Introduction to Android Studio for CSA class

If you want to install and set up Android Studio on your own to work with the files in this course…

Follow pages 5 through 25 of the file Sept19-New CSA Installation Guide exactly. I will have us pick up with configuring AS on all of our lab computers which will start around step 11 found on page 10.

Some tips for working with Android Studio (AS)

1. When you want to get a copy of a new AS project, set up a folder in your AndroidProjects folder with the name of the project you are going to get a copy of, then extract your download into this folder.

2. Whenever you are going to modify/extend a project, make a copy of it first. Then change the name of the project you are working on so that you will always have an untouched copy of the original project to fall back on.

3. “close project” to get back to the original starter menu where you can import a new project.

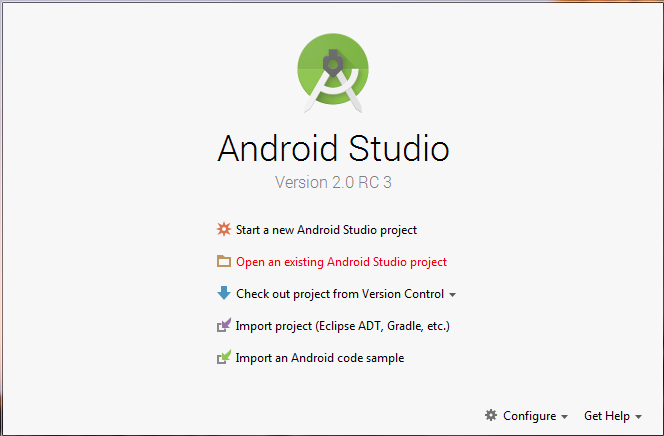
4. Click on updates in the Gradle Sync Messages panel (lower left); DO NOT click on pop-up update suggestions (upper right or other pop ups).

The end of 1.1.1 that we skipped…lets view an existing project

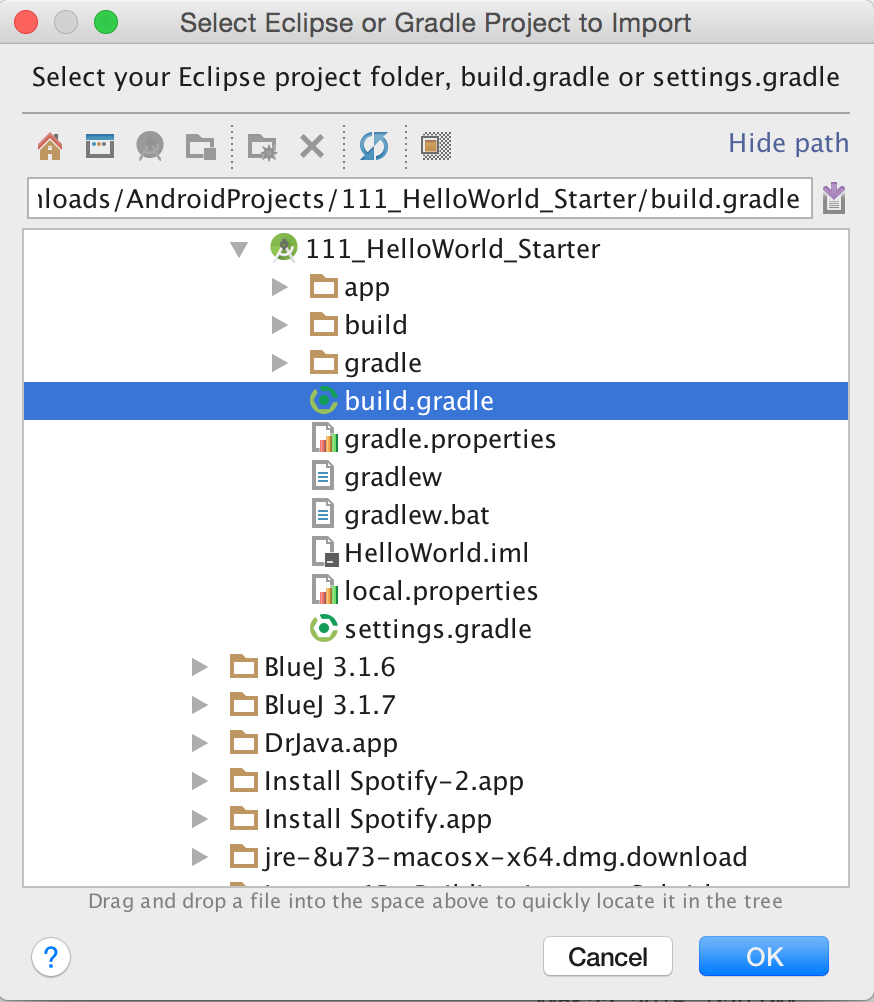
Part III: Hello Android

You will use Java to write programs called apps that can run on Android™ mobile devices, such as phones, tablets, and even wearables. To create these apps, you will use another IDE called Android Studio. You won’t be writing code in Android Studio quite yet; you need to master a few Java fundamentals in BlueJ first. For now, in this activity you will explore Android Studio to see what a “real world” IDE looks like.

