Aalto University
School of Science

Android development

T-110.5130 Mobile Systems programming

Contents

1. Android background
2. Design
3. Developing Android Applications
4. Android API
5. Other useful resources
6. Interactive demo

What is Android?

- **Android is the most popular mobile operating system in the world**
  - 84% of smartphones sold in 2014 run on Android
- **First Andoird phone sold in 2008**
- **Linux kernel**
- **Android RunTime virtual machine**
  - Old versions have Dalvik VM

Android

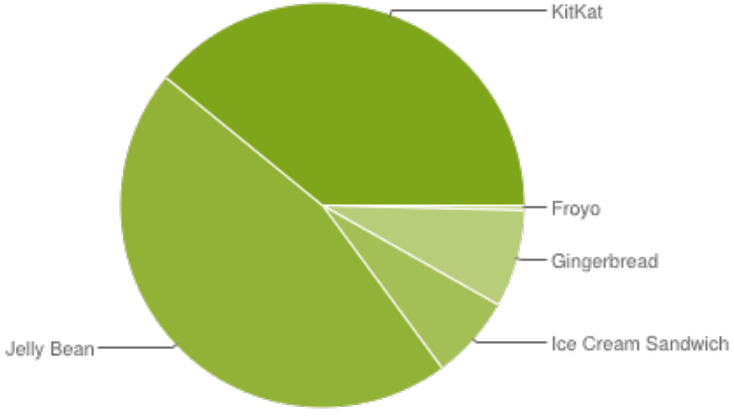
- Developed by Google
- Open source
- Free to use
- Free SDK
- Newest version Android 5.0 Lollipop, unveiled 25.6.2014

Google Play Store

- **Easiest way to distribute Android Apps**
  - But not the only one
- **25$ Starting fee for developers**
- **Income shared 70 / 30 between developer and Google**
- **No approval process, apps published in hours**
  - But can be removed without any warnings beforehand
- **Apps can be free, paid and / or include in-app purchases**
- **Over 50 billion total app downloads so far**
- **1.4 billion apps available**

Android versions

Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3-2.3.7	Gingerbread	10	7.8%
4.0.3-4.0.4	Ice Cream Sandwich	15	6.7%
4.1.x	Jelly bean	16	19.2%
4.2.x		17	20.3%
4.3		18	6.5%
4.4	KitKat	19	39.1%
5.0	Lollipop	21	<0.1%



Design

Material Design

- **New design language for all Google products across all platforms**
- **Introduced in Android 5.0 Lollipop**
  - Can be used with API level 7 support library (Android 2.2)
- **Way to build beautiful and intuitive user interfaces**
- **Everything you see in user interface should behave like a sheet of paper**
- **Video time! http://youtu.be/p4gmvHyuZzw**
- **More info: http://www.google.com/design/**

Things material design offers for you

- **Feedback animations**
- **Ready-made styles and guidelines for creating your visuals**
- **Layouts**
- **User interface components**
  - Cards, buttons, dialogs, list, menus, sliders, tabs…
- **Patterns**
- **Resources**
  - Color palettes, layout templates, fonts, general icons

