Prof. Dr. Peter Thiemann
Matthias Keil Winter Term 2012

Introduction to Android Smartphone Programming

http://proglang.informatik.uni-freiburg.de/teaching/androidpracticum/

Exercise Sheet 1

1 Installing Prerequisites

In this exercise you have to set up your development environment for writing and testing Android applications. Google's developer website provides a helpful and detailed step by step installation walk through[1], which you are recommended to look at whenever you don't know how to accomplish a task.

1.1 Java Runtime Environment and Eclipse

Java Runtime Environment / JRE Since Android development is done in the Java programming language it is essential to install the Java Runtime Environment[2]. Note for Linux users: OpenJDK also works fine for Android development and can be installed easily on command line following the advises on the OpenJDK website[3].

Eclipse The IDE used in this course is Eclipse[4].

1.2 Android Software Development Kit (SDK)

Get the latest Android SDK[5]. If you chose to download an archive, extract it and run *tools/android*. Note down where you installed the SDK because that might be needed later. Whenever you are asked what platform or Android version you should install, choose Android 2.1.

1.3 Android Development Tools: ADK Plugin for Eclipse

To use Eclipse for Android development the ADK plugin has to be installed. The instructions to do so can be found on the Android developer's website[6]. Please note that if Eclipse takes a long time downloading and does not show any progress, it can help to completely restart the program and try to install the ADK plugin again from the start.

[1] http://developer.android.com/sdk/installing.html
[2] http://java.sun.com/javase/downloads/index.jsp
[3] http://openjdk.java.net/install/index.html
[4] http://www.eclipse.org/downloads/
[5] http://developer.android.com/sdk/index.html
[6] http://developer.android.com/sdk/eclipse-adt.html#installing

1.4 Platform Configuration

The Android version used in this course is 2.1, which is the platform that has to be configured to finish the environment setup. This step might not be needed if you did all previous steps correctly, but it can still help checking the correct configuration.
Launch the Android SDK Manager, which can be found directly in Eclipse at *Window > Android SDK Manager*. There you should check the checkbox for Android 2.1 and install it. You might want to uncheck Android 4.0 or whatever different platform is installed on your PC to avoid confusion.

2 Hello Android (5 points)

If you finished the first exercise you should have an working development environment that is needed to accomplish this task. Here you will create a simple "Hello Android"-Application with a GUI and see that there are two different ways to create it.

2.1 Create an Android Virtual Device (AVD)

Before you can run any Android application you need to have a AVD, which describes the target platform an application can run on. To do so open Eclipse, open the AVD Manager at *Window > AVD Manager*, select *Virtual Devices* and choose to create a new one. Give it a name, choose Android 2.1 as target and create the AVD.

2.2 Create an Android Project

You can create an Android project in Eclipse through *File > New > Project...*. There you can choose Android Project. In the next step, enter the project name *exercise1_2*, then your just created own AVD. As package name use *androidlab.exercise1_2*. Note that the default name for the activity just created must not contain a dash, so change it to a underscore instead.

2.3 Change the GUI

In the last step Eclipse already created a fully working application, which shows a blank screen. To change this you have to modify the sourcecode of the file named *Exercise1_2Activity.java*, which is the starting point for your application. You can find it in the package explorer in *exercise1_2 > src > androidlab.exercise1_2*. There you see a class that inherits of *Activity*.

Make the line `setContentView(R.layout.main);` a comment and insert the following instead:

```
1 TextView tv = new TextView(this);
2 tv.setText("Hello␣Android!");
3 setContentView(tv);
```

Also add `import android.widget.TextView;` to your other package imports.
Run the application to test it. Please note that this step can take pretty long because the whole Android emulator has to be booted.

2.4 Try out XML-based GUI Design

To use the XML layout you have to change the *Exercise1_2Activity.java* again and bring it to the state it was before the changes you did in subexercise 2.3. Again, don't delete anything but make it a comment. The design is now located in a different file called *main.xml* which can be found in *exercise1_2 > res > layout*.
Delete the content of the file and enter the following text instead. If you see a graphical layout instead of text, click on the tab called *main.xml*.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
3   android:id="@+id/textview"
4   android:layout_width="fill_parent"
5   android:layout_height="fill_parent"
6   android:text="@string/hello"/>
```

This will create the exact same layout as before. The only thing missing is changing the string that will be displayed. The last line indicates what text will be displayed in the TextView, which is a string with the name *hello*. It can be found in the file *strings.xml*, which is located in *exercise1_2 > res > values*. To see that you really did change something, modify the string with name *hello* to something else and run the application again.

Submission

Deadline The submission deadline is **01.11.2012, 12:00 (noon)**. Late submissions will not be accepted.

Project Create an *Eclipse Project* **exercise1_⟨number⟩** for each exercise. Use **androidlab.exercise1_⟨number⟩** as package name. Make sure that your project include all source files.

Report Your solution will consist of a *pdf file* **report1_⟨number⟩.pdf** with a description. The description must be limited to one page per exercise. Submitting more than one page will lead to reduction in points. The description may be either in German or in English. Clear and understandable style is required.

Submission Submit your solution to the subversion repository. Your solution will consist of one *folder* **exercise1_⟨number⟩** for each exercise which include the eclipse project and the report.

You are strongly encouraged to test your solution. Provide your source code with comments to understand the intention. Clear and understandable style is required.