Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Teppo Tyrväinen, 000098957

LEARNING DIARY, MOBILE 2020-21 MODULE

**LEARNING DIARY**

10.11.2020  
I enrolled to the course and read course overview and general information. I noticed that working methods seem to be very different from MOOC courses (Java programming) in Helsingin yliopisto. After that I read about Git and other modern version control systems. I have been responsible for version control of large SW projects in Nokia, but it was years ago, and the tools were very different then. I decided to install GitHub to get things done easy.

16.11.2020  
I started the day by learning the “Introduction to GitHub” and doing exercise “Hello World” to create a project, commit changes, create branch, and merge branches with Pull request. Before that I watched the videos provided. I understood the ideas easily (since I have background in Nokia phone SW projects) but at first, I was confused about the learning material and the user interface of GitHub. I also installed GitHub desktop application which seems to be another way for performing actions from local machine.

Next step was to select a proper editor. There was a list of recommended editors. I tried to find pros and cons, but there was no clear difference for me. I had installed earlier Android Studio and done some “Hello world” type exercises, so I decided to stay with that editor.

Now the setup was completed, and I started doing My First App with the help of video “Android Studio for Beginners Part 1”. It seems that the video was done with earlier version of Android Studio (an also with Mac), so things worked a little bit differently. Finally, I got the “calculator” working. The last part of the video concerning debugging and breakpoints was tedious, because line numbers and icons were not visible by default. Also, the breakpoints won’t work unless they are set after the last debugging session. At the end – it worked 😊

17.11.2020

So far, I had worked with Android project files only in my hard drive. Next step was to add them to GitHub. I was naïve, and thought I was just a couple of button presses away from ready. I had to use couple of hours to solve the problems. I needed to create a token (two factor authentication) and add correct roles or privileges to it. First time I added only three which were mentioned in Android Studio. Repository was created but files were not added due to wrong password. On second try I added all possible privileges to the token, but it still did not work. Then I deleted those two repositories in GitHub to start over. Consequence: Android Studio thinks they still exist. Then I found a way to remove those, but still nothing works. Repository is created but then it fails. I have to say that I am now very frustrated to this authentication problem: in first phase my six-digit token is accepted but then I am asked to give credentials and password and it NEVER works. I have looked from many places and videos without success. This should be explained in detail in written or video format since this is painful to learn!

23.11.2020

Started again searching information about authentication problems between Android Studio and GitHub. No big success.

24.11.2020

Today I noticed an email from GitHub that said I should update Git to newer version. I had installed Git in September and GitHub in November. After re-installation, the problems started to disappear. Last week I was depressed due to authentication problems, but I decided to use all the time it needs to solve them, because I will need those tools in all coming SW projects. Finally, I managed to push files to repository “MyFirstApplication”.

25.11.2020

I watched the second video and created Quick App Launcher. In the video there is some older version of Android Studio in use with Mac. Therefore, all things do not match with newest version in Windows. I was able to overcome those difficulties and build everything as described. However, in the end I could not start the browser with “GOOGLE” button of the application.

1.12.2020

I returned to study the problem with Google not starting in emulator. I tried again and again, but I did not work. I got an idea: id it works with real device the problem is not in the code itself but in a setting somewhere. I proceeded the steps to install Google USB driver, connecting my device to that driver etc. and after Windows restart my Motorola G7 Play (shame on me after working 14 years for Nokia!) was connected. Then I run the code and it did work! Still, with the emulator it does not work (by debugging I noticed that condition

“**if** (gotoGoogle.resolveActivity(getPackageManager()) != **null**)”

is false). I must face this problem someday, but now I am not motivated for that because the real HW works.

3.12.2020

Today I watched the third video and created List App accordingly. There were again some problems due to different versions of Android Studio (on video 3.4.? vs. 4.1.1). Also difference between Mac and Windows cause some difficulty to follow the instructions. I managed to run it both in emulator and in real device. There is still something wrong with the layout since the prices are wrapped to second line in the real device. I do not still really understand how the constraints of visible elements are adjusted.

I will come back to this exercise tomorrow to set it right and push it to GitHub. I must get back to the code many, many times to really understand it all. I will do that when doing my own project.

4.12.2020

Problem solved. I had a constraint (an arrow) between name and price, and that affected to the width of price text field.

7.12.2020

Today I uploaded the second and third apps to GitHub. I did not put much attention to README etc. I will learn that properly while doing the real project.

22.12.2020

Christmas holiday started today, and I am going to create my project application during next weeks. I had couple of ideas, but probably it will be a to-do-list / task prioritization application. I did a similar one with Excel VBA when I was working in Nokia R&D.

28.12.2020

Tomorrow I will start really working for the project ***“Tasklist”***. I have already done some specification work in my mind. At this point, there will be only one task lists. Later, I will add support for multiple lists and different views. Idea is that the tasks related to work can be separated from tasks done in free time. However, both are important and need to be supported. The other lists can be set to task related work, hobbies, construction projects, hunting or whatever. There may be possibilities for this application to be shared in Google Play, if I can implement all good ideas on to it. For this course I will limit the functionality to the minimum to be used as a tool.

