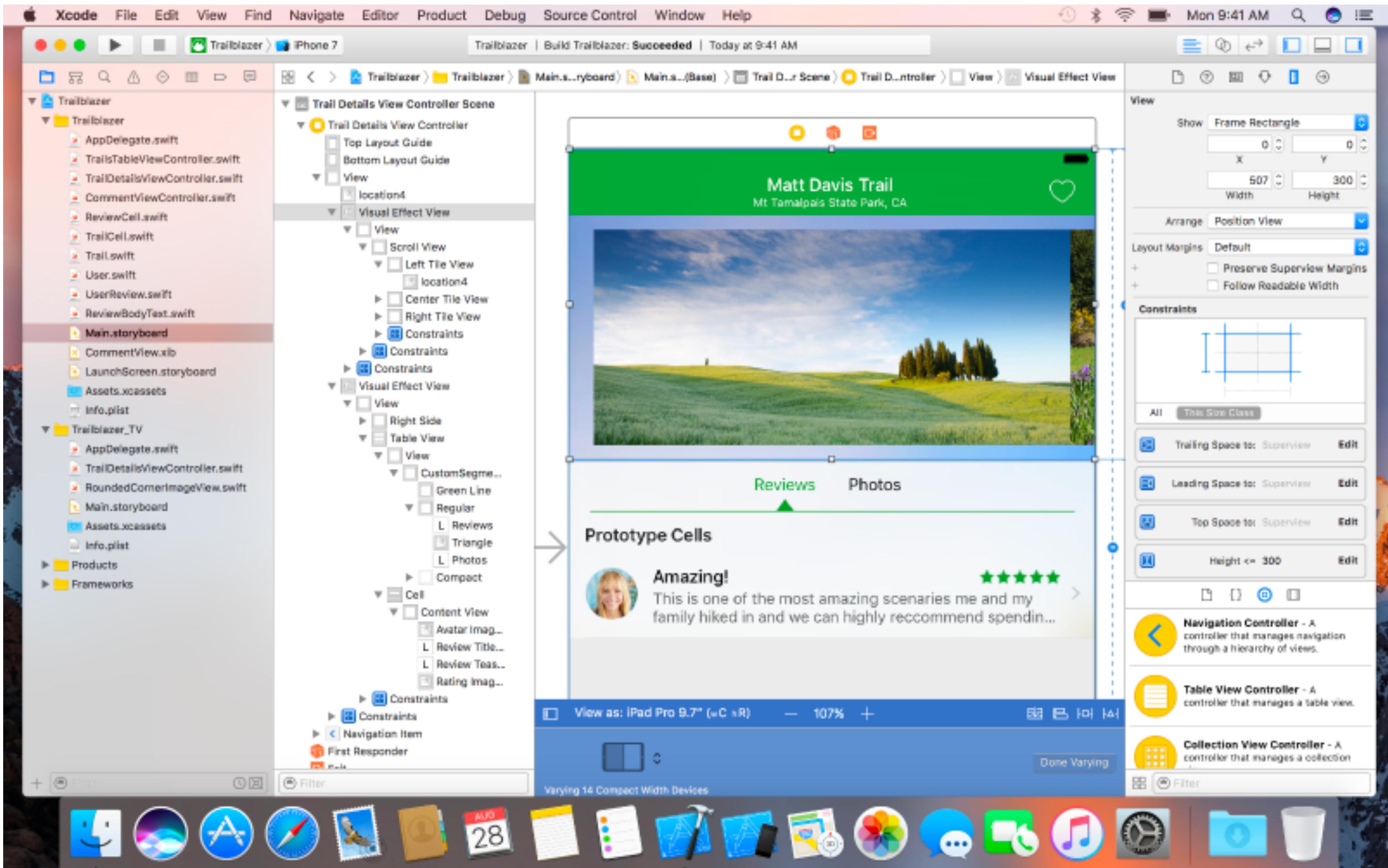
Storyboards

- Storyboards allow us to graphically lay out our app's view controllers and connections between them



Interface Builder

- Used to design and layout views on each screen
- Drag and drop user interface components, style them, resize them
- Set constraints on how views should be laid out on the screen (important to support multiple screen sizes!)



3

Gain experience building a simple
app!

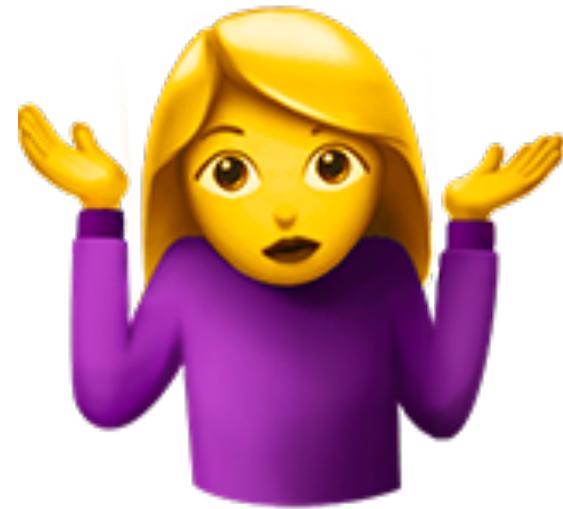
From: Jared Alexander <jared.marc.alexander@gmail.com>
Date: Sun, Oct 30, 2011 at 2:36 AM
Subject: DONE! Introducing JCalc, built from scratch. My first iPhone app!
To: Rosey Blais <rosey.blais@gmail.com>, Home <home@gmail.com>

