Katie Ochoa

Individual Summary

Our final was difficult in the fact that we had a well thought out plan, however, we did not know how to properly translate that into code language. First, I contributed by writing a noted version of our plan. Bullet pointed the requirements and how we would use that in our code. Rachel then proceeded to use that to write our proposal and we worked together to form our classes. From this, I researched different ways to write a battling game in code. We initially started with the idea that the user would pick three characters from a list of eight, and we would have a function to bring each character into battle after the first one dies. We ran into a problem because we wanted to do a stack but when we started writing our code we did not use it, we found it easier to let the user select one character. From this decision we also made the users health larger, they stood a better chance against the boss. As for the boss character we set a randomization to declare the bosses name, as well as impact of health his attacks has on the user. One issue we ran into here was we did not set the random numbers to change as the turns went along, so the same numbers were being used the whole round. We each kept messing around with the code and altering it to fix our vision. We ended up both sitting down to run the loops for the actual battle and the alternating turns. We got it to successfully work, however, we did not put our get’s or set’s from our classes in the code. We had to go back and reset all our definitions so our classes were represented properly. After the loop was set, all we had left was to code the input and output. We did struggle here because our first code did successfully open a file and write our saved number. However, after inserting if statements to help add up the total, the file would open itself, read the zero, then close it before we could write anything to it. It took us a while to figure out why it was not reading out anything, but after some help we were able to change our code to open and close it, but also open and close it after a second time with what we wanted to write to it. After this hurdle our code was running properly and meeting all requirements. From this final I learned a lot about implementing code and how it is very important to write out all you want to do before coding, or else you would have to heavily alter your ideas and take things out—or add—depending on what needs to be done.