Android Development Getting Started

Christopher Jenkins
Developer - Mubaloo

Who am I? Chris Jenkins

- Plymouth Graduate
  - BSc (Hons) Computing
- 3 Years of Java Development
  - C#, Objective-C, PHP, bash
  - MySQL,PL/SQL, XML
  - HTML,JavaScript, jQuery
  - NodeJS
- Android Developer 2 years+

- Developed
  - Weather/Maps - MetOffice
  - Property
  - Finance
  - Retail
  - Gambling

@chrisjenx

+Chris Jenkins

Mubaloo - App Developer 2012

@mubaloo - http://mubaloo.com

Summary

- Quick Start
- What I wish I knew 2 years ago!
- Designing for Android
- Common Practice & Do's
- Demos and Examples

Quick Start

Quick Start - What is Android

- Android is built on linux
- Its 4! (Born 20th of September 2008)
- Runs a Java virtual machine (dalvikvm)
- 480M Android Devices (1.3M activated daily)*
- Over 1400+ devices
- Clopen source - but very open community

*As of 5th September 2012

Quick Start -Android fundamentals

Quick Start -Android fundamentals

Some pros n' cons:

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework
- Resource framework (Extremely useful!)

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework
- Resource framework (Extremely useful!)

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework
- Resource framework (Extremely useful!)

Cons:

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework
- Resource framework (Extremely useful!)

Cons:

- Many devices (Phone's, Tablet's, TV's, Embedded, Camera's, etc..)

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework
- Resource framework (Extremely useful!)

Cons:

- Many devices (Phone's, Tablet's, TV's, Embedded, Camera's, etc..)
- Differentiation of hardware/software

Quick Start -Android fundamentals

Some pros n' cons:

Pros:

- Background tasks (Services)
- Intent system (Very Powerful)
- Permission framework
- Resource framework (Extremely useful!)

Cons:

- Many devices (Phone's, Tablet's, TV's, Embedded, Camera's, etc..)
- Differentiation of hardware/software
  - Vendor Specific 'Enhancements'

Quick Start - What you need/have (hopefully?)

Quick Start - What you need/have (hopefully?)

- Java JDK
- Android SDK - (*http://developer.android.com/sdk/index.html*)
- Eclipse - (Recommend the Juno release)
- Android ADT (eclipse plug-in)

Quick Start - What you need/have (hopefully?)

- Java JDK
- Android SDK - (*http://developer.android.com/sdk/index.html*)
- Eclipse - (Recommend the Juno release)
- Android ADT (eclipse plug-in)

- Windows, OSX and Linux

Quick Start - What you need/have (hopefully?)

- Java JDK
- Android SDK - (*http://developer.android.com/sdk/index.html*)
- Eclipse - (Recommend the Juno release)
- Android ADT (eclipse plug-in)

- Windows, OSX and Linux

Quick Start - What you need/have (hopefully?)

- Java JDK
- Android SDK - (*http://developer.android.com/sdk/index.html*)
- Eclipse - (Recommend the Juno release)
- Android ADT (eclipse plug-in)

- Windows, OSX and Linux

- Getting started - *http://developer.android.com/training/index.html*

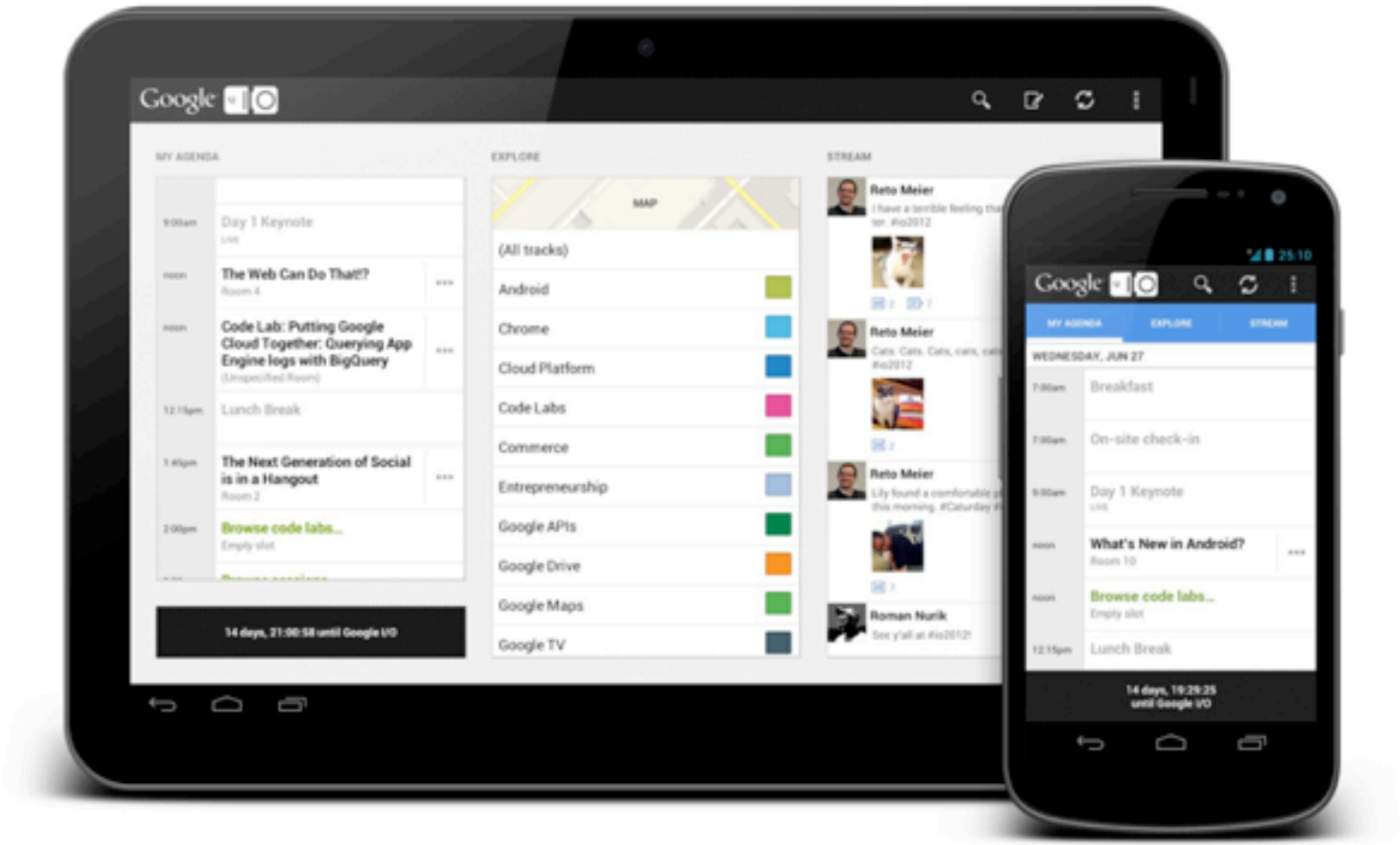
Quick Start - What you need/have (hopefully?)

- Java JDK
- Android SDK - (*http://developer.android.com/sdk/index.html*)
- Eclipse - (Recommend the Juno release)
- Android ADT (eclipse plug-in)

- Windows, OSX and Linux

- Getting started - *http://developer.android.com/training/index.html*

Quick Start - Forget your ruler...



What I wish I knew 2 years ago!

Have a plan

- Who is your market?
  - Young
  - Adults
  - Advanced
- Why will they care?
  - Whats unique?
- How will you communicate with them?
- Getting paid?
  - In-App Purchases
  - Ads
  - Single Sale

How will you distribute

- Direct download - http://mywebsite.com
- Play store - http://play.google.com
- Amazon - http://amazon.com/appstore
- NOOK

Target Devices

Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html

Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html
- Always Target/Build for Latest API (API 16)

Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html
- Always Target/Build for Latest API (API 16)
- Manifest.xml

```
<uses-sdk  android:minSdkVersion="8" android:targetSdkVersion="16" />
```

Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html
- Always Target/Build for Latest API (API 16)
- Manifest.xml

```
<uses-sdk  android:minSdkVersion="8" android:targetSdkVersion="16" />
```

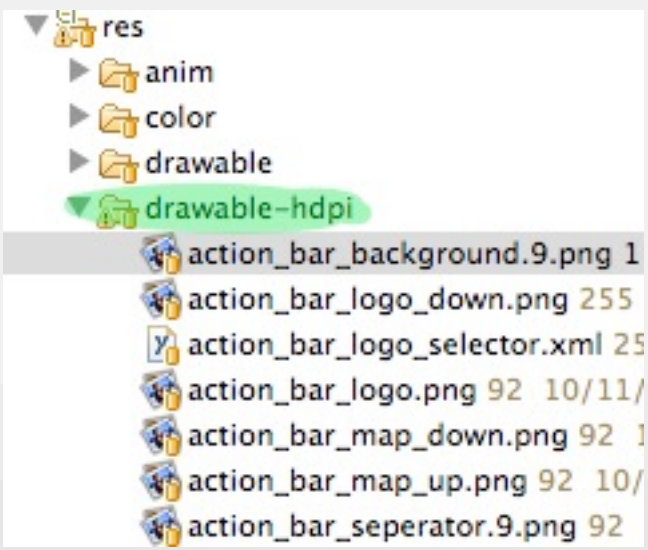
- normal - hdpi

Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html
- Always Target/Build for Latest API (API 16)
- Manifest.xml

```
<uses-sdk  android:minSdkVersion="8" android:targetSdkVersion="16" />
```

- normal - hdpi

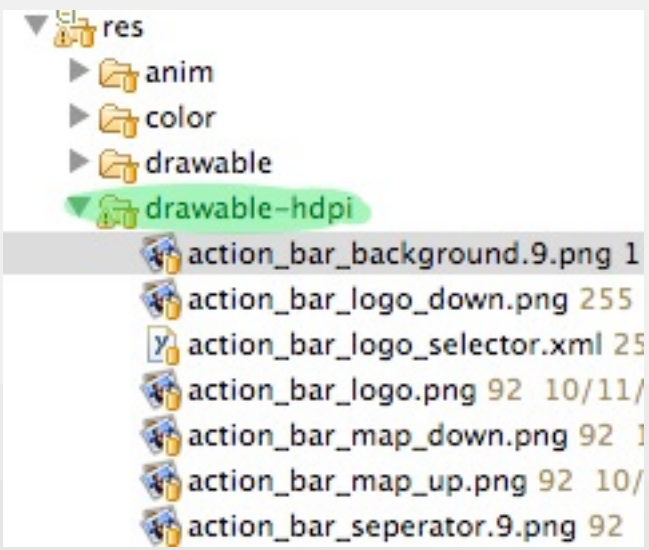


Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html
- Always Target/Build for Latest API (API 16)
- Manifest.xml