29.12.2020

Today I started to write Requirements Plan for my project. As I have plenty of experience from my former Nokia SW career, I decided to do this project properly. This means that everything is planned and documented. The first thing is to specify the features that will be implemented. However, this cannot be done too far before trying out what can be done in (a new) Android system and how it will be done. So, I also started to create the project and will update the documentation while trying and implementing things. First thing with a new project, is to select a template. Options are not very intuitive, so I used some time watching YouTube videos about Activities. I could not find much useful information, so I proceeded with Empty Activity. Since I need first one list of tasks, I started to copy things I had done in List App and followed the last video at the same time. I added something new – priority is of type **int** and not String. Otherwise, it was supposed to proceed nicely. At the end of the evening, it compiled without warnings, but did not work. Maybe it will be clear next morning.

30.12.2020

It took an hour or two to find the problem. Of course, I knew that introducing new data types may cause problems. The problem was caused by **int**. Changing *priority number* to String type solved the problem. Then I added more items to the list to see how the list works if there are more items than can be shown and how switching between portrait and landscape (in real HW) works. It was fine. At this point I started to create my own classes to Task and TaskList. I wanted to have an enumeration or like for task readiness, so I made a class for defining constants (CREATED, STARTED, HALFWAY, PROGRESSING, FINISHED). In the ListApp example the data in the list was predefined, so it was reasonable to define it in **strings.xml**. Because my task data will be created by the user, I needed to change that. When I carefully studied the code that I had copied from ListApp, it was clear that current code was far from efficient and clear. In ListApp all values are stored to String arrays where they are then picked by index. In my data one task contains everything (and more) that is needed for ItemAdapter. Therefore, I changed the parameters so that I will give the TaskList as a parameter instead of many String lists. And finally, it works!

4.1.2021

Time flies! Today I need to get one-task-view finalized. There will be two purposes for it: first to look at the details of one task and modify the information. Edit mode will be used also for creating new tasks. I assume I can use the same view for both purposes but let us see. Now I have managed to create a simple detail view for one task. There was and maybe is a problem with delivering data back and forth between list view and detail view. I would like to pass task items (my own class), but this seems to be hard. Therefore, I must use PutExtra and GetStringExtra methods which feels clumsy.

7.1.2021

I noticed that there is a 60-day limit for returning the final project. I am not sure where that time is calculated, but the time will end soon. I suppose if I return my project during this week it will be inside 60-days limit. Anyway, this means that a) I must rush and b) I need to reduce planned functionality to minimum to make it ready. When this course is completed, I will start (test) using this app for my own purposes and create new versions as needed. Those versions I will put to GitHub in ***private*** mode since there is a possibility that this app will evolve as a real app to be spread through Google Play.

9.1.2021

I was able to add jpg pictures showing readiness (5%, 25%, 50%, 75%, 100%) to task detail activity, but not to static ItemAdapter which populates the list view. So, I decided to give up and include the graphs only to detail view. Next problem was to decide how editing would be done: have both TextView and EditText components on the same activity on top of each other or have separate activities for showing and editing information. I selected the latter design.

10.1.2021

I had some problems adjusting the edit/view activities look similar enough, but at the end I got it working both in the emulator and in HW (Motorola Moto G7 play). Using ConstraintLayout needs time to understand and learn, but finally the view was good enough in both. I have one older Samsung phone which I wanted to use for additional testing. Sad thing is that I should have selected very old (4.4.4) Android version in the beginning – and I really do not remember if that was at all possible for this new SDK version. So, at this point, I will do testing only in emulator and my Motorola.

11.1.2021

Today I managed to make data transfer working between view and edit modes and saving changed values to TaskList object. While working with details views, I noticed that in some cases all fields do not fit to the screen when keypad is visible. I found a good video from YouTube (<https://www.youtube.com/watch?v=DpFNfQzhKQM>) which explained how to add a ScrollView around a ready-made ConstraintLayout. I did this to both detail views easily. Now I remembered that in the course requirements it was said that all links to *stack overflow* and such hints should be mentioned in the diary. I have not done this due the fact that there are tens or even hundreds of articles and videos I have used. Previous link was an exception to this. I still have not done the routines for reading and writing data to and from a file. I suppose those along with task deletion, list sorting etc. needs to be in TaskList class.

12.1.2021

Today I started by developing missing methods for TaskList. Those include deleting a task, sorting, and rearranging TaskList after one task has changed its priority. Passing data between activities was not very straightforward as I expected. It took hours to get the data pass form edit activity to main activity and finally rearranged to TaskList object. At the end, the problem was updating the list view (main view) after changing the data. The solution was to put *notifyDataSetChange* command to right place (this was also found from stack overflow).