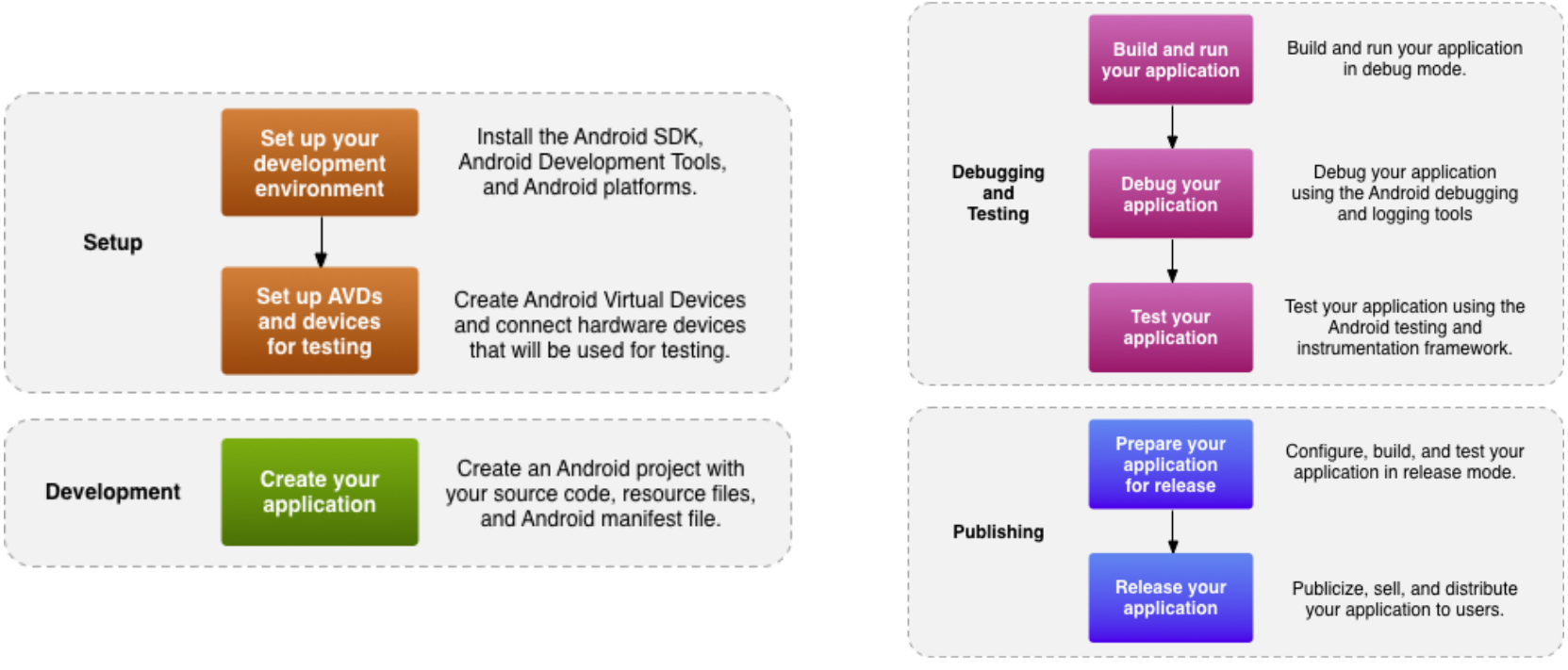
Designing you app

- **Follow Google guidelines**
  - Your app will feel instantly familiar to Android users
- **Do NOT try to mimic iOS / Windows Phone UI**
- **Use the buttons Android has (physical back button!)**
- **The most often used functions should require minimal amount of actions**

Developing Android apps

Aalto University
School of Science

Development workflow



Android Studio

- Code editor
- Layout editor and preview
- Virtual devices for testing apps
- Debugger
- Based on IntelliJ Idea
- Free to use
- Windows, Linux, OS X
- Get it: http://developer.android.com/sdk/index.html

Android emulator

- Run / debug apps without having real Android device
- Multiple Android versions
- Multiple screen sizes
- Limit network speed
- Emulate phone calls, sms messages, gps location
- Screenshots
- And lot more

Devices

- **Run / debug apps using real device**
- **You need to enable developer setting and USB debugging on your device**
  - http://developer.android.com/tools/help/adb.html#Enabling
- **Drivers required**
  - On OS X it works out of the box
  - On Windows and Linux see http://developer.android.com/tools/device.html

Testing / Debugging

- **Unit test support built in**
- **UI testing**
- **Graphical debugger**
  - Stop and examine application state
- **Tests run on emulator or device**
- **Logs can be helpful**
- **Test your app with the newest Android release and the oldest release your app supports!**

Publishing Apps

- **Build release version of your app and sign it**
- **Test the release version on real devices**
- **Select release method**
  - App market (Google Play, Amazon app store…)
  - Own website
  - Email
- **Set up release method**
  - Create account on release platform etc.
- **Distribute the .apk using selected release method(s)**

