

When I first started to work with Android Studio, I ran into a handful of problems and difficulties, especially since it was my first time navigating around on Android Studio. One of the first major challenges I came across was trying to understand how to set up a project without relying on any of the pre-configured activities that Android Studio would automatically provide. For example, even though there are a number of templates that can be used within the platform to start a new project, I chose the "No Activity" option, which by default forced me to manually configure both the layout and the Java files. This type of hands on/ under the hood process was quite overwhelming at first, to say the least. I am appreciative of how close in resemblance Android Studio and Visual Studio Code are, never the less it still took me some time to grasp exactly where the MainActivity.java file should be placed, and how exactly I was to correctly link it to the activity\_main.xml file to make sure that the project was able to function properly.

Another problem that I ran into was figuring out how to properly style the UI, especially when it came to customizing the button and the *ActionBar*. Adjusting the text and size of the button wasn't too hard to do, but for example, when I tried to change the background color, I started finding new challenges I didn't expect. At first, I thought I could just make the changes directly in the XML layout file, later I discovered that I was supposed to define colors and styles in the themes.xml or styles.xml files as it's a more efficient and consistent method. For example, I found that using the *android:backgroundTint* property was a good way to change the button's color. None the less, the *ActionBar* gave me a harder time because I had to adjust both the *colorPrimary* and *colorPrimaryVariant* attributes in the theme settings to get the black color I wanted. This process wasn't very straight forward for me, but after some research and troubleshooting, I was finally able to make the UI the way that I wanted it- which was quite honestly a decent conclusion for my fun pilot project.

Another thing I struggled with was how Android Studio shows the warnings and error messages, this type of syntax took a little bit of patience and time to get to understand. For example, I had a problem with the *ActionBar*'s color where it simply wouldn't change to black, no matter what, even though I thought I'd set it correctly in the XML file. Nevertheless, after debugging the problem for a while, I eventually found out that I also had to change both the *colorPrimary* and *colorPrimaryVariant* in the theme settings to get the color to update properly. This wasn't something that was straight forward from the error messages, and figuring out how themes and styles worked together in Android was definitely one of the tougher parts of this entire process.

None the less, all in all even though I ran into a fair number of challenges, the experience still helped me get much more familiar with using Android Studio, its features and how they work. I do still have a few questions, especially about managing the *AndroidManifest.xml* file, for example with using the new Android 12+ requirements.

Nevertheless, I'm feeling fairly confident that the more I work with this IDE, the more I will

learn and be more comfortable with navigating around and building projects. Even though there's definitely a learning curve to every new IDE, Android Studio resembles and follows a lot of similar layouts from Visual Studio Code, which makes it a little bit easier to learn the back end, rather than starting out at square one with a brand-new interface and not being able to navigate the IDE or a project at all at first. My most infamous and favorite example to use here is Eclipse and how using that IDE can be massively helpful in maybe 5% of very niche use cases, but in the other 95% is left forsaken. It takes time and practice, but it's nice to be able to flow into a new IDE without it being an experience like I had with Eclipse.

---

---

### *The code to the AndroidManifest.xml & the application running on an Android Emulator:*

