



UI Design and Development

+Roman Nurik

+Nick Butcher

# Agenda



1. Designing for Android
2. Layouts and resources
3. Tablet considerations
4. System UI integration
5. UI prototyping



# Designing for Android

# Design for...



- Touch
  - Interact primarily with your fingers
  - Expect direct manipulation
- Mobile
  - Often on the go
  - Often without network connectivity
- Heterogeneity
  - Different screen sizes and densities
  - Different hardware features
  - Different OS versions

# Key principles

1 + 2



"Pictures are faster than words."

"Only show what I need when I need it."

"Make the important things fast."

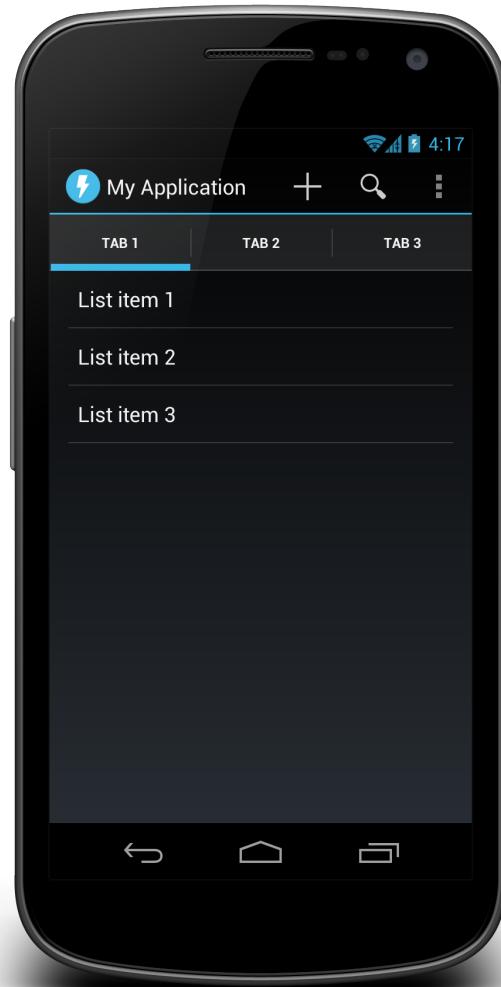
"Do the heavy lifting for me."



