Wordle

User Story

* I want a user-friendly interface that allows me to easily navigate the Wordle game.
* I want clear instructions on how to play the game so that I can understand the rules quickly.
* I want to be able to input my guesses for the word in a convenient and intuitive manner.
* I want feedback on each guess to know if I'm on the right track or not.
* I want to see the results of each guess displayed clearly, showing which letters are correct and in the right position.
* I want a clear indication of whether I've won or lost the game after making my final guess.
* I want the option to start a new game easily, so I can continue playing and improving my word-solving skills.

Pseudocode:

Define constants and variables.

* Define the target word for the game
* Define the maximum number of guesses allowed
* Define variables to track guesses made by the user
* Define variables to store the user's input and feedback

Define the app’s state variables, but don’t assign values to them.

* Initialize variables to track the current game state
* Initialize variables to store the user's guesses
* Initialize variables to store feedback for each guess

Select and save elements in variables that need to be accessed in the JavaScript code more than once.

* input field for the user's guesses
* submit button for submitting guesses
* feedback display area to show correct and incorrect guesses
* Select the message display area to show game outcome

Add event listeners - use delegated event listeners to listen to multiple elements with a single listener. Add an event listener to the submit button to process user guesses

* Initialize the target word for the game
* Initialize the maximum number of guesses allowed
* Initialize variables to track user's guesses

Create a word Array

* Randomly select a word from the list for the game.
* Choose a random word from the list of words
* Store the chosen word in the variable for the game word

DOM

* Render the input field for the user's guesses
* Render the submit button
* Render the feedback display area
* Render the message display area

Wait for the user to submit a guess.

* Get the user's input from the input field
* Validate the input to ensure it meets game requirements
* Update the feedback display area with the user's guess

Check the user's guess against the target word.

* Compare the user's guess to the target word
* Provide feedback on the guess
* Update the message display area with the game outcome

Check if the game is over.

* Check if the user has guessed the word correctly
* Check if the maximum number of guesses has been reached

EndGame

* If the game is not over, allow the user to make another guess. Continue providing feedback on the user's guesses
* If the game is over, provide a message to the user and allow them to play again. // Display a message indicating whether the user won or lost.
* Provide an option for the user to play again
* Reset the game state if the user chooses to play again

Top of Form

Bottom of Form