Cole Rutkowski
26 April 2024
CSCI 1300
Project 2 Report

To prepare for this project, I read through all the recitation study guides and prior homeworks to make sure I had a strong grasp of the concepts we learned. I also read through the project guide several times, breaking up each section into subsections to further organize the page. I also made several preliminary flowcharts to guide the order of what I would work on and the order of each looping function or component of the program. I started the project as early as I could, and by the time the code skeleton was due I had more than what was required. This allowed me to get ahead in the early stages of the project.

For the code skeleton, I used the previous homework as a guide, but completely revamped it. I started by making my structs for potions and weapons, then worked on my classes. First, I made getters and setters for all my private variables. Next, I worked on functions that would take in and print out information for those classes. After that, I focussed on making the map work. I did that quite successfully, getting my map to randomly generate with the correct amount of tiles, as well as keeping track of where each player is.

One of the biggest components I could improve upon is the structure and organization of my code. I began making header files for different aspects of the game because I didn't want all of my code to be in the game.cpp file. However, further along the process the purpose of my new files merged with other existing files and the code became clustered, when I may have been better off using less files altogether. I do wish I had more time to work on the project, but to really improve my code I would have needed a huge overhaul of my organization and may have needed to fully recode the project, using what I already made as an outline, to make something I was really happy with. I should mention that I spent approximately 35-40 hours working on the project, and I wasn't keen on doing muchmore. However, I go into projects wanting to make it perfect, which isn't entirely attainable given my coding experience and the ceiling of potential in the project, so otherwise I am quite content and proud of what I made.

I was quick out of the blocks on this project, and experienced no hiccups on the preliminary entity, game, and map header and cpp files. Next, I worked on player menus, dropping and swapping inventory, and potions. I spent a lot of time on these and put more focus than necessary on edge cases. I am very proud of those functions and how I made it readable for the user. It wasn't until after these were done that I hit serious issues. One was that I couldn't pass the map to my function that updates the players position after a dice roll, which meant that I couldn't (given my timeframe) make it impossible to skip all the islands. This also hindered one of the calamities that make a player go back 3 tiles, however I could get it to work when landing on the red tiles. This was a result of uninformed planning. I wouldn't say it was bad planning, just that some key components of the movement were single sentences spread throughout the directions, so I didn't have a complete understanding of exactly how movement should work when I started coding. I also made the mistake of stacking every file into the next with includes and only compiling the driver file, instead of all my cpp files. This didn't cause any direct issues, but I did learn that it isn't the most efficient way of coding. Also, at some point I needed to share information between two files, specifically map and game, but couldn't. This is the reason my player moving back tiles didn't work in my calamity function. Nearing the end of the project, I struggled with getting a player's stats to update and stay updated outside of the function. This

was because I was passing an Entity player to the function and updating the stats insteading of passing my players array. This was resolved with the help of TA Mohammed, who taught me what a reference was and how it worked. This was a huge help, as I didn't know how references worked before, and I was able to resolve the issue. Next time, however, I would design my functions to take in the player array instead of just the player entity. Other than general headaches and minor issues that were resolved with some debugging and TA help, the last big fix I had to my code had to do with the riddles file. Eventually, I learned that I couldn't just use the downloaded file for riddles, as it had a carriage return in it. The issue I had was with comparing a user's input for the riddle answer with the actual answer. When I tried to stoi the riddles that had integer answers, the comparison worked, but I was still having issues when comparing strings (when the user input was the correct answer). Fortunately, this was a quick fix and my last real setback. In the end, I had a very successful project with good user formatting and detailed edge case handling for my item shop and player selection. There is a lot that I would improve if I had much more time to work on it, but I am happy with the result of my game and I am glad my Odyssey is finally coming to an end.