

CMSC436 Fall 2013 – Week 2 Lab

Objectives:

Set up your Android development environment, familiarize yourself with it, and use it to create and debug simple applications.

Once you've completed this lab, you should have a initial working understanding of the tools developers use when they create Android applications. You should also know how to create and debug a simple Android application

Overview:

In this lab, you will set up your Android development environment by installing the Android Developer Tools Bundle, which includes the Android SDK, the Eclipse IDE and a number of other development tools and documentation.

You will use these tools to create and experiment with Android Virtual Devices, the Android emulator and Android applications.

You will need to create a Deliverables directory somewhere on your device. This directory will hold various work products that you will turn in to demonstrate that you've completed the lab.

Tasks:

- 1) Install the Android SDK and ADT (If you do not have it already)

Go to the Android developer's website and follow the instructions on the site to install ADT.

<http://developer.android.com/sdk/index.html>

You can, of course, use any IDE you want, we recommend using Eclipse for this course (we may unable to help if you are using a different IDE).

- 2) Create and experiment with Android Virtual Devices (AVDs) and the AVD manager.
 - a) Create 2 AVDs, each for a different style device (E.g., Nexus 4 and Nexus 7). For at least one of these AVDs, make sure that it supports a back-facing camera. Start that AVD in the Android emulator. Use the Camera application to take a picture.

Use the DDMS view to capture a screenshot of your emulator running the Camera application.

Save this screenshot as camera.png in your Deliverables directory.

- b) Start one of the AVDs. Open a telnet session to the emulator and issue the command "network speed edge". Open the Browser application in the emulator, Open the www.google.com website and measure how long it takes to fully load. Record this time.

Issue the command, "network speed full". In the Browser, go to the menu and hit the refresh button. Measure how long it takes to fully load the page. Record this time.

Repeat this process 4 more times.

What is the average load time under each network speed setting?

Save your data in the Deliverables directory under the title timing.txt.

- c) Connect to the console of the emulator instance and send an SMS message by issuing the following command, "sms send <senderPhoneNumber> <textmessage>:

You will enter your own senderPhoneNumber and textmessage.

Use the DDMS view to capture a screenshot of your emulator that has received your SMS message. Save this screenshot as SMS.png in your Deliverables directory.

3) Create and modify a HelloWorld application.

- a) Develop a simple HelloWorld app as presented in the lecture video. Once you've created it, run the app on the emulator.

Use the DDMS view to capture a screenshot of your emulator running the application. Save this screenshot as HelloAndroidInitial.png in your Deliverables directory.

- b) Now modify the app so that it says "Hello world! My name is [yourname]" Do this by modifying the string value in \res\values\string.xml.

Run the app on the emulator to see the new result. Use the DDMS view to capture a screenshot of your emulator running the application. Save this screenshot as HelloAndroidEnglish.png in your Deliverables directory.

- c) Create another string file for the Spanish language. Make the string value "Hola Mundo! Me llamo [yourname]."

Take a look at the following website, for further information.
<http://developer.android.com/training/basics/supporting-devices/languages.html>

Once you've created the string file, go to the Settings application and change the default language to Spanish. Rerun the app. You should now see the Spanish string, rather than the English string.

Use the DDMS view to capture a screenshot of your emulator running the application. Save this screenshot as HelloAndroidEspanol.png in your Deliverables directory.

4) HierarchyViewer

a) Use the HierarchyView tool (or the Dump View Hierarchy to UI Automator button in DDMS) to display and examine the user interface of the default Calculator App in your AVD. To do this:

- Start the AVD.
- Open the Calculator App.
- In Eclipse, open the DDMS view, then select the Dump View Hierarchy to UI Automator button, and select Calculator activity. Or use the hierarchyviewer program from the command line.
- Use your chosen tool to answer the following questions.

What is the height of the View tree?

Look at the view showing the plus (+) button.

What kind of View is its parent?

How many children Views does its parent contain?

What is the relative index of the plus button view in its parent?