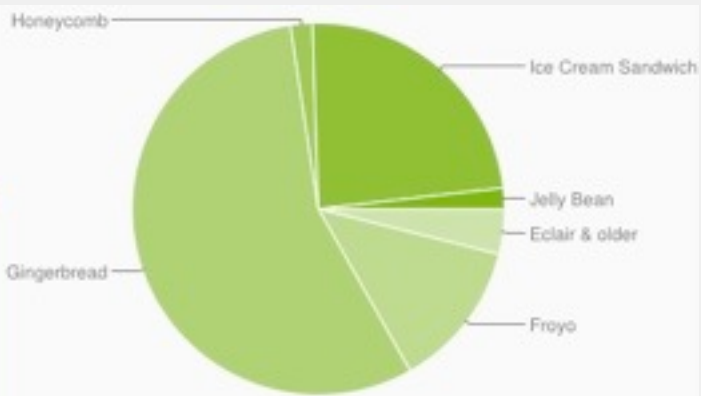
```
<uses-sdk  android:minSdkVersion="8" android:targetSdkVersion="16" />
```

- normal - hdpi



Version	Codename	API	Distribution
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.4%
2.1	Eclair	7	3.4%
2.2	Froyo	8	12.9%
2.3 - 2.3.2	Gingerbread	9	0.3%
2.3.3 - 2.3.7		10	55.5%
3.1	Honeycomb	12	0.4%
3.2		13	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	23.7%
4.1	Jelly Bean	16	1.8%

Data collected during a 14-day period ending on October 1, 2012



Target Devices

- Keep watching Android Dashboards
  http://developer.android.com/about/dashboards/index.html
- Always Target/Build for Latest API (API 16)
- Manifest.xml

```
<uses-sdk  android:minSdkVersion="8" android:targetSdkVersion="16" />
```

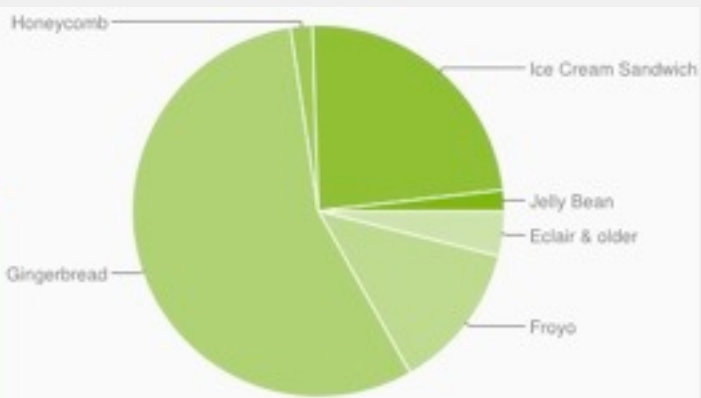
- normal - hdpi

	ldpi	mdpi	hdpi	xhdpi
small	1.7%		1.0%	
normal	0.4%	11%	50.1%	25.1%
large	0.1%	2.4%		3.6%
xlarge		4.6%		

Data collected during a 7-day period ending on October 1, 2012

Version	Codename	API	Distribution
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.4%
2.1	Eclair	7	3.4%
2.2	Froyo	8	12.9%
2.3 - 2.3.2	Gingerbread	9	0.3%
2.3.3 - 2.3.7		10	55.5%
3.1	Honeycomb	12	0.4%
3.2		13	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	23.7%
4.1	Jelly Bean	16	1.8%

ata collected during a 14-day period ending on October 1, 2012



Target Devices - cont.

- Phone
  - 3" - 5" devices most common
  - Gal S3/One X
- Tablets?
  - 7" Amazon Fire/Nexus 7
- Err 'Phablets'?

Fragments

Fragments

- Use them.

Fragments

- Use them.
  - Compatibility Library (if pre HoneyComb) https://developer.android.com/tools/extras/support-library.html

Fragments

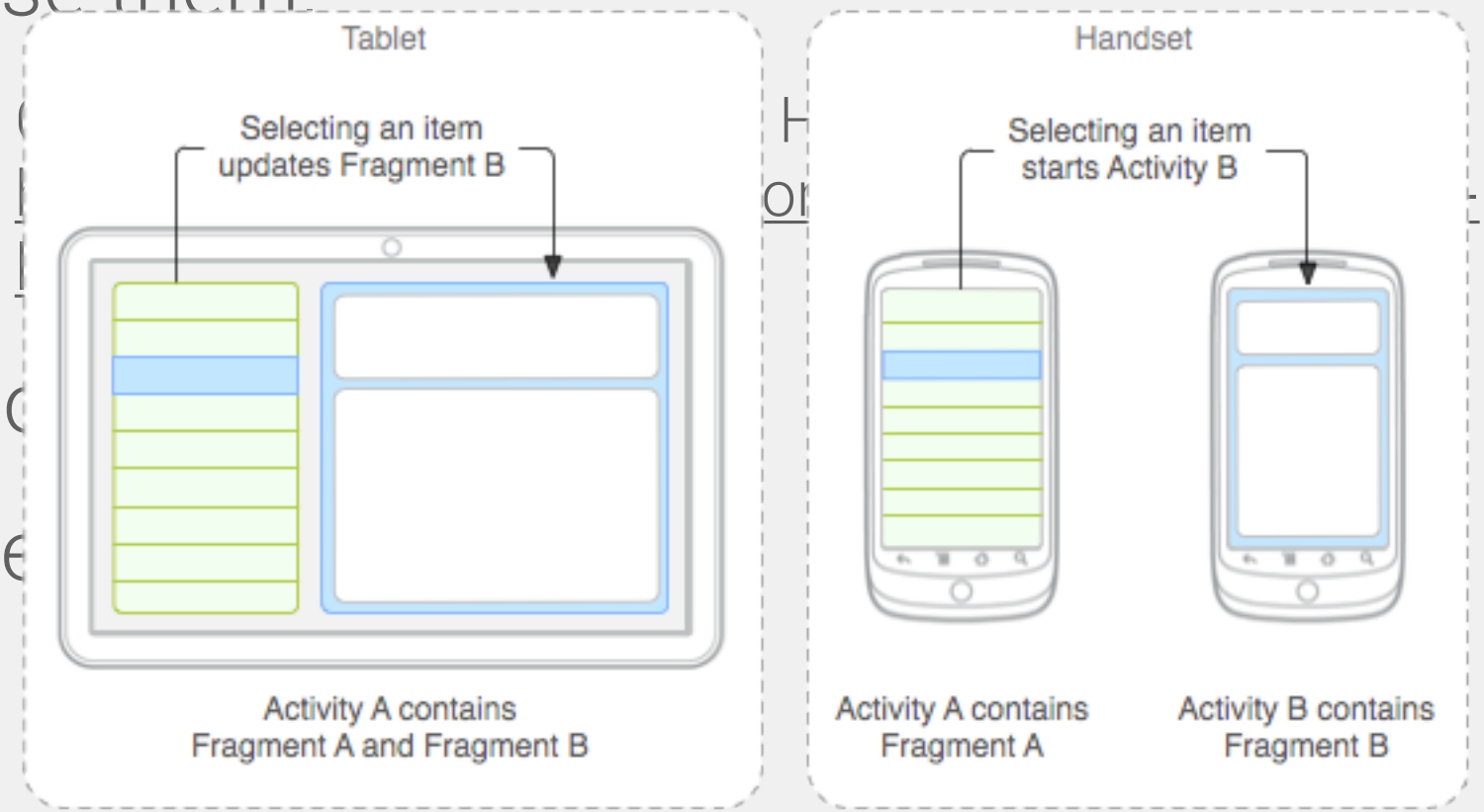
- Use them.
  - Compatibility Library (if pre HoneyComb) https://developer.android.com/tools/extras/support-library.html
- No really!

Fragments

- Use them.
  - Compatibility Library (if pre HoneyComb) https://developer.android.com/tools/extras/support-library.html
- No really!
- Demo later

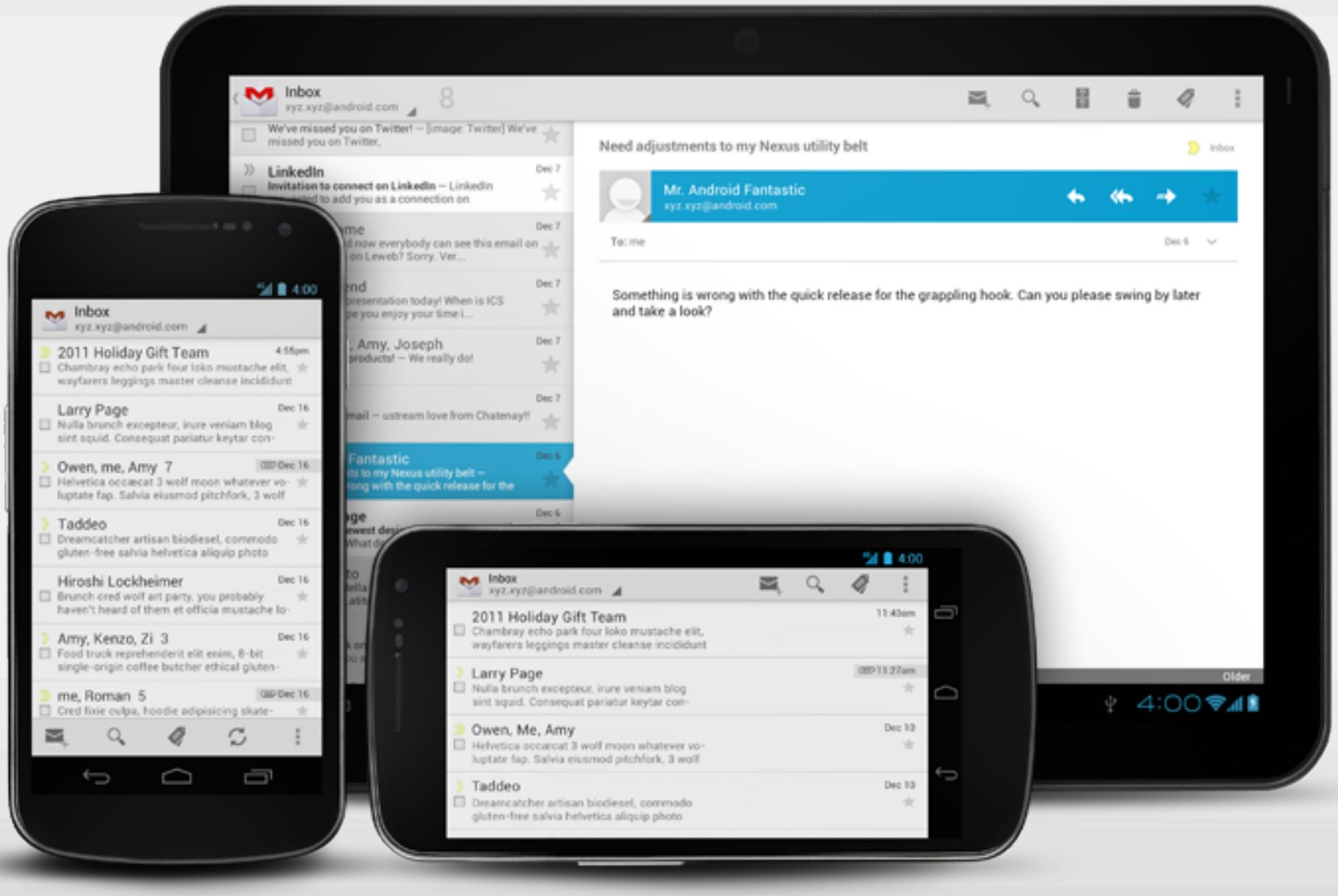
Fragments

- Use them
  - (
- No
- De



Action Bar

Action Bar


Inbox
xyz.xyz@android.com
Need adjustments to my Nexus utility belt
Mr. Android Fantastic
xyz.xyz@android.com
To: me
Something is wrong with the quick release for the grappling hook. Can you please swing by later and take a look?
LinkedIn
Invitation to connect on LinkedIn – LinkedIn
2011 Holiday Gift Team
Larry Page
Owen, me, Amy 7
Taddeo
Hiroshi Lockheimer
Amy, Kenzo, Zi 3
me, Roman 5
Owen, Me, Amy
Older
4:00

Action Bar

Action Bar

- Best practice

Action Bar

- Best practice
- Pre HoneyComb?

