Twesha Mitra

https://github.com/tweshamitra/CS1632_Deliverable2

CS 1632- Deliverable 2: Unit Testing CitySim9005

# Introduction

This deliverable taught me a lot about test driven development. When I started this assignment, I took a "cowboy coding approach" and this made it pretty difficult to modularize and test my code. I didn't really plan it before I sat down to code and this was definitely a detriment. Initially, I started out by writing tests before I wrote the code and it was clear to see the benefits of this. It allowed to me to write the bare minimum code that made the test pass. However, it was easy to see that test driven development takes more time than normal. And since I had not come up with a plan, it was taking even longer to test. So, I started to focus more on the program code and less on the testing to try to get the program to do what it needed to do. Then, once I got it to work, I wrote more unit tests. This, in turn, led me to refactoring a lot of my code because as I was testing, I would think of better and more modular ways to perform tasks. Initially, I had some unit tests that were depending on other classes that weren't being tested, which is why I had to refactor a lot of my code. There was unit test/method that I had problems refactoring. This was the test for getCurrentAve. It passes, but I wanted it to not call a method on the object that was returned. I spent a lot of time trying to fix the method it was calling, but I had trouble fixing it. I think this stemmed from the fact that I did not plan well before I started coding.

One thing I found liked about this deliverable was the amount of freedom we were given. Usually, we are told to use certain data structures and algorithms. But, with this environment, we were allowed to implement however we want as long as the functional requirements were satisfied. Additionally, I am not used to writing numerous tests for every method I write, so this was definitely a challenge.

Going forward, I will definitely create and follow a plan instead of just sitting down and coding. I can see that it's important to do at least a little bit of design work up front and follow that as you code. This makes testing and coding much easier in the long run.