Lappeenrannan teknillinen yliopisto
School of Business and Management

Sofware Development Skills

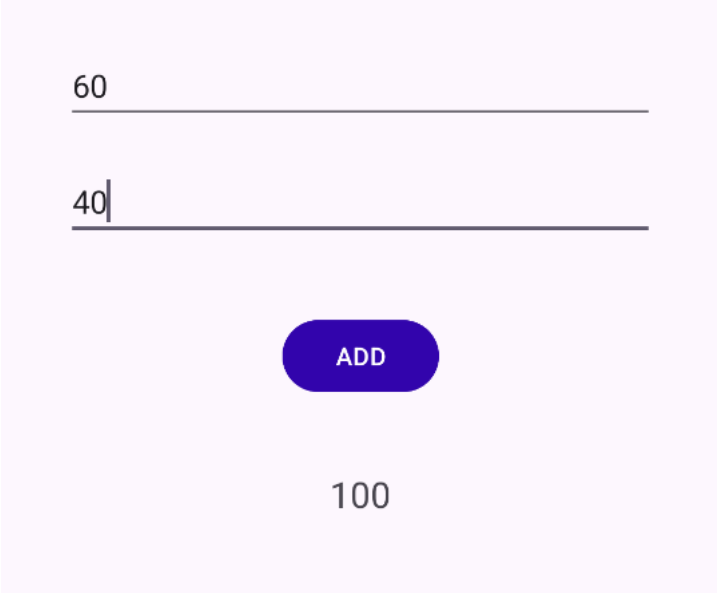
Aino Pöyhönen, 002675190

LEARNING DIARY, ANDROID MOBILE DEVELOPMENT MODULE

LEARNING DIARY

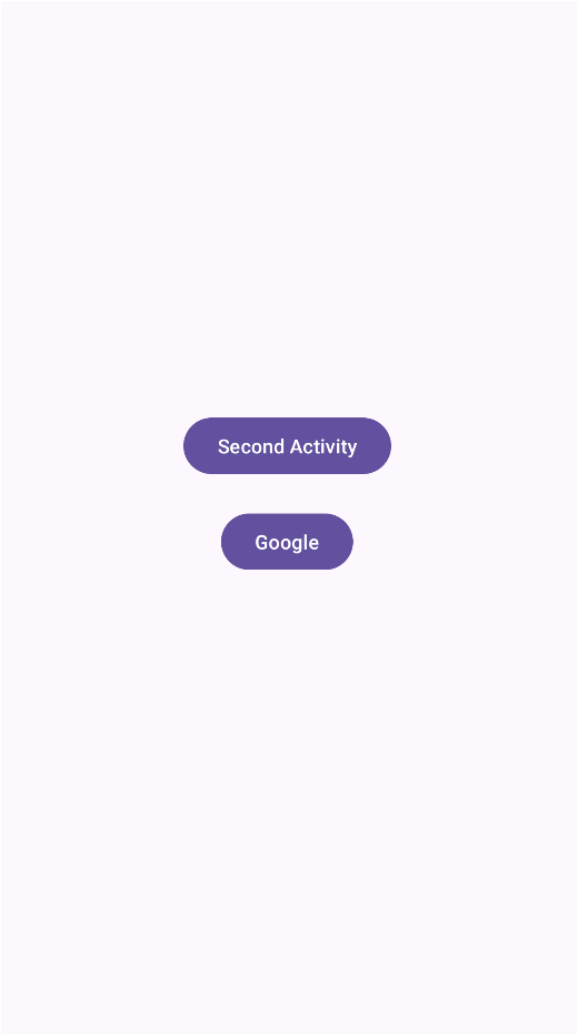
18.5.2024

I find the Moodle page a bit difficult to navigate, but I read through all the information, and I am ready to start working on the course assignments. I passed the Object-Oriented Programming course, so I am familiar with Java and Android Studio. As such, I went over the first Introduction YouTube-video quickly. I work with the xml-files very differently, I write in the xml-file and haven't used the palette in the Android Studio GUI. In the first YouTube-video a very simple two integer adder was made. Here is a screenshot of the adder I made by following the video:



Next up is the second YouTube-video. Launching another activity within our own app is something I'm already familiar with from the Object-Oriented Programming course, but launching another activity outside the app was new. The emulator didn't find the browser (Chrome) that is automatically installed, so the Google button didn't work. I ended up testing the app on my boyfriend's Samsung phone, which is a good skill to learn as well. I found

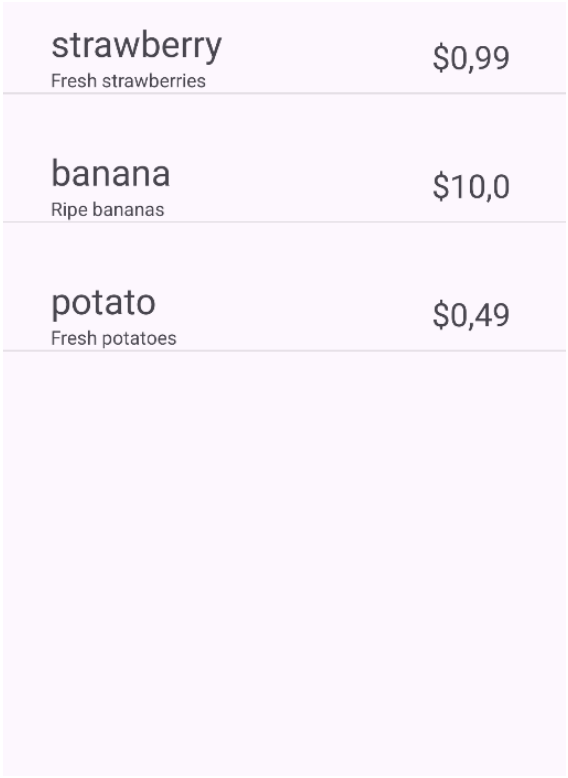
the YouTube-video to be quite useful, and here are screenshots of the app I produced following the instructions on the video, minus the google link I tested on a physical device:

