

Google™



# Pixel Perfect Code: How to Marry Interaction & Visual Design the Android Way

Chris Nesladek  
May 27, 2009

Post your questions for this talk on Google Moderator:  
[code.google.com/events/io/questions](http://code.google.com/events/io/questions)




## Note:

All examples that display a cursor are meant to represent a finger touching the screen. Our emulator can't do that...yet!



## My goal today:

Help you use the Android framework to enable richer user experiences when building your applications.

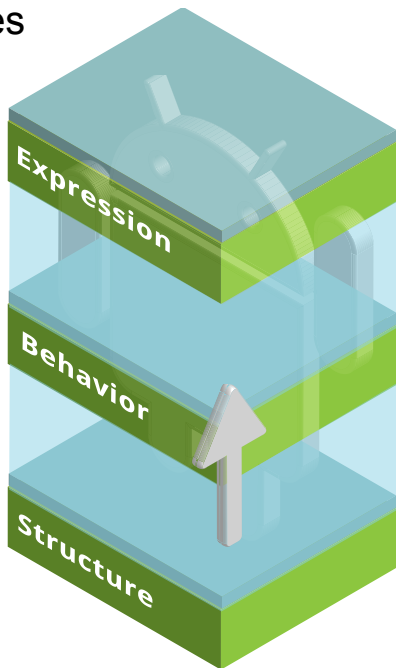


# Introduction

What we're going to explore together today...

- Framework Structure & Flow
  - Five things that make us unique
- Framework Basics
  - Hardware & Software
  - Portrait & Landscape
  - Focus & Menus
- Onscreen Behavior
  - Bringing screens to life with what we offer
  - Expected interactions & unexpected delight
- Expression
  - The 20% test
  - Tips and Tricks
- What our plans are for the future

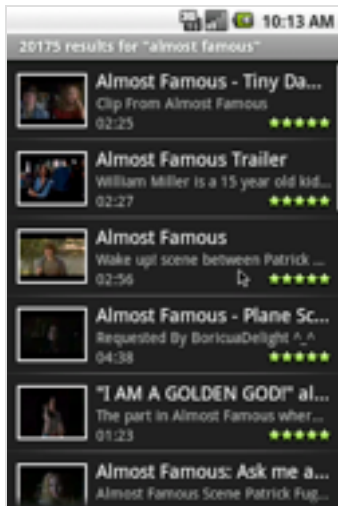
# Core Principles



# Framework Structure & Flow

What makes the Android User Experience unique

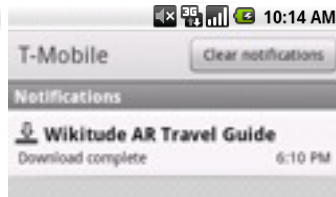
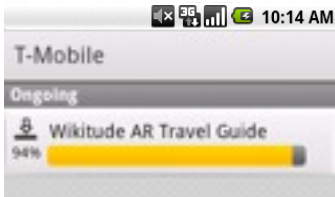
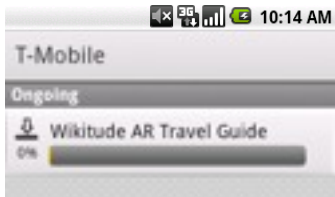
- Multi-tasking supported by background processing



