After creating and finishing my craigslist redesign midterm, I began to understand the thought process and amount of coding that was required for future web projects. In my midterm project, I aimed for a minimalist approach. I did not want to clutter the page with unnecessary text and unnecessary images and colors. Thus, in my final, I aimed to have an interesting background, with the user interacting with a box in the middle of the screen.

While the midterm did not require coding, the final did. Designing the layout (HTML and CSS) was not particularly difficult. After much coding during the semester, it was simply a matter of translating my sketches and wireframes into tags and selectors essentially. The main difficulty was translating my flowchart into code. This meant mastering JavaScript (jQuery) to make the page truly interactive. JavaScript was not an object-oriented language like C++. There were no objects in the sense of classes. However, there were branching statements, arrays, methods/functions, etc… like any other scripting language. Mastering these functions methods required a lot of research on the internet and experimenting.

Due to the short amount of time to work on the final and other obligations (work and other classes), it was necessary to revise my original complicated plan. In my original idea, I wanted to include puzzles. This would mean either actually coding the puzzle or incorporating a jQuery vendor into my code to do the puzzle for me. However, this took up too much time and the end result was a mess. I could not find a way to set a time limit for the user to finish these puzzles, so this plan was scrapped. However, this meant focusing exclusively on my riddle section. In my riddle section, the user gets five tries to solve each riddle and after solving all the riddles, they get a random picture. However, I had to introduce session cookies to keep track of the user's riddles since the page would refresh. Cookies were particularly difficult to master since they did not work on non-live sites. The sites had to be actually hosted on the web for the cookies to actually work.

The site was not as bad as I expected it to be. Although there may be some bugs in the website, for the amount of time allotted, it was good enough. In real-life, products will have bugs, but these are resolved later through patches or updates. My main concern was the fact that I had to resort to using a JavaScript prompt to ask the user for the answer to the riddle, instead of making them type in to the input field. It was difficult because of cookie and on click issues. Another issue was that sometimes the incorrect gif would pop up due to cookie issues again regarding answer histories. The design of the website was satisfactory. However, one design I wished to emulate was Stefon's quiz website since it took you to different pages, instead of leaving you on one page. Perhaps it could have solved the cookie problem, but it may be too late to tell.