1. To begin, review the 1.1.1 Introduction to Android Development presentation.
2. Similar to your BlueJProjects folder, create an AndroidProjects folder. Note there are no spaces in the folder name.
3. Get a copy of the 1.1.1HelloWorldApp Android project from your teacher. Copy or extract the files to a HelloWorldApp folder in your AndroidProjects folder.
4. Launch Android Studio. The first time you run Android Studio, it behaves differently than other times. You will see a welcome screen that asks you if you want to start a new project, open an existing project, etc.



1. Select Open an existing Android Studio project.
2. A dialog appears showing your file structure. Navigate to your AndroidProjects folder and then navigate to the location where you copied or extracted the HelloWorld project files. In the HelloWorldfile structure, select the file named build.gradle. It will be in the HelloWorldfolder, not in a subfolder.



1. Click **OK**.
2. You will see the Android Studio IDE window. You may see a dialog that states

The path to '\Users\...' does not belong to a directory.  
Android Studio will use this Android SDK instead: '\Users…'

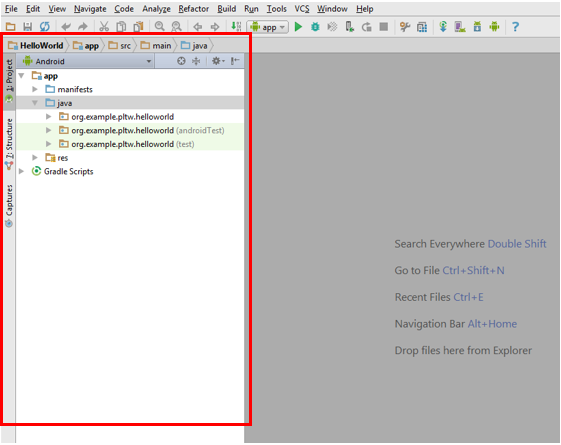
The *ellipsis* (…) indicates unnecessary information is not included. Click OK.

1. You will also see a “Tips” dialog. The Tips dialog is an optional feature that helps you learn the IDE. Feel free to read through some tips or dismiss the dialog. You may also choose to disable it for future sessions.

Gradle is a feature built in to Android Studio. It manages and organizes your project files and other files that are necessary for app development. Depending on your version of Android Studio, you may see one or more errors or warnings from Gradle in the Message Gradle Sync panel at the bottom of the Android Studio window, as shown here.



1. The specific errors and warnings along with their fixes are:
   1. “Plugin is too old…” Click “Fix plugin version and sync project” and follow the steps that are displayed.
   2. “Failed to find target…" Click “Install missing platform(s) and sync project”. You will be asked to accept a license agreement, do so.
   3. “Failed to find Build Tools….” Click “Install build Tools [*buildnum*] and sync project”. (Note the syntax [*buildnum*] means that a specific number will be displayed in place of [*buildnum*].)
2. Your HelloWorld project should be loaded and the Project panel should be showing:



If you do not see the Project panel (outlined in red above), select View > Tool Windows > Project.

Part IV: Observe Hello World

Running the Hello World app in Android Studio will help you become familiar with projects in Android Studio. After running the app and interacting with it, you will view the main activity file and the Java code it contains.

1. Start the emulator that your teacher has instructed you to use.
2. To run the app, select **Run** > **Run app** or click the green run arrow in the toolbar as shown.

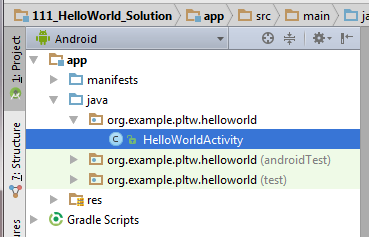


Note that it may take a few minutes for your app to load and run.

1. Enter your name. Click **Say Hello** and see the response.

Unlike your BlueJ program, an Android app does not have a main and therefore does not have a single entry point where the app begins. It does have a *default* entry point called a main activity. Android apps are based on activities, and you can view the main activity for HelloWorld called HelloWorldActiviy.java.

1. In the Android Studio Project panel, use the gray arrows to the left of appand the other folder names to expand the file structure and navigate to the project files.



Navigate to org.example.pltw.helloworld. Note: Do *not* navigate to the other structures org.example.pltw.helloworld (androidTest) or org.example.pltw.helloworld (test).

1. To open the HelloWorldActivity.javafile, double-click the filename. The code you see is much more complex than the code for your BlueJ project. For now, just note any similarity in overall structure.

Conclusion

1. What are some similarities you found between HelloWorld.java and MainActivty.java?

When you are finished, follow the instructions for making a similar app from scratch in the Getting Started with Android pdf doc.