SUMMER OF TECH

MOBILE 101

WHO WE ARE

Kate

Android Dev @ Paperkite
kate@paperkite.co.nz
twitter: @Mistyepd

Farlei

iOS Dev @ Paperkite
farlei@paperkite.co.nz
twitter: @farlei

PaperKite
http://paperkite.co.nz

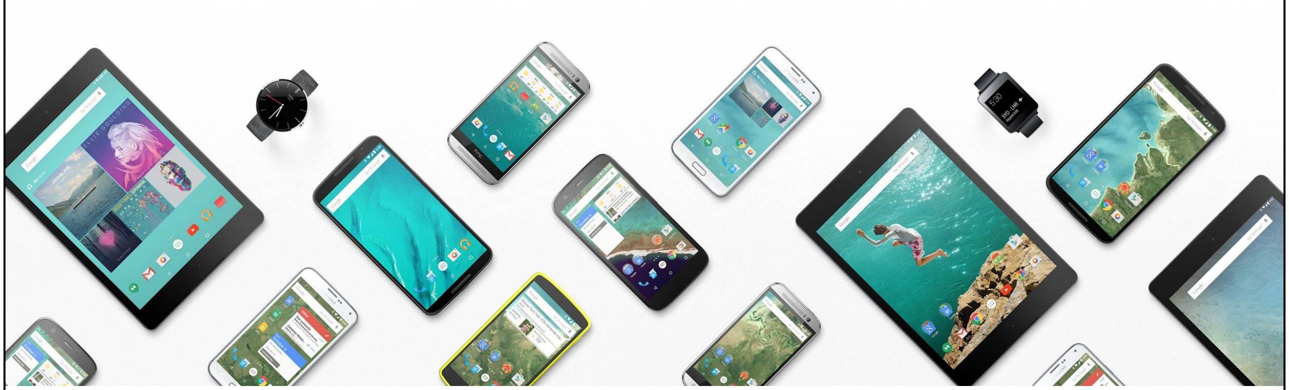
Farlei Heinen
iOS Dev @ Paperkite
farlei@paperkite.co.nz
twitter: @farlei

Kate
Android Dev @ Paperkite
kate@paperkite.co.nz
twitter: @Mistyepd

MOBILE PLATFORMS

MOBILE PLATFORMS

ANDROID



Developed by a company called Android Inc, that Google brought in 2005.

Google released in 2007, open source base, but most phones have proprietary software on top.

Programmed in Java and xml, though Swift was recently ported.

Also includes Android Wear, TV, and Auto

Strong points:

- Still biggest mobile OS
- Less barriers to entry
- Many options for customisation
- Range of devices

IOS

Proprietary operating system (using open-source components), running exclusively in Apple hardware.

The main programming language is Swift, but Objective-C still have full support from Apple.

Launch: 2007 - SDK in 2008

Strong points:

- lower fragmentation
- long upgrade cycle and more up-to-date installed base (79% iOS 9, iPhone 4S (from 2011) still get all updates)
- centralised application distribution ecosystem
- rigorous app review system

WINDOWS 10 MOBILE

One of the “editions” of the new Microsoft Windows 10 platform.

Successor to Windows Phone 8.1

Main language: C# (.NET Framework 4.6)

MULTI PLATFORM

- Xamarin
- PhoneGap
- Unity3D

Xamarin: MonoTouch based framework (C#), recently acquired by Microsoft

PhoneGap: Apache Cordova based framework (HTML, CSS & JS), it's an Adobe company

Unity3D: Powerful Game Engine that supports many mobile platforms - the main language is C# (MonoTouch)

Only one code base for many platforms.

Lower performance compared with the native alternatives.

DEVICE FRAGMENTATION

When developing apps for mobile devices we need to account for fragmentation. Devices can have many screen sizes and ratios, different capabilities, memory sizes and OS versions. Choosing the right one is tricky and depends on many factors:

- target audience;
- type of application;
- target price;
- available time and resources to develop the application.