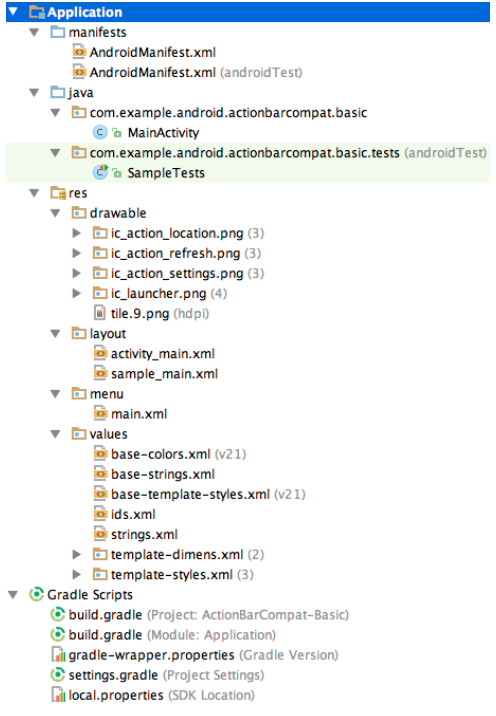
Other possible tools

- **Old Eclipse-based Android SDK**
  - ADT plugin
  - Deprecated
- **IntelliJ IDEA**
- **Vim, emacs and other text editors**
  - Build using ant or Gradle, other command line tools for managing (virtual) devices etc.
- **If you want to use these tools, course staff will not help you!**

Android API

Application Structure

- **Manifest**
- **Java code**
- **Resources**
  - Images
  - Layouts
  - Translations
- **Build scripts**