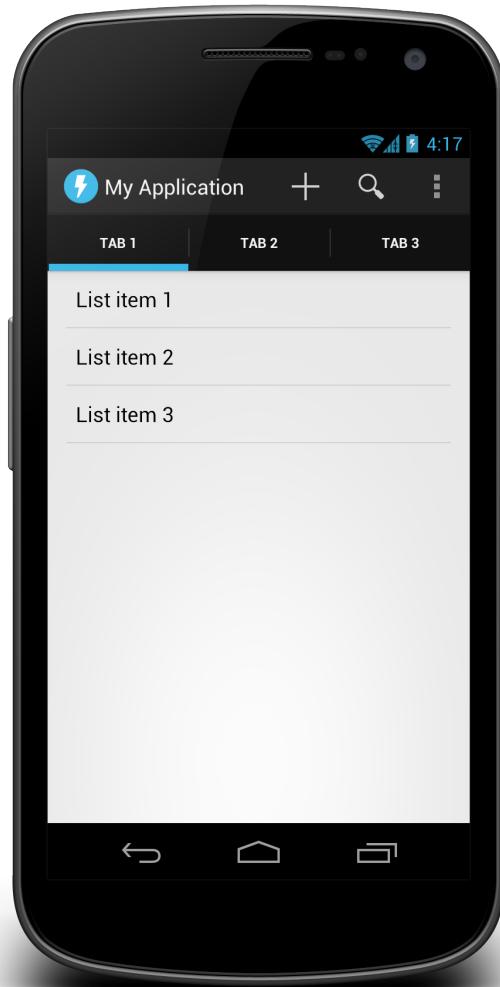
"Holo" visual language



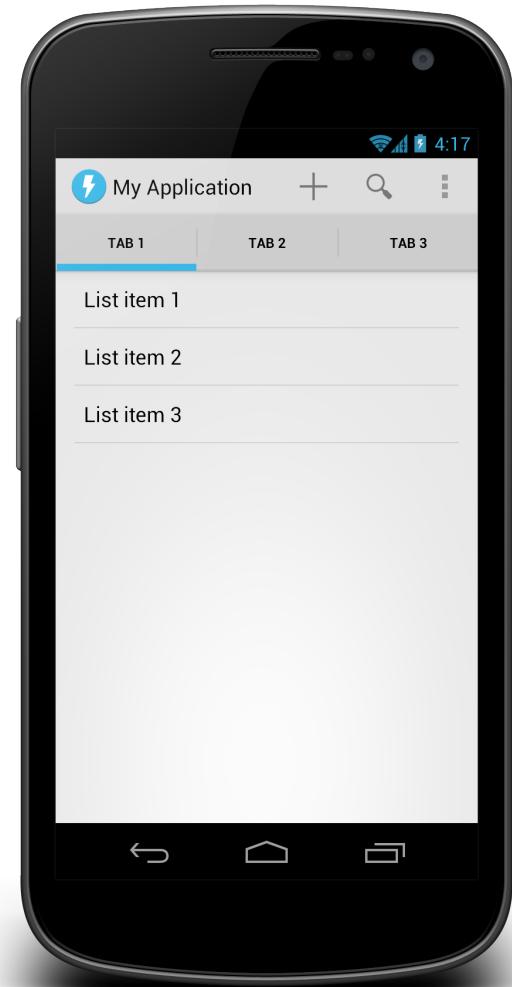
# Holo variations



Dark



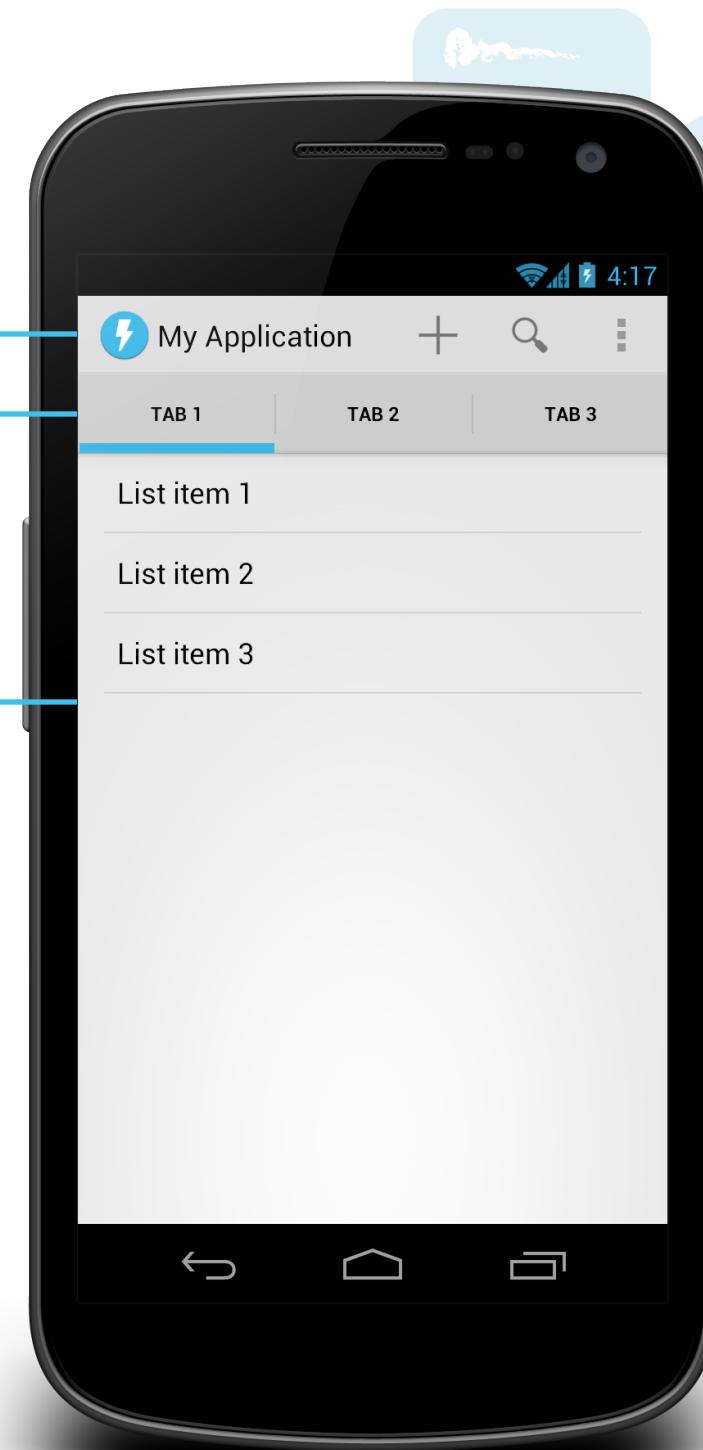
Dark Action Bar



Light

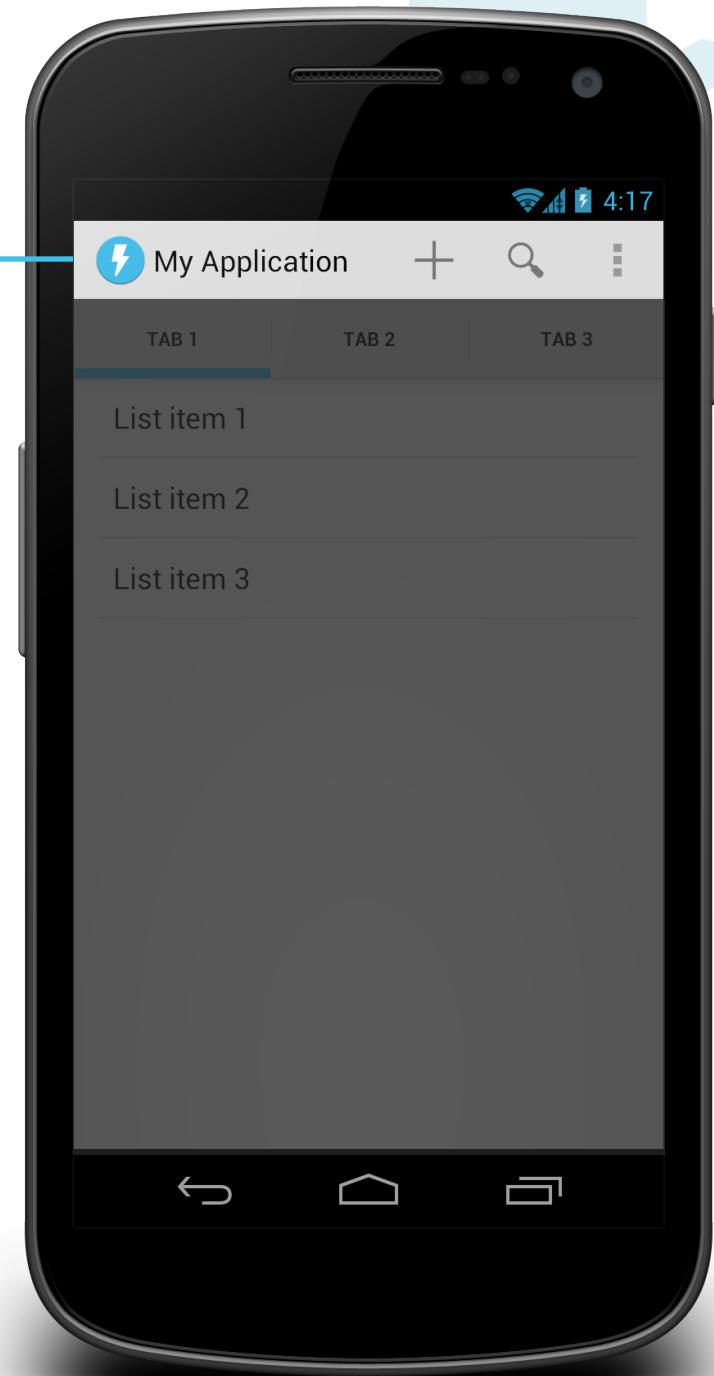
# Activity UI structure

Action bar  
Tabs  
Content  
(activity layout)

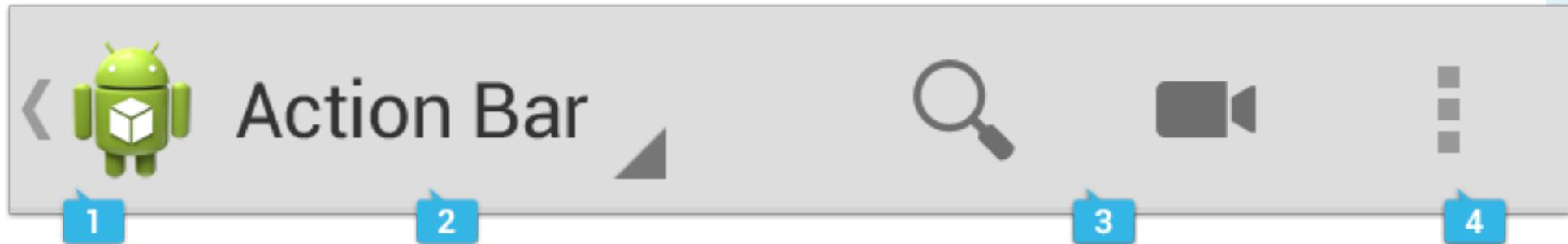


# Activity UI structure

Action bar

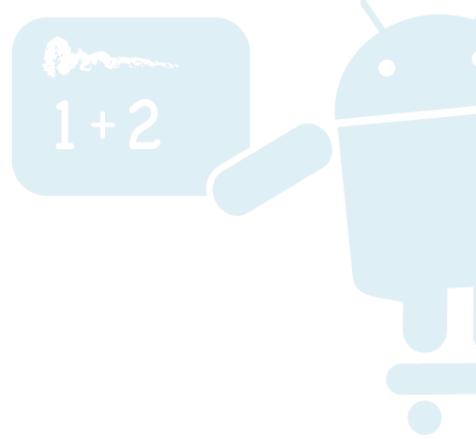


# Action bar



1. App icon and optional Up caret
2. View control (Title/tabs/dropdown)
3. Action buttons
4. Action overflow

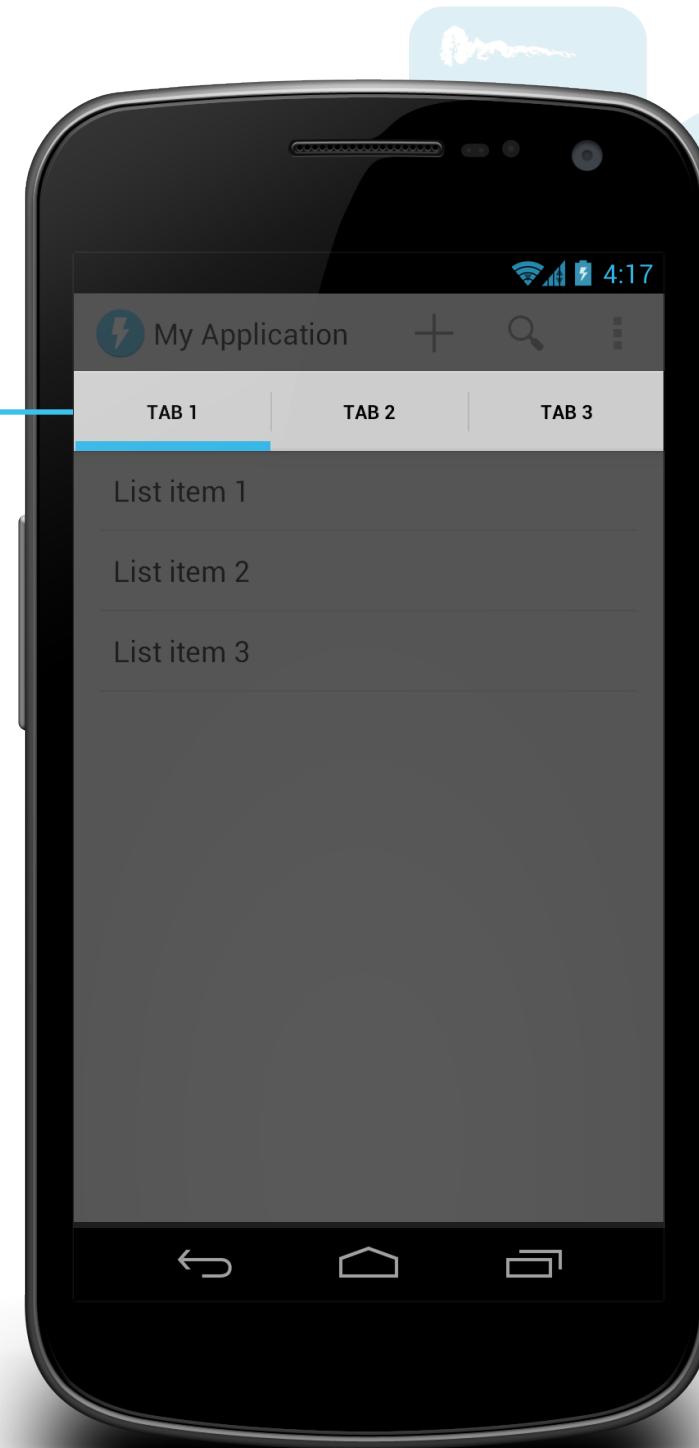
# Action bar



- Android 3.0 and above
- Automatically part of Holo themes
- Customize:
  - `getActionBar().setDisplayOptions()`
  - `getActionBar().setNavigationMode()`