DEVICE FRAGMENTATION

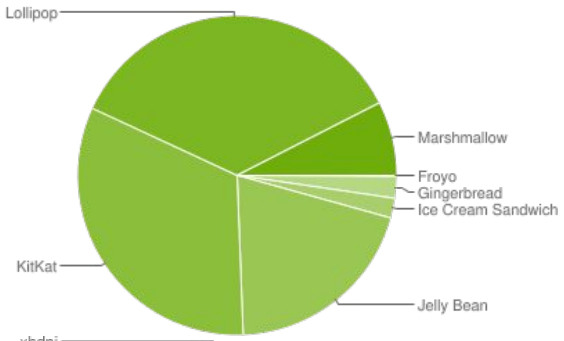
CASES

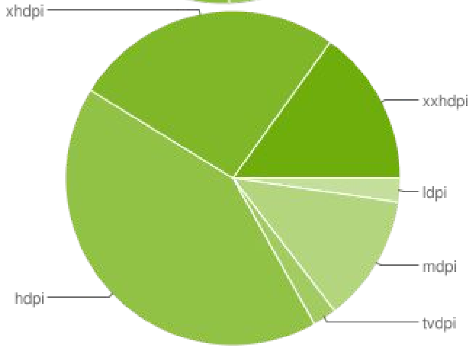
- AM/PM Scratch Power (USA)
  - iOS: 40%
  - Android: 60%
  - source: google analytics, 03/2016 (123K users)
- All Blacks (NZ)
  - iOS: 67%
  - Android 33%
  - source: google analytics, 03/2016 (21K users)

DEVICE FRAGMENTATION

ANDROID

- Version
  - 7.5% 6.0 Marshmallow
  - 19.4% 5.1 Lollipop
  - 16.2% 5.0 Lollipop
  - 32.5% 4.4 KitKat
  - 24.4% > 4.4
- Screen Size
  - 6 buckets
  - Thousands of different sizes





IOS

- iPhone
  - 4s - 320 x 480
  - 5, SE - 320 x 568
  - 6 - 375 x 667
  - 6Plus - 414 x 736
- iPad
  - Mini, Air - 768 x 1024
  - Pro - 1024 x 1366
- Watch
  - 272 x 340
  - 312 x 390
- iOS installed base
  - iOS 9: 79%
  - iOS 8: 16%
  - Earlier: 5%

source: Apple, 07/03/2016

PS:
- Showing current supported devices only
- iOS resolutions are represented in logical points

more info: http://iosres.com
install base: https://developer.apple.com/support/app-store/

DEVELOPER LICENSES

ANDROID

- US$25 Developer Licence
- That's it
- Google payment merchant account
  - If you want to sell priced apps or in app purchases

It's not required to have a licence to build the app, only to put it on the Play Store

You can upload free apps without the merchant account, but will need one if it's making money

More information: https://developer.android.com/distribute/googleplay/start.html

IOS

- Types
  - Free
  - Individual - US$ 99/pa
  - Organisation - US$ 99/pa
  - Enterprise - US$ 299/pa

Free: developer account allows access for all documentation and resources, including running the application on developer's device.

Individual: allows a single developer to submit the application to the Apple Store

Organisation: same as the individual account, but enable multiple developers using the same account using a team management system

Enterprise: targeted for big organisations that want to distribute applications internally

more info: https://developer.apple.com/support/compare-memberships/

OTHER PLATFORMS

- Windows
  - individual NZ$ 24 , company NZ$ 140 (one-time fee)
- Xamarin
  - free (custom quotes for Enterprise level)
- PhoneGap
  - free
- Unity3D
  - personal FREE, professional from US$ 75/mo

- Windows
more info: https://developer.microsoft.com/en-us/windows/programs/faq

- Xamarin
free after Microsoft acquisition
you still need the main platform licenses to publish (iOS, Android, Windows)
more info: https://store.xamarin.com

