Board game simulator in HTML

This template includes comments guiding students on what to fill in and is structured to be both fun and educational. The game is a simple word guessing game where the player guesses letters to form a word.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Word Guessing Game</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

margin: 50px;

}

#gameBoard {

margin: 20px 0;

}

.hiddenLetter {

display: inline-block;

margin: 0 5px;

padding: 10px;

border-bottom: 2px solid #000;

width: 20px;

text-align: center;

}

</style>

</head>

<body>

<h1>Word Guessing Game</h1>

<p>Guess the word one letter at a time.</p>

<div id="gameBoard"></div>

<input type="text" id="guessInput" maxlength="1">

<button onclick="makeGuess()">Guess</button>

<p id="message"></p>

<script>

// This is the word that the player needs to guess

const wordToGuess = 'EDUCATION'; // <-- Change this word to any word you like

// This array will store the letters guessed correctly

let guessedWord = [];

// This function initializes the game board

function initializeGame() {

for (let i = 0; i < wordToGuess.length; i++) {

guessedWord[i] = '\_'; // Fills the array with underscores for each letter in the word

}

displayWord();

}

// This function displays the current state of the guessed word

function displayWord() {

const gameBoard = document.getElementById('gameBoard');

gameBoard.innerHTML = '';

for (let i = 0; i < guessedWord.length; i++) {

const span = document.createElement('span');

span.className = 'hiddenLetter';

span.textContent = guessedWord[i];

gameBoard.appendChild(span);

}

}

// This function is called when the player makes a guess

function makeGuess() {

const guessInput = document.getElementById('guessInput');

const guess = guessInput.value.toUpperCase(); // Convert the guess to uppercase

// Check if the guess is a letter in the word

let correctGuess = false;

for (let i = 0; i < wordToGuess.length; i++) {

if (wordToGuess[i] === guess) {

guessedWord[i] = guess;

correctGuess = true;

}

}

// Show a message to the player

const message = document.getElementById('message');

if (correctGuess) {

message.textContent = `Good job! ${guess} is in the word.`;

} else {

message.textContent = `Sorry, ${guess} is not in the word.`;

}

// Clear the input field

guessInput.value = '';

// Update the displayed word

displayWord();

// Check if the player has guessed the whole word

if (!guessedWord.includes('\_')) {

message.textContent = 'Congratulations! You guessed the word!';

}

}

// Initialize the game when the page loads

window.onload = initializeGame;

</script>

</body>

</html>

**Explanation of the Template**

1. **HTML Structure**: The basic HTML structure includes a head section with metadata and styling, and a body section with game elements.
2. **CSS Styling**: Basic styling to center the text and style the guessed letters.
3. **JavaScript Functions**:
   * initializeGame(): Initializes the game by filling the guessedWord array with underscores.
   * displayWord(): Displays the current state of the guessed word on the game board.
   * makeGuess(): Handles the player's guess, updates the guessedWord array, and checks if the guess is correct. It also updates the message displayed to the player.

**Customization Instructions**

* **Change the Word**: In the JavaScript section, change the value of wordToGuess to any word you want the players to guess.
* **Add More Functionality**: Students can add more features like tracking the number of guesses, providing hints, or limiting the number of incorrect guesses.