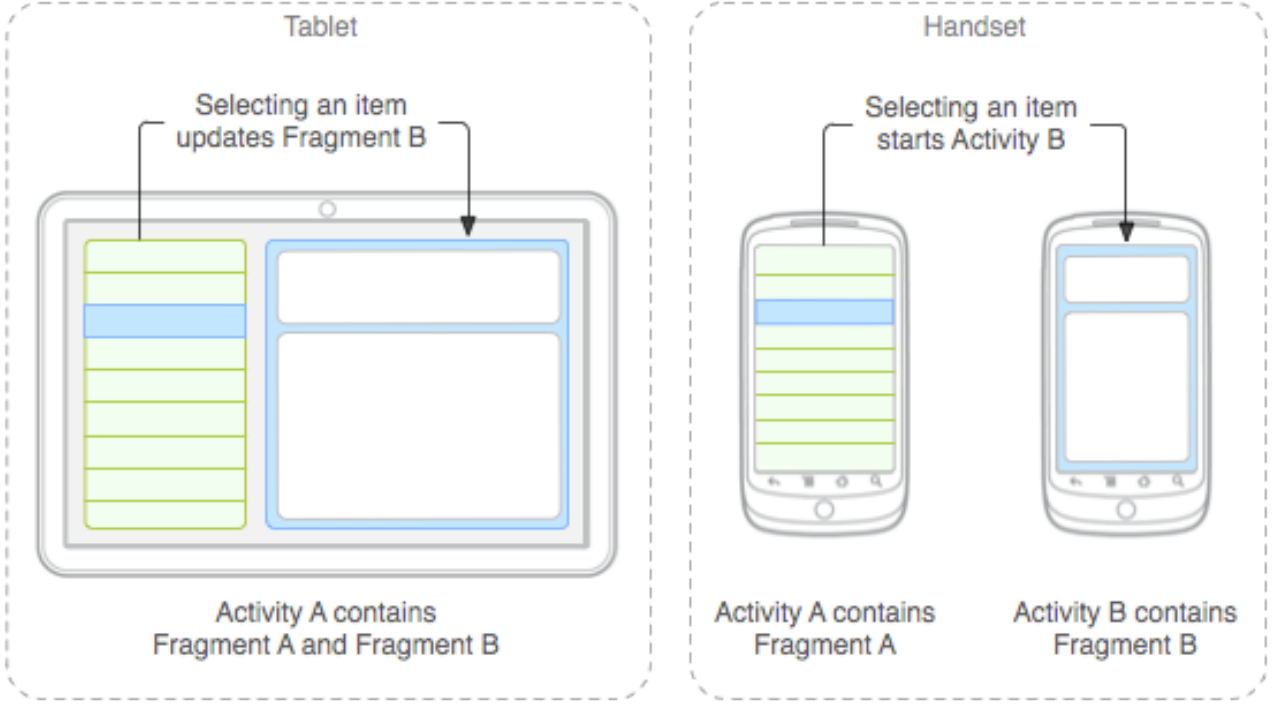
Manifest

- **Declare properties of your app**
  - Minimum Android version
  - Required permissions
  - External libraries
  - Activities
  - Services
  - Providers
  - Intents

Activity and Fragment

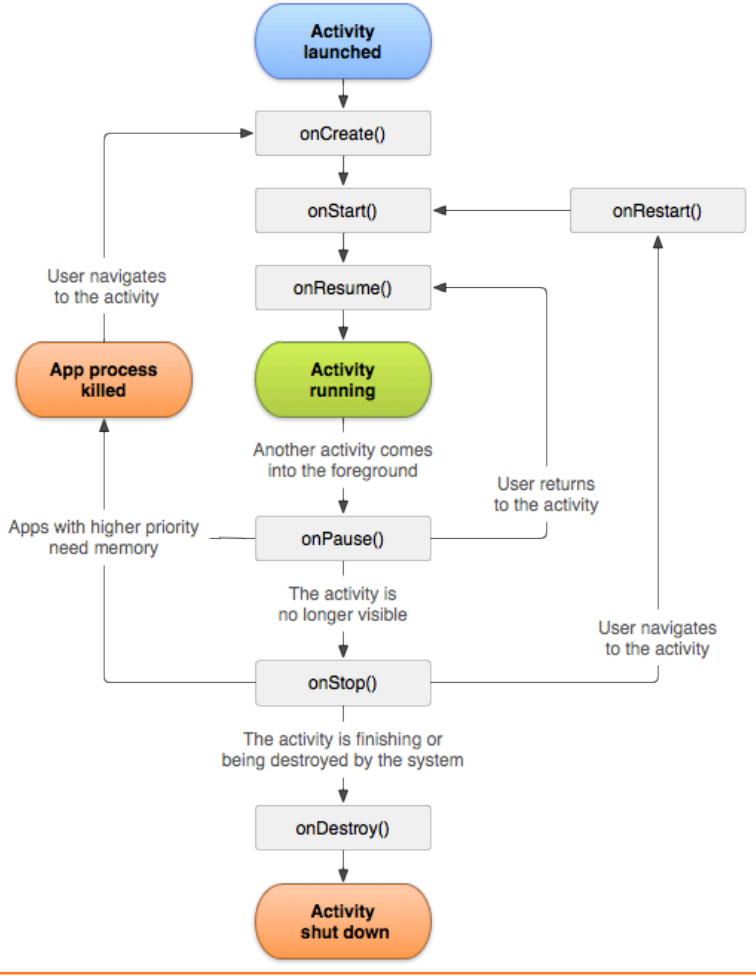
- Basic building blocks of Apps
- Activity is basically a single screen in your app
- Fragment is portion of user interface inside Activity
- Apps consists of multiple activities
- Activities consist of multiple fragments
- Both activities and fragments have layout, which indicates position for other components

Activity and Fragment



Activity Lifecycle

- **Activity can have multiple states**
- **You need to implement callback functions**
  - onCreate()
    - *When Activity is started*
    - *Set layout, initialize other components*
  - onPause()
    - *Save state*
- **Fragment has similar lifecycle**



