

Testing Discussion:

So as the way the program worked was using user input, I obviously couldn't write tests using user input. I tried using a `InputStream` and passing that in which worked. But it wasn't ideal, it would only allow me to pass one command and it meant I had to use my game loop which meant it never quit the test... infinite loop yay.

My solution was to make a new state for my program, 'testing'. This state allowed me to not use my game loop. Problem one solved. Due to the fact my Parser returns nodes that could be executed by my controller to alter my Model, it meant I could create different types of nodes and execute them to alter my game state. i.e `createNode` and `moveNode` etc. For each test, I added to this queue and ran the test which popped off all the nodes on the queue and executed them, altering my game state. Thus, no user input but workable testing.

I tested as I was going along, some bugs I just fixed and didn't add tests for because they were trivial, some I did. I could have done more tests but I spent a decent amount of time debugging alongside that I feel that the gameplay should be correct, or very accurate. I was struggling to create bugs.