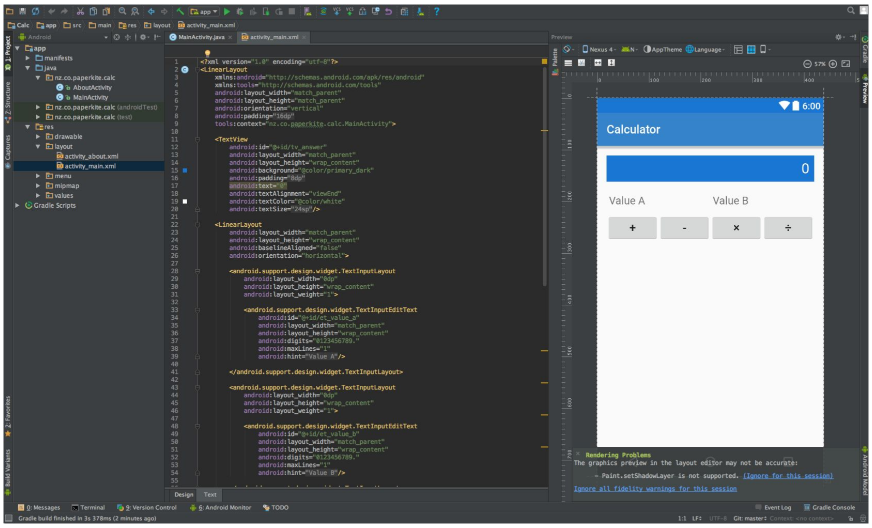
- Adobe PhoneGap
paid if using advanced features in PhoneGap Build Cloud service
you still need the main platform licenses to publish (iOS, Android, Windows)
more info: http://docs.phonegap.com/getting-started/5-going-further/

- Unity
* the free version adds a splash screen to your app (and don't have team and analytics features)

* Unity Professional customers who earned/received more than $100,000 in revenue/funding in the previous fiscal year must purchase iOS Pro and/or Android Pro deployment add-ons to deploy to these platforms.
* you still need the main platform licenses to publish (iOS, Android, Windows)
more info: https://unity3d.com/get-unity

DEVELOPMENT TOOLCHAIN

ANDROID STUDIO



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context="nz.co.paperkite.calc.MainActivity">

    <TextView
        android:id="@+id/tv_answer"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/primary_dark"
        android:padding="8dp"
        android:text="0"
        android:textAlignment="viewEnd"
        android:textColor="@color/white"
        android:textSize="24sp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:baselineAligned="false"
        android:orientation="horizontal">

        <android.support.design.widget.TextInputLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1">

            <android.support.design.widget.TextInputEditText
                android:id="@+id/et_value_a"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:digits="0123456789."
                android:maxLines="1"
                android:hint="Value A"/>

        </android.support.design.widget.TextInputLayout>

        <android.support.design.widget.TextInputLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1">

            <android.support.design.widget.TextInputEditText
                android:id="@+id/et_value_b"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:digits="0123456789."
                android:maxLines="1"
                android:hint="Value B"/>
```

```
Rendering Problems
The graphics preview in the layout editor may not be accurate:
  - Paint.setShadowLayer is not supported. (Ignore for this session)
Ignore all fidelity warnings for this session
```

