

## Debugging Dominion Nawwaf Almutairi

### First impression:

When I first started the class I was excited to learn a lot about all different kind of debugging. But after I saw Dominion code I thought it'll be impossible to debug such a thing. It took me a lot of time to understand the game and the code before I was able to do any debugging.

The first thing I did in the class was learning the game. I spent a lot of time trying to learn the game. I played it online with random people. I played it with my friends who already now the game. Finally, I read about it's rules online and by the end of week 2 I was able to play the game with no problems and I felt more comfortable reading the code.

The second thing I did was familiarize my self with the code. I start reading the code and try to make sense of all the mess. The first assignment helped me to get a good start with that, because it encouraged me to understand the code and get comfortable with it. after spending time reading the code I found that dominion.c is the main part of the code and it has all the main functionality of the game. Also, playdom had another main part of the code and both of files contain over 90% of the overall code structure of the game which is the first bug that I found with the program. So, after knowing both files and understanding them I decided to start solving the first assignment by refactoring some of the cards into function. The game logic is written in C. While C is a fast and simple language, I feel that the logic engine for the game would be more understandable and less error- prone in an object-oriented and memory-managed language. Removing the risk of manual memory management and having more expressive syntax for the custom data structures required by games would make the game logic much easier to visually debug.

The first assignment of the class wasn't hard. It didn't take me time to solve it but it was a great introductory assignment. The second assignment was a really interesting one for me. I always wanted to know how to write a unit test in c and this assignment helped me to achieve this goal. Also, it was a great experience learning how to use gcov. Through my computer science career I never heard about code coverage and that's because I never done any internships so learning how my test can cover my code is a really beneficial thing for me to know especially in the professional career. I opted for using the built in assert() function instead of implementing my own custom method. In hindsight, this was a mistake. Although my method resulted in slightly more readable and predictable tests, it also meant that the tests would terminate as soon as the first assertion was triggered. This mean that each test could only uncover one error before quitting, which was sometimes a pain.

Random test is another new concept of testing that I never done before too. For assignment three I had to write a random test generator for Dominion cards and it was an interesting assignment to do. It took me a lot of time to this assignment. I struggled for a while to write a

correct code that can actually do what I needed for the random test. The two functions I tested were the adventurer and village cards. Since there wasn't much documentation in the codebase, I looked into these functions and how they were used to ensure I tested the full range of their use. Next I wrote random generators that created sets of output within these ranges. Finally, it was time to actually test. The existing barebone tests only ensured that the program didn't crash with random inputs. I wanted to check that the program ran *correctly* with random inputs and, therefore, made various checks to the game's state to ensure it was following the expected behavior. In the end, I resulted with 44% code coverage.

In conclusion, this class taught me a lot of new and powerful skills to debug codes and test it. This knowledge will help me to be a better coder in the future and improve my skills in the professional career. I look forward to apply all theses new knowledge in a real life work that will help the humanity and make sure to minimize any mistakes that can hurt other people.