Action Bar

- Best practice
- Pre HoneyComb?
  - ActionBarSherlock

Action Bar

- Best practice
- Pre HoneyComb?
  - ActionBarSherlock
    - http://actionbarsherlock.com/

Action Bar

- Best practice
- Pre HoneyComb?
  - ActionBarSherlock
    - http://actionbarsherlock.com/
    - ActionBar back to Android 2.1

Action Bar

- Best practice
- Pre HoneyComb?
  - ActionBarSherlock
    - http://actionbarsherlock.com/
    - ActionBar back to Android 2.1
- Demo later

Make it look pretty

Make it look pretty

- Graphics

Make it look pretty

- Graphics
  - Ideally - Custom artwork

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes
  - Defacto as of ICS 4.0

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes
  - Defacto as of ICS 4.0
    - Light/DarkActionBar - `@style/Theme.Holo.Light.DarkActionBar`

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes
  - Defacto as of ICS 4.0
    - Light/DarkActionBar - `@style/Theme.Holo.Light.DarkActionBar`
    - Light(Dark text) - `@style/Theme.Holo.Light`

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes
  - Defacto as of ICS 4.0
    - Light/DarkActionBar - `@style/Theme.Holo.Light.DarkActionBar`
    - Light(Dark text) - `@style/Theme.Holo.Light`
    - Dark (Light text), also the default - `@style/Theme.Holo`

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes
  - Defacto as of ICS 4.0
    - Light/DarkActionBar - `@style/Theme.Holo.Light.DarkActionBar`
    - Light(Dark text) - `@style/Theme.Holo.Light`
    - Dark (Light text), also the default - `@style/Theme.Holo`

Make it look pretty

- Graphics
  - Ideally - Custom artwork
  - Make do - Android Asset Studio
    - http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html
- Themes
  - Defacto as of ICS 4.0
    - Light/DarkActionBar - `@style/Theme.Holo.Light.DarkActionBar`
    - Light(Dark text) - `@style/Theme.Holo.Light`
    - Dark (Light text), also the default - `@style/Theme.Holo`

Follow UI Patterns

Follow UI Patterns

Follow UI Patterns

Follow UI Patterns

- We now have guidelines
  - Design site
    https://developer.android.com/design/index.html
  - Follow them - but not religiously!

Follow UI Patterns

- We now have guidelines
  - Design site
    https://developer.android.com/design/index.html
  - Follow them - but not religiously!

- Best example preferences

Follow UI Patterns

- We now have guidelines
  - Design site
    https://developer.android.com/design/index.html
  - Follow them - but not religiously!

- Best example preferences

Follow UI Patterns

- We now have guidelines
  - Design site
    https://developer.android.com/design/index.html
  - Follow them - but not religiously!

- Best example preferences

- If you don't know the best way to do something - ask/google/examples/other apps etc

Sanity

Sanity

- Code validation
  - ADT Lint - Tells you when your doing it wrong!

Sanity

- Code validation
  - ADT Lint - Tells you when your doing it wrong!
- Testing
  - Robolelectric
  - Apkudo - online real device testing http://www.apkudo.com/

Help!

Help!

- Stack overflow - android tag
  http://stackoverflow.com/questions/tagged/android
- Android Developer Blog
  http://android-developers.blogspot.co.uk/
- Android Developer site
  https://developer.android.com/guide/components/index.html
- Android Developer Group
  https://groups.google.com/forum/?fromgroups#!forum/android-developers
- Github
  https://github.com/
- OpenIntents - Android Library Listing
  http://www.openintents.org/en/libraries
- Android Developer Meetups
  http://android-developers.meetup.com/
- Chrome Omni Extension
  https://chrome.google.com/webstore/detail/android-sdk-reference-sea/hgcbffeicehlpmgmnhnkjbjoldkfhoin

Help!

Help!

Google!

Designing for Android


Google
Google
Google

Design - It's all relative

Design - It's all relative

- No fixed screen size

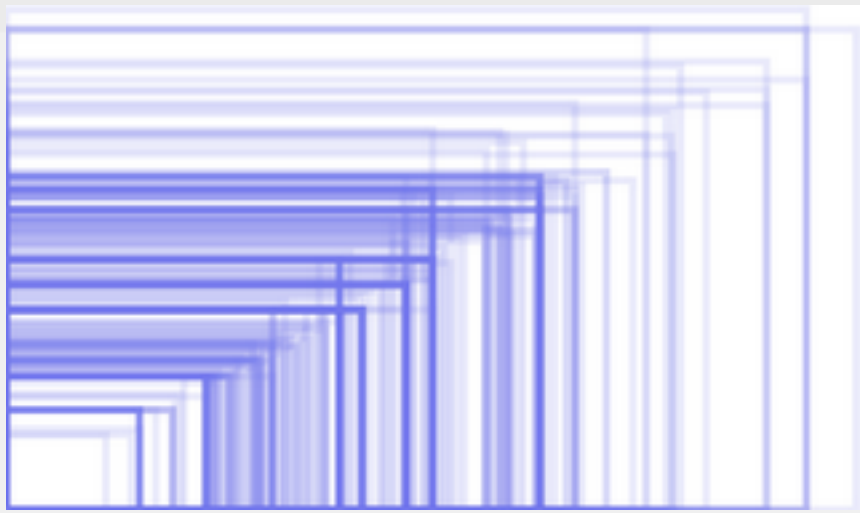
Design - It's all relative

- No fixed screen size

- We work in "**dip**s" **d**ensity **i**ndependent **p**ixels
  http://developer.android.com/training/multiscreen/screendensities.html
  - Galaxy Nexus = 720 x 1280 = 360dp x 640dp

Design - It's all relative

- No fixed screen size



- We work in "**dip**s" **d**ensity **i**ndependent **p**ixels
  http://developer.android.com/training/multiscreen/screendensities.html
  - Galaxy Nexus = 720 x 1280 = 360dp x 640dp

Design - dips

- Should I need to worry?
  - Yes - You will ONLY use them!
- Calculate?
  - 32dip = 32pixels @ mdpi (~160dpi) - *x1.0*
  - 32dip = 48pixels @ hdpi (~240dpi) - *x1.5*
  - 32dip = 64pixels @ xhdpi (~320dpi) - *x2.0*
- Example

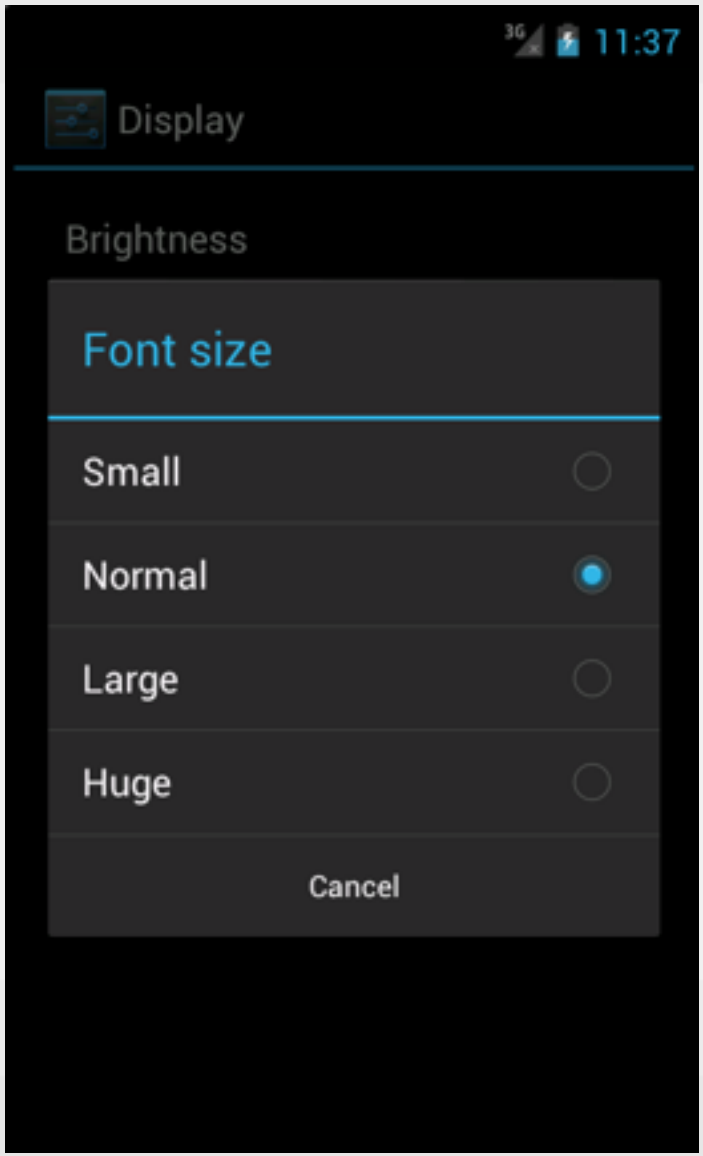
Design - scaled pixels

Design - scaled pixels

- ScaledPixels?
  - What happens when you change the device text size size?

