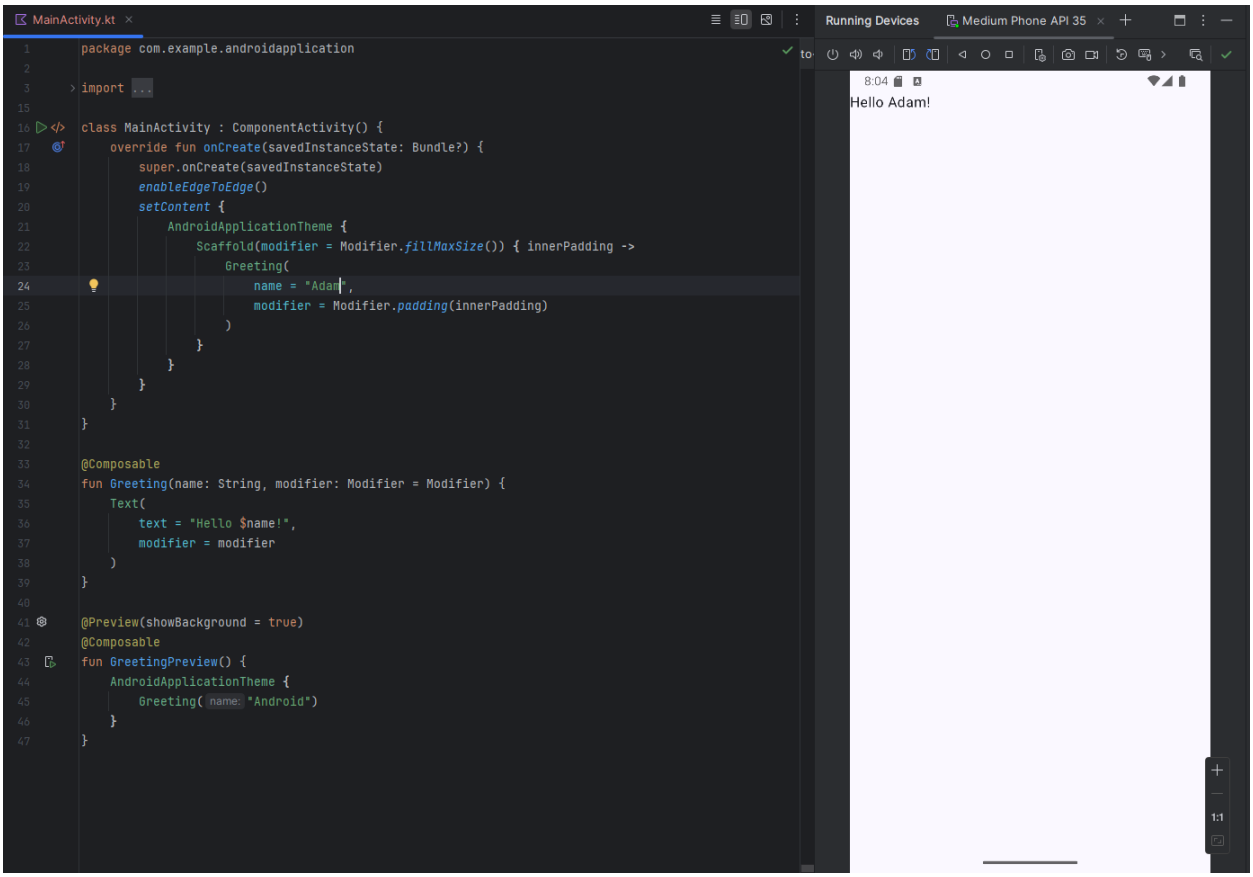
Journey With Android Studio

My experience with Android Studio goes back to high school. We used it to make very simple games that were visually unappealing. For this project, the professor had us take code we wrote for a java console app and try to apply it to work in android studio as an android app.

I started with the base program, which I modified with my name, as an intro “Hello World!” type project. This was all working well other than some issues I had to diagnose with my hardware, resulting in me having to enter my BIOs menu to turn on SVM mode. After this, I turned off the Windows Hypervisor platform. These issues were the root cause for my Android Emulator not working. After I got the emulator working, I started to implement my code into Android Studio.



Trying to merge my Pokemon assignment code into Android Studio came with many challenges. One of the first hurdles I faced was Kotlin being the primary language of Android Studio. I first

started migrating my code over to the Kotlin syntax, but this started coming with many difficulties. Kotlin is vastly different from Java in some of the features. I then found out I could use my Java code in the app directly without Kotlin, so I started to do this. I found some other issues involving android studio wanting to use an older version of Java, so I had it point to the version of Java I used for the original project. This was all working well. I next decided the two main changes I needed to make to the code was instead of using System.out.println() for my outputs, I had to map those outputs to a textarea. This was the easy part. I managed to get this part working no problem. The biggest issue I had with this, which has caused me to be unable to finish it, was trying to replace the Scanners. Normally, the program works as such:

```
Enter a name for Player 1.
Adam
Enter a name for Player 2.
Bobby
Adam, who will you play as your active?
Available Pokemon in Hand:

1. Name: Chatot || HP: 70
2. Name: Mudkip || HP: 70
3. Name: Arceus || HP: 80
1
Bobby, who will you play as your active?
Available Pokemon in Hand:

1. Name: Pikachu || HP: 60
1
What will you do, Adam?

1. Attack
2. Attach Energy
3. Play Trainer
4. Bench Pokemon from Hand
5. Switch Active with Benched
6. Display Game Info

0. End Turn
```

Where the program waits for user input before moving on to the next bit of game logic. Originally, I was going to try to use flags or game-states to try to have the program wait for input

from an EditText field and using a Button to submit the text. This got sloppy really quick and ended up not working. I tried a multitude of other things, including trying new methods I was unfamiliar with: callbacks. All of my planned methods ended up not working. I want to attribute this to potentially how my code was structured for the first half of the project. I do not think my implementation was bad per say, I just think it was not optimal for this project. I believe if the instructions for the first part of the project were clearer in alluding to this part of the project, I feel as if I would have changed it more to be ready for this kind of project. As something that works as a console app, using scanners as input, I feel it would have taken a lot to rewrite this, more than the allotted time allowed. The farthest I was able to advance this assignment was getting the first message to prompt, but it would not read the text I had given it. It would error out before, showing me that it was unable to pause and wait for input. I had made 3 projects as an attempt to get this work, with about 8-12 hours put into each with no sign of progress, so I have decided to forfeit this part of the project.