# Framework Structure & Flow

What makes the Android User Experience unique

- Ongoing and event-driven notifications



# Framework Structure & Flow

What makes the Android User Experience unique

- Real-time data views through widgets and live folders

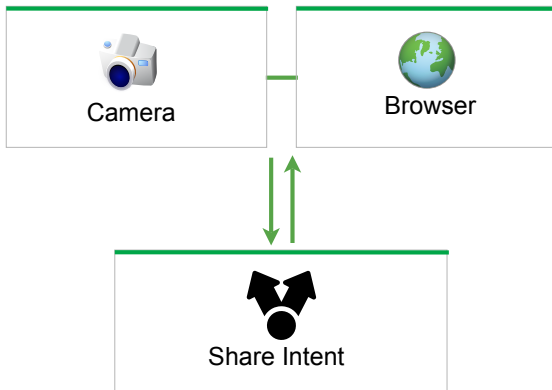




# Framework Structure & Flow

What makes the Android User Experience unique

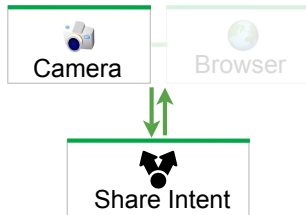
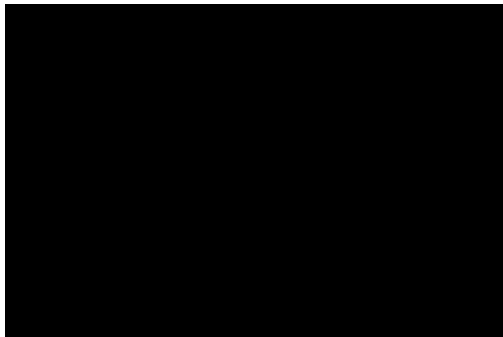
- Any application can pick and choose which intents they want to take



# Framework Structure & Flow

What makes the Android User Experience unique

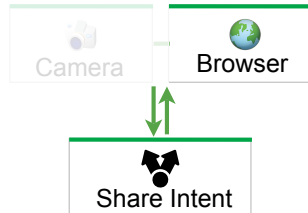
- Any application can pick and choose which intents they want to take



# Framework Structure & Flow

What makes the Android User Experience unique

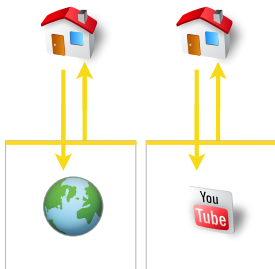
- Any application can allow the spread of data to other applications



# Framework Structure & Flow

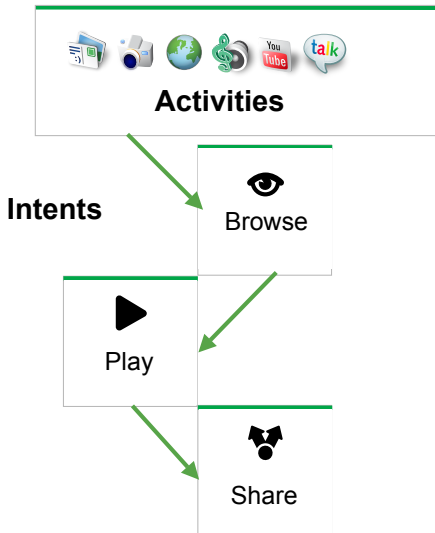
What makes the Android User Experience unique

- Use the hooks, re-use of components from intents is easy



vs.

