Logan Mann
CS1300
12/1/18

Oregon Trail Reflection

All in all, I feel I prepared very well for this project. I started early by typing up the organizational structure of my class objects and how the various data members required for the game would be grouped into the various classes. I then went straight into writing descriptions of the various functions I would need throughout the game, and completed my code skeleton. After that, I essentially went right into implementing my functions.

Writing my code skeleton was fairly useful because it forced me to think about the various tasks that would need to be accomplished throughout the game, and allowed me to group my data members into classes that made sense. It made actually coding the game quite straightforward because I had the basic functionality already laid out.

I think there are some redundancies in my code that could have been improved on, and some repetitive tasks could likely have been written into functions to make my code more readible, but overall I think I did a good job with commenting and overall, my code is pretty straightforward.

In terms of arriving at any false starts or problems when beginning to write the project, I had a clear vision from the start that it was going to be broken up into key game actions, like the shop, resting, raids, etc. and that these functions would update all the data members like food, money, etc. and the driver function would simply be a while loop that ran until a win or loss condition was reached, and simply asked the player what they wanted to do on any given turn, and called the appropriate action functions as necessary. This worked well for me without any significant problems or false starts, but there did come times where I had to implement a member

function for a given class because I didn't initially consider that I would need to access/update data members in a certain way, such as my functions which returned boolean values based on supply/mileage/health to show whether a win or loss condition had been reached. Implementing new functions as I needed them definitely made my code easier to read, because I didn't have random blocks of code thrown in when I needed to do something, I simply wrote a member function with a clear name and then called that function where needed. My driver function ended up being very clean looking this way. Overall, I'm proud of what I accomplished on this project.