

Project 2 - Documentation

Overview

The purpose of project2 was to create a drag-and-drop app written in vanilla JavaScript. The main aspects were as followed:

- Research: The majority of the project was learning what was needed to finish the project that was not already given to us.
- DOM manipulation/Dynamic HTML: The majority of the project was not static. Many moving parts had to be written dynamically using JavaScript.

Constraints

- Browser support is latest Chrome or Firefox.
- The use of frameworks or libraries such as jQuery or Angular were not allowed to be used.
- JSON being loaded in was to be loaded only once, and not reloaded on successive playthroughs.

Extra Points

- Having 5 rounds to the quiz to use all terms in memory used.

Background

I have 4 years of programming experience coming from an educational background. I have no prior experience to JavaScript, HTML or CSS. Although no experience in these fields, my experience in other languages allowed me to research and finish this project on time.

Considering the amount of time I had for this project, I completed what I felt was most important. A list from 1...X with shows what I felt to be most important, with 1 being the highest.

1. HTML/CSS: The HTML page was built to build the rest of the project on. Without this first page, there are no elements to be manipulated during the project.
2. Play Button: A simple press of the play button displays the quiz.
3. Drag-and-Drop: Making sure the elements could move and be dropped in the answer box.
4. AJAX: Having the items from memory load into the quiz.
5. Correct Pairing: Checking to see if the correct term was matched with the correct definition, and outputting that score to the screen once the game ended.
6. Undo: The ability to go back one move, and make sure one could only go back to before the first move.
7. Timer: Counting the hours:seconds:minutes of the time played.
8. Play Again: Implementing play again without refreshing the page/loading the ajax call a second time.
9. Next round: Adding up to 5 rounds using all the terms loaded from JSON.
10. Keep timer running throughout all rounds.

I tackled the project as per the list above. These were my observations on what I found to be most important and such reflected in my code. Many hours were given into each file, with the most in app.js. Estimation of around 20 hours was put into this file, while only 0-2 hours in the other files.

Diagram

Quiz Game 0.2

You got 3 correct answers in round 1.

00:00:23

Next Round



This is a simple screen of finishing round 1. The terms in the green and red boxed were originally on the left where the words "Empty" are displayed. The terms were able to be drug to the right and into a new box. Once the game was ended, the boxes turned either green for correct, or red for incorrect. The undo button is located below the score, but is not shown when the game is over or no moves have been made.

Potential Improvement

As for delivering a working project on time, the main functionality works. That being said, the project has much room for improvement given more time.

1. The use of `document.getElementById("").onclick` was used. While this is not best practice, it is the only way I was able to finish the project within the timeframe. Submit or `addEventListener` will be added to the next release.
2. The HTML and CSS scaffolding was provided to us. With more time, the quiz would have been more customizable.
3. Cut down on the use of global variables by passing more function parameters.
4. Unit testing for the functional code portions.
5. Adding the reusable code to the DMJS framework for use in the future, or creating a game framework.
6. Random number generator to choose which items appear on screen could be more optimized. Theoretically could run forever with 25 terms, and as complexity increases, so would load time without optimization.

Conclusion

Building bigger and more complex apps without the use of pre-existing frameworks is tedious. With my lack of knowledge in JavaScript, HTML, CSS, and DOM manipulation, this project took much time and research. Changing button functionality using `onclick`, hiding and showing the quiz along with buttons, and loading a json file with ajax were all new to me. The easiest part of the project was the drag-and-drop functionality, albeit it was already given to us from the professor/w3schools. The hardest part of this project personally was the play again button. My code was not designed well enough to implement new features in at first, and I had to restructure the program for more modularity. The final design of my code allowed for the optional part of the project to be complete, and I have a total count for how many questions you get right/wrong at the end of the quiz. W3schools and `developer.mozilla` were a great help, and documented in the code when used. With all that being said, the project was delivered on time with the basic requirements met. Having a list of goals on where to start and abstracting problems from the complete requirements was instrumental in finishing the project.

John Mills
Jgmills52@gmail.com