Lappeenrannan teknillinen yliopisto

Sofware Development Skills

MONTBLANC Emmanuel, 000740166

LEARNING DIARY, Software Development Skills: Mobile MODULE

**LEARNING DIARY**

14.4.2022

I checked the general information on the moodle page and downloaded android studio. I also did some research to remember the basics of GIT, that I already used before for other projects, and set up my repository.

I had to install java and java SDK that I didn’t have on this computer.

I then created my first project and familiarize myself with the IDE that is really similar to PyCharm, which I use a lot.

20.4.2022

I followed the tutorial and understood how to add a TextEdit, a button and a textView to an app.

I created a virtual device to be able to test the app. I add an error at first while launching the virtual device and after some research the problem was just that I didn’t have enough space on my disk.

I learned how to get access to the items in the app like a button or TextEdit and how to add an action to a button with a onClickListener.

I understood how to use the debugger to see the value of each variable and find precisely where is the error in the code.

I then added the learning diary and this first android app to the GitHub.

02.05.2022

I understood the meaning of the core elements to android development.

I created my second app following the second tutorial, that is composed of two buttons: one to launch another activity and one to open a browser and go to an URL.

I am starting to get familiar with how to position widgets on the screen, find the widgets in the activity with findViewById and with how to connect buttons to actions.

I discovered how to use intent to launch other activities or app and pass information to those other activities. Also, I learned how to check that there is an app or activity that can answer the intent on the phone before executing it.

Finally, I added this second app to my GitHub.

03.05.2022

I followed the third tutorial and discovered relative layout and listViews.

When constructing my relative layout, there was no arrows appearing allowing me to choose where to put my first textView that was then stuck to the top right corner, I couldn’t find in the parameters of the textView where to fix this, so I simply changed it in the xml file.

I struggled at first to understand what was a layoutinflater, but after some research I understood that it is simply a class to create a view element from a layout file.

I understood how to adapt items to a listView, as well as importing and scaling images.

I added this app with listViews to GitHub.

I saw that the listView were considered legacity and their usage were deprecated, so I looked up what is the replacement for it: RecyclerView.

I tried to implement the same app using recyclerView instead of ListView, by following simple RecyclerView examples (<https://stackoverflow.com/questions/40584424/simple-android-recyclerview-example>) and modifying it. I struggled a bit and realized that the problem was that I copy -pasted the activity to display the image, but doing so, the activity was not added to the manifest and caused errors. Once I got it fixed, I added this new app with recyclerViews to GitHub.

04.05.2020

I thought about an idea for my project. The idea would be to a basic tic tac toe app. The first activity would consist of two button to choose if we want to play versus another player (localy) or vs a IA. After choosing, a second activity would load, and we would pass as extra what kind of game was chosen. This second activity would be the main part where we play the game. I saw that recyclerViews can also be used to make grids, so the idea would be to make a grid, and detect when someone click on a case. After clicking on a case, it would display an image of either a cross or a circle on the case clicked. When the game is over, there would be a last activity for a victory screen, announcing the winner.

05.05.2022

I started my project. In a first time I worked on the first activity.

I created the two buttons, that open another activity and pass an Int representing the game mode chosen. For the first activity I also wanted to display a cross and circle as logo to make the app looks better, and I spent a lot of time trying to set the size of the two images to a percentage of the width of the screen. I found that you can simply use layout\_constraintWidth\_percent in the xml file and set it to the percentage wanted. But to use this, I also needed to add to Gradle file a new dependency.

I also created the second activity, that contains for now only a textView displaying the chosen mode to check that the information was transmitted.

06.05.2022

I did a lot of progress on my project.

I first created a recyclerView which served as a grid for the game. To make it a grid, I just needed to implement GridLayoutManager instead of a LinearLayoutManager and specify the number of columns. I quickly managed to make crosses and circles image show up in the cases after clicking on it, but my problem was that the onClick method was inside the viewHolder, which meant that I had no access to the state of the game for the other cells. I then found that you can implement the onClick method in the parent activity by using an interface, so I went with this solution. I was then able to implement the game in itself, like switching turns, tracking the state of the game and checking if the game is over.

I then went on to implement the ‘AI’ for the versus ai mode. In this mode, the opponent will just choose a random valid play, and not the most optimal one, as if it did, you would never be able to win. For this mode, I needed to change the image inside of the cell of the recyclerView after the ai play. To do so, I found that you can get a viewHolder inside the recyclerView from its position, and then get from the viewHolder the imageView.

After this day, all I needed to implement left was a victory screen displaying the winner (or draw), and the history of the turned played.

07.05.2022

I created a new activity for the victory screened, that is launched when the game is over, I passed to it as extras: the game mode, the result of the game, and the history of the turns played.

I implemented a textView to displays the result of the game.I then created a button to go back to the main menu (the first activity), so we can play again without needing to click the return button twice. I found that to do so, we can create an intent to the mainActivity normaly and just add a flag ‘Intent.FLAG\_ACTIVITY\_CLEAR\_TOP’ to the intent to close the others activities.

Then I implemented the history of the game with a recyclerView, displaying for each turn, the turn number, who played this turn and what was the play. I then did some testing and fixed some minors details and finished this project.

10.05.2022

I recorded the video of the project running and uploaded it to YouTube, I then added a file to the git repository with the link of the video. I then created the ReadMe to explain how to run the project and added it to the repository. Finally, I added the link to my repository for evaluation to Moodle.