My first iPhone app - JCalc

I built this in 2011 as part of "Handheld Device Software Development" course in grad school

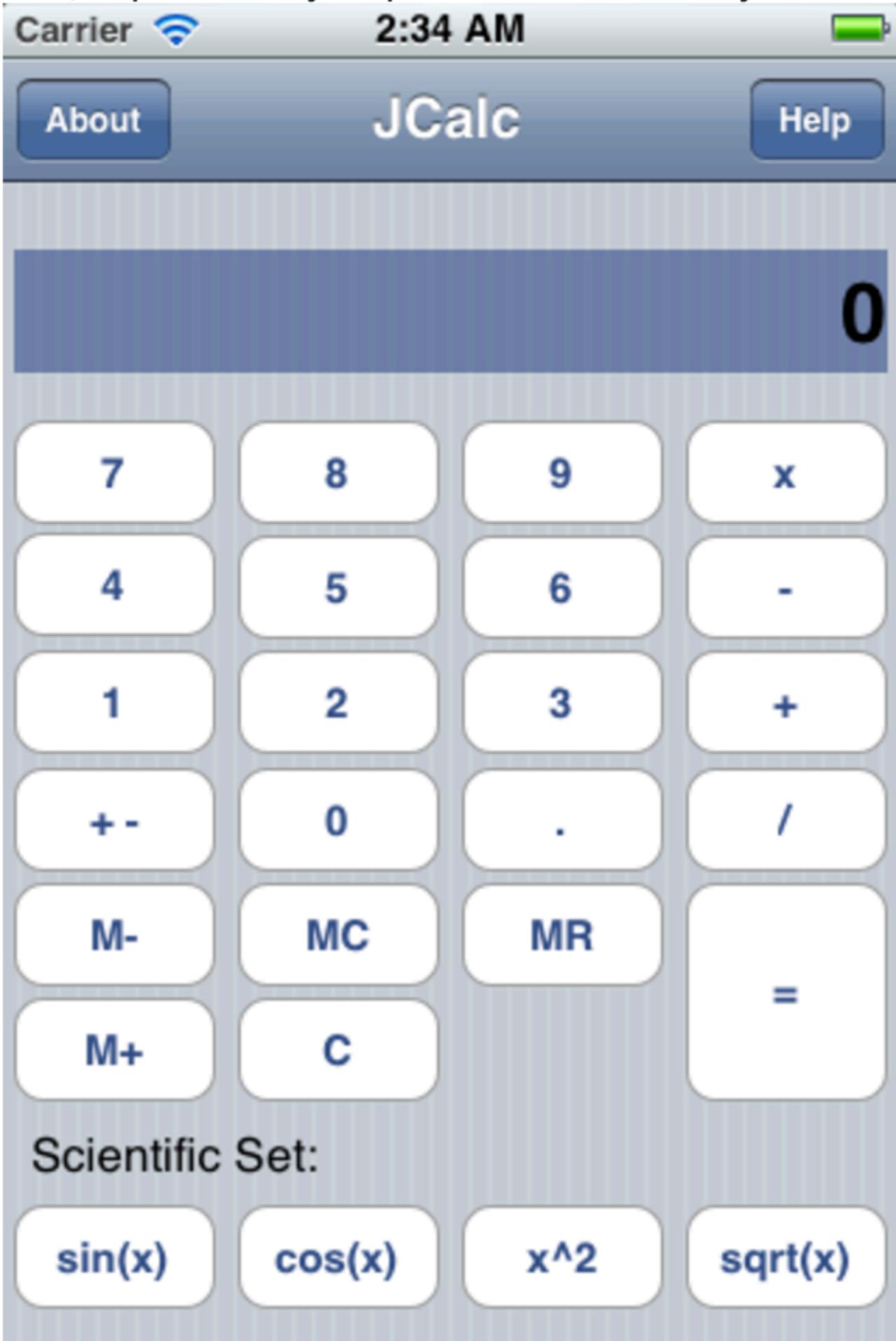
It's trickier to build than you might think!

Yes, iOS already has a calculator built in...



This is an assignment I had to do for class. Check it out:

Dad, I'll put this on your phone next time I see you.



Now let's build your first iOS app!



Xcode 9



Swift 4

<Hands-on coding activity begins>

So where do you go from here....?

Capital One Resources

- Tech College Hive – join the Mobile Gathering!

Free Public Resources

- [Intro to App Development with Swift](#) – Apple's official book (includes programming basics)
- [App Development with Swift](#) – part 2 of the above book
- [Free Stanford Course!](#) (will probably be updated to iOS 11 by January or so)
- [Ray Wenderlich](#) – great tutorials!