CMPS 121

Lecture 1

Luca de Alfaro
Basics

• TTh, 12-1:45, E2 Rm. 192
• TA: Huascar Sanchez (hsanchez@soe.ucsc.edu)
• Reader: Jon Gill (jonagill@gmail.com)

Office hours:
• Luca: Thursday 2-3:30
• Huascar: Thursdays
• Jon: TBA

Class discussion group (please JOIN IT)
Ecommons for assignments
Class material

- **Book:** Professional Android 2 Application Development, by Reto Meier

- **Laptop or PC:** You need some place where to install the development tools. We strongly advise laptops, so you can bring them to class / office hours.

- **Device:** An Android device (tablets ok) where you can install applications.
Homeworks

- **Homeworks** are assigned weekly on Tuesdays. Typical homework: implement something (smallish) related to what has been covered in class in the past week.
  - Due on Tuesdays end of day.
  - Individual.
  - We will disregard lowest homework grade (including 0).
  - If you do all the homeworks well, and only that, you can get up to B.
Project

• You can do a project (this way you can get up to an A grade).
• Projects can be done in groups.
• Choose projects that have some *simple* deliverable, and can grow. Don't get stuck into something complex!
• We will ask that you write a project proposal for us to approve. We will approve most proposals, and we may give advice on the scope.
• Project presentations will be done in the last day of class / finals.
• No final exam.
Why mobile?

- The main way in which people interact with information (and with “computers”).
- The most common type of computer owned.
- The most universal communication means.
- (I bet) the way future device interfaces will be.

I see only two types of development: web and mobile.
A disclaimer

• First time this class is given.
• The instructor is still learning Android!
• So is the TA; the most expert is Jon.

Upshot:
• If you take this class because you are interested in mobile, and are prepared to learn a bit by yourself, the class provides: structure, other students around you working on similar problems, as much advice as we can.
• If you take this class to fulfill a requirement, or because you want an easy and guided experience, I would advise you to take the class next year.

This year it's for the adventurous!
Homework 1: Due Tuesday April 10

• Write a “hello world” program for Android, which displays the message:
  \[ \text{Hello everybody! This is } \text{<your name>} \]
• Turn in the .zip and .apk, and install it on your device.
• (detailed instructions on class web site).
• Sounds simple? It is...
• ... but you will need to install all the Android development environments, and it will take you a few hours.
What is Android?

- A free, open-source OS.
- An open-source development environment.
- Devices that run Android.
What is Android? (More detail)

- A hardware reference design.
- A Linux-based OS, with several optimizations for mobile.
- Open-source libraries (webkit, SQLite, OpenGL, ...)
- A run-time, including the Dalvik virtual machine for Java bytecode.
- Libraries for core functionality.
- An application framework, where you can re-use application components (how to take a photo, send an email, place a call, ...)
- A UI framework
- Pre-installed applications and components.
What comes with Android?

- Touchscreen interface!! Camera!! Wifi!! All programmable!! (future of device interfaces??)
- Applications!!
  - Email, SMS, Contacts, Browser, Maps, ...
- A way to download more applications! (Market)
- Media displayers and players.
- Flash memory in (relative) abundance.
Android SDK Features

- No licensing, distribution, development fees (!)
- WiFi, GSM, EDGE, bluetooth, ...
- Sensors! (orientation, acceleration, microphones, ...)
- API for location services, GPS.
- Shared data storage (SQLite based)
- Camera.
- ...
- A mechanism for sharing application components.
Installing the SDK

Needed:

• Java DK (JDK)
  - Mac: already installed. Linux: sudo apt-get install sun-java6-jdk or something similar.

• Android SDK

• Java IDE (Eclipse is recommended)
  - http://www.eclipse.org/downloads
Installing the Android SDK

- Two main directories: tools and platform-tools
  
  export PATH=$PATH:~/android-sdk-ARCH/tools:~/android-sdk-ARCH/platform-tools

- You need to add at least one build target. This is a version of Android for which you want to build.
  - Type 'android &'
  - Install some development targets
Development targets
This is not compulsory, but it is strongly advised... and it's how I do it :-)

- http://www.eclipse.org/downloads
- Install one of:
  - Eclipse IDE for Java Developers (recommended)
  - Eclipse IDE for Java EE Developers
  - Install it in a local directory, not system-wide.
- Install also the Android Development Toolkit (ADT) plugin:
  - Select, in Eclipse: Help > Install New Software
  - Enter the following in Work With:
    - https://dl-ssl.google.com/android/eclipse/
  - (continued)
Eclipse (ADT Plugin)

- Select the ADT plugin (see figure 2.2). Click Next.
- Once the download is done, ensure both “Android DDMS” and “Android Developer Tools” plugins are selected; click Next.
- Accept the license... Next, ... then Finish.
- Restart Eclipse, Select Eclipse > Preferences, select Android from left panel. You need now to specify the location of the Android SDK.
- Click Browse... and navigate to where is the Android SDK, then click Apply. The list will update to display each of the available SDK targets (figure 2-3). Click OK to complete the installation.
Create an Android Virtual Device (AVD)

• See http://developer.android.com/guide/developing/tools/avd.html
• Start it with the 'android' tool from the command-line, or in Eclipse (Window > AVD Manager)
• You can create a virtual device, which is run by the emulator.
• The virtual device must have an API number at least equal to the one of the Android API you intend to test.
Creating a new Android Project

- File > New > Project
- Select what you want:
Create a launch configuration

- Use Run > Run (or Debug) Configurations
- Android Application > New
- Give a name to the configuration
- Enter Android Project, Target on which to run it.
A look to the hello world program
Ingredients: Activities and Intents

• An activity corresponds, roughly, to a screen. It is the unit of execution.
• An application is organized as a set of activities.
• Activities can be accessed individually also from other applications. E.g., you can start the activity to send an email from within your application.
• So, they are also a fundamental unit of re-use.
• Activities trigger one another via intents. An intent can either trigger a specific activity (specifying its class), or it can trigger any activity capable of handling a task (e.g., send a message).
Ingredients, cont.

- **Services**: run in the background, can trigger notifications, update content providers.
- **Content providers**: Shareable data stores. Defined by URIs. Preferred method for sharing data and achieving persistence. Based on SQLite.
- **Broadcast receivers**: receivers of intents, so you can configure your application to perform tasks for other applications.
- **Notifications**
- **Widgets**
The Application Manifest

Defines:

• Application name
• Minimum API, hardware requirements
• Permissions used
• Broadcast receivers

...
Process States

- Active processes
- Visible processes
- Started Service processes
- Background processes. (Activities that aren't visible and have no running services).
- Empty processes (placeholders to allow faster start-up).