CSE 5236: Android Studio Tips
Autumn 2019 Term
Instructor: Adam C. Champion, Ph.D.

Note: This document assumes you have installed the Java Development Kit and Android Studio; install them if you haven’t already done so. Android Studio is very CPU- and memory-intensive; builds can take a few minutes. If feasible, please add more RAM, an SSD, or both to your laptop (but **backup your data first** in case anything goes wrong).![image]

**Appearance:** Now launch Android Studio. Click on **Configure**, then **Preferences**. Select **Appearance & Behavior**, then **Appearance**. You should see a dialog like the one in Fig. 1.

![Figure 1: Android Studio Appearance Preferences](image1.png)

I recommend deselecting **Animate windows** (unless you have a fast machine) and selecting **Show memory indicator** so Android Studio’s memory use is visible. Make sure **Show tool window bars** and **Show tool window numbers** are both checked so you can select tool windows easily (e.g., Project, Structure, Android). If you have a widescreen monitor, check **Widescreen tool window layout**.

**Android SDK:** Now select **Appearance & Behavior**, **System Settings**, and **Android SDK** (in that order). Make sure you click **Show package details**. You should see a dialog like that in Fig. 2.

![Figure 2: Android Studio SDK Manager (Preferences)](image2.png)

There are numerous Android SDK packages that you can install, but you do not need to install all of them. Install at least one Android 5.x, 4.x, and 2.x SDK as well as Google Play services and Google repository (scroll to the bottom of the dialog). System images are provided for various ISAs (such as ARM, x86, and MIPS) and types of devices (such as autos, TVs, wearables). System images are **very large** and most of them are not needed for this course. I recommend installing only ARM system images for Android and Google APIs as many mobile devices use the ARM ISA. Downloading all these packages takes a **long time**. (Alternatively, clicking **Launch standalone SDK manager** will bring up a separate SDK manager process for installing packages.)

[1](https://sites.google.com/a/android.com/tools/tech-docs/configuration) shows Android Studio configuration notes. Android Studio 3.5 supports customizing its memory footprint: [https://developer.android.com/studio/preview/features](https://developer.android.com/studio/preview/features) I suggest a 1–2 GB maximum heap size for one large project.
**Code Style:** Once packages install, select Editor, Code Style, and Java (in that order). Click on Code Generation and enter m for fields and s for static fields. The result should look like Fig. 3.

![Android Studio Code Style](image3.png)

Figure 3: Android Studio Code Style

![Android Studio Line Numbers](image4.png)

Figure 4: Android Studio Line Numbers

**Line Numbers:** Next, select Editor, General, and Appearance (in that order). Select Enable line numbers in the dialog. The result should look like Fig. 4. Line numbers are very useful when debugging.

**Programming Font:** Finally, go to Editor, Colors & Fonts, and Font. You should see a dialog such as that in Fig 5. Ensure the programming font is visible and that similar characters such as 0, I, 0, 1, l, and ! can be distinguished. On MS Windows and Linux systems, the default font’s 0 (zero) and 0 (letter) are confusingly similar. On Windows systems, Consolas is a good programming font. On Linux systems, Cousine, DejaVu Sans Mono, and Hack are very readable fonts, and they’re free and open source. To make font changes, click Save As and enter a scheme name, then make your changes. Go to Editor, Colors & Fonts, and Console Font and repeat these steps.

![Android Studio Font](image5.png)

Figure 5: Android Studio Font

---