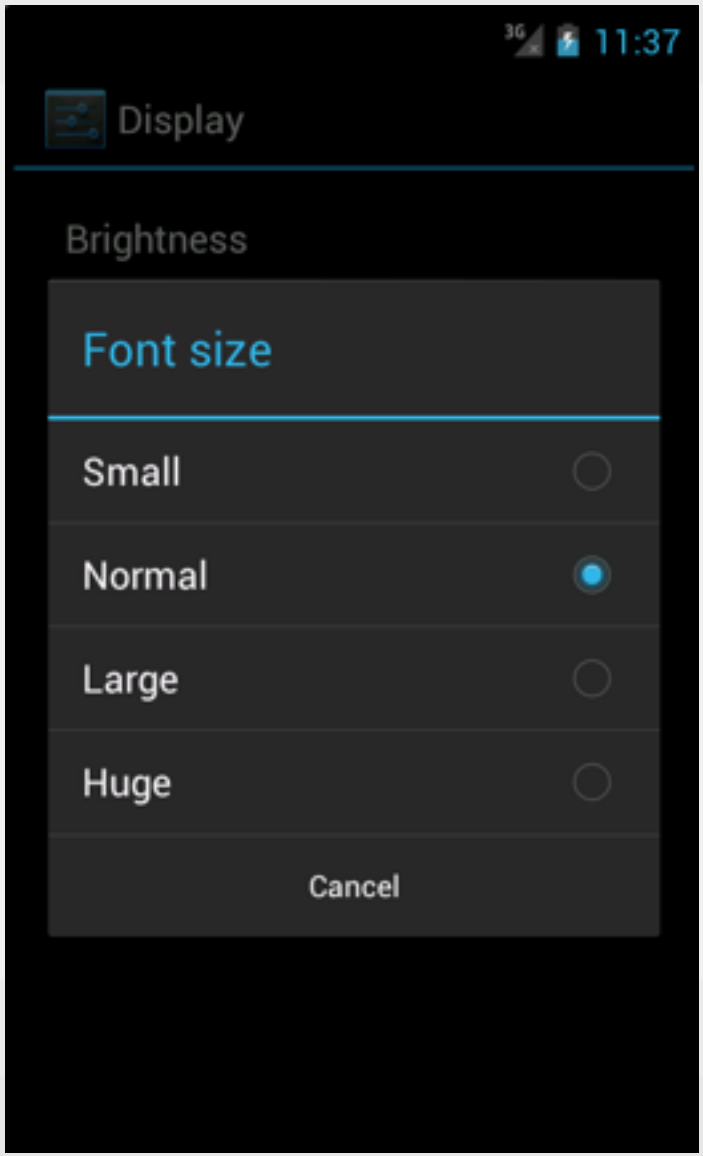
Design - scaled pixels

- ScaledPixels?
  - What happens when you change the device text size size?



Design - scaled pixels

- ScaledPixels?
  - What happens when you change the device text size size?
- Just like dips but multiplied by font size
- 1sp = 1dp at normal font size.



Design - Layouts

Design - Layouts

- Think Web!

Design - Layouts

- Think Web!
- Good:

Design - Layouts

- Think Web!
- Good:
  - Frame Layout

Design - Layouts

- Think Web!
- Good:
  - Frame Layout
  - Relative Layouts

Design - Layouts

- Think Web!
- Good:
  - Frame Layout
  - Relative Layouts
  - LinearLayouts

Design - Layouts

- Think Web!
- Good:
  - Frame Layout
  - Relative Layouts
  - LinearLayouts
    - Careful with nesting, If you layout is more than 4 layers deep, maybe rethink your design.

Design - Layouts

- Think Web!
- Good:
  - Frame Layout
  - Relative Layouts
  - LinearLayouts
    - Careful with nesting, If you layout is more than 4 layers deep, maybe rethink your design.
  - GridLayout (New API14+ or v7 acl)

Design - Layouts

- Think Web!
- Good:
  - Frame Layout
  - Relative Layouts
  - LinearLayouts
    - Careful with nesting, If you layout is more than 4 layers deep, maybe rethink your design.
  - GridLayout (New API14+ or v7 acl)
- DO NOT USE AbsoluteLayouts!

Design - Layouts

- Think Web!
- Good:
  - Frame Layout
  - Relative Layouts
  - LinearLayouts
    - Careful with nesting, If you layout is more than 4 layers deep, maybe rethink your design.
  - GridLayout (New API14+ or v7 acl)
- DO NOT USE AbsoluteLayouts!
- Read: http://developer.android.com/training/multiscreen/screensizes.html