```
Gradle build finished in 3s 378ms (2 minutes ago)
```

Free IDE based off IntelliJ, works on Windows, Linux and OSX.

Has a built in SDK(Software Dev Kit) manager to keep you up to date with the latest google provided libraries and tools

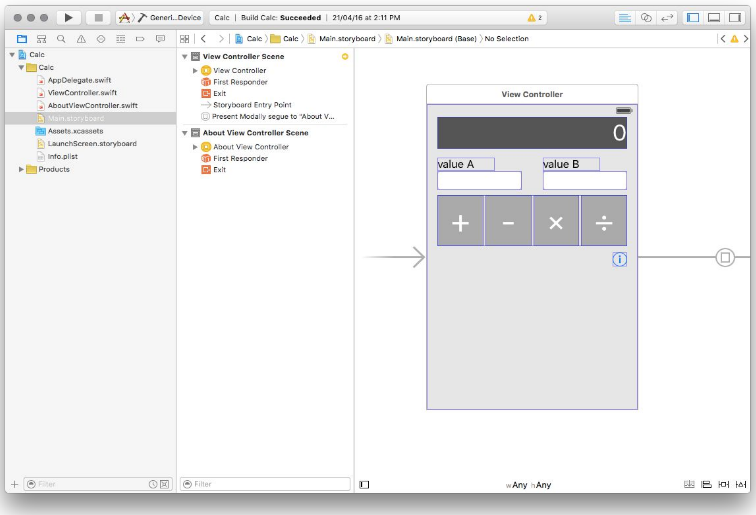
Used to use Eclipse with a plugin, android studio is a real upgrade

ANDROID RESOURCES

- Android developer docs
- Stack Overflow
- Android Arsenal for libraries https://android-arsenal.com
- Udacity for starting out learning
- MaterialUp for inspiration http://www.materialup.com
- Material Design docs

There's an Android NZ slack team and a wellington meetup here: http://www.meetup.com/Wellington-Android-Development-Meetup/

XCODE



Xcode is the IDE to develop for Apple iOS (and Mac OS X). It integrates all the tools and libraries needed to develop for iOS including an Interface Builder, debugger, performance analysis and deployment tools.

It also have a realtime interactive Swift language interpreter called Playgrounds.

You can download the latest version from the Mac App Store (Mac OS X only).

IOS ONLINE RESOURCES

- CocoaPods
- Apple Docs / Forums
- Stack overflow
- Websites:
  - Ray Wenderlich: https://www.raywenderlich.com/category/ios
  - Cocoa Controls: https://www.cocoacontrols.com

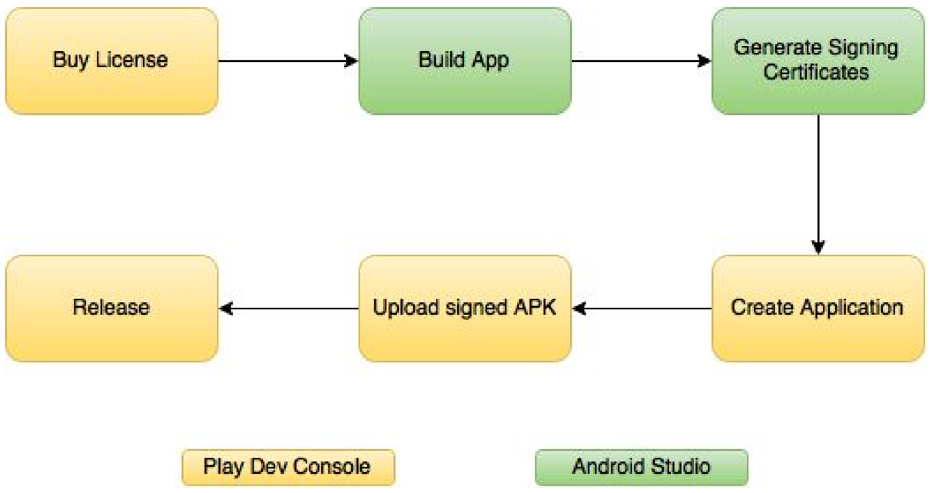
Cocoapods: library dependency manager for Swift and Objective-C

Apple Forums:
https://forums.developer.apple.com

There's an iOS Wellington meetup here: http://www.meetup.com/Cocoaheads-Wellington/

APP DEPLOYMENT

ANDROID



- Steps for Play store release:

1: Buy a Google Developer License

2: Create the app

3: Generate the signing certificates (These are so the signature matches for all the APKs, and to give the key to APIs for security

4: Build the app

5: Create a new Application in the Play Developer Console

6: Upload the signed APK

7: When Google has completed its checks, release it