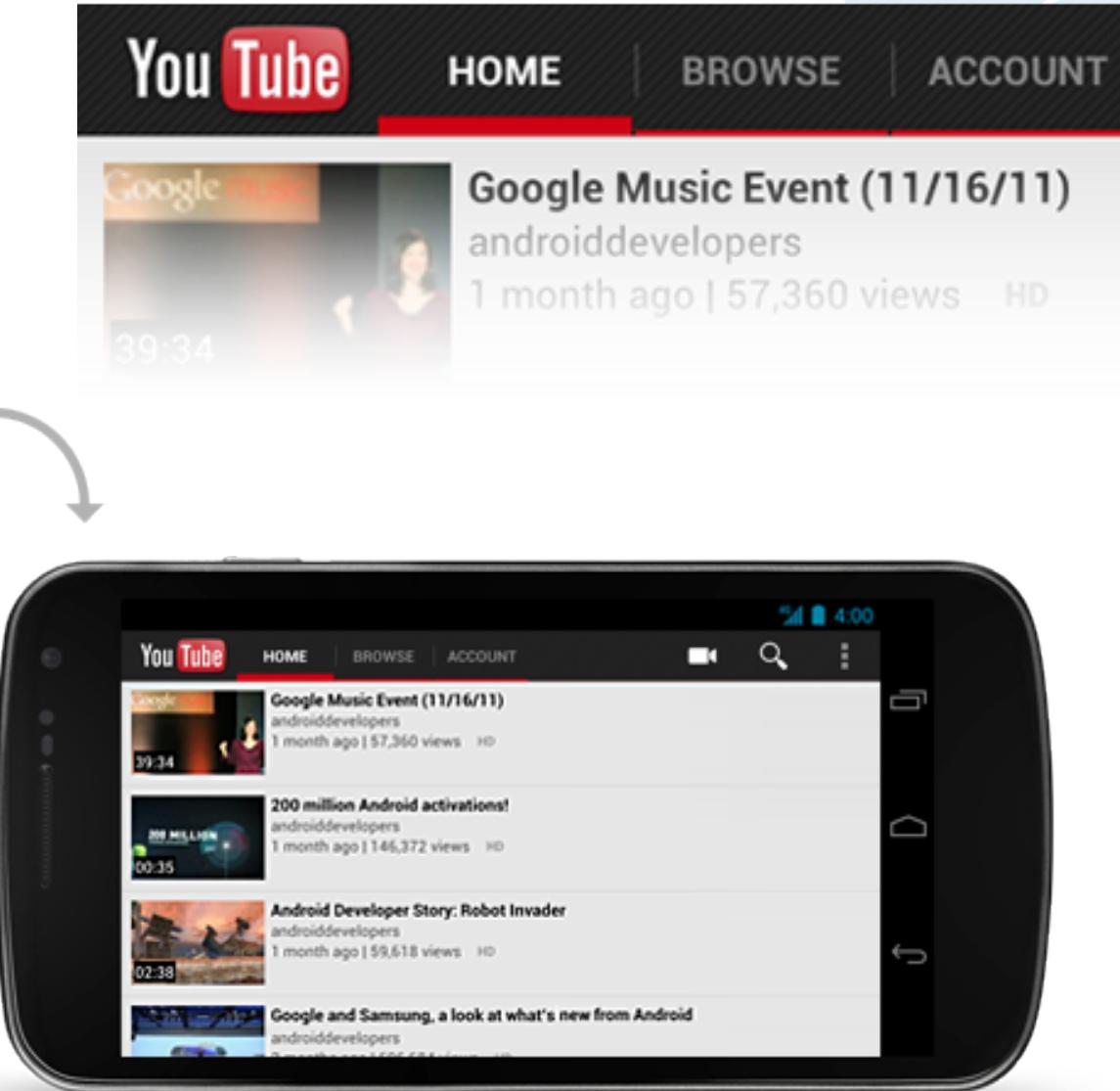
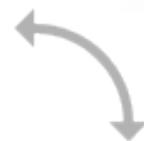
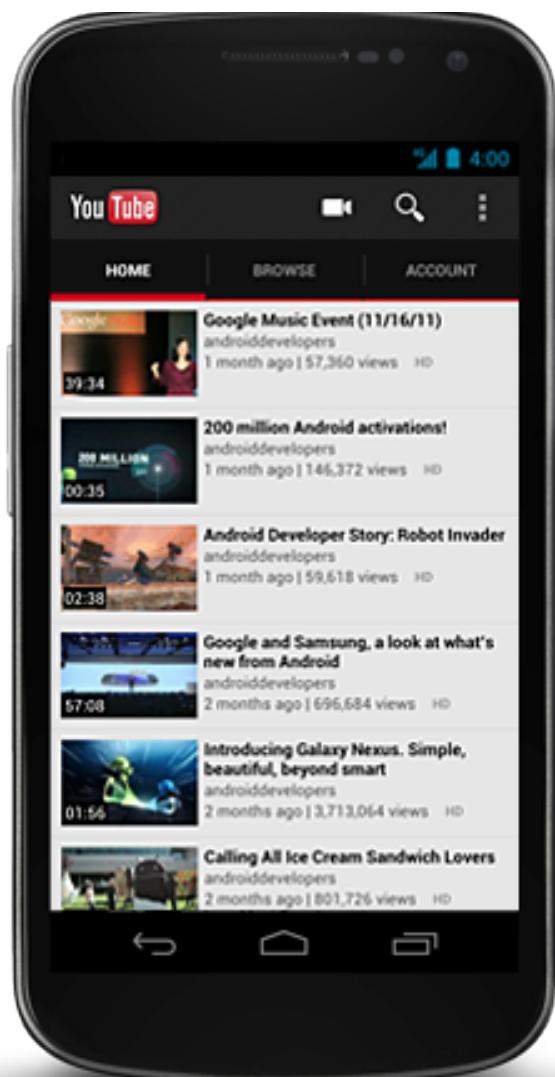
# Activity UI structure