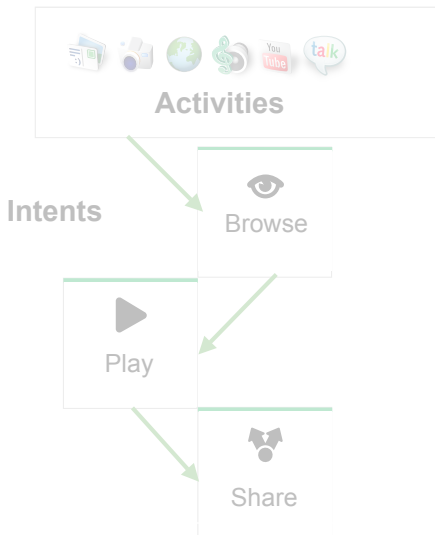
In and Out



# Framework Structure & Flow

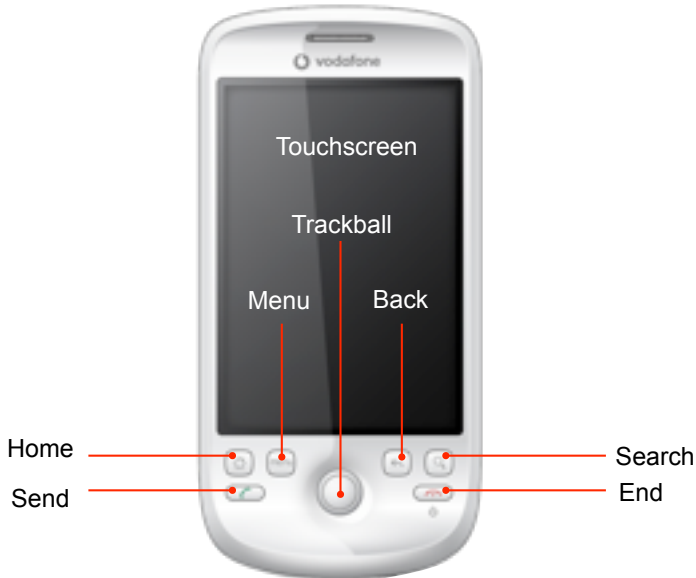
What makes the Android User Experience unique

- Use the hooks, re-use of components from intents is easy



# Framework Basics

- The representative Android device is a happy marriage of hardware and software



# Framework Basics

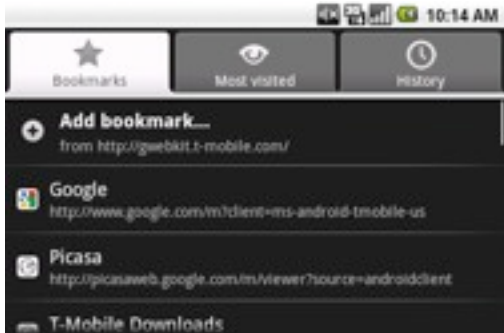
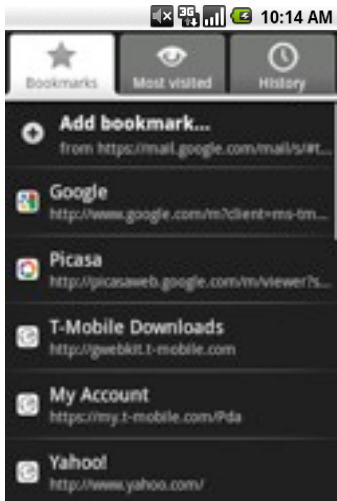
- Portrait & Landscape



- Definitely keep it in mind
- SDK can support different screen sizes and densities
- 99% of our building block layouts will accommodate landscape

# Framework Basics

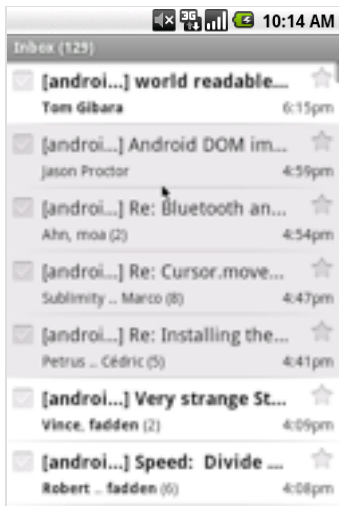
- Portrait & Landscape





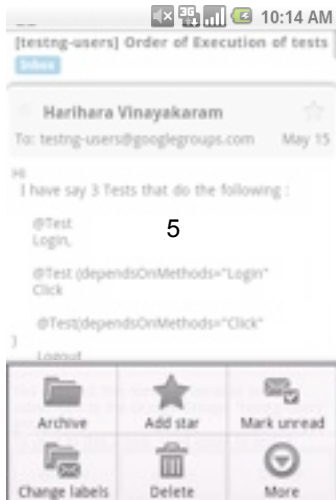
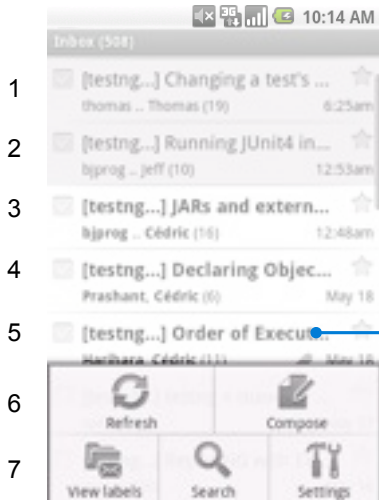
# Framework Basics

