Building Android Applications
Agenda

• Overview & Inspiration
• Application components
• Practical matters
A Powerful Trend

Why?
A Powerful Trend
A Powerful Trend

Internet and Mobile Phone Users, Worldwide

- Internet users (PC-based)
- Mobile phone users

2000 2001 2002 2003 2004 2005 2006 2007
Open Handset Alliance Members
Open Handset Alliance Members

Mobile Operators

- China Mobile
- KDDI
- DoCoMo
- Sprint
- Telecom Italia
- Telefónica
- T-Mobile
Open Handset Alliance Members

Mobile Operators

Handset Manufacturers

© Copyright 2008, Google
Open Handset Alliance Members

Handset Manufacturers

- Motorola
- HTC
- Samsung
- LG Electronics
- Ascender Corporation
- eBay
- esmertec
- Google
- livewire
- C Living Image
- NUANCE
- pv
- Skypod
- SONiVOX

Mobile Operators

- Mobilcom
- Sprint
- Telecom Italia
- Telefónica
- T-Mobile

Software
Open Handset Alliance Members

Mobile Operators
- China Mobile
- KDDI
- NTT DoCoMo
- Sprint
- Telefonica
- T-Mobile

Handset Manufacturers
- Motorola
- HTC
- Samsung
- LG Electronics

Software
- Ascender Corporation
- eBay
- esmertec
- Google
- livewire
- Living Image
- Nuance
- PV
- SkypePOD
- SONiVOX
- Texas Instruments

Semiconductor
- Audience
- Broadcom
- Intel
- Marvell
- NVIDIA
- Qualcomm
- SirF
- Synaptics
- Texas Instruments

Android
**Application Framework**

- Activity Manager
- Window Manager
- Content Providers
- View System
- Notification Manager
- Package Manager
- Telephony Manager
- Resource Manager
- Location Manager
- GTalk Service

**Libraries**

- Surface Manager
- Media Framework
- SQLite
- OpenGL | ES
- FreeType
- WebKit
- SGL
- SSL
- libc

**Android Runtime**

- Core Libraries
  - Dalvik Virtual Machine

**Linux Kernel**

- Display Driver
- Camera Driver
- Bluetooth Driver
- Flash Memory Driver
- Binder (IPC) Driver
- USB Driver
- Keypad Driver
- WiFi Driver
- Audio Drivers
- Power Management
## Application Building Blocks

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>UI component typically corresponding to one screen.</td>
</tr>
<tr>
<td>IntentReceiver</td>
<td>Set and respond to notifications or status changes. Can wake up your app.</td>
</tr>
<tr>
<td>Service</td>
<td>Faceless task that runs in the background.</td>
</tr>
<tr>
<td>ContentProvider</td>
<td>Enable applications to share data.</td>
</tr>
</tbody>
</table>
Emulator

- QEMU-based ARM emulator runs same system image as a device
- Use same toolchain to work with devices or emulator
Eclipse Plugin

Project template

New Android Project
Creates a new Android Project resource.

- Project name:
- Package Name:
- Activity Name:
- Application Name:

- Use default location

Location: /Users/mcleron/Documents/workspace-x

Finish
Debugging

- Call stack
- Examine variables
- Breakpoints, single stepping
Industry
Industry

- Software stack open-sourced under Apache 2.0 license
- Source available after first handsets ship
- Anyone will be able to build a system image
Industry
Industry

Users
• Users have control of their experience
• They control what gets installed
• They choose the defaults
Developers

Industry

Users
- Don’t need permission to ship an application
- No hidden or privileged framework APIs
- Can integrate, extend, and replace existing components
Integrate
Integrate
Integrate

Extend
Replace
Replace

- Android Factory
  +1-555-4242
- Andy Rubin
  +1-555-1234
- Dan Morrill
  +1-555-5678
- Eric Chu
  +1-555-0987
- Jason Chen
  +1-555-5555
- Steve Horowitz
  +1-555-4321
Replace

Phonebook 2.0

My Contacts (56)

- Anna
- Amy
- Arnold
- Babeth
- Brian
- Bob

Get into the groove!