Tabs

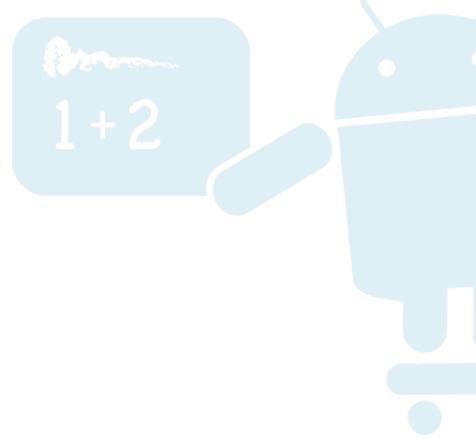


# Tabs

1 + 2



# Tabs

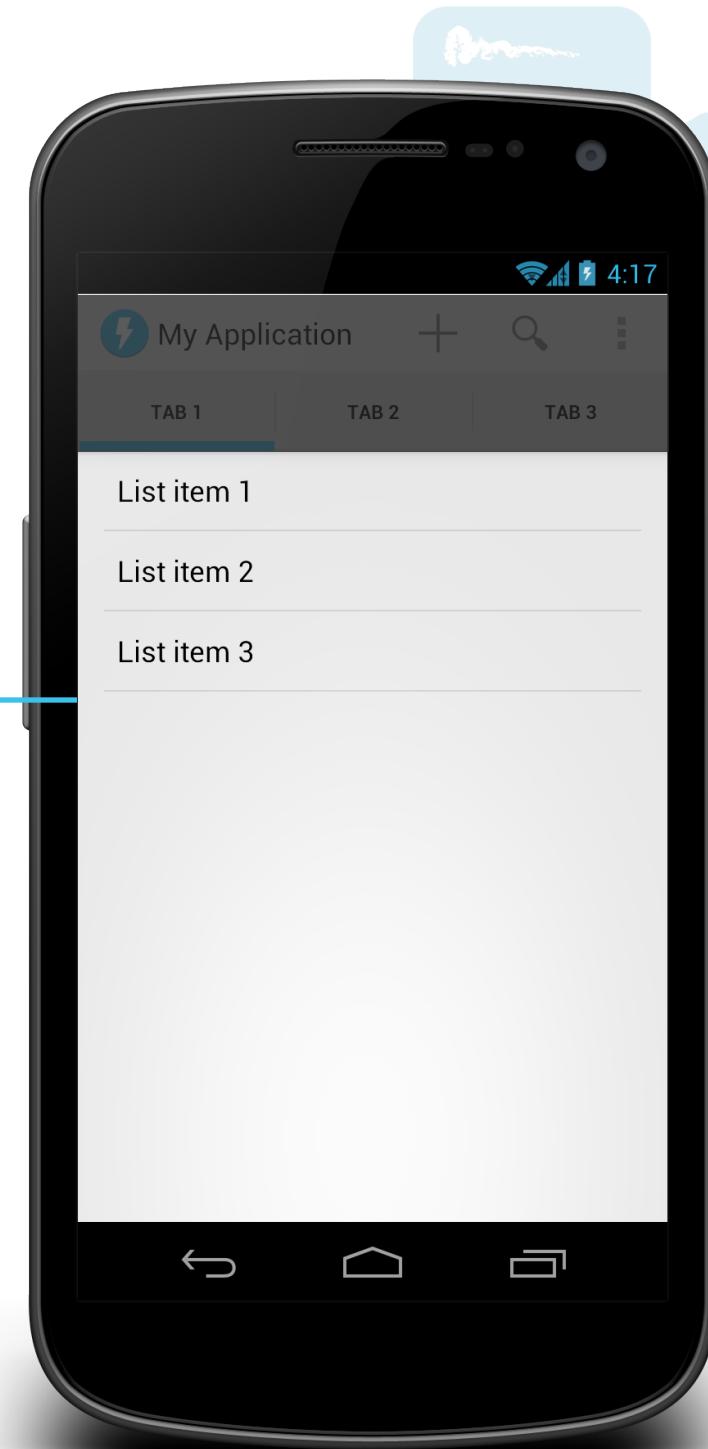


- Part of the ActionBar APIs
- Usually gesture-enabled using ViewPager

```
getActionBar().setNavigationMode(NAVIGATION_MODE_TABS);  
  
ActionBar.Tab tab = actionBar.newTab();  
tab.setText("Tab 1");  
tab.setTabListener(this);  
getActionBar().addTab(tab);
```

# Activity UI structure

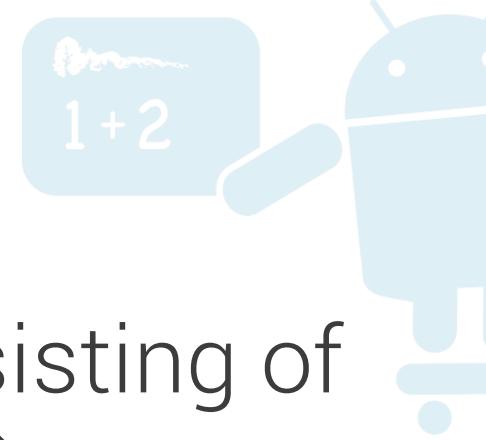
Content  
(activity layout)



# Layouts and resources



# Layout system



- The UI for an activity is a tree consisting of view groups and views (leaf nodes).

```
<view group>
    <view group>
        <view>
    <view group>
        <view>
        <view>
```

- Most commonly defined in XML under res/layout/.

# Views and View Groups



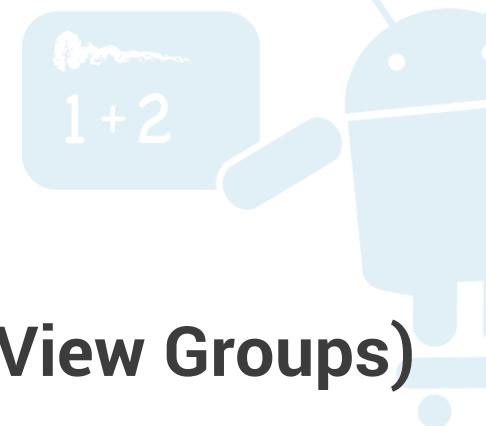
## Views

- Reusable individual UI components
- Optionally interactive (clickable/focusable/etc.)
- Bare minimum functionality is to draw themselves

## View Groups

- Ordered list of Views and View Groups
- In charge of positioning and sizing their child views and layouts
- Simple layouts and more complex groups (e.g. ListView)

# Views and View Groups



## Views

- TextView
- EditText
- Spinner
- ImageView
- Button
- WebView
- SurfaceView
- Your own custom views

## Layouts (simple View Groups)

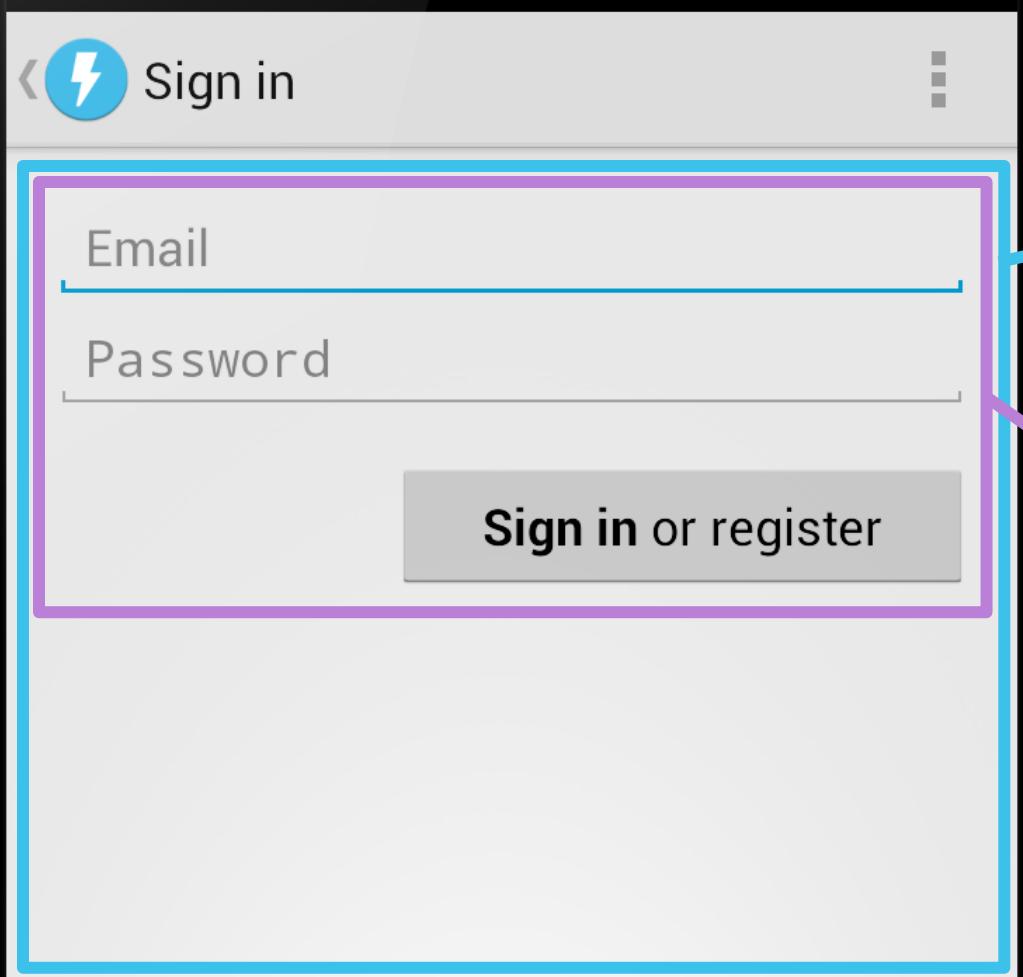
- FrameLayout
- LinearLayout
- RelativeLayout
- GridLayout
- Your own custom layouts