## Focus & Menus



# Framework Basics

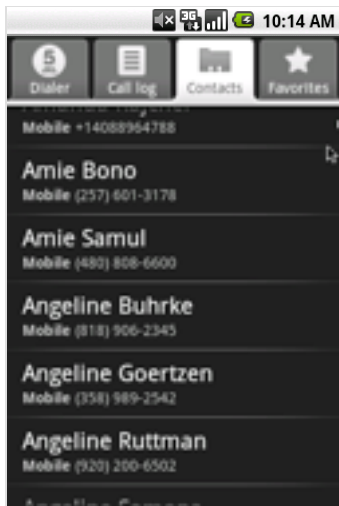
Main menus are global; they relate to an activity as a whole



More menu fits 5 if 6 use more

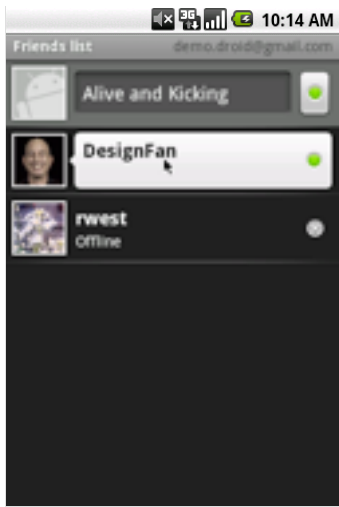
# Framework Basics

Contextual menus aka Long Press are focused on a specific item



# Framework Basics

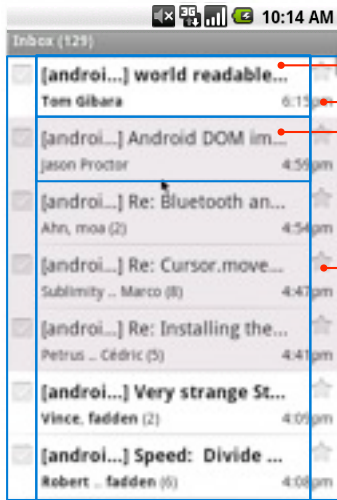
More menus let you add less relevant tertiary information



# Framework Structure

A basic example

- Gmail



Unread

Time

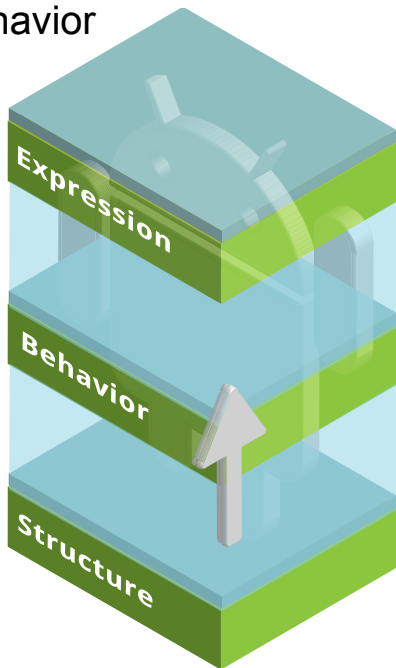
Read

Importance

## Some things to keep in mind

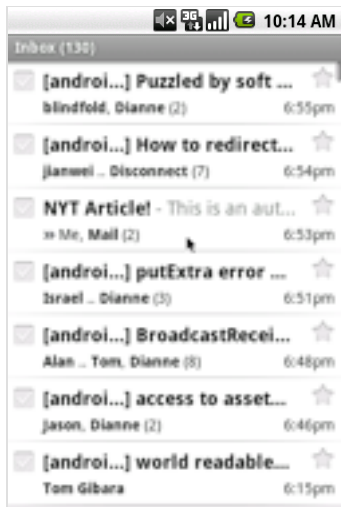
- Design for speed and simplicity
  - Keep hierarchy as flat as possible
  - Minimize onscreen actions
  - Load data rather than asking a user to wait to see a fully fleshed out page aka lazy load
- Think about activity streams as opposed to linear actions
  - Design your app to have hooks across the framework
    - Make it so users can act on the data they see

