

## Project 4

The first step into the development was to see what the playing field looked like, as reading into the functions of the existing classes. Then what I did was figure out a way to run the app as I had trouble doing this in the past. When successful I started to complete the TODO's listed in the code as a way to guide me into creating runnable code, while staying on task, using it as a tip in where I must add code. Once that was done I edited how the android app should look changing to have only 2 digits in the UI rather than the previous 00:00. Once I was able to see that what I was changing was reflecting onto the app I started to work on the functions. First thing I did was delete what seemed to be unnecessary, this was anything that involved with laps as this would be a timer with an alarm decreasing the timerValue instead of resetting and starting over. This is where I faced a challenge within the state of the timer, as I was struggling on figuring out how to connect each file from another. While looking at the DefaultStopWatchStateMachine class I figured that I needed to change the names of the things that I was doing and not doing as I can use it in other classes. So I deleted anything related to laps. While testing I kept receiving errors upon every change which chained onto others. This is because I created an Alarm class and Incrementing class. The reason why I wanted to add these classes is because it is easy to call back to the classes, for example when the alarm is called it will play or stop the audio, when the time adds another second it will add it to the counter and display it. This came to prove more challenging and the more I try to add the more errors and disconnect played out.

The similarities between the extended machine and this project is that both follow the same path as they are both timers, in which both timers have a trigger and an event reaction. The difference is that the project has a set time in which it would stop and you are able to set when it stops between 0-99. Another difference is that the code is able to run the model with the modifications.

I would say that it is more effective to create the code first which would run the basis of what the program goals are. Then once the function is running smoothly since the project runs 2 digits instead of 4 you can remove the other stuff and make it look pretty. This is something I will start doing as the things I would change in this model is the DefaultStopWatchStateMachine and figure out how to run it into other classes while worrying about it affecting if it can run or not. The challenging part was I couldn't see what the issues were as the android app would not run and the terminal would blank at pointing at a class.