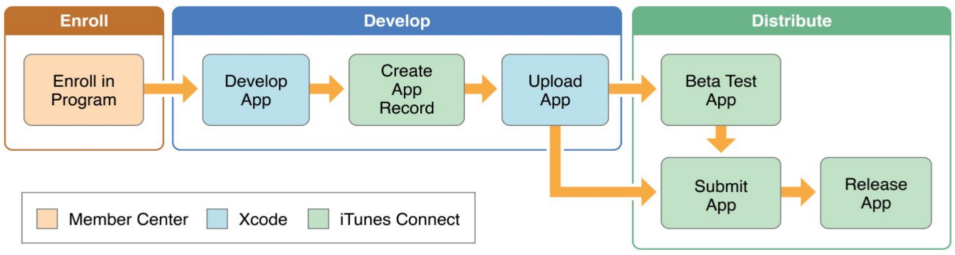
- Size limits
Max App Size: 100mb (but more data can be downloaded after the app

is installed)

https://developer.android.com/distribute/tools/launch-checklist.html

IOS

- Apple Store Submission Process



- Steps for Application Submission to the App Store:

1: Create an Apple ID and enrol as an Apple Developer (Paid version)

2: Develop the App using Xcode (testing in the simulator or own device)

2.1: in this step is necessary to generate your developer certificate. Xcode generate all the certificates for you the first time you try to run in a device or build the app for submission

3: Create an App entry in iTunesConnect

4: upload the App using Xcode

5: submit the App for Apple approval

6: release the App

- Size limits

Max App Size: 4GB (but more data can be downloaded after the app is installed)
Max over 3G/4G: 100MB (if bigger it can only be downloaded over WIFI)

- Approval Process

All new App and Updates go through an approval process. It usually takes around 2 days (so, plan ahead if you want to release at a specific date)

Links:
- Apple Dev
    https://developer.apple.com

- iTunes Connect
    https://itunesconnect.apple.com

PRICING MODEL

- Free
- Ads
- Freemium
- Paid

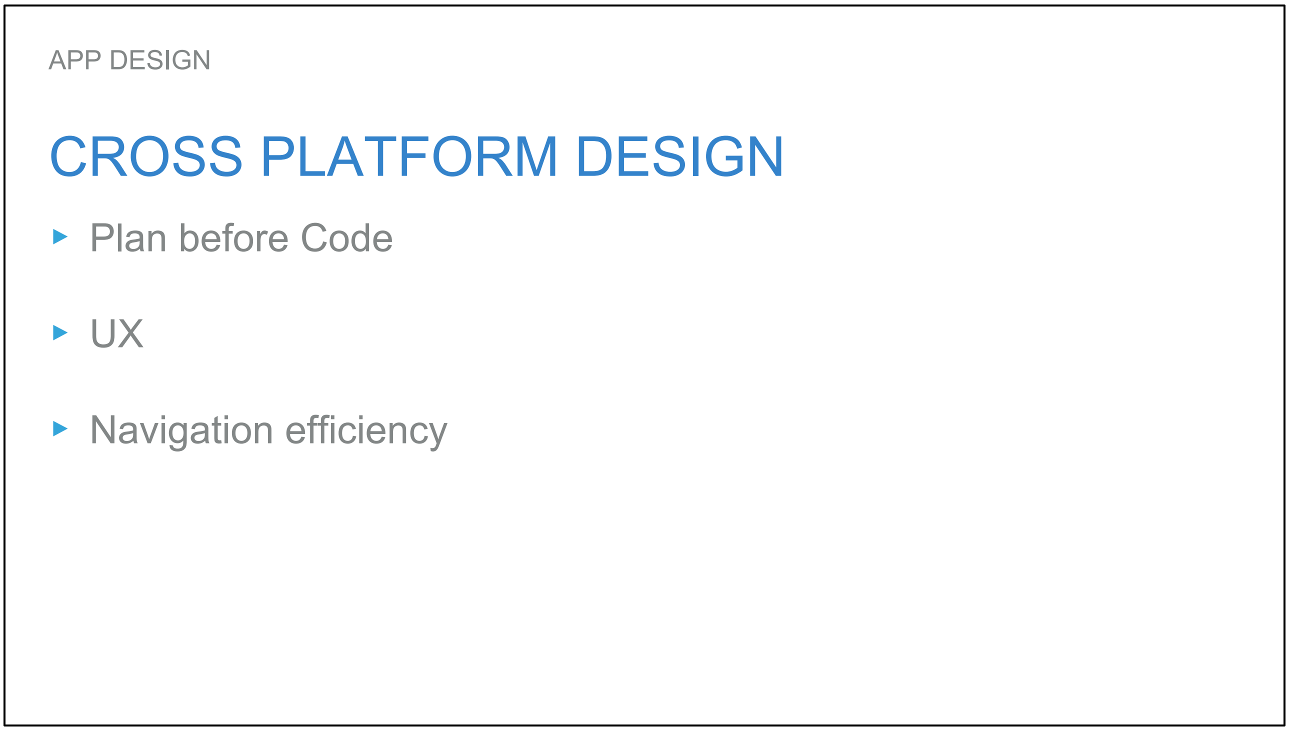
Free: The App and all features are free.

Supported by Ads: The App is Free, but display Ads as a source of revenue.

Micro-transactions: The App can be downloaded for free, but the user need to pay to enable premium features: remove Ads, additional functionality, internal currency (“coins” in games for example)

Paid: The user pay to download the App, and all features are enabled.

APP DESIGN


APP DESIGN

CROSS PLATFORM DESIGN

- Plan before Code
- UX
- Navigation efficiency

- Before start to think about the Design for a specific platform, plan ahead the App objectives and the expected user interaction/experience.