# Onscreen Behavior



# Onscreen Behavior

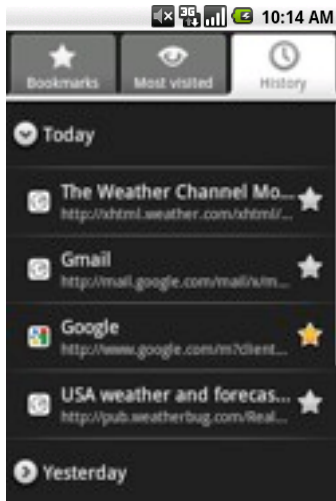
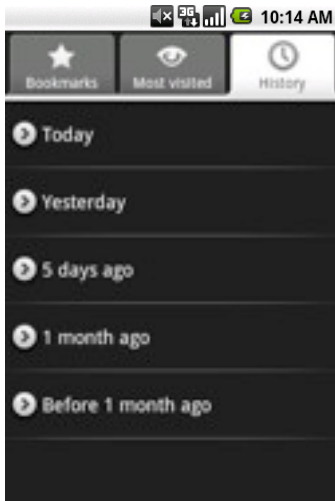
Bringing screens to life with what we offer





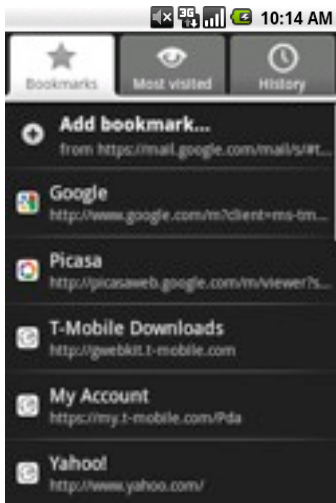
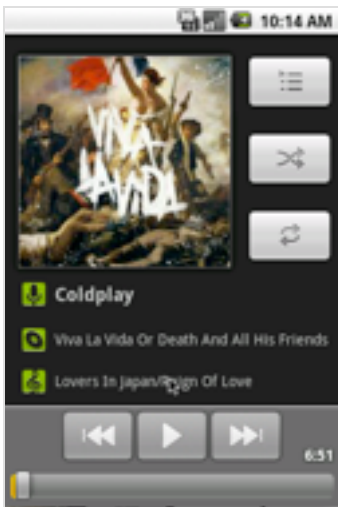
# Onscreen Behavior

Bringing screens to life with what we offer



# Onscreen Behavior

Bringing screens to life with what we offer



# Onscreen Behavior

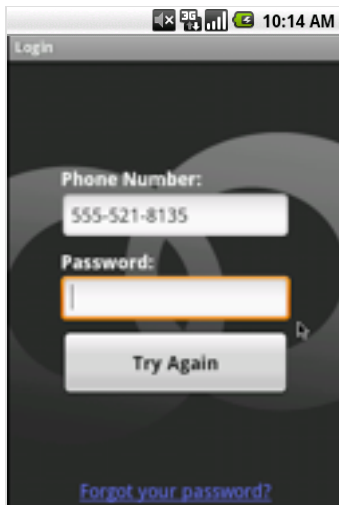
Bringing screens to life with what we offer

- A word about Toasts



# Onscreen Behavior

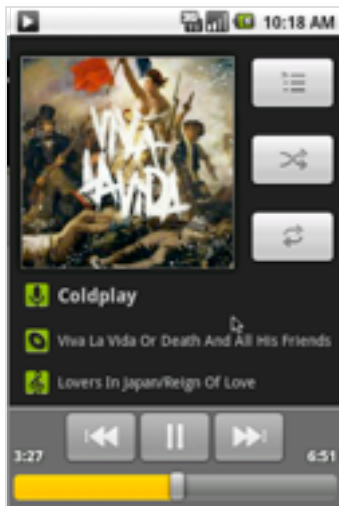
Expected interaction and unexpected delight



