Android Experiences

Tomi Mickelsson
Senior Technology Specialist @ Idean

http://overtheair.org/
April 4, 2008
Agenda

- Android Intro
- Framework
- SDK and tools
- UI, components, layouts
- UI goodies
- Custom UI
- Sample Prototypes
  - Contacts & Splash
- Good and Bad
- Wishlist
Idean

- User-centric mobile design
  - User experience design
  - User and market research
  - Interaction design
  - Visual design
- Prototyping and development of user interfaces
  - Low-tech and high-tech
- Strong focus on mobile
- Finland, Helsinki
Why Prototype?

- Get user feedback at an early stage
- Verify that the design meets the set requirements
- Get to know technology limitations
- Evaluate technologies
- Reduce risks, costs
- Great communication medium, beats docs
Android

- “Hot blonde that everybody wants to be with"
- Buzz: Linux, Java, Google, open-source
- First phones 2H2008?
- SDK open sourced eventually
  - truly open-source: Apache-license
- Backed by OHA, Open Handset Alliance
  - mobile operators, device manufacturers, SW/HW companies
Android Just Now

• Pre-Alpha!
• Google is the architect
  – Gathers feedback, improves
  – Avoid design by committee
• Not yet open
• Play with emulator and discover...
• Big changes between SDKs
  – m3 vs m5 SDKs
• Android Developer Challenge, ADC
  – 10 M$
Android Architecture

- Linux
- Java as a native language
  - safe bet: millions of developers
  - Java but not J2ME
  - no JVM, a new Dalvik Runtime
- Currently no native access to OS
- OS is customizable
  - replace system apps, even home screen
- A whole new UI
- SQLite database
Android Java Framework

- Anatomy of an application:
  - Activity
  - Intent Receiver
  - Service
  - Content Provider

- A new, innovative way to construct apps
  - Learn it, don't fight against

- Building screens and services, not monolithic black-boxes
  - Loosely coupled, addressable parts
  - Isn't this like the web? Back-button too!
  - Rapid prototyping

- OS or an ecosystem?
SDK (m5-rc15)

- Windows, OS X, Linux!
- 124MB of HTML-docs
- 183 Java sample source files
- New UI since m3
  - from QVGA to HVGA
- Decent and simple
  - just unzip to a folder
  - Docs, samples, binaries included
SDK Tools

- **Emulator**
- **Eclipse**
  - ADT: Android Development Tools plugin
  - DDMS: Dalvik Debug Monitor Server tool
- **Command line tools**
- **No user interface editor**
  - Edit, compile, test cycle takes 10s
  - www.droiddraw.org
Technologies for prototyping on Android

- Java
  - but not MIDP, there's a new UI
- WebKit
  - standalone browser
  - embedded component
  - but no access to the system?
- OpenGL ES
- No C++
- No Flash /Lite (yet)
- No Python (yet?)
  - but Guido works at Google…
  - used in SDK tools: activityCreator.py
Application Design Philosophy

• "Successful applications will offer an outstanding end-user experience"
  - Fast
  - Responsive
  - Seamless

• The industry takes user experience seriously!
User Interface

- Rich UI model, huge number of components
- Do not code, use XML
  - Declarative programming ala web, Adobe Flex, Mozilla XUL
- New APIs, no J2ME, AWT, Swing, LCDUI
- No other UIs: a single UI a good thing
- Both keypad and touch modes
- Layouts and components
UI components, basic

- Button, ImageButton, CheckBox, Chronometer, CompoundButton, EditText, ImageView, ProgressBar, SurfaceView, TextView, ZoomSlider, ExpandableListView, ListMenuitemView, ListView, RadioButton, RadioGroup, ScrollView, Spinner, Toast
UI components, high-level

- AutoCompleteTextView, AnalogClock, DigitalClock, MapView, DatePicker, Gallery, TabWidget, TextSwitcher, Ticker, TimePicker, TwoLineListItem, VideoView, WebView
UI components, MapView

- Location-based apps will be hot
- Android makes location-stuff easy
  - Map-component built-in
- Based on Google Maps
UI components, WebView

- Essential component
  - So much is in web
  - Mashup layouts
- Industry standard WebKit
  - Safari, S60 browser, Adobe AIR, QT...
- Javascript, AJAX
- Embedded assets
  - `<img src="file:///android_asset/myimage.png">`
UI Layouts

• Good set of layouts
  – AbsoluteLayout, FrameLayout, Gallery, GridView, LinearLayout, ListView, RelativeLayout, ScrollView, Spinner, SurfaceView, TabHost, TableLayout, ViewFlipper, ViewSwitcher

• You should understand layouts well
  – "wrap_content", "fill_parent", "layout_weight"

• Use relative layouts, fits into many screens

• Margins, paddings ala CSS
Custom UI components

- First try use existing
  - Existing components already customizable
- Framework very open for extra customization
- Strategies:
  - Inherit and override a base component
  - Reuse and combine existing components
  - Fully draw yourself - `onDraw(Canvas canvas)`
- However, avoid overkill customization
  - Variance among devices!
  - Small screen vs big screen
Ui goodies

- Touch (single, not multi)
- List fading at ends
-Scrollbar on off, fancy skins
- Context menu with touch
- Scrolling
- Rounded corners
- Themes
- Focus management
- Progressbar, primary, secondary, in title too
UI goodies: Animation

- Flash-alike stuff!
- Tweened animation: controlled by XML-files
- Frame-by-frame animation: controlled by code
- Shaders, shapes, gradients, pivot, path effects
- Drawing caches
- OpenGL too
UI goodies: 9-Patch Stretchable

- For custom look'n'feel
- Innovative, no code required
- Improves designer-developer-co-operation
Sample Prototype: Contacts

- Tabs
- Custom view with relativelayout
- WebKit browser component
- Text shadow
Sample Prototype: Splash Anim

- Flash-like animation
  - Scale, rotate, alpha
- PNG-graphics with alpha
- Dynamic layout
  - View.GONE, View.VISIBLE
Prototyping on Android - Good

- Java and Eclipse
  - Familiar language, familiar tools
- Rich UI model, customization features
- Declarative programming, XMLs
- Applications expose an ecosystem of components
  - Reuse existing, mashup screens (activities)
- SDK runs on all mainstream platforms
- Active support forum
Prototyping on Android - Bad

- Just pre-alpha:
  - SDK a moving target
- Need to learn a big new UI API
- No interface editor yet
- No real phones, just an emulator
- Not yet open source
Problems/Critique:

- Bugs
  - "This is a bug that will be fixed in an future SDK."
  - "I just wasted 5 days for the challenge banging my head trying to figure out why..."
- Deprecated or not?
- Missing functionality
- Animation bugs
- Nested layouts broken
- Samples with no code
  - How to really program tabs?
Detective Work

- Playing Sherlock, digging SDK internals...
Wishlist

• Perfect the UI, make it coherent and simple
  – Finetune API, drop odd classes
  – Have a clear MVC
• UI editor
• Animation between activities
• Animation to change layouts
  – Like Adobe Flex state transitions…
• Drag'n'drop
• Apply themes dynamically
• Intuitive touch, multi-touch
• Handle layout scaling, small to big screens
• Fix bugs
The End

• Thank You!