Paris | 8:45 pm
34°F
Replace
Basics of User Interfaces
Creating UIs

- Two ways to create GUIs: in XML or in code
  - Declarative route via XML has advantages
- A lot of your GUI-related work will take place in:
  - `res/layout`
  - `res/values`
- `@id/name_for_component` gives you handle for referencing XML declarations in code
Views are building blocks

Examples:

- Can be as basic as: TextView, EditText, or ListView
- Fancier views: ImageView, MapView, WebView
Layouts

- Controls how Views are laid out
  - FrameLayout: each child a layer
  - LinearLayout: single row or column
  - RelativeLayout: relative to other Views
  - TableLayout: rows and columns
  - AbsoluteLayout: <x,y> coordinates
Layouts re-sized

480x320

240x320

320x240

9 - 11 PM Android Tech Talk, Tunis (B43, 2nd Floor)
Layout Parameters

• Specify many aspects of what’s being rendered

• Examples:
  • android:layout_height
  • android:layout_width

• Tip: start with documentation for a specific View or Layout and then look at what’s inherited from parent class
Demo: Hello, World!
Application Components
### Basic components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities</td>
<td>UI component typically corresponding to one screen.</td>
</tr>
<tr>
<td>IntentReceivers</td>
<td>Respond to broadcast Intents.</td>
</tr>
<tr>
<td>Services</td>
<td>Faceless tasks that run in the background.</td>
</tr>
<tr>
<td>ContentProviders</td>
<td>Enable applications to share data.</td>
</tr>
</tbody>
</table>
Activities

- Typically correspond to one screen in a UI
- But, they can:
  - be faceless
  - be in a floating window
  - return a value
Intent Receivers

- Components designed to respond to broadcast Intents
- Think of them as a way to respond to external notifications or alarms
- Applications can invent and broadcast their own Intents as well
Intents

- Think of Intents as a verb and object; a description of what you want done

- **Examples:** VIEW, CALL, PLAY, etc.

- System matches Intent with Activity that can best provide that service

- Activities and IntentReceivers describe what Intents they can service in their IntentFilters (via AndroidManifest.xml)
Intents

- Home
- Contacts
- GMail
- Chat
Intents

Client component makes a request for a specific action
Client component makes a request for a specific action

“Pick photo”
Intents

“Pick photo”

System picks best component for that action
Intents

System picks best component for that action

“Pick photo”
Components can be replaced any time
Intents

Components can be replaced any time
Intents

New components can use existing functionality

“Pick photo”
Services

- Faceless components that run in the background
- Example: music player, network download, etc.
- Bind your code to functionality provided by another APK
  - Interfaces defined by a remote-able IDL
ContentProviders

- Enables sharing of data across applications
- Examples: address book, photo gallery, etc.
- Provides uniform APIs for:
  - querying (returns a Cursor)
  - delete, update, and insert rows
- Content is represented by URI and MIME type
Practical matters
Storage and Persistence

• Multiple options:
  • Preferences
  • Flat file
  • SQLite
  • ContentProvider
Application Packaging

• Think of .apk files as Android packages
• APKs contain everything the application needs
• Basically a glorified ZIP file
Resources

- `res/layout`: declarative layout files
- `res/drawable`: intended for drawing
- `res/anim`: bitmaps, animations for transitions
- `res/values`: externalized values for things like strings, colors, styles, etc.
- `res/xml`: general XML files used at runtime
- `res/raw`: binary files (e.g. sound)
Assets

• Similar to Resources

• Differences:
  • Read-only
  • InputStream access to assets

• Any kind of file
  • Be mindful of file sizes
Application Lifecycle

- Application lifecycle is managed by the system
- Application start/stop is transparent to the user
- End-user only sees that they are moving between screens
- Read documentation for `android.app.Activity`
- More details in Part 2 later today
Code Tour: WikiNotes
Q & A

- [http://code.google.com/android](http://code.google.com/android)