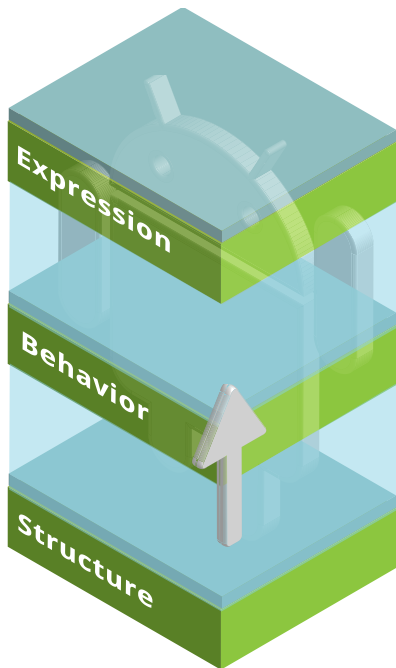
The image shows a mobile application interface for a login screen. At the top, there is a status bar with icons for signal strength, 3G connectivity, and battery level, along with the time 10:14 AM. Below the status bar is a header labeled "Login". The main content area has a dark background with a large, faint "8" watermark. It contains two input fields: "Phone Number:" with the value "555-521-8135" and "Password:" which is currently empty. Below the password field is a button labeled "Try Again". At the bottom, there is a link that says "Forgot your password?". The "Try Again" button is highlighted with a thick orange border, indicating it is the focus of the interaction.

# Onscreen Behavior

Expected interaction and unexpected delight



# Expression



# Expression

- The aesthetics of an application help call attention to the key tasks you have made the core of your application experience



# Expression

## Tips and Tricks

- API demos are a great place to start your toolkit

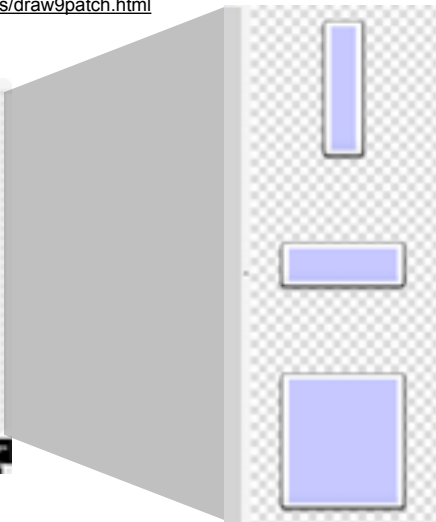
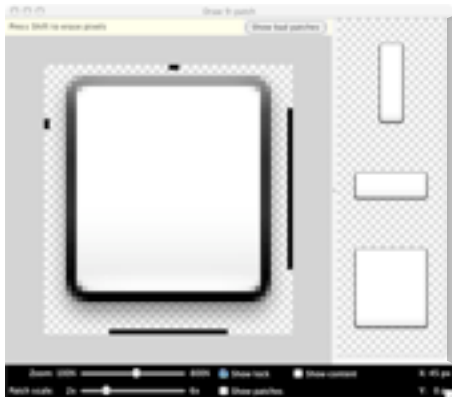


# Expression

## Tips and Tricks

- Consider using 9 patches or 9 patch techniques when you can

<http://developer.android.com/guide/developing/tools/draw9patch.html>



# Expression

## Tips and Tricks

- Want to reduce banding in gradients?

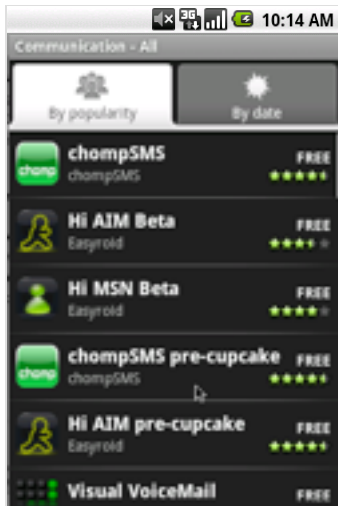


```
<nine-patch xmlns:android="http://  
schemas.android.com/apk/res/android"  
    android:src="@drawable/  
activity_title_bar"  
    android:dither="true"  
/>
```

