COMPSCI 381-01 – JavaScript and DHTML – Spring 2025

Evolutionary Leap Project Review

Andy Brandt (BrandtA13@uww.edu)

Overall, I am impressed with my product. I had a lot of difficulty selecting a topic, as I felt that I didn't know enough about web design and JavaScript to create a project I would be satisfied with. I was under the impression that this project should be useful to the public, and I had a lot of difficulty finding a topic that interested me. It was about three-quarters of the way through the semester when I realized that I could make a project for myself. After that, I quickly decided on creating a Leap manager for *Plants vs. Zombies Heroes*, a card game I often play on my phone. I've been playing this game for the past 7 years, but that doesn't guarantee that I know the different card pools when Leaping my zombies. I knew that there was only two 8-cost cards, and only one 9 and 10-cost card, but I was unsure on what is in the 7-cost pools and below.

Once the project began, I realized a large portion of the time spent on the project would be spent on creating the database. I ended up importing all 1000+ cards into my program, but only about 415 cards would be used, as only certain types of cards can be Leaped. I was fine with this, because this is project I would like to come back to in the future, adding new features that will assist me in winning games.

Once this was completed, much of the program wrote itself. The biggest problem I ran into was creating the flex boxes to show the query results. We only talked about formatting a website for one or two lectures, so I didn't have much to go off of from class, and I spent a lot of time deciding on what should be passed to the two functions.

Another difficulty I had was when turning the team radio into selecting which list I query from. I wanted to have the value of the radio's be the two arrays, but I couldn't get it to work. I'm sure with more time I can get this working and cut back on many variables that are used only once. I was also stumped when creating the 'effective check boxes' portion. I developed this portion last, and I believe it's quite clear by the way it is structured. I'm sure there is a more efficient way to filter with effectiveStrength/effectiveHealth instead of strength/health if a checkbox is checked, but simply overwriting the strength variable meant I didn't have to change anything else in the program. Additionally, without the else statement resetting the strength/health values, they increased strength/health would persist between queries. I'm not sure why this is, as I am editing a copy of the array at this point in the program, but I can assume that JavaScript crates pointers to copied objects in arrays, and that is why the changed statistics persist.