Design - Styling

- styles.xml
  - ```
    <style name="Text.Small">
        <item name="android:textColor">@color/greyish</item>
        <item name="android:textSize">@dimen/text_size_normal</item>
    </style>
    ```
- dimens.xml
  - ```
    <dimen name="text_size_small">16sp</dimen>
    ```
- colors.xml
  - ```
    <color name="greyish">#111111</color>
    ```
- More Reading
  - https://developer.android.com/guide/topics/ui/themes.html
  - http://stylingandroid.com/
    ```

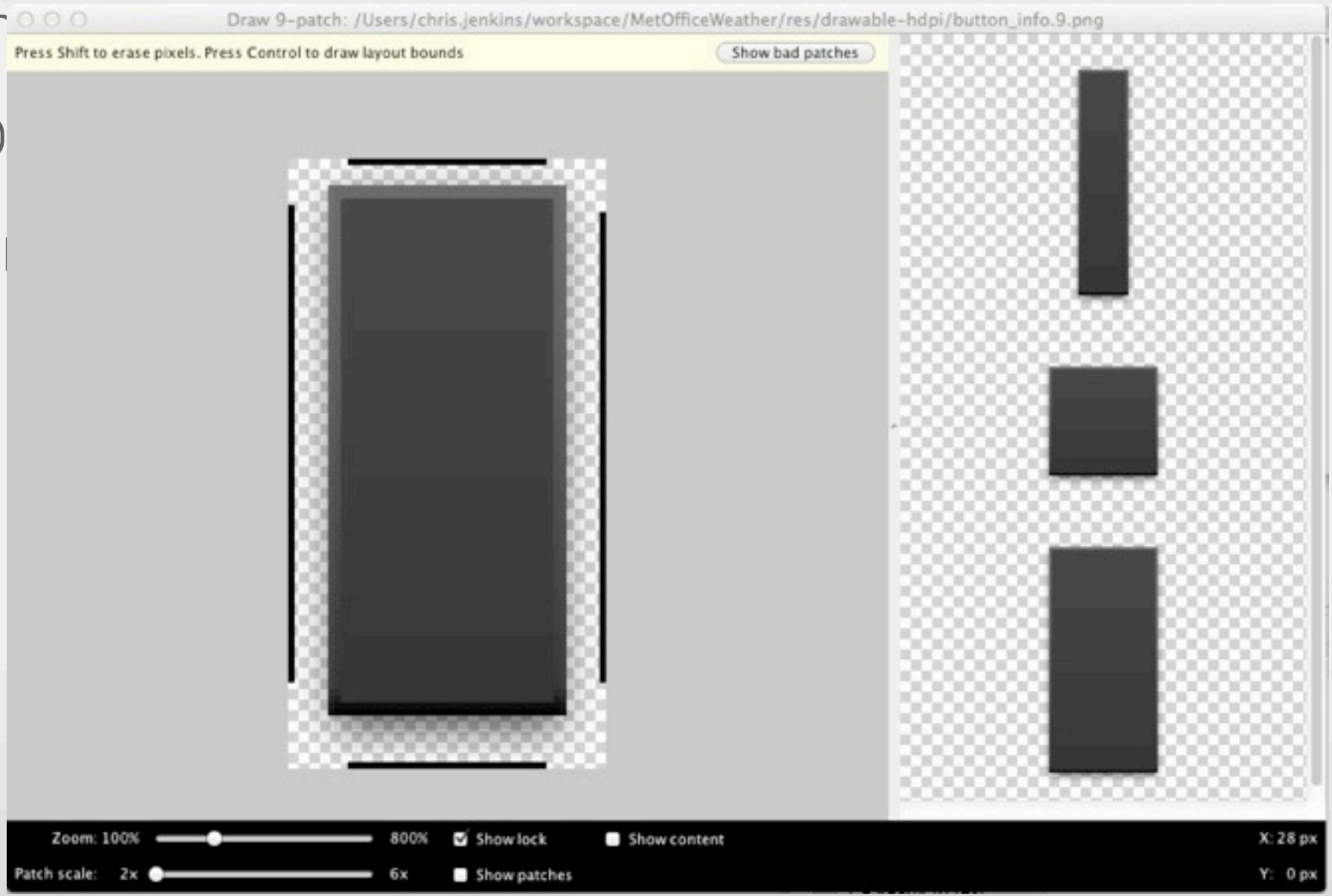
Design - 9 Patchs

Design - 9 Patchs

- Images
- Definable stretchable regions
- draw9patch tool

Design - 9 Patchs

- Ir
- D
- d



Design - Pet peeves

- Don't port/copy from iOS!
- Not using native ActionBar (Or ActionBarSherlock)
  - Example of bad ActionBar implementation - *Olympics 2012 Results App*
- No splash screen (well try to avoid it!)

Design - Resources

- Android Design
  http://developer.android.com/design/index.html
- 9-Patch
  http://radleymarx.com/blog/simple-guide-to-9-patch/
- UI Patterns
  http://www.androiduipatterns.com/

Do's n' Don'ts

Threading - ANR

Threading - ANR

- ANR?

Threading - ANR

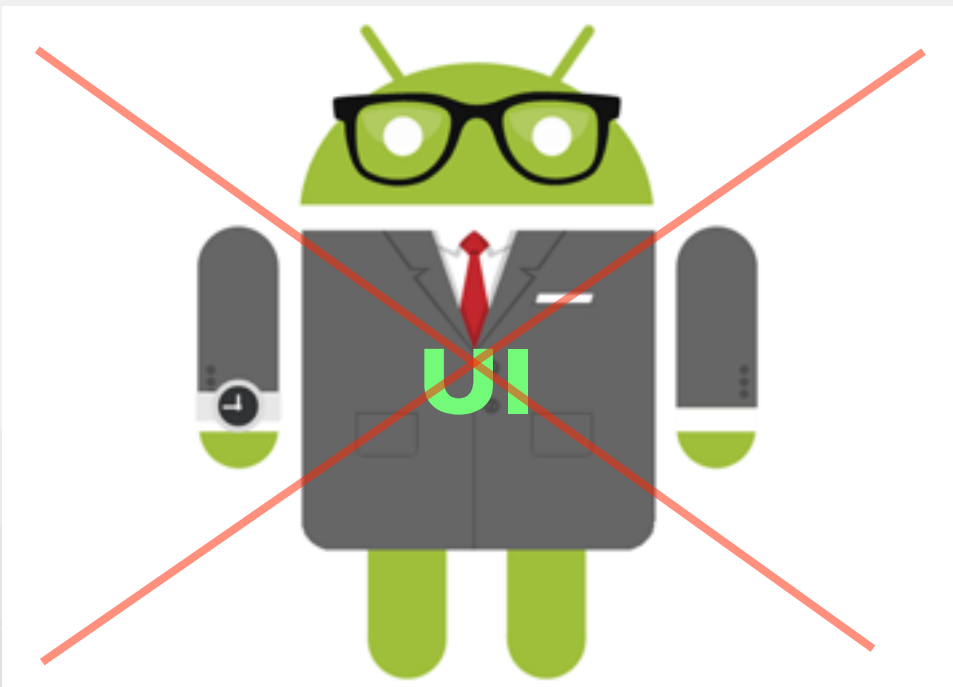
- ANR?
  - Activity Not Responding.

Threading - ANR

- ANR?
  - Activity Not Responding.
- Do not work on the UI thread!

Threading - ANR

- ANR?
  - Activity Not Responding.
- Do not work on the UI thread!



Threading

- Simplest use
  - `AsyncTask<Params, Progress, Result>`
  - Managed Thread pool
  - doInBackground runs on a background thread.
  - PreExecute & PostExecute run on the UI thread.
- What should you thread?
  - Web requests - Data/Images etc
  - Database
  - IO/Preferences etc

Threading - cont'd

- **ContentProviders**
  - Will listen to database changes and asynchronously update UI.
    https://developer.android.com/reference/android/content/ContentProvider.html
- **IntentService** - Queued Background Threaded Service
- **WakeFullIntentService** - Does work in the background service but will wake the device to do work
  MarkMurphy - https://github.com/commonsguy/cwac-wakeful

Fragments

- ModularUI
  - Not just for Phone&Tablet apps
  - Break up UI into manageable buckets/ modules
  - No Gargantuan Activities!
  - Ideally don't need to be linked, ContentProviders + Fragment lifecycle.
