Project 3 Report
Case Chrisbacher
CSCI 1300

For the third and final project Phillip Gerhardstein and I created a game in which you were a pirate on Treasure Island attempting to fight through enemies to get the treasure at the end. You faced enemies such as skeletons, spiders, pirates, and the final boss: Black Beard. With options such as fight, run, and hide (among others), you attempted to pass your enemies however you want, generally playing to your chosen characters strengths, and maybe even the current enemies' weaknesses, that is unless they have no weaknesses (AKA Black Beard.) We used a pirates, enemies, and treasureisland class in order to complete this game.

Other than the homework and projects leading up to the final project, we honestly didn't do a whole lot of preparing for this project. One thing Phillip and I did do was a couple day long brainstorm. We knew we wanted to do some sort of story where there were obstacles along the way, but we didn't know exactly how we wanted to present this. We knew that our coding skill would be most effective in a game type such as Oregon Trail. Our first idea was a bank robbery game but having police as the bad guys seemed like a questionable decision. This idea was easily morphed into a treasure hunting game, changing the robbers into pirates and police into various enemies.

Our code skeleton was simply the earliest stage of our code. Phillip and I did a zoom call and essentially set up and talked about how we wanted the project to run and how we were going to hit all of the requirements. The code skeleton in which we used for our meeting consisted of all of the .h files, outlines of the .cpp files, and the outline of the main function. We used our code skeleton by simply filling out the blank spots after the meeting since our code skeleton was simply the early stages of our code. Writing out this code skeleton was very useful, it made it so our project was well formatted and easier to read.

One thing that would have made our project more efficient is if either of us understood how to use arrays of type class which we created. Although we knew how to create it, we spent a lot of unneeded time on trying to call it, but more on that later. Also, being able to work in person with Phillip would have made this run a lot smoother. Sending code back and forth over email was effective, but every time either of us put work into the project, the other needed to look through the code for a while to understand what the other had done. Working with a partner definitely decreased the workload, but everyone codes different so it was challenging to understand everything that he wrote immediately. One thing that we could have done better is make the game a little harder or added difficulty settings. The game is very easy to beat, even if you pick the weakest character "Smallfry." Your character will not necessarily be able to run or hide every single time, but odds are your character will not die before your character successfully runs or hides (or kills the enemy with fight.) So, a way to make it better is just to make the enemies do more damage to your character and balance out the game a little bit.

One night of coding, I went down a path only to turn back later on. I spent many hours trying to understand how to access arrays of a type class which we created. The first attempt was the enemy array. I spent a lot of time trying to make this array work and honestly, I still am unsure why it never worked. Essentially what I was attempting to do was create an array of type Enemies (which I successfully did), call, and subsequently output different values from inside the array. After many hours of testing and trying slight variations of the same exact thing I gave up on it. After getting rid of the Enemies array, we were missing a requirement for the project so I thought it would be a good idea to create a difficulty array. I wanted this array to be of size three and hold different multipliers to make the enemies more powerful. I thought this was a glorious idea and decided to commit the rest of my night (and a lot of the early morning) to it and failed miserably. At that point I was feeling pretty defeated, so I went back to project two to see

how we used arrays of different classes. In that project we used arrays solely to output (cout) information. With this newfound knowledge, I deleted all of my code I had written for difficulty and remade the Enemies array and set it to simply fill up and output (cout) that information. I had this up and running in about twenty minutes after draining hours and hours into trying to attempt to call the array using a variety of complicated functions. These twenty minutes of coding fulfilled all project requirements and it all could have been avoided if I knew how class arrays worked.

Overall, I am very proud of how our project turned out. Each of us let our roommates play and try their hardest to try and break it. Although they did succeed in the earlier stages, our final project is unbreakable from my knowledge. It runs very well and is actually somewhat fun to play. Finally, coding with a partner was a good experience and exposed me to different things in which I generally don't use in my code.