## Complex View Groups

- ScrollView
- ListView

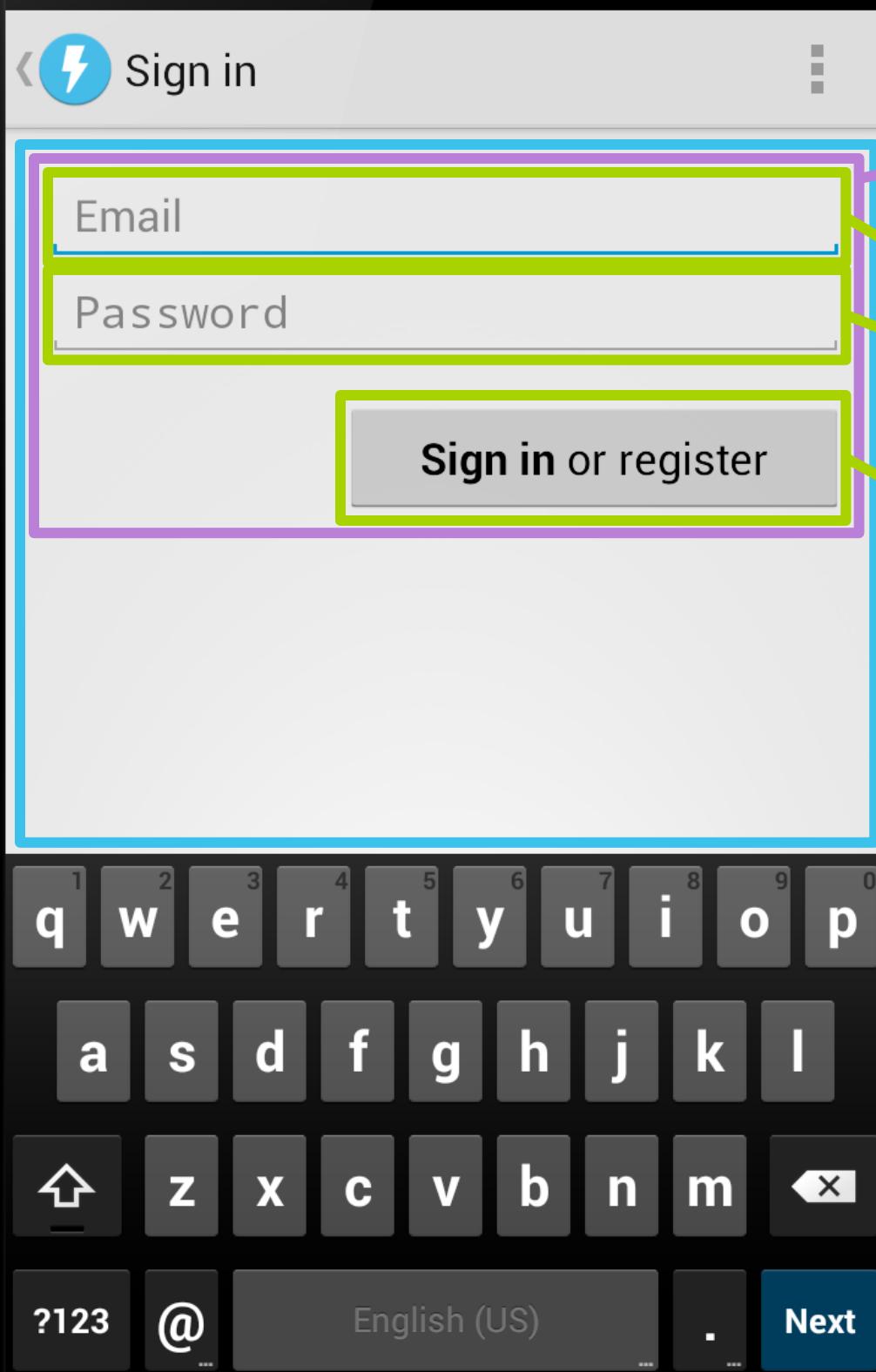
# Anatomy of a simple layout





<FrameLayout>

<LinearLayout  
    orientation="vertical">



`<LinearLayout  
    orientation="vertical">`

`<EditText>`

`<Button>`

`<ScrollView>`

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <EditText
            android:id="@+id/email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_email"
            android:inputType="textEmailAddress"
            android:singleLine="true" />

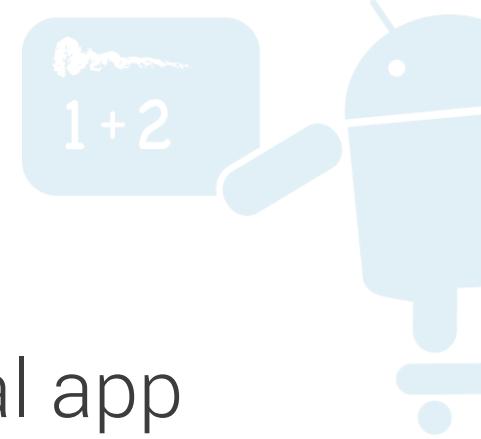
        <EditText
            android:id="@+id/password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_password"
            android:inputType="textPassword"
            android:maxLines="2"
            android:singleLine="true" />

        <Button
            android:id="@+id/sign_in_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_marginTop="16dp"
            android:paddingLeft="32dp"
            android:paddingRight="32dp"
            android:text="@string/action_sign_in_register" />
    </LinearLayout>
</ScrollView>

```



# App resources



```
▼ res
  ► drawable
  ► drawable-hdpi
  ► drawable-mdpi
  ► layout
  ► layout-port
  ► layout-land
  ► layout-large-land
  ► layout-large-port
```

- One universal app binary contains all resources
- System chooses at runtime which resources to use

res	
►  <b>drawable</b>	Drawable XML
►  <b>drawable-hdpi</b>	PNGs, 9-patch PNGs, optimized for multi. densities
►  <b>drawable-mdpi</b>	
►  <b>layout</b>	
►  <b>layout-port</b>	Layout XML optimized for physical size and orientation
►  <b>layout-land</b>	
►  <b>layout-large-land</b>	
►  <b>layout-large-port</b>	
►  <b>values</b>	Strings, styles, themes, etc.
►  <b>values-en</b>	
►  <b>values-fr</b>	Strings XML localized for your target regions
►  <b>values-ja</b>	

# Screen density and DIP units



DIP units keep things the  
**same physical size** across any screen.

1 dip = 1 pixel @ MDPI (160 dpi)

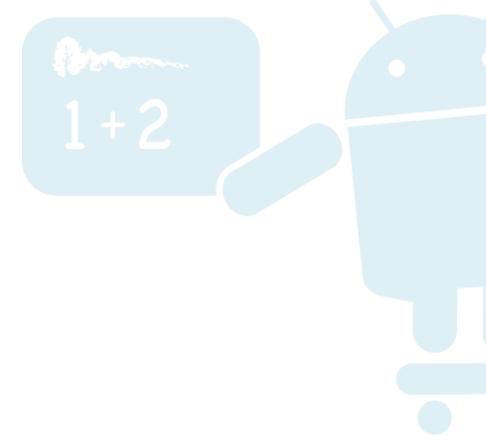
1 dip = 2 pixels @ XHDPI (320 dpi)



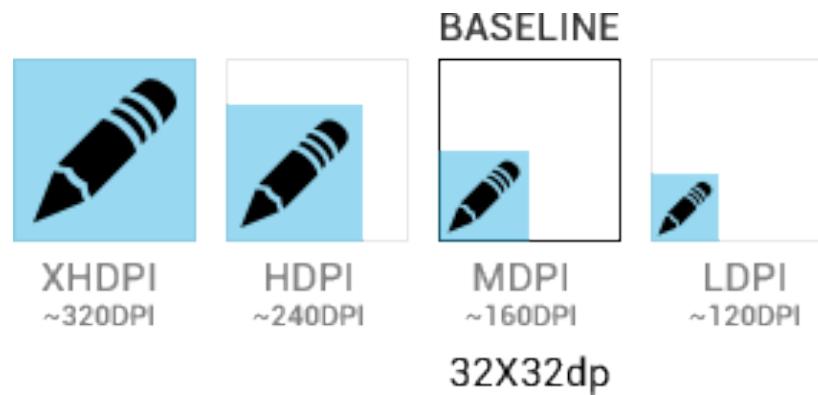
**Q:** What is the Nexus 7's screen resolution in DIPs if it's **1280x800 px** and **213dpi**?

**A:** **~960x600 dip**

# Screen density and DIP units



Icons and other PNG files should generally be provided for multiple densities

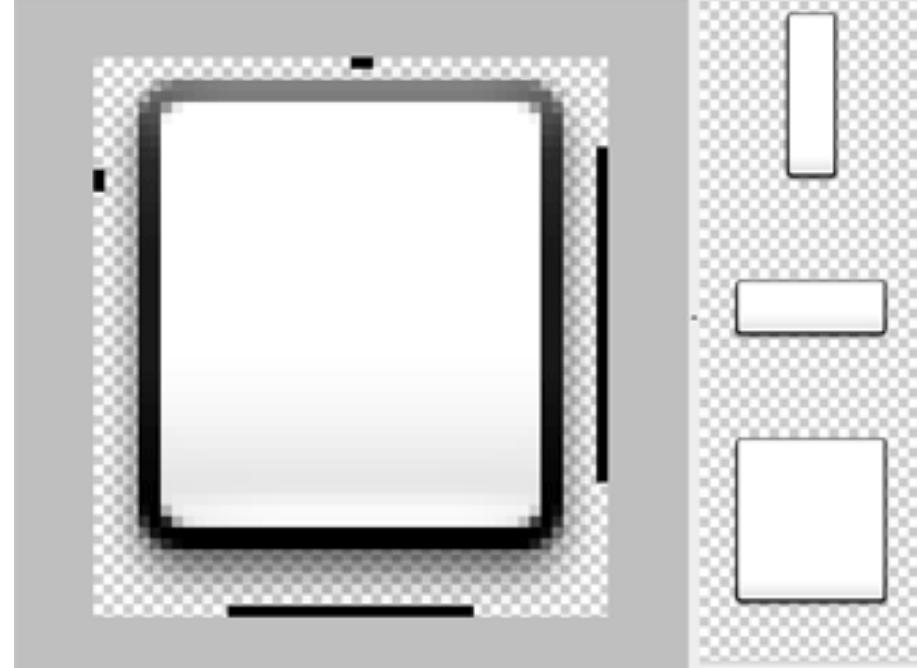


# Drawables



- Bitmaps (.png)
- 9-patches (.9.png)
- State Lists (.xml)
- Basic Vector Shapes (.xml)
- Color Drawables (.xml)

## 9-patches – foo.9.png



- Border pixels indicate stretchable regions
- Make density-specific versions (-**xhdpi**)

# State Lists (selector)



## ► **drawable**

foo.xml:

```
<selector>
    <item android:drawable="@drawable/foo_disabled"
          android:state_enabled="false" ... />
    <item android:drawable="@drawable/foo_pressed"
          android:state_pressed="true" ... />
    <item android:drawable="@drawable/foo_focused"
          android:state_focused="true" ... />
    <item android:drawable="@drawable/foo_default" />
</selector>
```

# State Lists (selector)



## ► **drawable-mdpi**

foo\_default.png



foo\_disabled.png



foo\_focused.png



foo\_pressed.png



## ► **drawable-hdpi**

foo\_default.png



foo\_disabled.png



foo\_focused.png



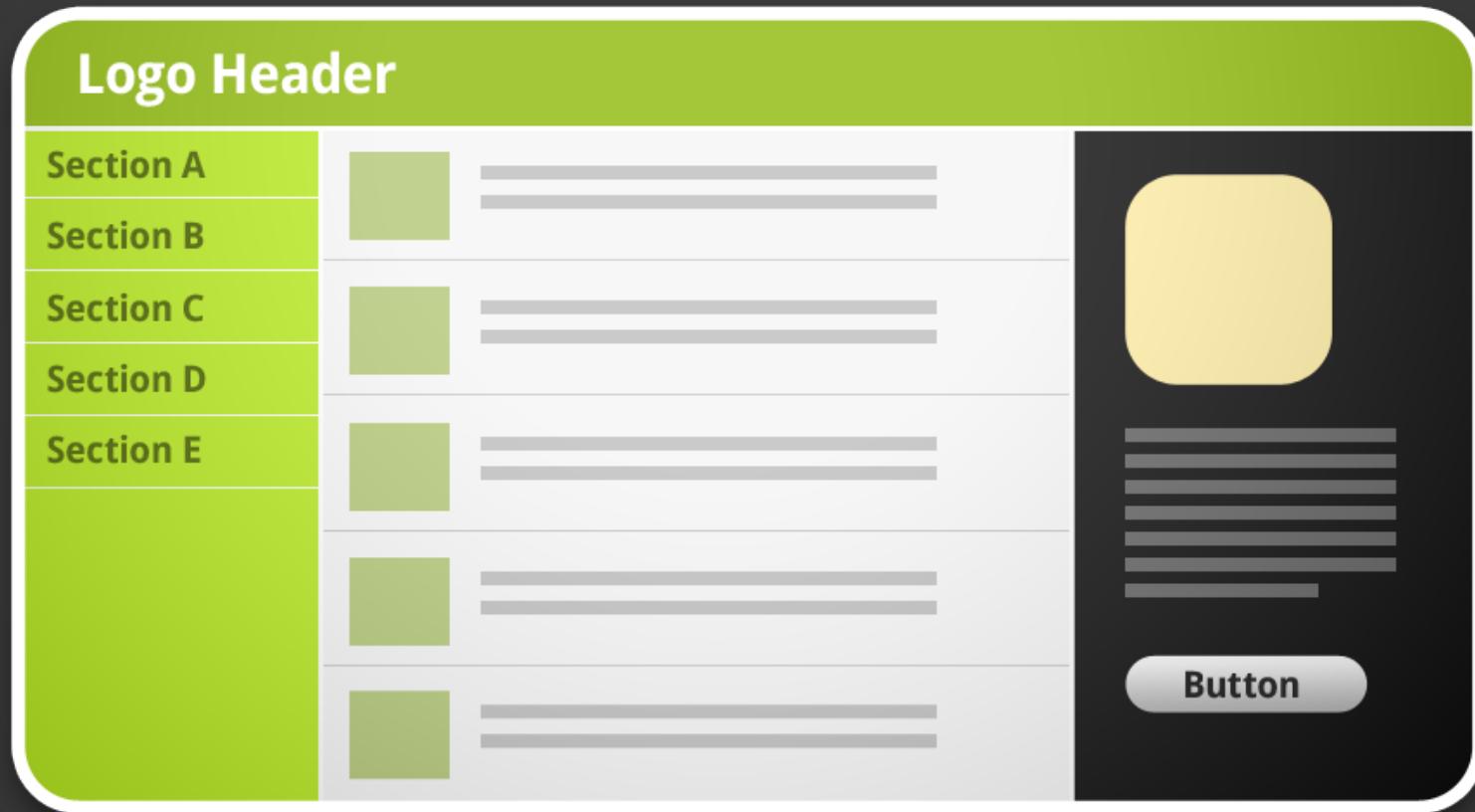
foo\_pressed.png





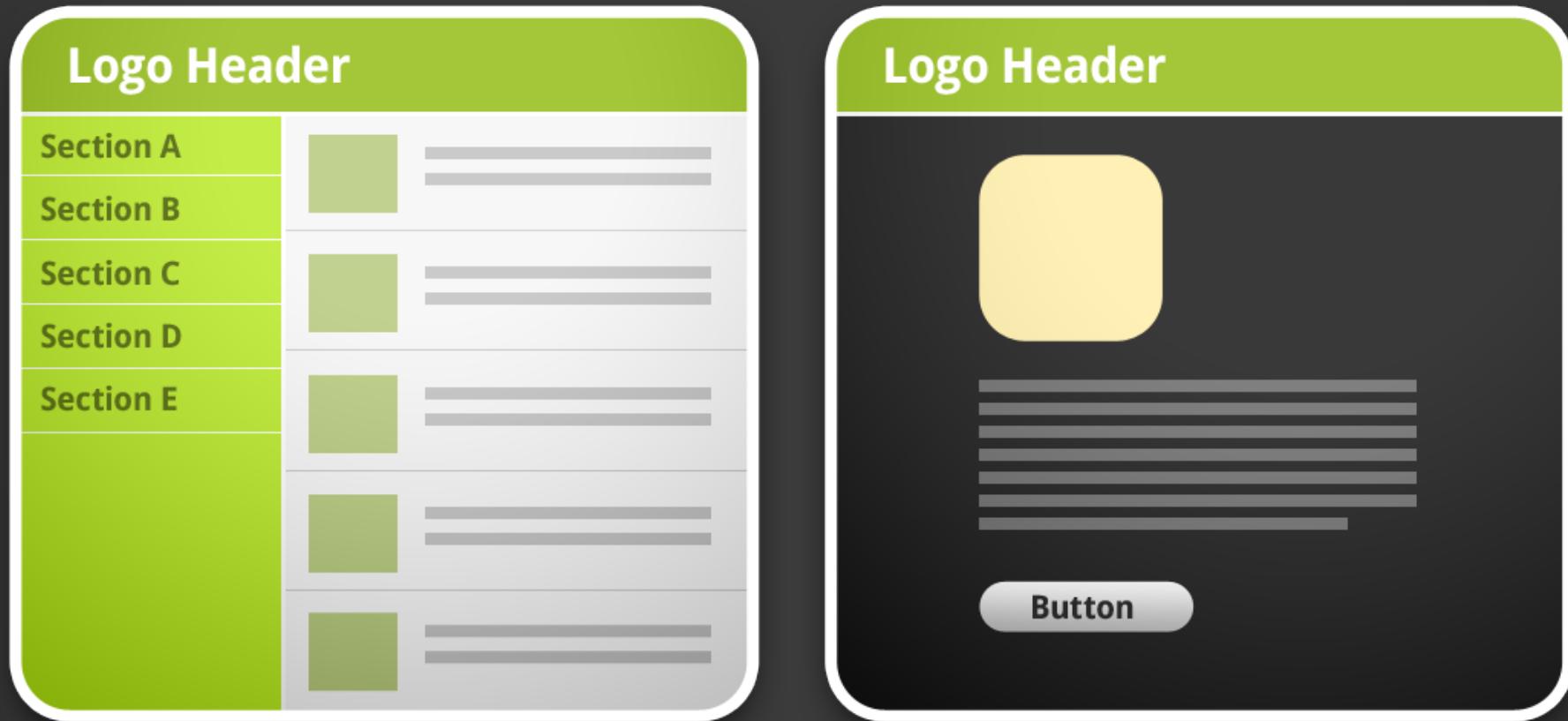
# Tablet considerations

# Information hierarchy and flow



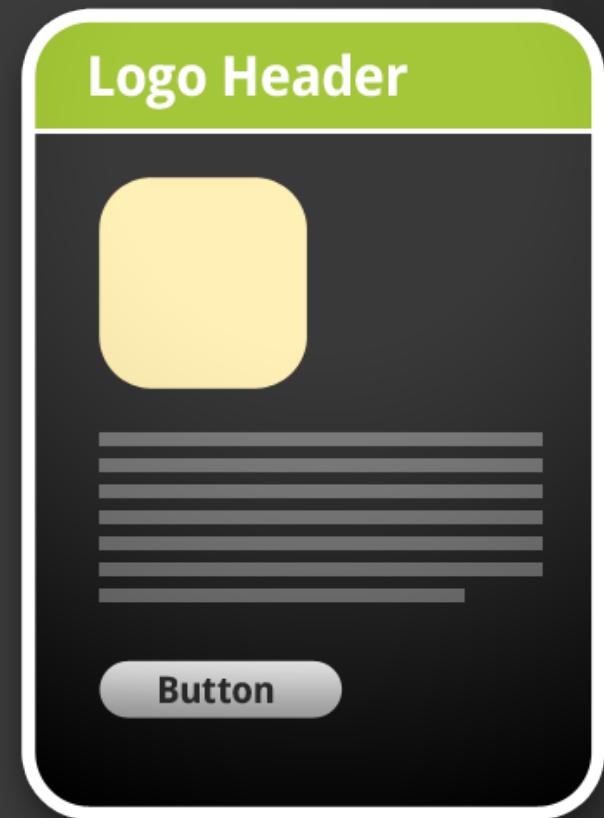
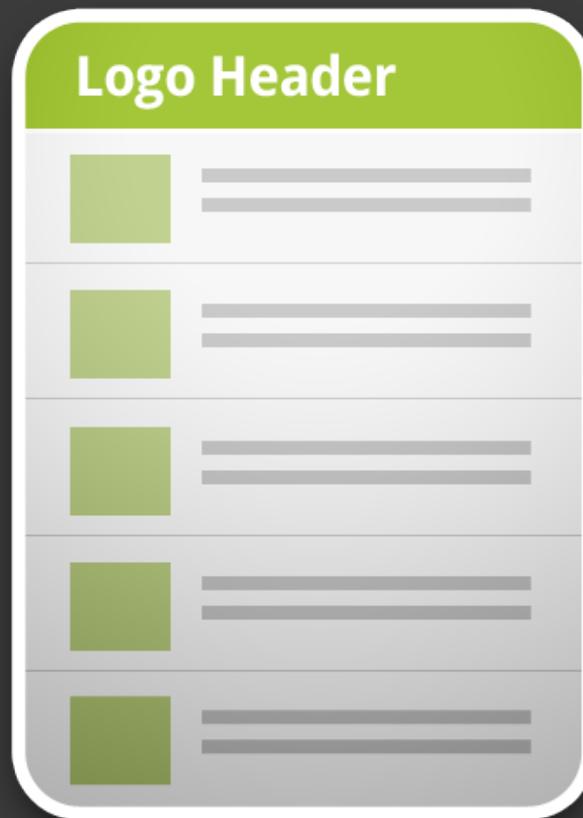
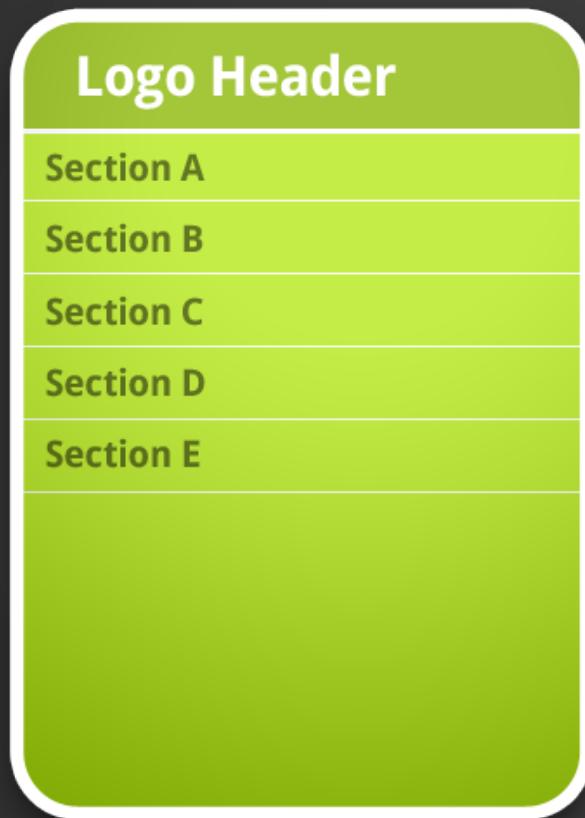
Traditional desktop app or website