    - Similar lifecycle to Activities.

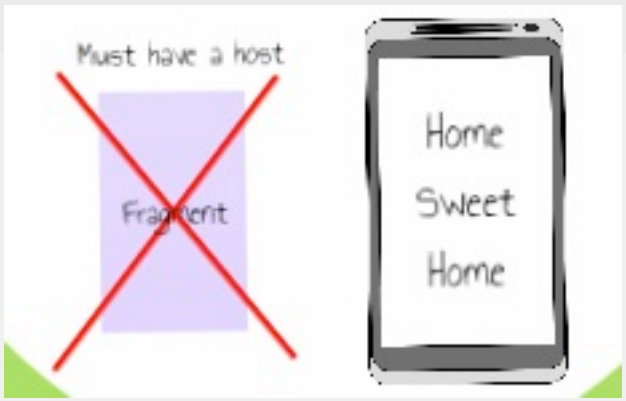
Fragments - Cont'd

- “If your not writing fragments, your writing legacy code" - Berry Drinkwater

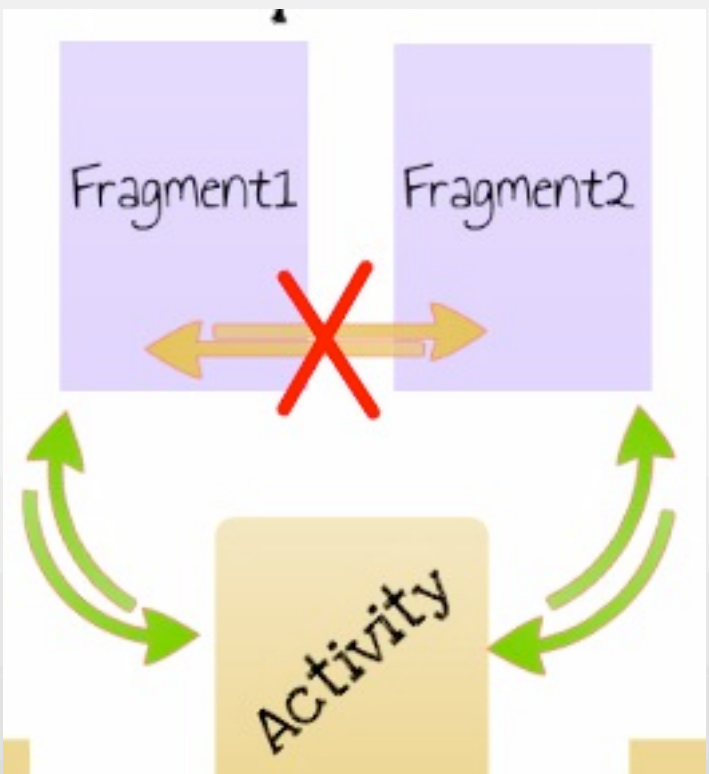
- Opens up new possibilities and API's
  - ActionBar + Tabs
  - Transition Animations
  - Automatic BackStack management
  - ViewPager

Fragment Idea

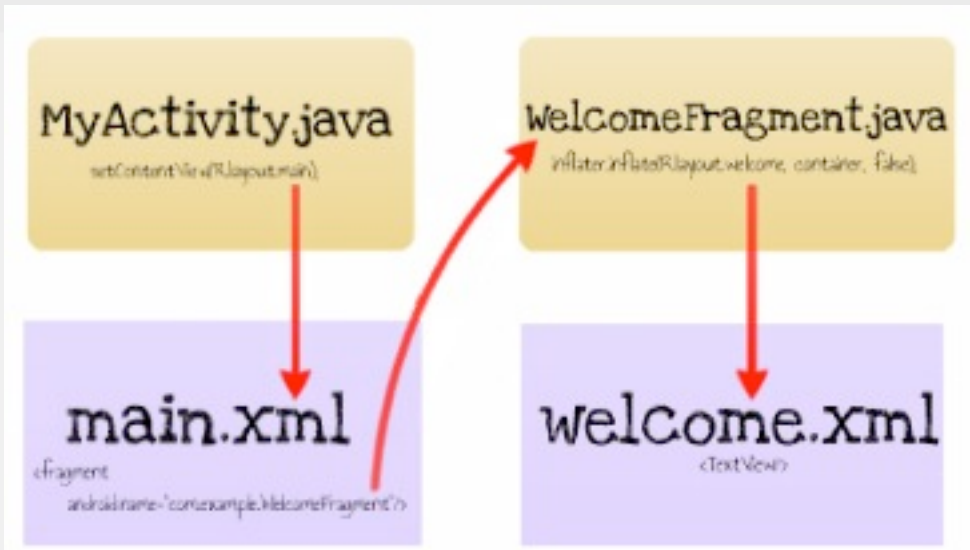
- Sub-Activity?



- Independent

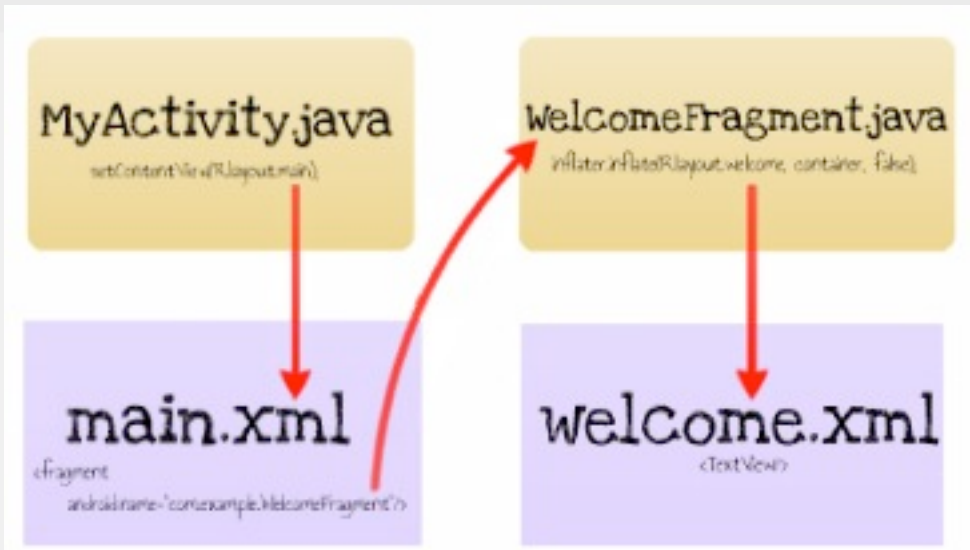


Fragments...


MyActivity.java
WelcomeFragment.java
main.xml
welcome.xml

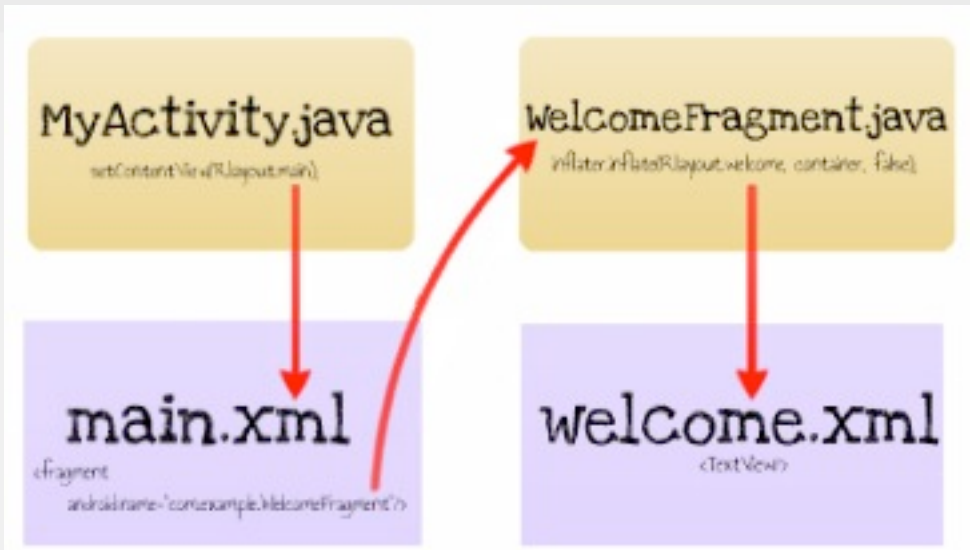
Fragments...

- **Simple Idea**



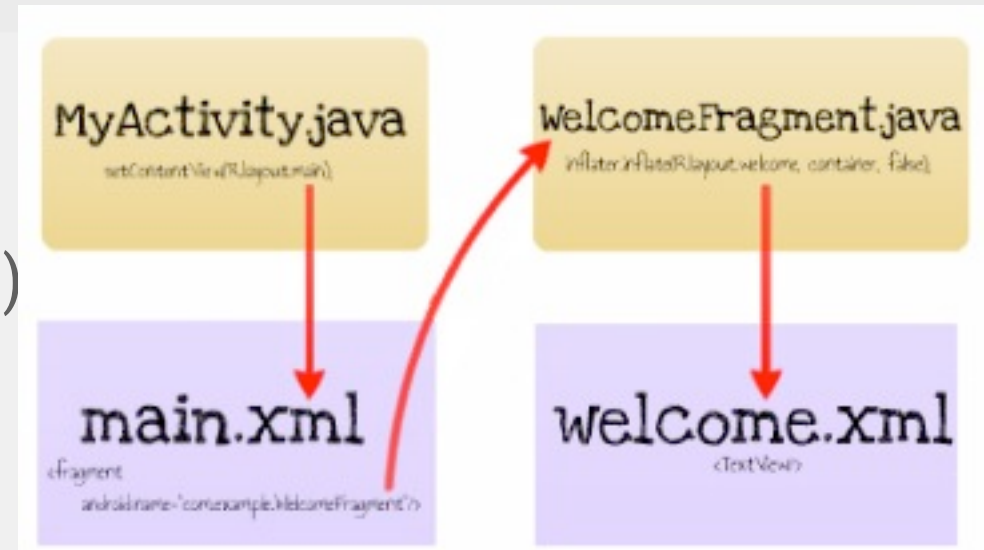
Fragments...

- **Simple Idea**
- MainActivity.java



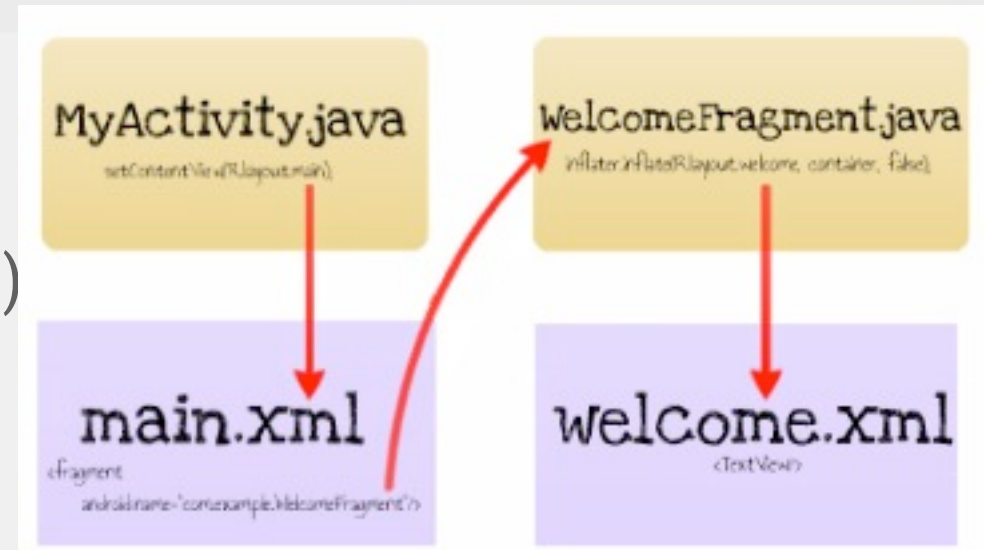
Fragments...

- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)



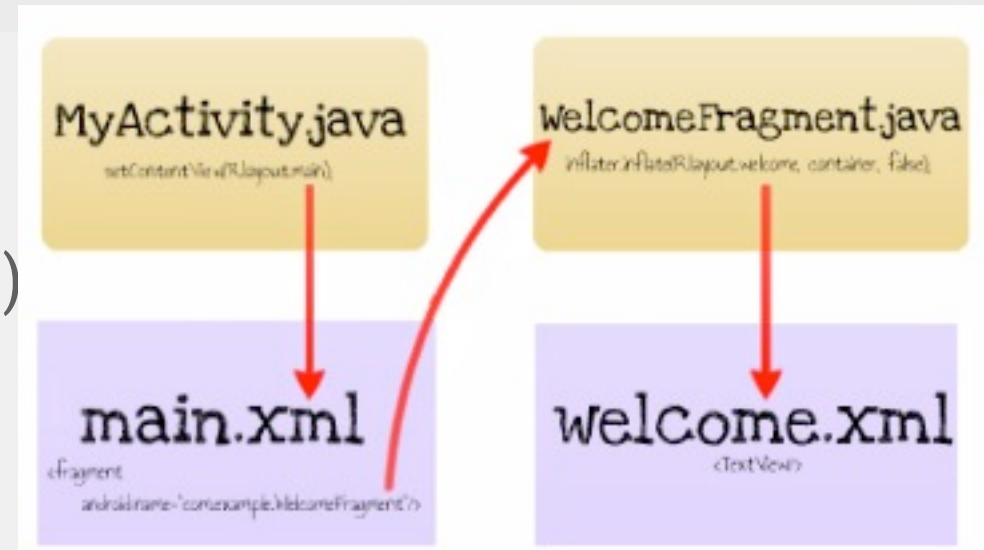
Fragments...

- **Simple Idea**
- MainActivity.java
  - **setContentView(R.layout.main)**
- main.xml



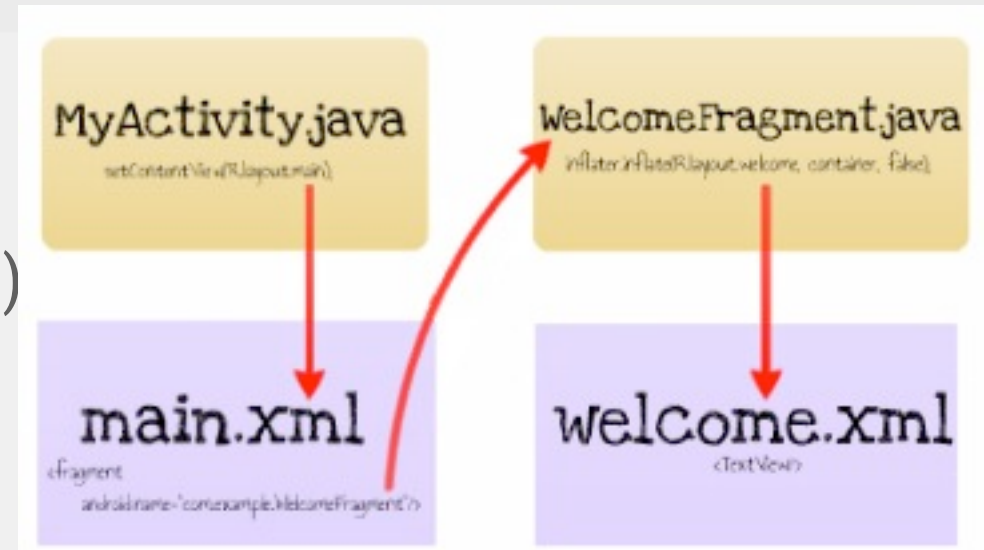
Fragments...

- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)
- main.xml
  - <fragment android:name="com.example.fragments.WelcomeFragment"/>



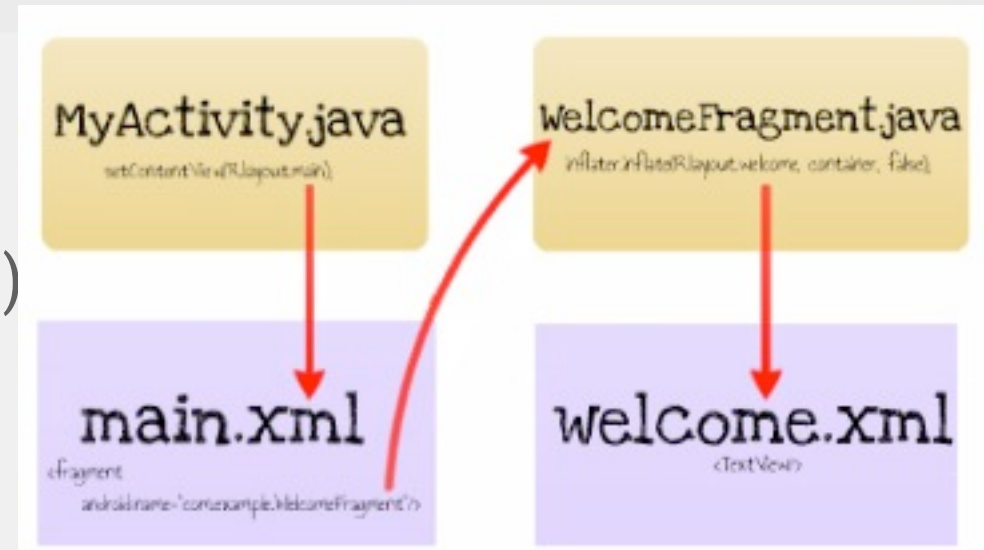
Fragments...

- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)
- main.xml
  - <fragment android:name="com.example.fragments.WelcomeFragment"/>
- WelcomeFragment.java



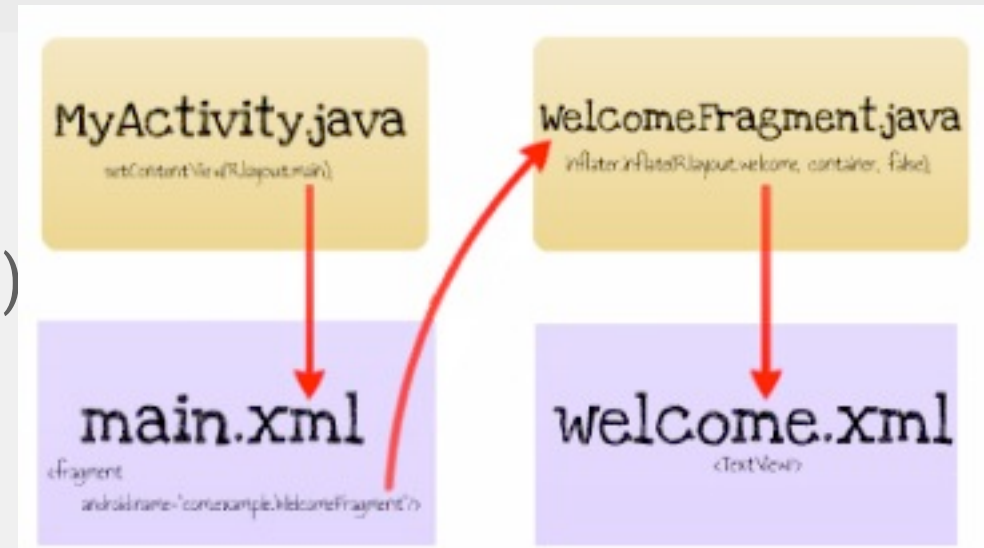
Fragments...

- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)
- main.xml
  - <fragment android:name="com.example.fragments.WelcomeFragment"/>
- WelcomeFragment.java
  - inflater.inflate(R.layout.frag_welcome, container, false);



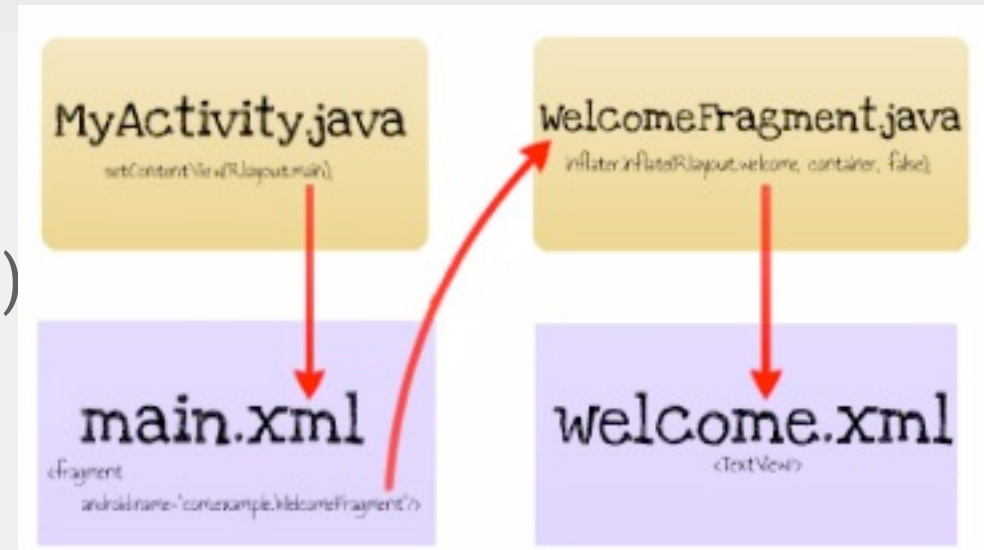
Fragments...

- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)
- main.xml
  - <fragment android:name="com.example.fragments.WelcomeFragment"/>
- WelcomeFragment.java
  - inflater.inflate(R.layout.frag_welcome, container, false);
- frag_welcome.xml



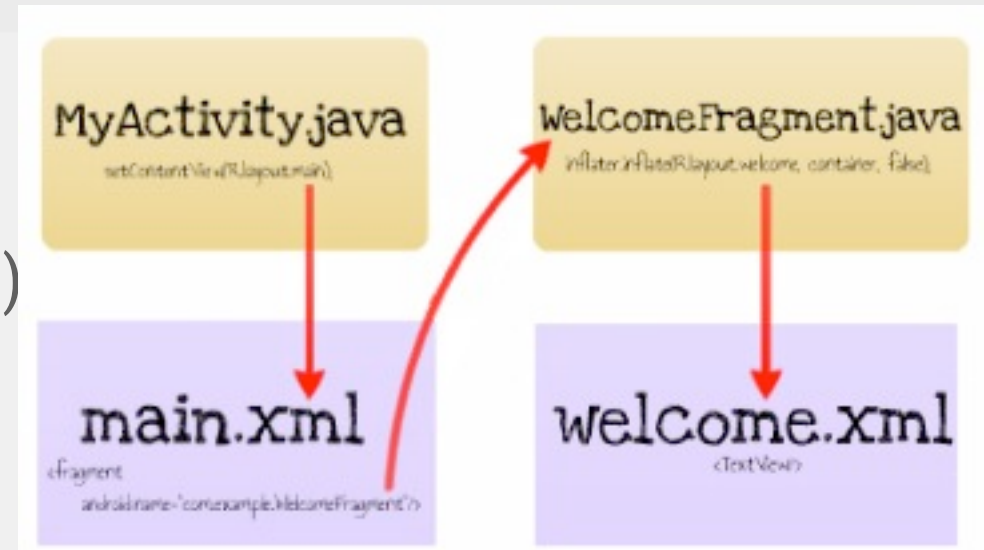
Fragments...

- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)
- main.xml
  - <fragment android:name="com.example.fragments.WelcomeFragment"/>
- WelcomeFragment.java
  - inflater.inflate(R.layout.frag_welcome, container, false);
