

Aanvi Guliani

CSCI 1300

Dr. Fleming

2 December 2018

Project Report- Oregon Trail

For our Final Project for CS 1300, we were tasked with creating a simplified version of the 1971 computer game Oregon Trail. I had never heard of this game before, and I knew that creating it would be a challenge, but not impossible.

Because I'd never played Oregon Trail before, I knew I couldn't understand its game mechanics without at least playing the original game a few times. That's why the first thing I did upon receiving this project was play the game three or four times a night. I began to understand what elements were dynamic, what were static, and how to go about possibly implementing them in C++.

To start off, I created a code skeleton for the project, creating .h files for classes and its respective .cpp files, and wrote the getter and setter methods in the .cpp files. I created multiple classes, including Hunting, Miles, Misfortune, Player, Supplies, Milestone, and Raider Attack. While the majority of my classes didn't end up being the same as the code skeleton, one of the skeleton classes wasn't used in the end, and one class wasn't in the skeleton to begin with, it was helpful to at least get started in thinking about what functions I would end up needing for different aspects of the game. For example, my Misfortune class in the skeleton looked completely different from what it actually was in the final deliverable. In the skeleton, I was treating it as if every misfortune could occur all at once, when in reality you choose one random misfortune, and if it's a sickness, it occurs to one player. It was things like this that made the difference between my code skeleton and my final project classes. Regardless of the end differences, I still found it useful to create a code skeleton and wouldn't change the fact that I did.

One of the things that I could've improved on was the general organization of my code. While all of it works (to my knowledge), some of my code was repetitive and messy in terms of placement, and had I had more time, I would probably work on fixing that first. Another thing I

would work on is how my code is presented. If I want to improve on this in the future, I would add a user interface and make it more similar to the original game. Of course, this wasn't super feasible to do in the matter of just a few weeks, but if I want to improve upon my code in the future, I would look more into creating user interfaces and apply the learned skills to this project. Additionally, I would try to make sure that my code looks professional. I'm 100% sure that there are random capitalizations, extra lines between things that are being printed, etc. To improve this, I would just clean that up.

My code skeleton definitely set me off on a few false starts. One of the reasons this happened is simply because I didn't completely understand the game mechanics from the get go. Rather, I learned more and more about how the game worked as I was working on the project. Because of this, I was always starting code and scratching it in the end because I was going about creating the game incorrectly. As I mentioned earlier, my code skeleton for the Misfortune class was one of my false starts. I figured out that I was implementing the misfortunes inaccurately just a few days before the project was due. Additionally, I had a few false starts because of the fact that I wasn't completely sure how to organize my code, and how to update all of the supplies after each turn component. I fixed this by passing Supplies objects, Miles objects, and Players objects for classes like Misfortune, Hunting, and Milestones.

Overall, this project was tough, but the thought of recreating such an iconic game made it more enjoyable for me. This was one of the first few computer games ever invented, and it's cool to know what goes into it.