- Remember: each platform have its own UX.

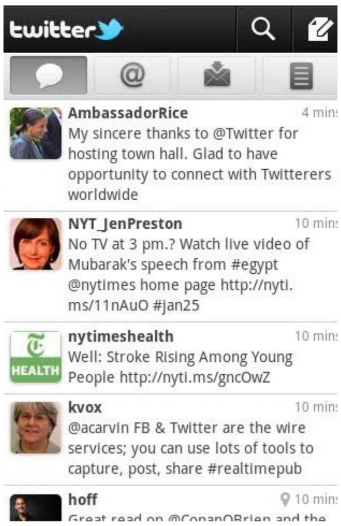
MATERIAL DESIGN

- All about 3D interactions
- Solid
- Layers and shadows
- Components that can split and join
- Move on any axis

The material design docs are your best resource.

https://design.google.com
https://www.google.com/design/spec/material-design/introduction.html

MATERIAL EVOLUTION



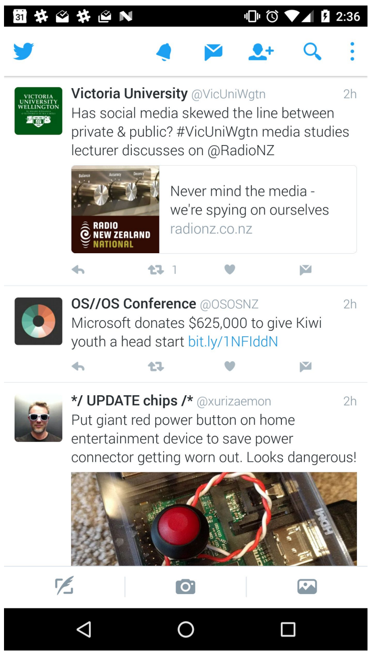
AmbassadorRice 4 mins
My sincere thanks to @Twitter for hosting town hall. Glad to have opportunity to connect with Twitterers worldwide

NYT_JenPreston 10 mins
No TV at 3 pm.? Watch live video of Mubarak's speech from #egypt @nytimes home page http://nyti.ms/11nAuO #jan25

nytimeshealth 10 mins
Well: Stroke Rising Among Young People http://nyti.ms/gncOwZ

kvox 10 mins
@acarvin FB & Twitter are the wire services; you can use lots of tools to capture, post, share #realtimepub

hoff 10 mins
Great read on @ConanOBrien and the



Victoria University @VicUniWgtn 2h
Has social media skewed the line between private & public? #VicUniWgtn media studies lecturer discusses on @RadioNZ

Never mind the media - we're spying on ourselves
radionz.co.nz

1

OS//OS Conference @OSOSNZ 2h
Microsoft donates $625,000 to give Kiwi youth a head start bit.ly/1NFIddN

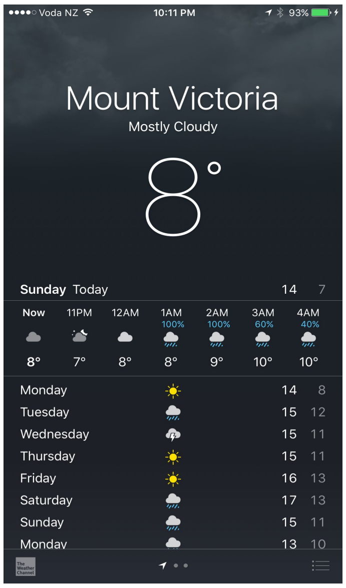
/ UPDATE chips / @xurizaemon 2h
Put giant red power button on home entertainment device to save power connector getting worn out. Looks dangerous!

2011 on the left, 2016 on the right
The material design docs are your best resource.

https://design.google.com
https://www.google.com/design/spec/material-design/introduction.html

IOS HIG

- Deference
  - Defer to content
- Clarity
  - provide clarity
- Depth
  - use depth to communicate



Mount Victoria

Mostly Cloudy

8°

Sunday Today 14 7

Now	11PM	12AM	1AM	2AM	3AM	4AM
			100%	100%	60%	40%
8°	7°	8°	8°	9°	10°	10°