# Expression

## Tips and Tricks

- Marquee is an option for long text strings



`<TextView`

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
    android:singleLine="true"  
    android:ellipsize="marquee"  
    android:fadingEdge="horizontal" />
```

# Expression

## Tips and Tricks

- Consider using the built-in scalable pixel units typography

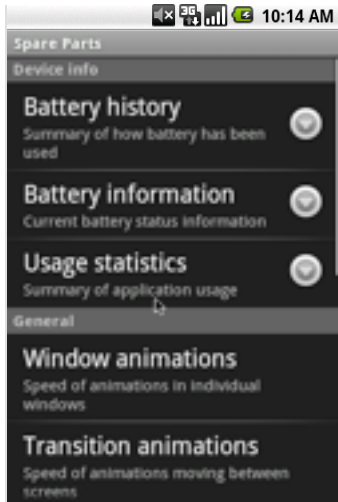
```
android:textAppearance="?android:attr/textAppearanceLarge"  
android:textAppearance="?android:attr/textAppearanceMedium"  
android:textAppearance="?android:attr/textAppearanceSmall"
```

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/  
textAppearanceMedium" />
```

# Expression

## Tips and Tricks

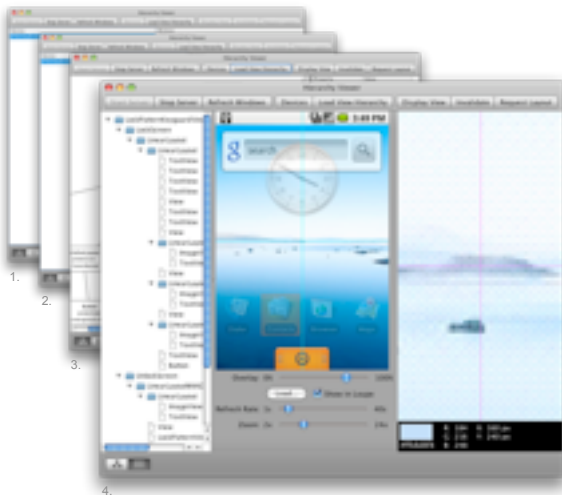
- Consider using the built-in scalable pixel units typography



# Expression

## Tips and Tricks

- Want to match a pixel perfect layout? Use Hierarchy Viewer



Picture 1. Select the device, a window (leave the selection blank to grab the focused window) and click "Load View Hierarchy"

Picture 2. In the new view, click the second button from the left, at the bottom of the window (the one with a grid) to enter the PixelPerfect tool

Picture 3. This tool shows, from left to right, the view hierarchy, a live view of the device, a live zoomed view of the device. Clicking a View object on the left shows its bounds in the live view. You can control the refresh rate and the zoom level. Click the Load button to load an overlay and choose the pixel perfect file

Picture 4. The pixel perfect shows on top of the live view, you can control the translucency of the overlay and you can click the "Show in loupe" box to display the overlay in the zoomed view on the right

# Expression

## Tips and Tricks

- Want to learn more?
  - Check out Romain Guy, Joe Onarato, and David Sparks's presentations
  - Today
    - David: Mastering the Android Media Framework, 4:15pm Moscone West Rm 7
  - Tomm.
    - Romain: Turbo-charge your UI 2:30 Moscone West Rm 7
    - Joe: Supporting Multiple Binaries 3:45pm Moscone West Rm 4

# Our plans for the future

We're continuing to innovate with each release

- UI Guidelines
  - First section of living UI Guidelines is available online today
- Make sure to update the SDK
  - All assets we use are in source tree and available for use
  - We've recently published Icon and Widget Guidelines as well that provide the links to these assets
    - go to <http://android-developers.blogspot.com/>



# Q & A

Post your [questions](https://code.google.com/events/io/questions) for this talk on Google Moderator:  
**[code.google.com/events/io/questions](https://code.google.com/events/io/questions)**

