LUT

School of Engineering Science

Software Development Skills

Jani Heinikoski, 0541122

LEARNING DIARY, MOBILE MODULE

**LEARNING DIARY**

03.11.2020

I read the course overview and the general course information sections from the course Moodle page. I learned that this course is intended to give me an edge in the industry by providing tools to create unique projects which reflect my current skills as a software developer.

Furthermore, I learned that the main environment used in this course is Android Studio which I have used for approximately 300 hours before this course. I found out that the version control system used in this course is Git which I am already familiar and comfortable with. I made a new public Git repository to GitHub and added this document to it.

After briefly checking out the course module task-list and watching the first video, I realized that there is not going to be much new for me in the “beginner level” videos. This learning diary might thus be a little empty regarding the task modules but when I get to the course project, many entries in this diary are to be expected. I will be sure to come up with new challenges and take this course as a chance to expand my skillset regarding native Android mobile development.

04.11.2020

While working on the Introduction module I learned that the EditText object method getText will not return a null value in any case. Instead, it returns an empty string if the String inside of the widget is empty. Another concept I learned during the Introduction module was theming and styling while following the Material design guidelines (documented in <https://material.io/>). Themes and styles in native Android development are complex concepts due to the levels of indirection (compared to e.g. CSS). I added support for a light (default) theme and a dark theme using Android developers documentation (available: <https://developer.android.com/guide/topics/ui/look-and-feel/themes>, referred 04.11.2020).

While watching the Core Elements -module’s video, I learned that IntentServices handle Intents (I have created my previous applications mainly as a single Activity using fragments which is the best practice nowadays).

After playing around couple of hours with animations, selectors, animated-selectors, shapes, lists etc. trying to create an animation for a custom button by using only XML, I stumbled upon a nice thread in Stackoverflow <https://stackoverflow.com/questions/35400318/custom-button-animation-in-android>. It was extremely hard to find from the Android developer’s documentation how to properly animate a View (specifically a button) using only XML. I have used sort of a bubble gum approach when animating buttons in the past (using mixed XML and Java code). This is the first solution which allowed to fairly simply attach a StateListAnimator to a button.

05.11.2020

As I watched the third module video, I did not really learn anything new. I have previously used RecyclerView instead of ListView in my projects since it is the best practice for displaying large and/or dynamic (set size not yet known) datasets. ListView is much easier to implement and has less overhead so it is acceptable to use it in this case since we only have an (ahead-of-time) known small dataset to display. Furthermore I only use SVG -vector based image assets as my image assets which can automatically scale to fit a container while maintaining same quality (no fear for app crashing or need to scale write own scale-down functions).

As I created the base project for the third module (initial commit to module 3), I encountered an error I have not seen before:

* “AAPT: error: failed writing to ‘~\GitHub\sds\_mobile\modules\ListViewDemo\app\build\intermediates\runtime\_symbol\_list\debug\R.txt': The data is invalid. (13).”

I did not find any solid fix for the problem by Googling it so I tried to downgrade my Gradle -plugin version from 4.1.0 to 4.0.2 which solved the problem.

While working on the module I learned how to use Dialogs by fiddling around with them for a while. Because it was extremely inefficient to change the Activity to display a picture about the clicked item, I used a dialog to display the clicked item.