Monday	14	8
Tuesday	15	12
Wednesday	15	11
Thursday	15	11
Friday	16	13
Saturday	17	13
Sunday	15	11
Monday	13	10

Human Interface Guidelines

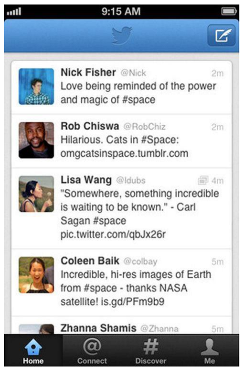
Deference. The UI helps people understand and interact with the content, but never competes with it.

Clarity. Text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design.

Depth. Visual layers and realistic motion impart vitality and heighten people's delight and understanding.

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/

IOS HIG - EVOLUTION



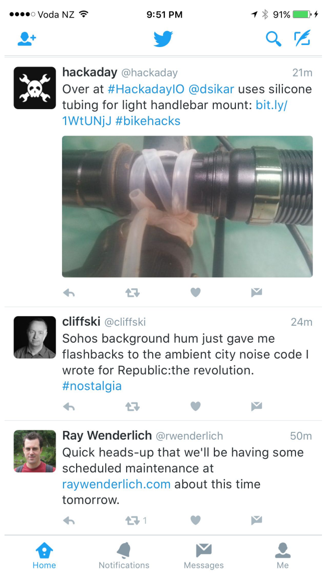
Nick Fisher @Nick 2m
Love being reminded of the power and magic of #space

Rob Chiswa @RobChiz 2m
Hilarious. Cats in #Space: omgcatsinspace.tumblr.com

Lisa Wang @ldubs 4m
"Somewhere, something incredible is waiting to be known." - Carl Sagan #space pic.twitter.com/qbJx26r

Coleen Baik @colbay 5m
Incredible, hi-res images of Earth from #space - thanks NASA satellite! is.gd/PFm9b9

Zhanna Shamis @Zhanna 5m



hackaday @hackaday 21m
Over at #HackadayIO @dsikar uses silicone tubing for light handlebar mount: bit.ly/1WtUNjJ #bikehacks

cliffski @cliffski 24m
Sohos background hum just gave me flashbacks to the ambient city noise code I wrote for Republic:the revolution. #nostalgia

Ray Wenderlich @rwenderlich 50m
Quick heads-up that we'll be having some scheduled maintenance at raywenderlich.com about this time tomorrow.

iOS Design Example:

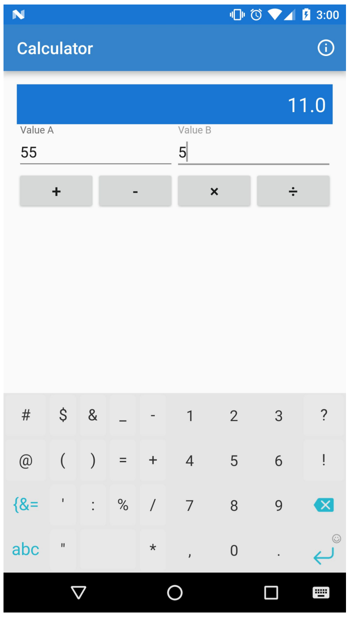
Twitter evolution from iOS 3 to iOS 9 (from Skeuomorphism to Flat Design)

APP ARCHITECTURE

more info: https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html#//apple_ref/doc/uid/TP40007072-CH2-SW1

ANDROID

- Activity
- Fragment
- Custom Views
- Layouts (Linear, Relative, Coordinator)
- Components



Activity:
The main controller of the view state.

Fragment:
