Lappeenrannan teknillinen yliopisto

School of Business and Management

Software Development Skills

Marta Xiaoyang Moraga Hernández, 001101267

LEARNING DIARY, MOBILE MODULE

**LEARNING DIARY**

**08.02.2023**

Android Studio For Beginners

In this tutorial I learnt about the basics of Android Studio, like adding components to the view and basic scripting, and debugging. Debugging is similar to other programs. I did the simple app example.

I set up the repository as I did for other courses with *git* init and *git remote add* and downloaded Android Studio. The program has changed a lot from the video examples, for instance, to change to the preferences recommended in the tutorial, I had to access *settings*, which were represented as a gear symbol. While, on the tutorial, said options are accessed by pressing the program title.

Furthermore, in the current version it is necessary to confirm the refactor when changing the id of an app element and the options for debugging are in *Run*->*Debugging* *actions…*

Instead of concatenating *result+””* to show the result in the *TextView* as the tutorial did, I used *Integer.toString(result)*, because the program showed a warning.

**09.02.2023**

Android Studio For Beginners Part 2

I followed the tutorial and learnt about Activities and Intentions, and reinforced the knowledge from the first video. Android Studio has changed a bit from the tutorial version, it asks for more confirmations when adding an Activity than in the video, but it works the same.

The biggest issue I encountered was that the button supposed to open *google.com* was not working, probably because the difference in versions. After looking online I added the following filter to *AndroidManifest.xml*:

<activity ...>  
    <intent-filter>  
        <action android:name="android.intent.action.VIEW" />  
        <!-- Include the host attribute if you want your app to respond  
             only to URLs with your app's domain. -->  
        <data android:scheme="http" android:host="www.example.com" />  
        <category android:name="android.intent.category.DEFAULT" />  
        <!-- The BROWSABLE category is required to get links from web pages. -->  
        <category android:name="android.intent.category.BROWSABLE" />  
    </intent-filter>  
</activity>

This appears in the documentation for Android Studio: <https://developer.android.com/guide/components/intents-common#ViewUrl>

About how to load a web URL.

I also learnt that you can send data between Activities as key-value pairs and how to use said data.

**10.02.2023**

Android Studio For Beginners Part 3

I followed the tutorial and learnt about *ListView* and *ImageView*, and reinforced the knowledge from the other videos. As always there were some differences because of the Android Studio version, but everything worked the same. It seems nowadays *ListView* is not used as much, as it appeared in the *Legacy* section of the palette.

When creating the first *my\_listview\_detail.xml* layout I changed the text size by adding *android:textSize="32dp"* manually instead of editing an attribute, as I did not find the option. <https://stackoverflow.com/questions/13264794/font-size-of-textview-in-android-application-changes-on-changing-font-size-from>

Thanks to that I know learnt a bit about changing attributes by changing the script instead of looking for it in the attributes section.

The video also teaches how to add files directly to the folder structure. I accessed the drawable folder on my computer by right clicking the folder on Android Studio and copying the path/reference. I do not own the images used; they are the first results of looking up the words on Google.

I understand better the usual workflow, organisation, and development structure.

Personally, the most interesting parts were working on *DetailActivity.java*, since it reminds me of Object-Oriented Programming and Web Development. In this section I also learnt about useful features such as *BitmapFactory.decodeResource(),* which allows for a resource to be checked without loading it, or checking the system display.