Mobile Applications and Services
FALL 2013

- Android Design Pattern!

Navid Nikaein
Mobile Communication Department

This work is licensed under a CC attribution Share-Alike 3.0 Unported license.
ANDROID DESIGN PATTERN & PRINCIPALS

http://developer.android.com/design/index.html
3-Objectives

1. **Beauty and fast**
   - Combine beauty, simplicity and reactivity
   - Create a magical experience that is effortless and powerful

2. **Easy and smart**
   - Simplify my task and easy to understand and work regardless of age/culture
   - Transparent file management, sync, and configuration, as well as context-aware and smart (adaptive learning)

3. **Fun and personal**
   - Do and combine new things in different ways
Design principles for a better user experience

- Beauty and fast
  - Beautiful surface, carefully placed animation, or a well-timed sound effect
  - Real objects vs button and menus
  - Best default configuration and optional customization
  - Learn from people interactions and preferences over time (history), to provide fast operation (reactive)
Design principles for a better user experience

- **Easy and smart**
  - Use short phrases and simple words
  - Use pictures to explain ideas as they are more efficient than words
  - Use history to take your best guess (act rather than asking first), and let the user have a final say (allow undo/back operation)
  - Break the information into small and digestible chunks, and hide options that are not essentials (teach while using)
  - Places in your app look distinct and use transitions to show relationships among screens
  - Remember data, settings across devices (easy upgrade)
  - Interrupt when it’s critical and time-sensitive
Design principles for a better user experience

- Fun and personal
  - Leverage common visual patterns and gestures (swipe)
  - Be gentle in how you prompt users to make changes
  - Break complex tasks into smaller steps, and feedback on actions
  - Enable to combine actions to build more complex operation
  - Not all actions are equal
Common App UI

1. **Menu action bar**
   - Control center of your app

2. **View control**
   - Switch between different views

3. **Content area**

4. **Split action bar**
   - Move important actions to the button
Device and display

- **Basics**
  - Stretch and compress your layouts to accommodate various heights and widths
  - Optimize layout for each screen
  - Provide resources for different screen densities (DPI)

- **Strategy**
  - Base standard (normal size and MDPI) and scale it up or down for the other buckets
  - Start with the device with the largest screen size, and then scale down
Touch Feedback and Visual Response

- Which object was touched and that your app is "listening"
  - Use color and illumination to respond to touches
  - Reinforce the resulting behaviors of gestures
  - Indicate what actions are enabled and disabled

- Note:
  - Most of Android's UI elements have touch-feedback built in
Screen and density

- Device varies in physical size and also in screen density (Dot-Per-Inch, DPI)
- **Size bucket**
  - handset (smaller than 600dp) and tablet (larger than or equal 600dp)
- **Density bucket**
  - LDPI (120 dpi), MDPI (160 dpi), HDPI (240 dpi), and XHDPI (320 dpi)
- **Touchable UI components are generally laid out along 48dp units (physical size 9mm, recommended 7-10mm)**
  - users will be able to reliably and accurately target
- **Spacing between each UI element is 8dp.**
# Android Units

- Pixel = width * dpi
- Pixels = dps * (density / 160)

<table>
<thead>
<tr>
<th></th>
<th>Lower resolution screen</th>
<th>Higher resolution, same size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Width</td>
<td>1.5 inches</td>
<td>1.5 inches</td>
</tr>
<tr>
<td>Dots Per Inch (“dpi”)</td>
<td>160</td>
<td>240</td>
</tr>
<tr>
<td>Pixels (=width*dpi)</td>
<td>240</td>
<td>360</td>
</tr>
<tr>
<td>Density (factor of baseline 160)</td>
<td>1.0</td>
<td>1.5</td>
</tr>
<tr>
<td>Density-independent Pixels (“dip” or “dp” or “dps”)</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>Scale-independent pixels (“sip” or “sp”)</td>
<td>Depends on user font size settings</td>
<td>same</td>
</tr>
</tbody>
</table>
Screen and density: an Example

[Diagram showing various sections and bars with text and dates]

©Navid Nikaein 2013
Iconography

- Visual representation of your app on the Home or All Apps screen, should be visible on any type of BG
  - 48x48 dp on mobile device
  - 512x512 pixels on Google play

Notifications 24x24 dp

Contextual 16x16 dp

Action bar 32x32 dp
Writing style

- Keep it brief (30 character limit including spaces)

- Keep it simple. Use short words, active verbs, and common nouns.