A kind of sub controller to the Activity. Used where interaction involves swiping through fullscreen views, or changing fullscreen views in an activity

***Custom*View:**
A group of smaller views put together to make a reusable component. Often you'll have custom dialogs and components that appear on multiple screens
Will usually contain it's own logic, separate to the rest of the app.

LinearLayout/RelativeLayout:
Used to layout your components on the screen, a layout type will be the root view of each layout file.

Components: (TextView, Button, EditText, Switch etc)

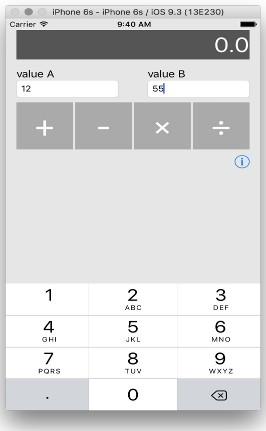
The lowest level of the view, prebuilt and with a single function usually

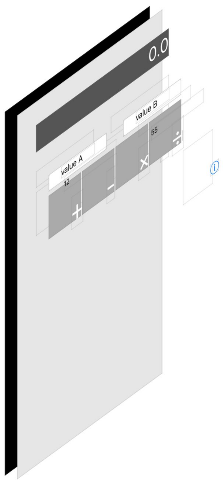
An amazing document about all this, slightly outdated but still useful:
https://inthecheesefactory.com/aca/The%20Android%20Cheese%20Sheet%20rev%204.pdf

IOS

- View Hierarchy

- Navigation Flow





- View Hierarchy

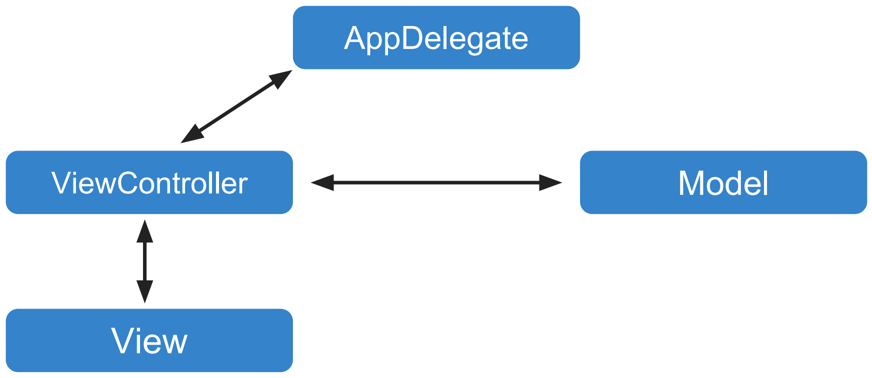
The visual components of an iOS App are represented as a hierarchy. The root is called Window and represent a physical display (local or remote) and all the additional components are Views, each one with a different function (navigation between views, display some information or provide interaction with the user).

- Navigation Flow

Each “Screen” in an iOS App is called ViewController, and is possible to navigate through many ViewControllers in different ways: linear flow (navigation controller), tabs, split screen (master/detail), modal presentation, and a few others.

IOS

- iOS App Structure



AppDelegate: starting point for an iOS app

Model: documents and data model objects

ViewController: manage the presentation of the app content and handles the user interaction and navigation flow

View: visual representation of the app content

QUESTIONS?

RESOURCES

- **github**
  - presentation PDF
  - Android and iOS source code examples

https://github.com/paperkite/sot-mobile-workshop