CS378 - Mobile Computing

Android Overview and Android Development Environment
What is Android?

• A software stack for mobile devices that includes
  – An operating system
  – Middleware
  – Key Applications

• Uses Linux to provide core system services
  – Security
  – Memory management
  – Process management
  – Power management
  – Hardware drivers
Android Features

- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source WebKit engine
- Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- SQLite for structured data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

A Short History Of Android

• 2001 Palm Kyocera 6035, combing PDA and phone
• 2003 - Blackberry smartphone released
• 2005
  – Google acquires startup Android Inc. to start Android platform.
  – Work on Dalvik VM begins
• 2007
  – Open Handset Alliance announced
  – Early look at SDK
  – June, iPhone released
• 2008
  – Google sponsors 1st Android Developer Challenge
  – T-Mobile G1 announced, released fall
  – SDK 1.0 released
  – Android released open source (Apache License)
  – Android Dev Phone 1 released
Short History cont.

• 2009
  – SDK 1.5 (Cupcake)
    • New soft keyboard with “autocomplete” feature
  – SDK 1.6 (Donut)
    • Support Wide VGA
  – SDK 2.0/2.0.1/2.1 (Eclair)
    • Revamped UI, browser

• 2010
  – Nexus One released to the public
  – SDK 2.2 (Froyo)
    • Flash support, tethering
  – SDK 2.3 (Gingerbread)
    • UI update, system-wide copy-paste
Short History cont.

• 2011
  – SDK 3.0 (Honeycomb) for tablets only
    • New UI for tablets, support multi-core processors, fragments
  – SDK 3.1 and 3.2
    • Hardware support and UI improvements
  – SDK 4.0 (Ice Cream Sandwich)
    • For Q4, combination of Gingerbread Honeycomb
Short History cont.

• 2012
  – Android 4.1, "Jelly Bean" released in July

• 2013
  – Most recent version 4.3 API level 18
Device Distribution Jan 2012

- Based on active devices
- Forward compatible
- Not necessarily backward compatible

1.5 Cupcake: 0.6%
1.6 Donut: 1.1%
2.1 Ecliar: 8.5%
2.2 Froyo: 30.4%
2.3 Gingerbread: 56%
3.X Honeycomb: 3.3%
4.x Ice Cream Sand: 0.6%

http://developer.android.com/resources/dashboard/platform-versions.html
Device Distribution July 2012

Current Distribution

The following pie chart and table is based on the number of Android devices that have accessed Google Play within a 14-day period ending on the data collection date noted below.

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API Level</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>Cupcake</td>
<td>3</td>
<td>0.2%</td>
</tr>
<tr>
<td>1.6</td>
<td>Donut</td>
<td>4</td>
<td>0.5%</td>
</tr>
<tr>
<td>2.1</td>
<td>Eclair</td>
<td>7</td>
<td>4.7%</td>
</tr>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>17.3%</td>
</tr>
<tr>
<td>2.3 - 2.3.2</td>
<td>Gingerbread</td>
<td>9</td>
<td>0.4%</td>
</tr>
<tr>
<td>2.3.3 - 2.3.7</td>
<td></td>
<td>10</td>
<td>63.6%</td>
</tr>
<tr>
<td>3.1</td>
<td>Honeycomb</td>
<td>12</td>
<td>0.5%</td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td>13</td>
<td>1.9%</td>
</tr>
<tr>
<td>4.0 - 4.0.2</td>
<td>Ice Cream Sandwich</td>
<td>14</td>
<td>0.2%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td></td>
<td>15</td>
<td>10.7%</td>
</tr>
</tbody>
</table>
### Version, Codename, API Level, Distribution

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API Level</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>Cupcake</td>
<td>3</td>
<td>0.2%</td>
</tr>
<tr>
<td>1.6</td>
<td>Donut</td>
<td>4</td>
<td>0.5%</td>
</tr>
<tr>
<td>2.1</td>
<td>Eclair</td>
<td>7</td>
<td>4.2%</td>
</tr>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>15.5%</td>
</tr>
<tr>
<td>2.3 - 2.3.2</td>
<td>Gingerbread</td>
<td>9</td>
<td>0.3%</td>
</tr>
<tr>
<td>2.3.3 - 2.3.7</td>
<td></td>
<td>10</td>
<td>60.3%</td>
</tr>
<tr>
<td>3.1</td>
<td>Honeycomb</td>
<td>12</td>
<td>0.5%</td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td>13</td>
<td>1.8%</td>
</tr>
<tr>
<td>4.0 - 4.0.2</td>
<td>Ice Cream Sandwich</td>
<td>14</td>
<td>0.1%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td></td>
<td>15</td>
<td>15.8%</td>
</tr>
<tr>
<td>4.1</td>
<td>Jelly Bean</td>
<td>16</td>
<td>0.8%</td>
</tr>
</tbody>
</table>

