CSCD 372: Android Development
Lab #0: Getting Started and Using the Empty Wizard

There are two objectives for this lab. The first is to install and explore Android Studio, which is probably a bit more involved than other IDEs you have used. The second is to modify the code generated by the Empty Activity wizard so that it has nearly the same beginning functionality as that generated by the Blank Activity Wizard. We will discuss the differences between the code generated by these two Wizards in class.

1) Download Android Studio (should be easy to find from a web search) and install it. If the installer still gives the opportunity, I recommend that you specify your own install paths, e.g.: C:\android-studio and C:\android-sdk, because there will be occasions when you need to get to the sdk directory, and the default path is not very convenient.

2) Goto Quick Start/Configure/Settings/System Settings/Updates. Install updates if available. I recommend you stick to the Stable Channel.

3) Goto Quick Start/Configure/SDK Manager. I prefer launching the Standalone SDK manager as it lets you select subsets of the SDKs. You do not need any Android Wear or Android TV components. Under Tools, make sure you have SDK Tools, SDK Platform-Tools, and the latest Build tools. Do not take any tools from the Preview Channel. Under SDK Platforms, take API 23, 21, and 16 (we will be developing with the latest but testing under API 16). Do not take the Android N preview. Under Extras, make sure you have the Android Support Library, Google Repository, USB Driver, and (if your machine is Intel based) the HAXM Accelerator. You

4) Goto Quick Start / New Project. Specify something like "pschimpfLab0" as your app name and "paul.schimpf.com" as the company domain. The idea here is that your app and domain names should make your app uniquely identifiable in the universe of Android apps.

5) Press Next and choose a Phone app with a min SDK of API 16. Press Next, choose "Empty Activity" and accept all remaining defaults.

6) Goto Tools→Android→AVD Manager. Press New to create a virtual device with the following settings (you can create additional emulators if you like):

   - Device: Nexus 4 (press Next)
   - System Image: Jelly Bean, API 16, x86 (press Next)
   - If you have decent GPU acceleration for OpenGL, try the "Use Host GPU" option, otherwise check the "Snapshot" option (the former will speed up emulated processing, while the latter will make your emulator load faster – unfortunately the two cannot be combined, but HAXM will also speed up your emulation)
   - accept other defaults and press Finish

8) Run your project on the emulator. It should run with the "Hello World" message. If your system is Intel-based, then launching the emulator should give a message that HAXM is up and running. If that does not happen, find your Android SDK installation directory, and goto: …\extras\intel\Hardware_Accelerated_Execution_Manager. Then run silent_install.bat and/or

the .exe that you find there (you have previously changed your Windows Explorer View settings to always show file extensions, right?)

9) Find the resource file that specifies the content of your app's screen (res/layout/activity_main.xml). Change the text for "Hello World" so that it contains your name.

10) Add a file called menu_main.xml to your project resources (in a directory called menu). You can do this by right-clicking on the res directory and selecting New→Android Resource File, and hen setting the resource type to menu. Give it the following content:

```
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".MainActivity">
    <item
        android:id="@+id/action_about"
        android:orderInCategory="100"
        android:title="@string/action_about"
        app:showAsAction="never" />
</menu>
```

10) Add a string to your strings.xml file (in res/values) with the name "action_about" and the value "About".

11) In your MainActivity class, add overrides for onCreateOptionsMenu() and onOptionsItemSelected(). You can look up the method signatures or find example code for these in lecture module 2 or in code generated by the Blank Activity Wizard. Note that lecture modules are posted on the course website, and the URL and access code for that are in the course syllabus.

12) In response to the About selection, pop up a Toast or a Snackbar containing your name, lab assignment, and the course term. You can find example code for Toasts and Snackbars at the end of lecture module 2.

13) Run your project and make sure the About selection works.