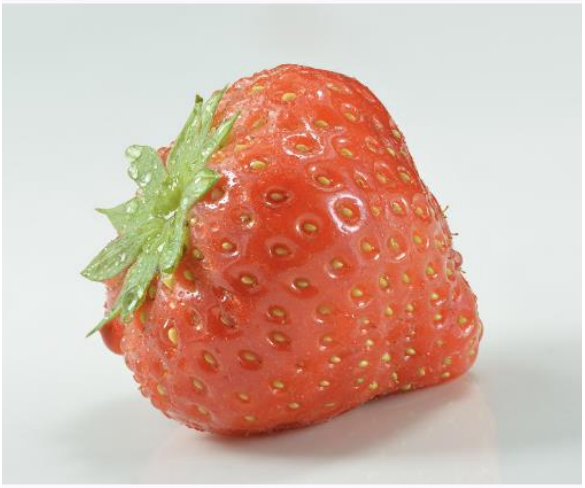


26.5.2025

Finally, I have reached the third YouTube-video. This one was very interesting, because even though I have messed around with it a bit, I haven't utilized strings.xml in any of my finalized projects before (instead I have ended up hardcoding all the strings). I haven't used ListView either. However it is worth mentioning that according to Android documentation, RecyclerView seems to be the recommended view for displaying lists [1]. The displaying and manipulation of the images was also something I wasn't familiar with. Overall, I had to

follow the video this time very carefully, and making the app took longer than the other two assignments. Here are screenshots of the app I made following the instructions on the video:





I also had a bit of trouble getting all my projects to the same GitHub repository. I tried to rename these Android Studio projects with more accurate names to help with course evaluation, but the package names still have the old project names used (sds1, sds2, sds3). I hope that this does not cause any issues when running the apps on other devices.

I have started working on the project task. My project will be the guessing game I tried to do for the OOP-course, but back then I felt like I wouldn't have enough time to do it and coded a Lutemon game instead. Here is the introduction I had in my OOP-course project proposal: "This document proposes a course project topic for the course Object-Oriented Programming (OOP) 2025. The course project proposed is a mobile version of LoLdle – a quiz game related to the popular League of Legends (LoL) video game. LoLdle has a web application/game freely available in Appendix 1 as well as a mobile game available in Google Play for Android, and in AppStore for iOS. The game's core idea is to guess characters (champions) based on various clues related to them.

The scope of this project regarding the gameplay is more limited than LoLdle – the focus is more on good design and OOP principles. In the proposed version, the user only guesses champions based on a full or partial splash art, depending on the difficulty chosen."

I will only do the hard difficulty, but this time I will try to add a statistics page, with similar data to the one original LoLdle has. Today I designed the UI and started with the data fetching from Riot's Data Dragon. Working with the Json-files in from Data Dragon isn't easy, but each line of code is closer to making the full splash art appear for the user.

27.5.2025

My app is now successfully fetching and displaying whole splash arts and the language can be changed in-app between finnish and english! Next up, I need to figure out the actual game logic and how to display only a part of the splash art, that zooms out a bit each try.

5.6.2025

I have now progressed quite a bit and probably should start writing to the learning diary a bit more often. I'm done with the game logic for now. I just need to make the app more visually appealing by showing wrong guesses in RecyclerView and giving better feedback from a correct guess (like showing the full splash art). It took me a long time to research and figure out how to randomize the crop and make it successfully zoom out with each time. At one point it randomized the location with each zoom, but it only needed another parameter to track center x and center y for the zoom out to work smoothly. I also needed to be careful with the coordinates going out of the bounds of the original bitmap, as this would cause a crash.

The problems that fetching a random splash art brings is that some champions share a splash art and with my implementation (the splash art fetching is from Riot's Data Dragon), there isn't much I can do. That is why I implemented a generous "solve this for me" button. Additionally, randomizing a crop will sometimes result in showing the champions face, and then the game may be too easy. As the game is fun otherwise and some people can't tell the champions even from a face, I'm not worrying too much about these shortcomings right now. I'll try to build a working app first.

12.6.2025

Past week I haven't really done anything, but I changed around some methods to better line with object-oriented programming principles. I also realized that the method that I currently have for cropping the splash could be split into smaller methods, so I transferred it to its own class in utils package. I'm planning to forget about the statistics view completely. Unfortunately, I'm having a hard time understanding Room and SQL and I realized I don't have that much time left for the course. I'll now try to make the app work as smoothly and look as good as possible.

16.6.2025

I have now added a working RecyclerView that displays previous wrong guesses and for now commented away all code regarding the statistics view. Since I will need to do multiple views for the final project and so far I only have one that works, I'm thinking about creating an extra game mode where the user guesses the champion based on a random skill image.

17.6.2025

I have implemented a mode where the user guesses the champion based on a random skill image. It was very easy to implement, since I mostly just copied code from the splash art mode. To make the mode harder, the skill image is greyscale and rotated randomly. I am now ready to return this project. It isn't exactly as impressive as I first had in mind, but it reflects my current skills and I am looking forward to continuing improving with each project.