CS371m - 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
• **Integrated browser** based on the open source [WebKit](http://developer.android.com/guide/basics/what-is-android.html) 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, combining PDA and phone
  – PDA = personal data assistant, PalmPilot
• 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) after Alpha and Beta
    • 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

https://en.wikipedia.org/wiki/Android_version_history
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
  - Android 4.4, KitKat released October 31, 2013

### Top Smartphone Platforms

<table>
<thead>
<tr>
<th></th>
<th>Share (%) of Smartphone Subscribers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Feb-12</td>
</tr>
<tr>
<td><strong>Total Smartphone Subscribers</strong></td>
<td>100.0%</td>
</tr>
<tr>
<td>Google</td>
<td>50.1%</td>
</tr>
<tr>
<td>Apple</td>
<td>30.2%</td>
</tr>
<tr>
<td>RIM</td>
<td>13.4%</td>
</tr>
<tr>
<td>Microsoft</td>
<td>3.9%</td>
</tr>
<tr>
<td>Symbian</td>
<td>1.5%</td>
</tr>
</tbody>
</table>

Source: comScore MobiLens
Short History (Getting Longer)

• November, 2014
  Android 5.0 Lollipop released.
  API level 21
  "Material Design"

• October, 2015
  Android 6.0 Marshmallow
  API level 23
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 Eclair 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
### August 1, 2012

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>4.2%</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</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

**OS Version, API Level, Nickname**

4.1, API Level 16, Jelly Bean
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.
January 8, 2014

- 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>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>1.3%</td>
</tr>
<tr>
<td>2.3.3-2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>21.2%</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>16.9%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>35.9%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>15.4%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>7.8%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>1.4%</td>
</tr>
</tbody>
</table>
Based on unique devices that visit the Google Play Store.
January 2015

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>0.4%</td>
</tr>
<tr>
<td>2.3.3-2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>7.8%</td>
</tr>
<tr>
<td>4.0.3-4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>6.7%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>19.2%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>20.3%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>6.5%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>39.1%</td>
</tr>
</tbody>
</table>

Where is Lollipop?
January 2016

Took a year for Lollipop to gain roughly a third of device share.
Android Fragmentation

- August 2014
- open signal app for Android and iOS
- 5m - 10m Android downloads

<table>
<thead>
<tr>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18,796</td>
<td>Distinct Android devices seen this year</td>
</tr>
<tr>
<td>11,868</td>
<td>Distinct Android devices seen last year</td>
</tr>
<tr>
<td>682,000</td>
<td>Devices surveyed for this report</td>
</tr>
<tr>
<td>43%</td>
<td>Samsung's share of those devices</td>
</tr>
<tr>
<td>20.9%</td>
<td>Android users on KitKat</td>
</tr>
</tbody>
</table>
The white line shows the market share of the leading API level at any time.
## Android Fragmentation

- **August 2015**
- **10m - 50m Android downloads**

### Distinct Android Devices
- **This Year**: 24,093
- **Last Year**: 18,796
- **Total Devices Surveyed**: 682,000
- **Samsung's Share**: 37.8% of those devices
- **Device Brands Seen This Year**: 1,294
Brand Fragmentation

http://opensignal.com/reports/2015/08/android-fragmentation/
The white line shows the market share of the leading API level at any time.
Android - iOS comparison

COMPARISON WITH IOS

- iOS 8 (85%)
- iOS 7 (13%)
- Earlier Version (2%)

- 5.1 (Lollipop) (2.6%)
- 5.0 (Lollipop) (15.5%)
- 4.4 (Kit Kat) (39.3%)
- 4.3 (Jelly Bean) (4.7%)
- 4.2 (Jelly Bean) (15.9%)
- 4.1 (Jelly Bean) (13%)
- 4.0.3 - 4.0.4 (ICS) (4.1%)
- 2.3.3-2.3.7 (Gingerbread) (4.6%)
- 2.2 (Froyo) (0.3%)
Developer Revenues