# Appendix



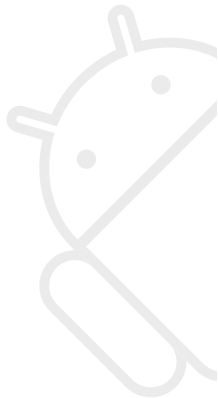
# Reading materials

- Books

- Designing Interfaces: Patterns for Effective Interaction Design Jenifer Tidwell
- Information Architecture for Designers Peter Van Dijck
- The Visual Display of Quantitative Information Edward Tufte
- The Elements of Typographic Style Robert Bringhurst
- The Elements of Color Johannes Itten
- Icon Design Steve Caplin

- Websites

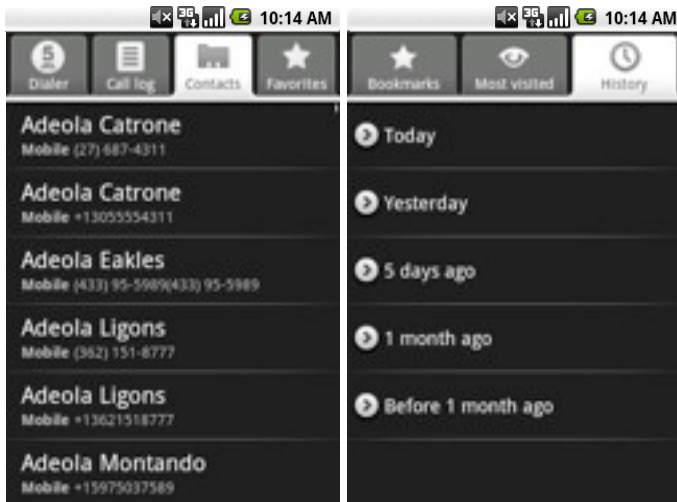
- <http://www.patternlanguage.com>
- <http://www.designinginterfaces.com>
- <http://www.37signals.com/gettingreal>
- <http://www.welie.com/>



# Toolkit starter templates

- Views

- Lists
- Expandable lists
- Tabs

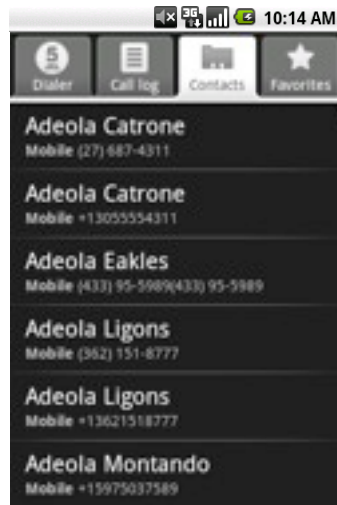


# Toolkit starter templates

- Views
  - Controls



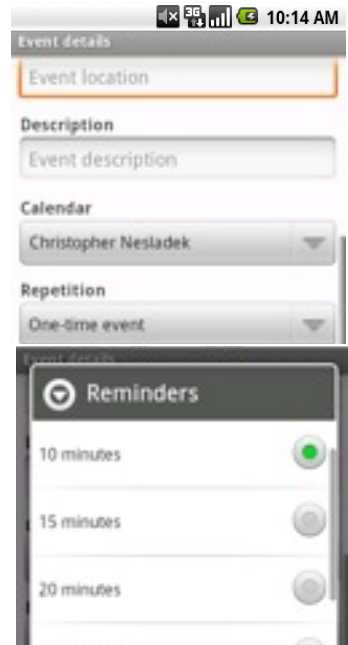
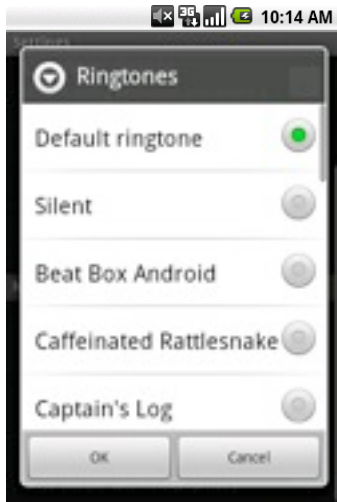
Light Theme



Dark Theme

# Toolkit starter templates

- Apps
  - Dialogs, Pulldowns



Google™