*Data collected during a 14-day period ending on August 1, 2012*
August 1, 2013

- Based on device visits to Google Play

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6</td>
<td>Donut</td>
<td>4</td>
<td>0.1%</td>
</tr>
<tr>
<td>2.1</td>
<td>Eclair</td>
<td>7</td>
<td>1.2%</td>
</tr>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>2.5%</td>
</tr>
<tr>
<td>2.3 - 2.3.2</td>
<td>Gingerbread</td>
<td>9</td>
<td>0.1%</td>
</tr>
<tr>
<td>2.3.3 - 2.3.7</td>
<td></td>
<td>10</td>
<td>33.0%</td>
</tr>
<tr>
<td>3.2</td>
<td>Honeycomb</td>
<td>13</td>
<td>0.1%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>22.5%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>34.0%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>6.5%</td>
</tr>
</tbody>
</table>

Data collected during a 14-day period ending on August 1, 2013. Any versions with less than 0.1% distribution are not shown.
Devices and Apps

• Estimated 900M activated devices
  – 2012 - 400M, 2011 - 100M

• Estimated 1.5M new activations per day
  – 2012 - 1M

• Google Play (formerly Android Market)
  – > 1,000,000 apps
  – 600,000 apps, June 2012
  – 2/3 free, 1/3 paid
  – Apple App Store, >825,000 apps April 2013
  – Apple and Google each claim 50,000,000,000 downloads

• What's old is new - Mac vs. PC
  iPhone vs. Android???
Developer Revenues

• Business Strategy: attract developers with comparison of revenue generated by applications, average revenue per user, etc.
Apple Still Dominating Revenue

Google Play downloads 10% higher than iOS App Store downloads, while iOS App Store generated 2.3x the app revenue of Google Play in Q2 2013.

Search Trends fall 2013

World wide
Search Trends fall 2013

US Only
Setup Development Environment

• Install JDK 7
• Install Eclipse IDE (version 4.3 - Kepler)
  – recommended "Eclipse Standard"
• Download and unpack the Android SDK
• Install Android Development Tools (ADT) plugin for Eclipse
• Detailed install instructions available on Android site
Android Emulator or AVD

• Emulator is essential to testing app but is not a substitute for a real device
• Emulators are called **Android Virtual Devices** (AVDs)
• Android SDK and AVD Manager allows you to create AVDs that target any Android API level
• AVD have configurable resolutions, RAM, SD cards, skins, and other hardware
Android Emulator: 1.6
Android Emulator: 2.2
Android Emulator: 3.0
Android Emulator: 4.0
Emulator Basics

• Host computer’s keyboard works
• Host’s mouse acts as finger
• Uses host’s Internet connection
• Other buttons work: Home, Menu, Back, Search, volume up and down, etc.
• Ctrl-F11 toggle landscape → portrait
• Alt-Enter toggle full-screen mode
Emulator Limitations

• No support for placing or receiving actual phone calls
  – Simulate phone calls (placed and received) through the emulator console
• No support for USB connections
• No support for camera/video capture (input)
• No support for device-attached headphones
• No support for determining connected state
• No support for determining battery charge level and AC charging state
• No support for determining SD card insert/eject
• No support for Bluetooth
• No support for simulating the accelerometer
  – Use OpenIntents’s Sensor Simulator

That's why we need the dev phones and tablets!
Create an AVD using AVD Manager

or use the command line

Android Runtime: Dalvik VM

• Subset of Java developed by Google
• Optimized for mobile devices (better memory management, battery utilization, etc.)
• Dalvik runs .dex files that are compiled from .class files
• Introduces new libraries
• Does not support some Java libraries like AWT, Swing
Or From the Command Line

C:\android-sdk-windows\tools>**android create avd** -n MyDevice -t android-8
Android 2.2 is a basic Android platform.
Do you wish to create a custom hardware profile [no]
Created AVD 'MyDevice2' based on Android 2.2, with the following hardware config:
hw.lcd.density=240
vm.heapSize=24

C:\android-sdk-windows\tools>**emulator** -avd MyDevice

More info:
Applications Are Boxed

• By default, each app is run in its own Linux process
  – Process started when app’s code needs to be executed
  – Threads can be started to handle time-consuming operations
• Each process has its own Dalvik VM
• By default, each app is assigned unique Linux ID
  – Permissions are set so app’s files are only visible to that app
Producing an Android App

Java code → Byte code

.java → .class

Other .class files

Byte code → DALvik exe

.classes.dex

<xml>

AndroidManifest.xml

Resources

<str>

aapt

.apk
Other Dev Tools

• Android Debug Bridge
• Part of SDK
• command line tool to communicate with an emulator or connected Android device
  – check devices attached / running
  – install apk's, Android PacKage files, "executables", can find samples on places besides Android Market (security?)
  – and more!

Dalvik Debug Monitor Server

- DDMS
- debugging tool
- "provides, screen capture on the device, thread and heap information on the device, logcat, process, and radio state information, incoming call and SMS spoofing, location data spoofing, and more."
- can interact with DDMS via Eclipse plugin, another view in Eclipse