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

• http://blog.appannie.com/app-annie-index-market-q2-2013/
• In-app purchases per install higher in iOS as well.
2015 App Revenue

Annual Worldwide App Revenue

Indexed Revenue

2013 2014 2015

iOS App Store  Google Play

App Annie
Search Trends
January 2015

Interest over time

World wide
Search Trends January 2015

US only
Search Trends January 2016

World wide
Search Trends January 2016

US only
Mobile Development

• Mobile "shops" (contract to develop mobile apps)
  – Mutual Mobile, Chaotic Moon, Nerd Ranch
• Companies tied to mobile
  – Bee Cave Games, Waze, Snapchat, Instagram
• Companies with major mobile apps
  – Facebook, Ebay
• Companies that want mobile apps for customers
  – banks, everybody??
• Companies that want mobile apps for internal use
  – everybody?
ANDROID DEVELOPMENT TOOLS
Setup Development Environment

• Install JDK 8
• Install Android Studio
  – includes API level 23
• Use SDK manager to download lower API levels
  – I suggest down to 15
• Detailed install instructions available on Android site
  http://developer.android.com/sdk/installing.html
Elements of Android Projects

• **Application Name**
  – seen by users on app chooser, app list, store

• **Project Name**
  – in IDE, can be different, often directory

• **Package Name**
  – Java package name, not using default package

• **Minimum SDK Level**
  – how far back do you support, ~15 Jan 2016

• **Target SDK Level**
  – device / api you had in mind for app, most recent?

• **Theme**
  – look and feel of app, color scheme, various built in themes such as Theme, Holo, Material (Design)
Android Projects

- Creating a project results in multiple files and resources being created
ANDROID PROJECT COMPONENTS
Android Projects - Components

Manifest

- AndroidManifest.xml
- Like a table of contents for your app
- Main activity
- Target and min SDK
- Declare all the parts of your apps:
  - activities, services
- Request permissions
  - network, location, ...
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.utexas.scottm.bplteams" >
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="BPL Teams"
        android:theme="@style/AppTheme">
        <!-- Your application content here -->
    </application>
</manifest>
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="BPL Teams"
    android:theme="@style/AppTheme">
    <activity
        android:name=".BPL_Activity"
        android:label="BPL Teams">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>
</manifest>
Android Projects - Components
Java Source Code

- Source Code:
- In java directory in Android Project View
- Actually in src directory on system
Android Projects - Components

Resources

- Resources or the res directory
- non source code resources for the app
- packaged up with app
- large role and use in development of app
Resource Directories

- res/drawable for graphic images such as png, jpeg
- res/layout for xml files that define the layout of user interfaces inside the app
- res/menu for xml based menu specifications
- res/values for lists of strings, dimensions, colors, lists of data
- res/raw for other kinds of files such as audio clips, video clips, csv files, raw text
- res/xml for other general purpose xml files
Gradle

• .apk files, Android Package Kit
  – Android executables
• Development environment takes, source code, manifest, libraries, resources, etc and packages them together in an APK
• some things known and set
• some things variable and configurable
• Gradle
Gradle

- Gradle is the build engine that Android Studio uses to convert your project into an APK
- What needs to be created and how to do it
- Like
  - make for C/C++
  - Ant/Maven for Java
- build.gradle file
sample build.gradle file - PROJECT

```groovy
// Top-level build file where you can add
// configuration options common to all sub-projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.0.0'
        // NOTE: Do not place your application dependencies here,
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "19.1.0"

    defaultConfig {
        applicationId "edu.utexas.scottm.bplteams"
        minSdkVersion 15
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.
        }
    }
}
EMULATORS
Android Emulator or AVD

• Emulator is useful for testing apps 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
• More info at
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!
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
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
- Byte code
- Other .class files
- AndroidManifest.xml
- Class files
- Resources
- .apk

Processes:

- javac
- dx
- aapt
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 Google Play (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 Android Studio