*Exercise 1: Corrent (Game.cpp)*

***Explained***

* ***This exercise had us debug and correct a simple “cout” spelling error. From this I realized that the difference between regression testing and unit testing.*** 
  + ***Regression Testing –*** Regression tests are designed to check the functionality of an entire system to ensure that unintended side effects do not occur.
  + This week’s regression test file seeded a random number generator with 50 different seeds. The output from each of these respective seeds was then printed to a .txt file (golden\_data).
  + Now if the developer (me) were to make changes to the code, they would then be able to run a regression test. The test would then produce 50 additional .txt files using the same seeds as before but with the newly augmented code.
  + Once completed, these 2 respective sets of 50 .txt files are compared for discrepancies. If all 100 of the .txt files are exactly the same, everything is good (the regression tests is considered to have passed)! If anything is different amongst these 100 files (meaning a code change essentially effected the functionality/output of the program), this is considered to be a failed regression test.
  + ***Unit Testing –*** These are tests that are usually done to a specific module or piece of code. These tend to focus on ensuring that the outcome works as intended.
* This first part was pretty straightforward as there was only 1 miniscule spelling error that needed to be corrected. This process was as follows:
  1. Correct the spelling error in the “**src/game\_lib/Game.cpp**” file.
  2. Save the changes in Vim.
  3. Navigate to the build folder and re-build the game using: “**cmake –build .**”
  4. Staying in the build folder, run the trivia file using “./**build/src/game\_runner/trivia”** to ensure that there are no errors.
  5. Still staying in the build folder we must rewrite our regression test cases using the new code output by calling: **“./test/regression/create\_golden\_data.pl**”
  6. Now with regression test cases that match our updated code, we can run the test to see if it is working using: **“./test/regression/test\_\_regression**”
     + Done in the above order, line 5 works as intended. What is unclear to me is why if you “cd” into the “test/regression” folder and then try to manually run ./test\_\_regression”, it always fails…

Graphical user interface, text

Description automatically generated