14.1.2021

Last missing crucial parts of this application are now the read/write file operations. That is my agenda today. After that I will make some more testing and start finalizing the project.

My purpose is to save tasks as **objects** – not as a coded text file. This causes some difficulties. This might be something I have never done and not mentioned in basic programming courses. *(I remember one project done back in year 1991 where I used Object Pascal (Turbo Pascal 6), but I probably saved those objects in dBase IV format).* It was fair easy to find examples of **FileOutputStream** and **ObjectOutputStream**, but those did not work. Errors seem to link to directory structure which I do not know. In Windows, DOS, Unix or Linux you can easily list directories and files. Then I found (with the help of Google) *Device File Explorer* from Android Studio. At the end this was only nice-to-know – I got it working with a simple command context.openFileOutput and correct try-catch frame.

15.1.2021

So far, I had tested this app only with sample data (created in program). After removing this, I noticed that this app does not have add functionality. I added a FloatingActionButton to the main activity (on top of the list view). It needed some logic programming since the User can change the default priority (last task) before saving the new task. However, finalizing this was quite straightforward. Next thing I realized was that the app needs also delete functionality. I had implemented delete to TaskList, but in the UI there was nothing.

18.1.2021

I started to create a new activity for the dialog “Do you really want to delete this task permanently? Cancel/Delete”. Before I got it ready, I realized that I wanted to create a Popup window – not an activity using the whole screen. After browsing the net for a while, I found a promising Popup Window example from a site called ***codelocker.net***:

<https://www.codelocker.net/p/101/android-create-a-popup-window-with-buttons/>. This example uses LinearLayout instead of ConstraintLayout which seems to be the default in my layouts. I do not fully understand the differences between them, but I will continue with my default. Unfortunately, the code I created did not show any window. After searching more, I found a new type of object, Dialog (You tube id: ARezg1D9Zd0). I managed to get popup window visible and get “button” events back to the parent activity (not the main activity). The problem is, how to get main activity (list view) visible again. I am stuck with this now and need a break. Next: home office renovation!

29.1.2021

After renovating my home office and finalizing MOOC course “Building AI” (Helsingin yliopisto) I have time to continue with this project. I really hope that 60-days’ time limit is only a threat for lazy students. I was planning to learn Android Studio on my own, but it is a shame if all this work will be invisible to others (no points).

1.2.2021

At this point I forgot to update this diary along my doings, so it will be a lot shorter in this period. After working hours with Dialog views, I managed to get the right activities visible after the User presses selection in Dialog popup. The problem was that in one case the information must be given directly to the previous activity (view) and in the other there was another activity in the middle.

2.2.2021

Now it was time to test file reading and writing of the TaskList class. Surprise, there were errors which I debugged in the next hours. I started to go clean the layout files. Most of the texts were hardcoded. I created layout texts as keys for values in *strings* file. Since I knew the idea and technique of localization (from the Symbian times in Nokia) it was easy to create both Finnish and English versions with Android Studios string editor. It would be easy to add maybe German and Swedish versions, but I do not see the value at this point.

3.2.2021

The default launch icon for new applications is an Android symbol. I wanted to try to make my own. Target was to have numeral list visible in the icon. The challenge is of course the small size – 72x72 pixels. I am not an artist or even talented in drawing, so I was forced to “steal” a screen capture of a larger picture and resize very small. I have learnt earlier to use Gimp, so I managed to copy only the right information and make the rest transparent. Then I added a green circle where I added the list. The result is not something I could use in a real product, but I met the target to have own unique icon.

9.2.2021

Today I learnt to create README.md file. There were fine examples what this file could include, but here I have used a minimal approach. I hope everything crucial is included.

10.2.2021

Video time! I am planning to use both Android phone and Android tablet in the video. On phone I have English version and in tablet Finnish version. I am taking the video with a “real” digital Nikon camera, since that can be attached firmly to camera tripod.

Before I started to shoot the video, I understood that there needs to be some kind of plan or story which will be followed. It was unbelievable how tedious it was to make a good plan. However, I wrote it with Word like a Movie script, so it would be as simple as possible to do the filming. I will add this script file to Git. When I started to use the camera, I noticed that even with a good camera (Nikon is!) it is extremely difficult to take a video from phone screen with miniature fonts. I decided to make most of the video from emulator (using OBS Studio) and add then short videos from a physical phone and Samsung tablet.

11.2.2021

This is – I hope – the last day I spend with this project. Today I started to edit and combine those four videos I made yesterday. I tried couple of video editing programs before I found VideoPad. It is easy enough to use for an unexperienced user but has many capabilities. When adding the last video shot from Samsung tablet running Finnish version, I noticed there was a bug. Command strings in popup windows were hardcoded in English. After correcting this I had to take the video again from the tablet.

12.2.2021

Yesterday was not the last day on this course! I have spent many hours for tuning up the video. I must admit, the result is not bad. I am not ashamed to upload it to YouTube. Next, I will upload this file to Git and finalize whatever there was to do in Moodle.