CSCI 1300 Project 3
Ryan Taylor
Recitation Tuesday @ 2:00

I felt very prepared going into this project. With the vast amount of different types of functions we've written between homeworks, recitation work, practicums, and other projects, I've amassed quite the collection of examples to most of what we've learned in this class thus far. The one thing that scared me going in was the immensity of this project. I told many other people in class that nothing in the project was explicitly difficult, there was simply alot to do and alot of bases to cover. Figuring out specifically for myself what organization and specific header files made sense in my head and would make working easy was quite the challenge in and of itself.

I wrote a code skeleton out on paper before I started writing any code C9. I started by writing out every single variable I would need and categorizing them into different possible class names such as players, inventory, and miscellaneous functions. I then took those and started looking more broadly and thinking what classes I would have to include in my cpp files and how to pass things between all my functions. Only after this did I start writing out my header files in C9 in preparation for my initial project meeting. This skeleton was immensely helpful throughout my entire project, knowing where all my different function were saved specifically, and knowing what headers each implementation file needed. Although time consuming, for any large program such as this project I would take the time to write a skeleton for how beneficial it was throughout the entire process.

I am well aware that at this point in my computer science career, my code is extremely messy. I often had the problem of creating new unnecessary variables for only one-time-uses, and would forget which type each of my variables was, which resulted in alot of unnecessary errors when it came to compiling the entire game. I also would have been much happier had I started the project even earlier than I did. I worked some over Fall Break, however looking back there's much more I could've done and it would've made the few days and weeks leading up to the due date much less stressful.

While I had some definite road blocks in writing my implementation files, I would not say I had any major false starts or wrong turns. I attribute this greatly to my creation of written out and physical code skeleton. Being able to visualize the organization of all my classes made it very easy to make changes quickly to a specific set of variables. For example, I created the “gameOver()” function in my miscellaneous function very early on, and whenever I would get to another game-ending part in the driver function, it was very simple to add another switch case and keep working. The use of switch cases also makes adding or removing outcomes very easy. Throughout the creation of this game I went back to switch and nested switch cases to change the cases and switch makes that very painless.