Brady Esplin

CS133J Online

**Final Project Diary and Remarks**

KEY:

**Date and Time (24Hr Format) Time Elapsed (HH:MM:SS Format)**

Activity

**5/17/2020 1040-1141 01:01:00**

Write Final Project Outline, Submitted to Moodle

**5/21/2020 1307-1840 05:33:03**

Research into games to code, followed tutorial for a “Breakout” style game. I was able to code the game and grasp the concepts but realized that we would need at least another term of instruction before being ready to make a game on our own. The current level of training was inadequate to prepare us for a task of this magnitude. Played/inspected around 12 “JavaScript games” found online for inspiration. Looked at the code making the games run by hitting Ctrl + Shift + J to inspect the backend code in Google Chrome. All of the games, even the simplest graphically were far beyond my comprehension and coding ability. Found tutorial for “Memory style” card game, looking at code, it is still beyond my ability and training bit I believe I might be able to code a game like this.

Researched design solutions for accessibility. Found these infographics:

<https://accessibility.blog.gov.uk/2016/09/02/dos-and-donts-on-designing-for-accessibility/>

Found font developed by the Royal National Institute for the Blind called Tiresias, named for a blind figure from Greek Mythology. Font was developed for persons with little sight and has been helpful for those with dyslexia. Downloaded the .ttf files and converted them to .woff and .woff2 file types using:

<https://www.fontsquirrel.com>

Picked colors for my games based on the recommendations from the above accessibility guides. Light blue, shades of light grey and black, for contrast and to be friendly to those with anxiety, and those on the autism spectrum. Searched for around 40 minutes to find free pictures to use for matching. Was finally able to find a set of emojis that I could use for educational purposes without having to buy a license or membership. They can be found at:

<https://www.joypixels.com/licenses/free>

The next several hours were spent puzzling through different web pages finding out how to make my game perform as desired, including the HTML, CSS, and JavaScript code necessary to even start making the game.

At 6:40pm, decided that I was not making any more progress due to burnout and went to bed.

**5/22/2020 0917-2007 10:50:00**

Majority of this time was spent in frustration trying to figure out the code necessary to make the game look and work as intended. Lots of help from:

<https://stackoverflow.com/>

<https://www.w3schools.com/>

<https://www.geeksforgeeks.org/>

<https://www.javascript.com/>

and random tutorials on [www.youtube.com](http://www.youtube.com)

Major hurdles included: getting the cards to animate properly when flipping over, random bugs in code, and I was stumped for around 45 minutes when I had a function indented wrong.

At 8:07pm, realized I was staring blankly at my screen and decided to wrap up and go to bed.

**5/23/2020 0851-1857 10:06:00**

Got game working as intended, ironed out minor issues with code, inserted graphics onto the front and back of cards. Got animations working as intended with help from Web Dev Simplified on YouTube. Sorted out how to have the cards “shuffle” with tutorial help from PORTEXE on YouTube, algorithm from Wikipedia:

<https://en.wikipedia.org/wiki/Fisher-Yates_shuffle>

Had a bug on my code somewhere went through line by line through every one of the 509 lines of code. Found a missing < at the end of my HTML file that was causing the entire game to break. It took me a literal 2 hours + to find and fix it.

Project rough draft submitted at 11:41pm May 24th on Moodle.

**1/7/2020 1236-1400 01:23:48**

Researching ways to make the game more accessible. With prior planning, I could have written my code in a way that would make altering it easier. At this point to make the game playable with a screen reader and keyboard would require me to start from literal line 1 and write all code over again. To make the page usable with ARIA, I used correct html tags, but doing anything else beyond that was not recommended. I would have liked to have found a way to disable the timer as to be accessible to those with anxiety, but too many of the game functions require the timer to be present and running. As of right now I would again have to start from scratch and rewrite every line of code. As for increasing and decreasing difficulty levels, again the way that my code was written does not permit an adjustment. I would be required to rewrite all code from my starting point onwards. The game will need to be turned in in its current form.

**Lesson Learned:** Coding for accessibility and flexibility is required from the planning stage forward and must be an integral part of your code. The coding knowledge to plan, design, build, and implement the code necessary for these ideals is far beyond my capabilities at this point.

Writing up the diary for turning in with final. Pruning the development diary for relevance, correcting spelling mistakes, properly citing sources of software and information, expounding on points as needed.

Adding instructions to game to meet grading rubric requirements.

Checking code comments for spelling, legibility, correctness.

Last minute check for completeness and compatibility with grading rubric. Zipping and submitting file.

**REMARKS:**

I learned during this project that while I enjoy playing video games, I absolutely despise having to make them. The coding ability required to make this game was far beyond what I currently know, and I only got through this assignment by the kindness of the coding gods, w3schools, and stackoverflow. I believe at least another full term of instruction if not another year of instruction would be necessary for me to possess the knowledge to make a game on my own without the help of the aforementioned websites and deities.

That being said, it was a valuable learning experience, teaching me that to design for accessibility must be part of the process through all stages of the coding. You must keep in mind the need to have dynamic code and build accordingly.