Put your answers in a text file called, hierarchyviewer.txt. Save this file in your Deliverables directory. Some additional information is at:
<http://developer.android.com/tools/debugging/debugging-ui.html>

5) Download the source code for the Android 4.2 DeskClock application at:

<http://www.cs.umd.edu/class/fall2013/cmsc436/Extras/Labs/Week2/DeskClock.zip>

Unzip the file and then import the project into Eclipse. Don't worry about any compilation or syntax errors that might appear. Create a file called deskclock.txt. Write your answers to the following questions into that file and save it in your Deliverables directory.

- i) Open the project's AndroidManifest.xml file. The file has tags corresponding to all of the 4 fundamental component classes. Identify one instance of each fundamental component. For each one, identify its corresponding source code file and copy the line of code that defines the class as a subtype of one of the fundamental components (e.g., class X extends Activity, class Y extends Service, etc.)
- ii) To help you with this problem, read the following blog post.
<http://android-developers.blogspot.com/2011/07/new-tools-for-managing-screen-sizes.html>

Look in the res directory. How many layout directories are there? In a few short sentences, describe a specific device size and/or orientation that would use each of the different layouts you've found.

- iii) Look in the res directory. How many different values directories are there? Find two examples of resource strings that can be configured for different languages. Copy the xml definition for each string, from at least two different values directories.

- iv) Look at R.java. There is a field in the id class called "cities". In which function and class does this field get used? How does it get referred?
Hint: Use Eclipse's Open Call Hierarchy feature.

- 6) Logcat. Download the source code for the WikiNotes application at:

<http://www.cs.umd.edu/class/fall2013/cmsc436/Extras/Labs/Week2/WikiNotes.zip>

Using Logcat feature of DDMS, filter out messages relating to the WikiNotes application. Create a file called logcat.txt. Write your answers to the following questions into that file and save it in your Deliverables directory.

- a) What is the log tag used by this application's main Activity? Where do you see that in the application's source code?
- b) What is the message associated with the first Log entry emitted when this application starts running?

For more information, read the following webpage:

<http://developer.android.com/tools/debugging/debugging-log.html>

- 7) Heapview. Use DDMS to view heap usage for the WikiNotes process. In the Devices tab, select the WikiNotes process (com.google.android.wikinotes).

Click Update Heap button.

In the Heap tab, click CauseGC to enable the collection of heap data.

Create a file called heapview.txt. Write your answers to the following questions into that file and save it in your Deliverables directory.

How many objects have been created?

Which type of Object has been allocated most frequently?

How much memory has been allocated for this type of Object?

- 8) Method profiling. Use DDMS to view method execution sequences for the WikiNotes process. In the Devices tab, select the WikiNotes process.

Click the Start Method Profiling button.

Interact with the WikiNote application.

Click the Stop Method Profiling button.

Create a file called traceview.txt. Write your answers to the following questions into that file and save it in your Deliverables directory.

- a) Look at the profile panel. Find a method that is part of the WikiNotes application (in the com.google.android.wikinotes package).
- i) Record the name of the method?
 - ii) In what file is the source code for this method?
 - iii) What are the exclusive real time and the inclusive real time spent in the this method?
 - iv) Why are they different?

See <http://developer.android.com/tools/debugging/debugging-tracing.html> for more information on traceview.

- 9) Download the source code for the simple calculator application with its corresponding JUnit Test project at:

<http://code.google.com/p/simple-calc-unit-testing/downloads/detail?name=FullCodeDownload.zip&can=2&q=>

Then import the two projects into Eclipse and do following steps:

Run the SimpleCal app in the emulator.

In the SimpleCalTest, set a breakpoint at the line 91 (Function testSubtractValues()) of MathValidation.java

Start debugging the project (right click on the SimpleCalTest project, select Debug As, then Android JUnit Test)

Step in the function, debug it and use the watch windows to keep track the value of the variable "mathResult"

Create a file called debug.txt and then answer the following questions. Save the file in your Deliverables directory.

- a) What are the values of "mathResult" before and after line 94?
- b) Is the later value a correct result?

10) Compress your Deliverables directory (by using zip). Log in to the submit server at <https://submit.cs.umd.edu> and submit the zipped directory as your solution to project W2-Lab.