# Information hierarchy and flow



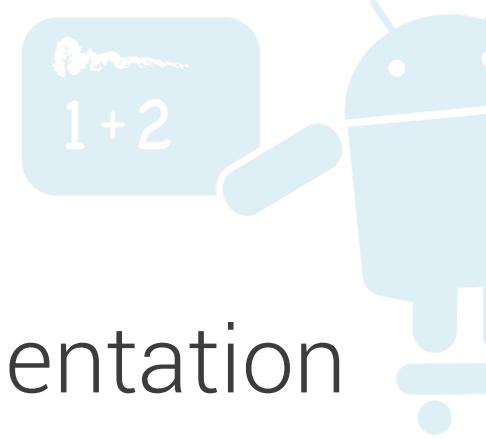
Tablet or mini desktop app

# Information hierarchy and flow



Mobile phone

# Fragments

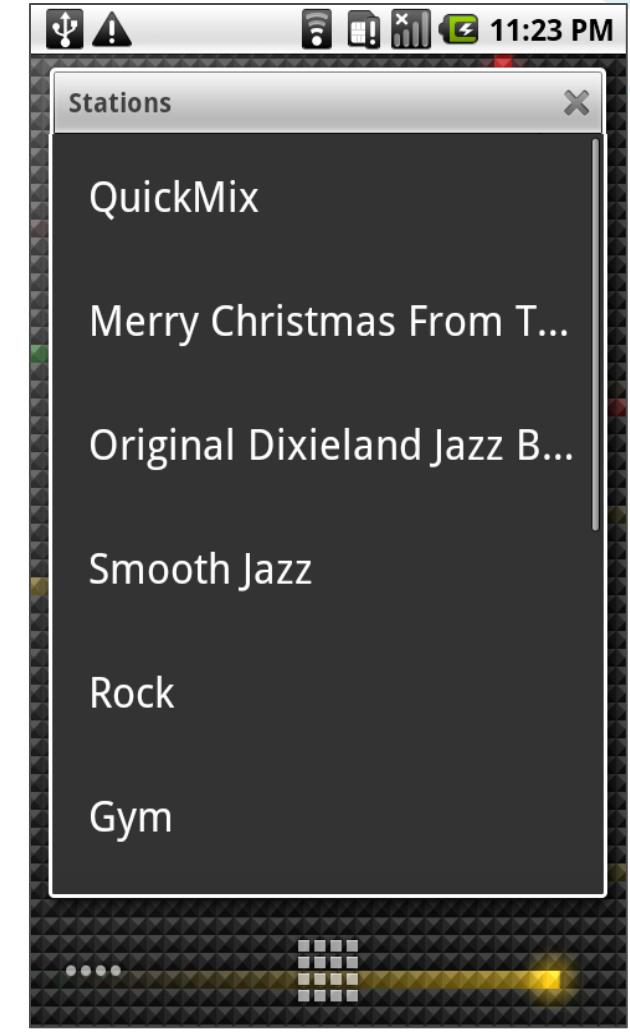
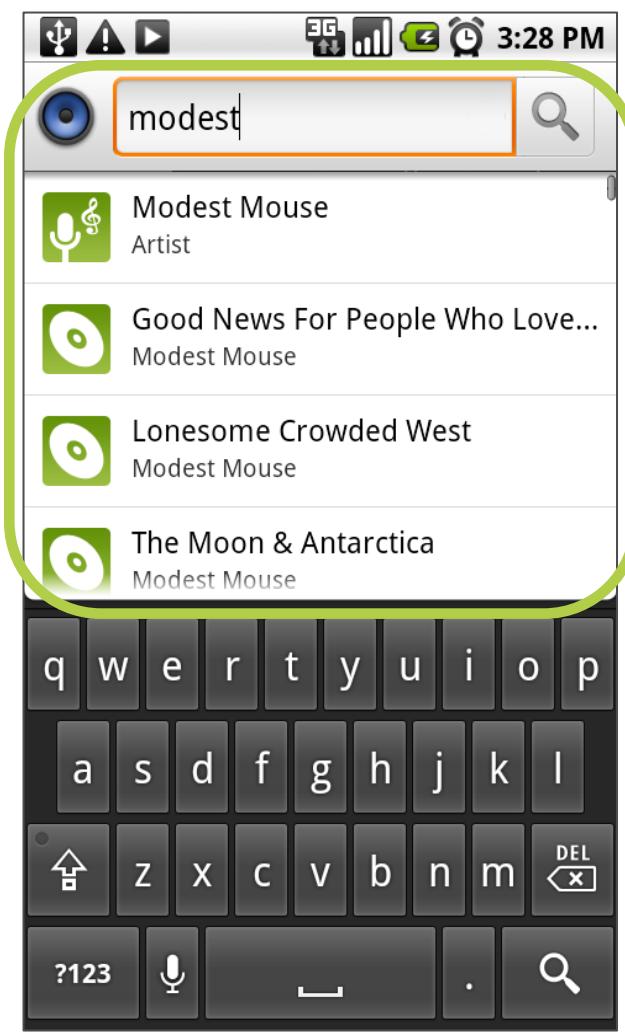
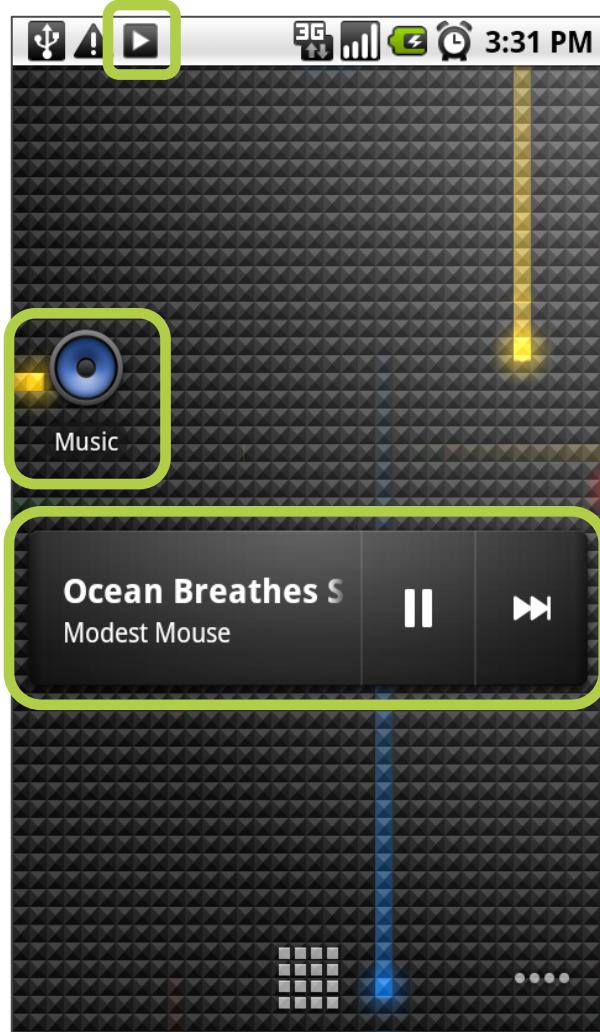


- A single piece of UI and/or implementation of an Activity, defined in its own class
- Help with supporting phones + tablets
- <fragment> in layout XML
  - Automatically instantiates the given fragment
  - Acts as a placeholder for the fragment's view to be inserted in that part of the layout tree



# System UI integration

# System UI integration



# UI prototyping



# Why prototype?



- Record your ideas and asses their real-world feasibility
- Test your ideas and **rapidly iterate**
  - See which work and which don't, evolve them
- Map out user flow and activity diagrams
  - Re-arrange/add/remove interactions quickly
  - Scope UI complexity