- frag_welcome.xml
  - <TextView android:text="Hello World" />

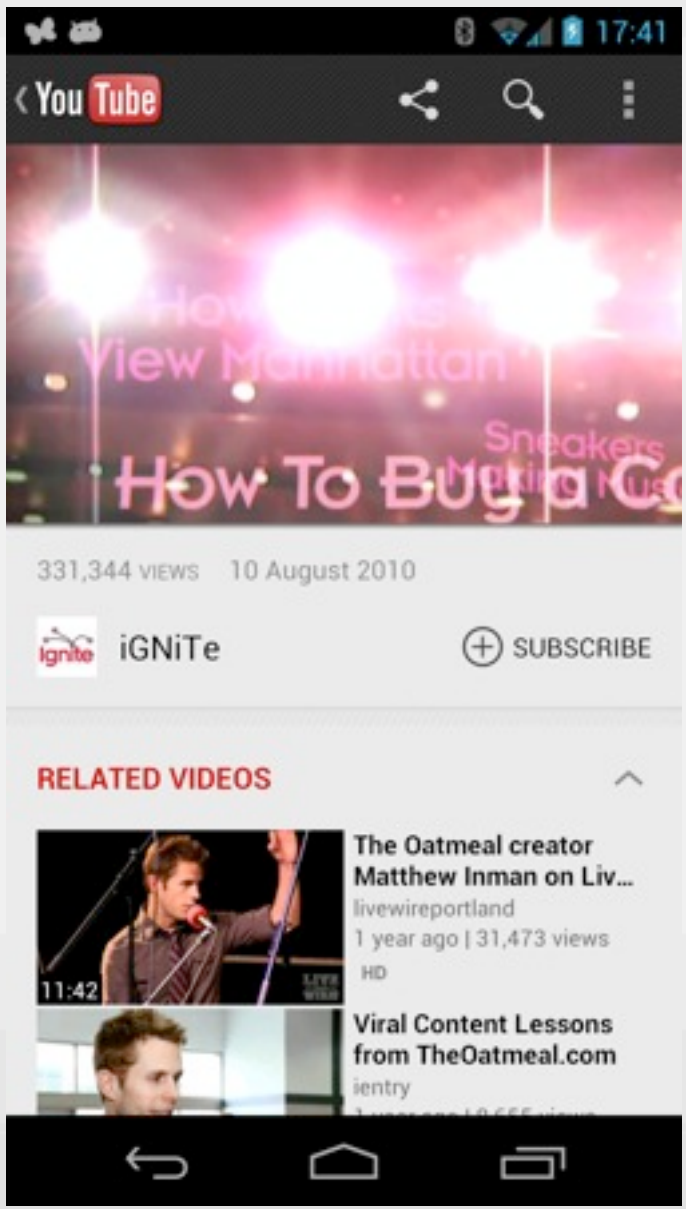


Fragments...

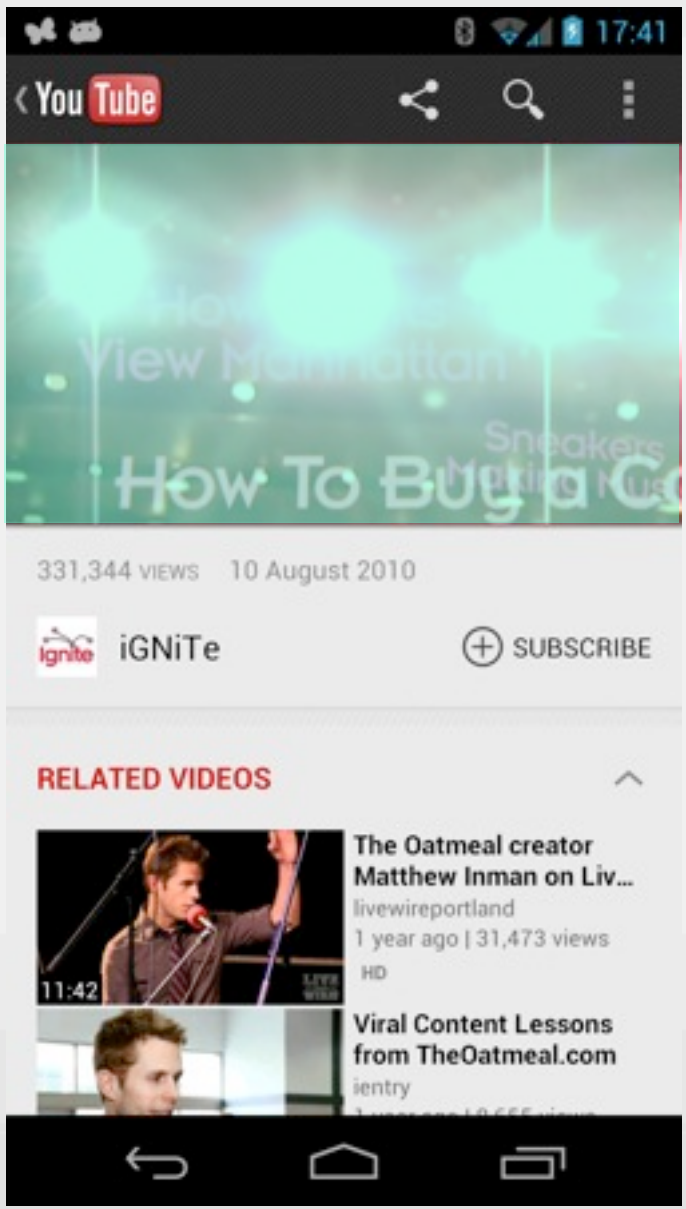
- **Simple Idea**
- MainActivity.java
  - setContentView(R.layout.main)
- main.xml
  - <fragment android:name="com.example.fragments.WelcomeFragment"/>
- WelcomeFragment.java
  - inflater.inflate(R.layout.frag_welcome, container, false);
- frag_welcome.xml
  - <TextView android:text="Hello World" />
- Example..



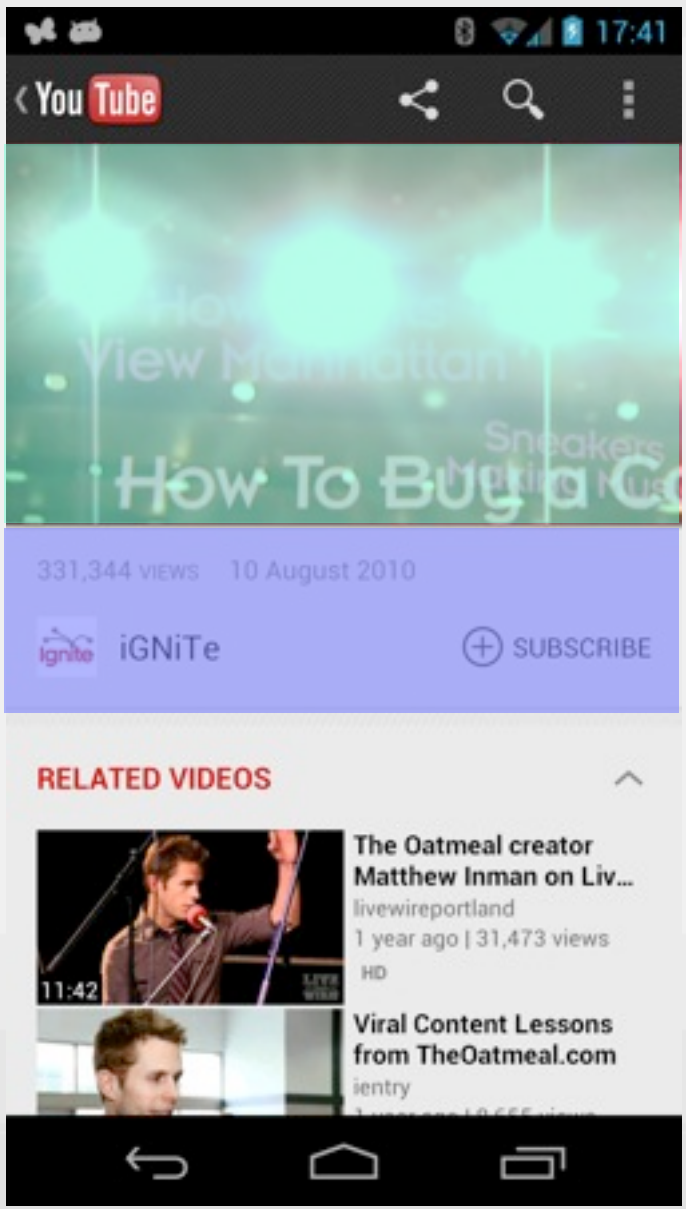
Fragments - Bigger Picture



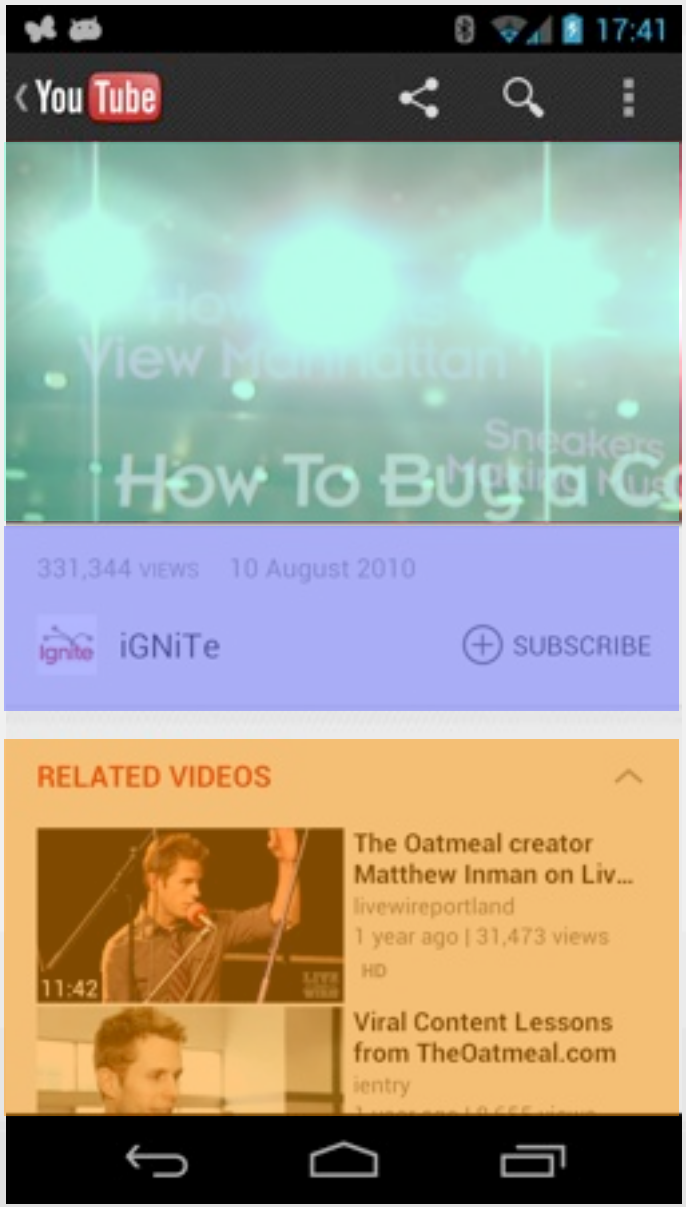
Fragments - Bigger Picture



Fragments - Bigger Picture



Fragments - Bigger Picture



Fragments - Dialog

- **DialogFragment**
  - Stop using **Dialog**!

- DialogFragments maintain their lifecycle and modularise your code - win/win.
  - Can convert your old Dialogs using:
    - **onDialogCreate()** method.

- http://prezi.com/fh2uslbr1xs3/the-fragment-transition/
- https://developer.android.com/reference/android/app/DialogFragment.html

Performance

Performance

- Background thread - of course...
  - avoid - **new Thread();**

Performance

- Background thread - of course...
  - avoid - **new Thread();**
- Don't create objects unnecessarily
  - Smaller device ~ Smaller Resources
  - Less room to allocate memory - OOM Errors!

Performance

- Background thread - of course...
  - avoid - **new Thread();**
- Don't create objects unnecessarily
  - Smaller device ~ Smaller Resources
  - Less room to allocate memory - OOM Errors!
- Static over Virtual
  - About 15-20% faster invocation
  - ```
    private static String staticRef = "SomeValue";
    ```
  - ```
    private String virtualRef = "SomeValue";
    ```

Performance

- Background thread - of course...
  - avoid - **new Thread();**
- Don't create objects unnecessarily
  - Smaller device ~ Smaller Resources
  - Less room to allocate memory - OOM Errors!
- Static over Virtual
  - About 15-20% faster invocation
  - ```
    private static String staticRef = "SomeValue";
    ```
  - ```
    private String virtualRef = "SomeValue";
    ```
- Getters n' Setters - Best Programming practice, but slower than accessing a field.

Performance Cont.

Performance Cont.

- For loops
  - ```
    public void two() {