Intent

- **Start other Activities**
- **Navigation inside app**
- **Start other apps**
- **Get some resources from (other) app**
  - Image from camera
- **Share content using other app**
  - Upload image to Dropbox from gallery

Layout

- **What user interface component should be where**
- **XML file, graphical editor**
- **Different layouts for different screen sizes**
- **Layout inside layout**
  - Reuse parts of layouts easily

ActionBar

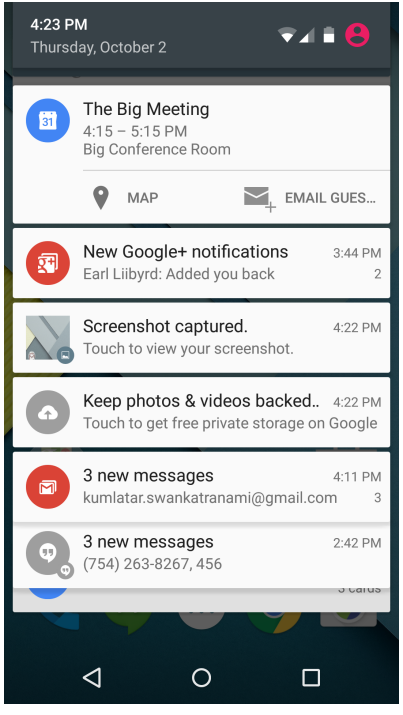
- **Show location in app**
- **Navigation inside app**
  - Tabs, dropdown
  - Up-button
    - *Different from the hardware back button!*
- **Place for important actions**

UI elements

- **Input controls**
  - Buttons
  - Text fields
  - Pickers
  - Etc.
- **Views**
  - Lists
  - Cards
- **Menus**
- **Dialogs**
- **Settings view**
- **Toasts**

Notifications

- **Display something to user outside of app UI**
  - Notify about incoming email
- **Simple actions for common tasks**
  - Delete email
- **Sound, vibration, light**



Network

- **Get content from network**
- **Hard to get it right**
  - App should not freeze
  - Connection should not stop when rotating device
- **Network connections can not happen in main thread**
- **Multiple helper classes**
  - HTTP Clients
  - Asynchronous task runners
  - Even background services

Other APIs

- **API for all the functions in the device**
- **Media and Camera**
- **Sensors**
  - Location (GPS), motion (gyroscope)…
- **Bluetooth**
- **USB**

Google Services

- Google Maps
- Google+ (sing in)
- Google Mobile Ads
- Google Cloud Platform
- Google Play In-App Billing
- Google Drive
- And a lot more…

Support Library

- **Use new features with older Android versions**
- **Multiple versions**
  - Version name tells what is the minimum required Android version
- **V4: Fragments, Rich Notifications, UI elements**
  - Supported by roughly 100% of active Android devices
- **V7: ActionBar and Material Design**
  - Supported by ~99.9% of active Android devices
- **V13: FragmentCompat**
  - Supported by ~92% of active Android devices

Native Development Kit (NDK)

- High-performance C-code for Android
- 3D-games, physics, other computational operations
- Must build separately for different CPU architectures
- Increases complexity
- Use only if you absolutely must!

Resources

Documentation

- **Official Android developer site: http://developer.android.com/index.html**
  - API documentation
  - Basic guides
  - Code examples
- **http://www.androiduipatterns.com/**
  - UI design hints

Useful libraries

- **Google GSON**
  - Convert JSON to Java objects
  - https://code.google.com/p/google-gson/
- **Google Volley**
  - Easier network data handling
  - http://developer.android.com/training/volley/index.html

Help for Android project

- **IRC: KimiA@IRCnet (probably fastest way to get help)**
  - If I don't answer immediately, do not repeat your question multiple times. I'll answer as soon as I have time for you
  - Do not ask if I can help, ask the question directly!
- **Email: kimmo.ahokas@aalto.fi**
  - Include "T-110.5130" in the subject line
- **Include all the possible information about your problem**
  - Error log, description what you were doing, even link to source code
  - Screen capture with text "Application stopped unexpectedly" is not enough!