- PRESENTATION IS DUE NEXT TUESDAY
- Cycle two is all about testing
    - Tests should be based around the target in the design document
        - Nor based around the code
    - If tests and code are written from design decisions then that’s good
- Documentation should have planned tests for each feature
    - This should be shown in the presentation
- Test documentation must show that it refers to a set of files with the error code
    - Showing the test.txt that the Vm is reading from
    - Show the expected value in the design document
        - Instead of giving a specific error a better idea would be just to give the line where the error is
            - Telling what the error is would be hard to test and write
        - Might be nice to print out what the error word was in addition to the fact that there was an error
    - Out current test doc looks like it is left up the discretion of the tester and that isn’t good
        - Capturing the output of the program would be a good idea because then you don’t have to look at each test manually
    - We should make some way to catch exceptions
- Put unexpected things learned in the tests into the presentation
    - Show in the presentation a side by side code and document breakdown.
- Make sure to have something to show as a demo
    - If we dont have anything new to visually demo make sure to show the amount of changes in backend code
    - Side by side comparison in git possibly.
- JUnit is only good for unit tests
    - Great for a class with lots of data modification methods
    - Not so good for something with command-line arguments and visual looks
        - This could be accomplished with a batch script
- Tons of tools for GUI tests
    - He shared a wiki link
- Unit tests can be very powerful

Possible test to show

Get video of track breaking and then if fixed show the track not breaking