      int sum = 0;
      for (Foo a : mArray) {
        sum += a.mSplat;
      }
    }
    ```
    - Has the most efficient execution on pre/post JIT devices.
- Avoid InnerClass to OuterClass Field access in high performance code spots.
- https://developer.android.com/guide/practices/performance.html
    ```

Base Classes

Base Classes

- MainActivity extends BaseActivity

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment
- What goes in their?

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment
- What goes in their?
  - protected final MyApplication mApp = MyApplication.get();

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment
- What goes in their?
  - protected final MyApplication mApp = MyApplication.get();
  - protected final MyPrefs mPrefs = mApp.getPrefs();

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment
- What goes in their?
  - protected final MyApplication mApp = MyApplication.get();
  - protected final MyPrefs mPrefs = mApp.getPrefs();
  - protected final DatabaseHelper mDB = mApp.getDatabase();

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment
- What goes in their?
  - protected final MyApplication mApp = MyApplication.get();
  - protected final MyPrefs mPrefs = mApp.getPrefs();
  - protected final DatabaseHelper mDB = mApp.getDatabase();
- Singleton pattern is popular in Android.

Base Classes

- MainActivity extends BaseActivity
- ScoreFragment extends BaseFragment
- What goes in their?
  - protected final MyApplication mApp = MyApplication.get();
  - protected final MyPrefs mPrefs = mApp.getPrefs();
  - protected final DatabaseHelper mDB = mApp.getDatabase();
- Singleton pattern is popular in Android.
  - Careful not to hold reference to unneeded classes!

Any questions..?

Examples and Demo's

ActionBar

- Standard UI/UX Experience
- Play/Facebook/Twitter/Maps/Google+ etc..
- ActionBarSherlock - *Jake Wharton*
  http://actionbarsherlock.com/
- Always use it (Unless 3.0+ only)
- Solves menu deprecation issues

- Example

Draggable Sort ListView

- DragSortListView - *Carl A. Bauer*
  https://github.com/bauerca/drag-sort-listview

- Example..

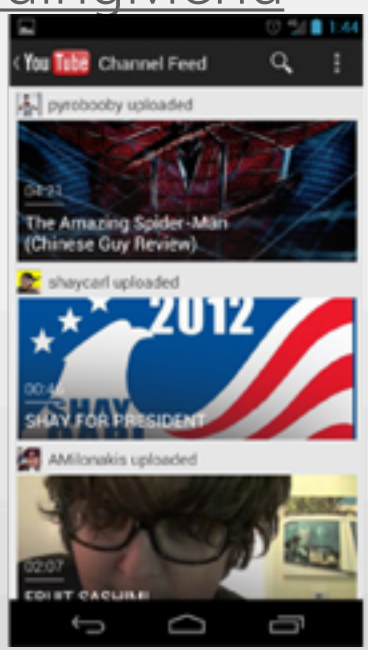
- Other
  - cwac-touchlist - *Mark Murphy*
    https://github.com/commonsguy/cwac-touchlist
    Although ***Depreciated*** now

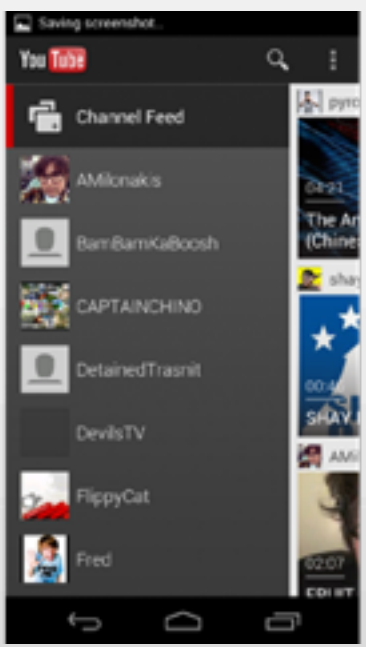
RoboSpice - async network lib

- Async Done Right
- Built of Google 2010 IO thought process
  - Decoupling REST and UI components correctly
- https://github.com/octo-online/robospice
- https://play.google.com/store/apps/details?id=com.octo.android.robospice.motivations

SlidingMenu

- Creates a singular parallel user flow
- Google Plus / You tube / Facebook
- Plays nice with ActionBarSherlock
- Jeremy Feinstein / SlidingMenu
  https://github.com/jfeinstein10/SlidingMenu





Bluetooth example

- SDK Sample

Books

- http://commonsware.com/
  Almost wrote the book on android... regularly updated and LOTs of smaller examples
- http://www.amazon.co.uk/Professional-Android-Application-Development-Guides/dp/1118102274/ref=ntt_at_ep_dpt_1
  Professional Android 2 App Development - By Reto Meier
  One of the Android evangelists at google, knows his stuff.
- http://www.amazon.co.uk/Smashing-Android-UI-Magazine-Book/dp/1118387287/ref=sr_1_1?s=books&ie=UTF8&qid=1351591674&sr=1-1
  the 'Smashing' books are good too although a little less well established.

Thanks to

- Presentation Ideas
  - @mubaloo
  - @scottyab
  - @commonsguy
  - @JakeWharton
  - @corey_latislaw

Questions?