  - Plan out intra-app “Intent-based API”

$1 + 2$

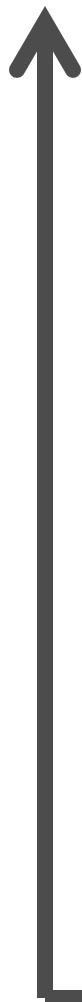


Without prototyping,  
you'll find yourself  
doing a lot of  
UI refactoring. ☹

# Lots of wireframing tools



Time/Effort



Photoshop

Fireworks

OmniGraffle  
(Mac)

Keynote/  
Powerpoint

Balsamiq

Pencil  
(Firefox addon)

Eclipse  
Layout  
Editor

Sketches

Fidelity

# Interactive (clickable) wireframes



**Time/Effort**



Photoshop

Fireworks

OmniGraffle  
(Mac)

Keynote/  
Powerpoint

Balsamiq

Pencil  
(Firefox addon)

Sketches

Eclipse  
Layout  
Editor

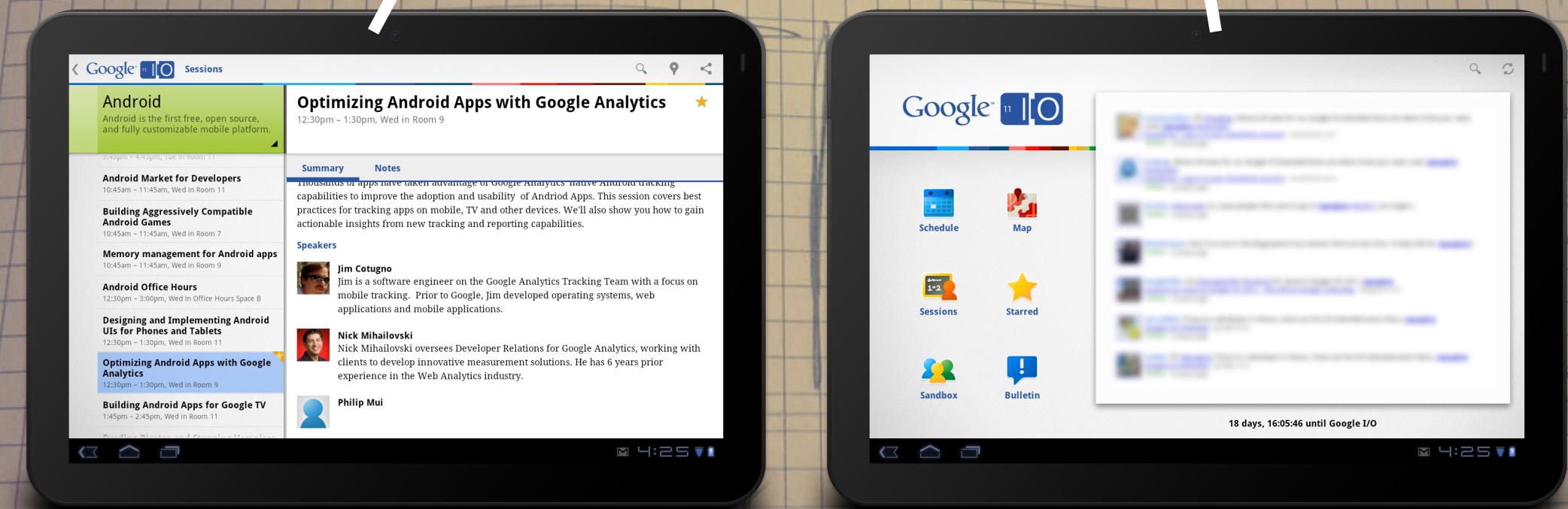
**Fidelity**

$1 + 2$

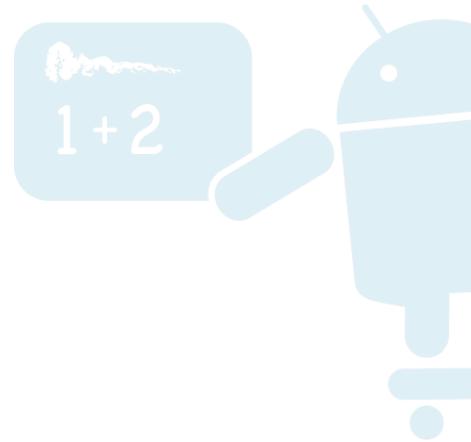


Always start with  
pencil and paper.

(or a whiteboard)



# Demos



1. "Pencil" (Firefox add-on)
  - Android stencils at <http://j.mp/androiduiutils>
2. Fireworks-generated interactive PDF
3. Eclipse Android Layout Editor

Exercise:

Begin sketching!