Joe Tingstad CS 172 | 6/27/2018

For my project, I chose to do a word scramble game, which calculates the amount of time it takes for a user (or users) to unscramble ten words. To start, I created a class “Player” that made up almost the entire functionality of my project. The player class contained functions that carried out all of the essential parts of the project, from filling a vector with a list of 6000 of the most commonly used words in the English language, to calculating which player won based on their score. As far as working outside of class, I had to do a bit of research into how to calculate elapsed time, because that was not something that I was familiar with. Although I ended up finding it to be fairly simple (just a matter of creating two variables and calculating the difference in time between the two), this was something that I had never worked with before and so was something that I had to do a bit of research on. Something that surprised me the most about this individual project was the amount of thought it took prior to writing any code. My previous experience with projects and homework has been that the qualifications and the requirements for the project are laid out in a very simple and direct fashion, and you simply have to figure out how to do what the problem is telling you to do. With this project, however, I found that the hardest and most time-consuming part was not the implementation of the code, but rather the thought that went into planning a game that I could code using my limited knowledge of the C++ language, as well as utilizing the concepts we learned in class. Instead of having the requirements on the page of a textbook, I was setting my own requirements, which I’m sure is much more realistic compared to “real-world” coding but was a new and challenging experience for me. If I were to do this project over, I think I would have taken more time to really think through the requirements and the basic skeleton of the code before I began to try to put code into visual studio. I spent a lot of my time really thinking about “how do I even begin to make the program do what I want it to do” rather than “how do I code it to do what I need it to do?” If I were to do a version 2 of my project, I would actually carry out the idea I had that I would like to check it against all real words, rather than just the one that the generator spit out. Additionally, I would like to have been able to give hints, such as un-scrambling the first couple letters or something like that, as long as the number of times the user could ask for a hint would be limited. Overall, I was fairly surprised at how easily I was able to put the code into the program once I had gotten through the hours of trying to plan and make it into something I knew how to code, and I feel encouraged that my coding abilities were more limited by my lack of ideas than by my knowledge of the syntax and interworking’s of the language.