Designing for the mobile form factor

Kirill Grouchnikov
Android, Google Inc.
Overview

• Characteristics of a great UI
• Why mobile is different
• From design to implementation
Overview

• Characteristics of a great UI
• Why mobile is different
• From design to implementation
What is a great UI?

- User friendly
- Responsive
- Polished
Square

Seesmic

doubleTwist

TweetDeck
User friendliness

• Simple flows that assist in completing tasks
• Task-oriented instead of feature-oriented
• Guide the user instead of fight the user
• Build on user’s knowledge of the domain
Responsiveness

- Visual confirmation of a user action
- Long tasks not blocking the UI
- Progress of long running tasks
Visual polish

- Visual appeal as important as functionality
- Macro-level as well as micro-level
- Visual consistency to anchor the flows
Is mobile different?

• Similar problems
• Similar solutions
• Different form factor
Overview

• Characteristics of a great UI
• Why mobile is different
• From design to implementation
Form factor

- Smaller screens, bigger controls
- Rotation and ratio change
- User interaction
- Screen density
- Limited resources
Form factor

- Smaller screens, bigger controls
- Rotation and ratio change
- User interaction
- Screen density
- Limited resources
Screen size
Screen size

- Finger interaction
- Small controls - user frustration
- Onscreen keyboard - even less space for controls
Screen size

- Hide optional controls
- Split long forms into separate screens
- Annotate fields for better keyboard modes
Form factor

• Smaller screens, bigger controls
• Rotation and ratio change
• User interaction
• Screen density
• Limited resources
Rotation

- Smaller form factor
- No desktop “rigidity”
- Different “natural” orientation
- Varying user preference
Landscape mode
Landscape mode
Landscape mode

• Don’t blindly reuse portrait layout
• Action bar / footer - precious vertical space
• More actionable content above the fold
• Don’t leave gigantic unused “holes”
• But don’t cram too much content either
Form factor

- Smaller screens, bigger controls
- Rotation and ratio change
- User interaction
- Screen density
- Limited resources
User interaction

• No mouse. No stylus. No rollovers.
• Optional physical keyboard.
• Touch is king.
Touch

- Touch modes
- Tap / long press / move / fling
- Multi-touch
- Pinch / zoom / rotate / tilt
Touch
Form factor

• Smaller screens, bigger controls
• Rotation and ratio change
• User interaction
• Screen density
• Limited resources
Resolution / DPI

1280 pixels / 13.3 inch

800 pixels / 8.3 inch

800 pixels / 8.3 inch = 96 pixels per inch

Artwork by studiomx.eu
Resolution matters

300dpi

96dpi
Screen density - LDPI

Motorola Charm

Sony Ericsson Xperia X10 Mini

Motorola Citrus

Motorola Flipout

HTC Tattoo

HTC Wildfire
Screen density - MDPI

- Motorola Charm
- HTC Magic (myTouch 3G)
- Motorola CLIQ
- HTC Droid Eris
- Motorola Backflip
- HTC Aria
Screen density - HDPI

Motorola Droid

Sony Ericsson Xperia X10

Samsung Nexus S

HTC Nexus One

HTC Evo 4G

HTC G2
100% difference between LDPI and HDPI
Screen density

- No hard coded pixel values
- Use dips and DisplayMetrics
- No single set of images
- Bundle multiple resolutions
Form factor

- Smaller screens, bigger controls
- Rotation and ratio change
- User interaction
- Screen density
- Limited resources
Resources

- Less powerful CPUs
- GPU / hardware acceleration optional
- Much less memory
Resources

- Large bitmaps - out of memory errors
- Frequent allocation of small objects - garbage collection pauses
- Use DDMS and “Allocation tracker” tab
Mobile is different

- Smaller screens, bigger controls
- Rotation and ratio change
- User interaction
- Screen density
- Limited resources
Overview

• Characteristics of a great UI
• Why mobile is different
• From design to implementation
What is a great application?

- User friendly
- Responsive
- Polished
But much more important
A clear, simple, user-oriented goal
Design process

- Formulate and finalize the product goal
- Identify major scenarios
- Build user flows
- Pixel-perfect visual design
- Implementation
User scenarios

- IA, IaD, UxD, UX, HCI
- Define features (not the other way around)
- Sketch high-level wireframes
- Define consistent navigation model
As the implementor

- Know hard platform limitations
- Identify layouts, components, event handling
- Flesh out major building blocks
  - anchor areas, navigation controls, interaction patterns
Visual design

- Colors, textures, typefaces, layouts
- Lighting models, anti-aliasing, drop shadows, soft edges
- Translucency, overlapping, non-rectangular components
As the implementor

- Know hard platform limitations
- Weigh performance considerations
- Consider target hardware limitations (colors, memory, screen size, ...)
- Two-way communication and early validation
Pixel perfection

- Visual separation between sections
- Layout depends on the screen size
Pixel perfection

- Thumbnail drop shadow
- Text drop shadow
- Vertical alignment
- Scaling text size to fit
Working with designers

• Know the platform capabilities and limitations
• Trust that interaction and visual designers know what they’re doing
• Engage early and often
• Provide specific feedback
• You are not the user
Below the surface

- UI layer is just one piece
- Data models, persistence, wire protocols, caching, binding
- It’s easy to “cut corners” on pixel level
Remember what’s important
What is a great UI?

• User friendly
• Responsive
• Polished
Everything else can be good enough
Q&A

• @kirillcool
• pushing-pixels.org
• kirillcool@yahoo.com
• android-developers.blogspot.com