Marcus Padmore
Final Project Report
TA: Arcadia Zhang

This project required more time and C++ knowledge than any other project I have worked on however, with proper preparation it was not too much. To prepare for the completed projected I had to organize my time and break the project into manageable chunks instead of trying to finish the whole thing in one go. First I created the pseudocode for the main driver program so that I could get an idea of the classes and functions I would need to fulfil all the requirements. After that I tried to figure out what classes would be best. I came up with a class for date, players, and inventory. Before making the final decision to use these classes I asked other students what classes they used so that I could determine whether I had chosen the right classes or perhaps there was a class that was more efficient. In the end I decided I liked my classes and would stick with them. I then created my classes and their member functions as the next step in my project.

After creating my classes and pseudo code I felt my project was going great. I felt the code skeleton really helped me organize my project and take it step by step because I knew how I wanted my code to run and the order everything should run. Next I started from the main function in my projects driver file and started to write actual code. I figured it would be easiest to try and get the main function coded so that I could test my code periodically. About ¼ of the way through I realized I would have to write some of the other functions before I could really test the main function. The main function ended up being the last function I completed. I used my skeleton code to write

and test each function separately. In a separate file I wrote the code for a function and tested it to make sure it did what it should and then copied it into my project. The only obstacle was that I had to create variables in order to test it and then change them later. The code skeleton was extremely helpful in completing this project.

When I finally finished my code all of my functions worked but the order of them was a little messed up and I had to tweak a lot of things to get it working properly. Eventually I got all the bugs worked out and the code ran smoothly doing what it was intended too. After reflecting I realize while my project is good there are somethings that could make it much better. For instance I didn't have much time to optimize my code after it was finished. I felt as if I needed another week dedicated to improving my code and making it more efficient. We weren't given much time for the project so when doing the project I was more focused on things working than in them being efficient. There are some parts of my code with likely twice the if and else statements that were needed or where I could've used a loop instead of extra code. I think the way I approached the project was the quickest and most efficient way but that more time was required to really make a project that is above and beyond.

Additional Paragraph

I had some false “false starts” while working on this project. There were times when I thought I would need to go backwards and recode some things but I ended up working around those. An example of this is when I couldn't figure out how to deal with milestones. I initially thought I would have to create a class to create milestone objects but instead I created arrays to hold the milestones information and simply searched through them when needed. I had plenty of moments like this and other then those my project went surprisingly smoothly. I think it went so smoothly mainly because of my code skeleton. Writing skeleton code allowed my to look at my code before writing it and get a good idea of whether things would work or not. Any false start I would of had was prevented by writing skeleton code before the actual code.