Jason Chandler
Probability & Applied Statistics

I have honestly never understood the appeal of Androids. My friends who are more into technology have raved about it, but I did not understand what they meant until this project. The entire program being in the Java output is limiting. It keeps all of my hard work contained within a small box inside the IDE. This project made me realize that by showing me how Android apps work.

To start, I downloaded Android Studio. It was easy and free to download, but there were some extra hurdles I had to go through to get everything else working. For starters, I had to enable the hypervisor in my PC's BIOS. I had done something in those settings before when prompted for an update, so this was not that bad. The hard part was figuring out what the issue was, as the IDE did not explain it well. I would just go to start a VM and get an error. Then, when I did eventually find the hypervisor to download, it would not work upon multiple downloads. Eventually, I enabled it in BIOS and my Windows Features.

After enabling the hypervisor, I had to download the VM and make sure everything was set up. This took a ton of messing around with the SDK manager. I downloaded and redownloaded the key components until everything ran properly. From there, I tried messing around with the IDE myself, but nothing looked familiar to me. I made an empty project and got even more confused. I was met with Kotlin, XML, and coms. It was nothing like the Eclipse I am used to, the 2021 version I downloaded then.

To understand how to use the IDE to make a Hello World app, I followed this tutorial: https://www.youtube.com/watch?v=J9uahzEYg8s. It was super quick and taught me exactly what I was wondering at first: How do I start? I made a new Empty Views project. This made a default app with Hello World in the center. To change this text, edit the text option in TextView. I changed it to a bunch of exclamation points.

As a simpler alternative to Android Studio, I transferred my Pokemon game app to my iPhone by running it on jDoodle. jDoodle is an online IDE that allows users to run code straight from the browser. This makes this IDE ideal for simply handling my code as is. Everything prints and scans normally.

This method is much simpler than the Android Studio way. It allows me to bypass converting anything into a usable Android version. The XML file in Android Studio particularly confused me, and the visualization element made it worse for me. It almost looked too easy. I am used to and like Java's more rigid nature, whereas I felt Android Studio IDE was UI-focused rather than logic-focused.

I found using Android Studio frustrating. It looks like Java, but the XML stuff makes it entirely different. The way it functions is unlike Java or any other language I have experience in. Following Part 2, I know various languages now, and I cannot begin to relate how Android Studio functions to them.

I really enjoyed this class as a whole. I like how many opportunities for me to guide what I learn. Assignments were a bit vague, which forced me as a student to get stuck and figure things out. I have always preferred that style of learning and working. Furthermore, the assignments allowed for a lot of customization, like the original Pokemon game and the Report from this project. I felt like those two in particular have challenged many facets of my knowledge. I had to conceptualize and execute plans to ensure I covered all aspects of the tasks. At the end of each of them, I had a finalized project that I was really proud of. It gives me hope for my future in programming and my education, and I am excited to go into next semester with this new outlook.