- Be friendly. Use the second person ("you").
Writing style

- Put the most important thing first

  - Top news last
    - 77 other people +1’d this, including Larry Page.
  - Top news first
    - Larry Page and 77 others +1’d this.

- Describe only what's necessary, and no more

  - Task last
    - Touch Next to complete setup using a Wi-Fi connection.
  - Task first
    - To finish setup using Wi-Fi, touch Next.

- Avoid repetition

  - From a Setup Wizard screen
    - Signing in...
      - Your phone needs to communicate with Google servers to sign in to your account. This may take up to five minutes.
    - Signing in...
      - Your phone is contacting Google. This can take up to 5 minutes.
Gestures

- **Touch**: press, lift
- **Long press**: press, wait, lift
- **Double touch**: two quick touches

- **Swipe**: press, move, lift
- **Drag**: long press, move, lift
- **Pinch open**: 2-finger press, move outwards, lift
- **Pinch close**: 2-finger press, move inwards, lift

©Navid Nikaein 2013 - p 16
App Structure

- Single activity
  - Calculator, camera

- Multi independent activity
  - Phone

- Detail view and deep navigation
  - Gmail
Navigation

- **Up button**: navigate within an app based on the hierarchical relationships between screens

- **Back Button**: navigate in reverse chronological order, through the history of screen
  - Dismisses floating windows (dialogs, popups)
  - Dismisses contextual action bars, and removes the highlight from the selected items
  - Hides the onscreen keyboard (IME)
Navigation: examples
Action Bar

1. App icon/identity: up button
2. View control (opt): display data in different view
3. Action button: most often used actions
4. Action overflow: less often used actions

Location of action bar:
1. Main action bar
2. Top bar
3. Bottom bar

Contextual action bar (CAB)
Action Bar: Adapt to Screen Rotation
**Action Bar : View Control**

- **Tabs**
  - Fixed tabs
  - Scrollable tabs

- **Spinners**

- **Drawers**

- **Most important activities are action buttons**

- **Use the FIT scheme to prioritizing actions on the action bar**
  - Frequent, Important, Typical
Confirming & Acknowledging

- Avoid uncertainty and mistake in user action

Confirming

No Confirmation or Acknowledgment

Acknowledging
Notifications

- Keep the user informed about the events
  - Chat, Calendar

- Structure
  1. Title
  2. Large icon
  3. Text
  4. Content info
  5. Small icon
  6. Time stamp

- Expand and contract

- Actions

- Stack notification

- Prioritize notification
Settings

Decide for me but let me have the final say

- How your app should behave and look like
  - Set through user preferences

- Avoid the temptation to make everything a setting

- Initial access through the action overflow
  - Group the setting
    - Section divider
    - Separate sub screen

- Optimal default values

- Clear and concise labels

1. Is it actually a user preference?
   - Information and actions are not user preferences.
     - NO
       - Don't make it a setting.
         - If it's static information about the app (e.g. version number, terms of service), organize it within a Help screen.
         - If it's an action (e.g. refresh), find a place for it in the action bar.
     - YES

2. Would users change it a lot?
   - Frequently enough that they'd feel burdened by taking 3 steps (at least) to access it every time?
     - NO
       - Don't make it a setting.
         - Give it higher prominence on the action bar so it's more easily accessible.
     - YES

3. Is there a right answer for the majority of users?
   - Would at least 80% of users set this control to the same exact value and never change it again?
     - NO
       - Make it a setting!
     - YES or NOT SURE
   - If the right answer were implemented as fixed behavior in your app, would it cause harm to the minority who wouldn't be able to change it?
Setting Design Pattern

- Checkbox
- Multiple choice
- Slider
- Date and time
- Subscreen navigation
- List subscreen
- Master on/off switch
- Individual on/off switch
- Dependency
Help

- Do not show unsolicited help
- Offer help in the action overflow → Help is part of the UI
- Provide straight the answer
- Help menu if other information is required
  - copyright info, credits, terms of service, and privacy policy
Web, mobile, and enterprise developers building applications in the cloud with Google and open web technologies

Products and technologies to be featured at I/O include App Engine, Android, Google+, Google Chrome, HTML5, AJAX, Maps and Data APIs, Google TV, and more.

https://developers.google.com/events/io/