```
1    /**
2       * Array with image source links to be retrieved and inserted into the HTML when
3       */
4
5    const cardDeck = [
6        "castle.jpg",
7        "cat.jpg",
8        "dragon.jpg",
9        "frog.jpg",
10       "gingerbread.jpg",
11       "owl.jpg",
12       "rabbit.jpg",
13       "rainbow.jpg",
14   ];
15
16   const gameId = 'board-game';
17   const delayBeforeRemovingCards = 100;
18   const maxTopScores = 10;
19   const gameTime = 120; // Total time in seconds that the player has to match all of
20
21   class BoardGame {
22       constructor(totalTime) {
23           this.fullDeck = [];
24           this.totalTurns = 0;
25           this.totalTime = totalTime;
26           this.timeLeft = totalTime;
27           this.turns = document.getElementById("turns");
28           this.timer = document.getElementById("time-left");
29           this.configuration = null;
30           this.playerPanel = document.getElementById("playerPanel");
31           this.boardPanel = document.getElementById("main-gameboard");
32           this.checkCard = null; // Sets the card as the card to be matched
33           this.addListeners();
34       }
35
36       /**
37          * Starts the BoardGame application - it is called when an instance of the c
38          */
39
40       start() {
41           this.loadConfiguration();
42           this.showPlayerPanel();
43       }
44
45       addListeners() {
46           let playerForm = document.getElementById("playerForm");
47           playerForm.addEventListener("submit", this.onStartGameHandler.bind(this));
```

JSHint

version 2.11.1
(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

```
48          }
49  
50          /**
51             * Loads the game configuration from localStorage. If the configuration does
52             */
53  
54          loadConfiguration() {
55              this.configuration = JSON.parse(localStorage.getItem(gameId));
56              if (!this.configuration) { // Sets configuration to the default value
57                  this.configuration = {
58                      playerName: "",
59                      scores: [],
60                  };
61              }
62          }
63  
64          /**
65             * Sets the game cards, timer, turn counter and card actions for the game sc
66             */
67  
68          startGame() {
69              this.checkCard = null;
70              this.totalTurns = 0;
71              this.timeLeft = this.totalTime;
72              this.matchedCards = []; // Array which will store the matched cards as the
73              this.busy = true;
74              setTimeout(() => {
75                  this.shuffleDeck(this.fullDeck);
76                  this.countDown = this.startCountDown();
77                  this.busy = false;
78              }, 500);
79              this.hideCards();
80              this.timer.innerText = this.timeLeft;
81              this.turns.innerText = this.totalTurns;
82              this.showBoardPanel();
83              this.appendCards();
84              this.subscribeButton();
85          }
86  
87          /**
88             * Loads the player name and score from localStorage
89             */
90  
91          showPlayerPanel() {
92              document.getElementById("playerName").value = this.configuration.playerNam
93              this.renderScores();
94              this.boardPanel.classList.toggle("d-none", true);
95              this.playerPanel.classList.toggle("d-none", false);
96          }
```

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

version 2.11.1
(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

```
97
98          showBoardPanel() {
99              this.playerPanel.classList.toggle("d-none", true);
100             this.boardPanel.classList.toggle("d-none", false);
101         }
102
103         /**
104          * Renders the score table and appends a table to the scores container.
105          * If the table already exists, it is removed and created from scratch
106          */
107
108         renderScores() {
109             let scoresContainer = document.getElementById("scores");
110             if (scoresContainer.firstElementChild) {
111                 scoresContainer.firstElementChild.remove();
112             }
113             let table = document.createElement("table");
114             let header = table.createTHead();
115             let headers = header.insertRow(0);
116             headers.innerHTML = `<th class="position">Position</th>
117                                 <th class="player-name">Player Name</th>
118                                 <th class="flips">Total Turns</th>
119                                 <th class="total-time">Total Time</th>`;
120             let tblBody = document.createElement('tbody');
121             this.configuration.scores.forEach((score, index) => {
122                 let tr = document.createElement("tr");
123                 tr.innerHTML = `<td class="position">${index + 1}</td>
124                                 <td class="player-name">${score.playerName}</td>
125                                 <td class="flips">${score.flips}</td>
126                                 <td class="total-time">${score.totalTime}</td>`;
127                 if (score.currentPlayer) {
128                     tr.classList.add('last-game');
129                 }
130                 tblBody.appendChild(tr);
131             });
132             table.appendChild(tblBody);
133             scoresContainer.appendChild(table);
134         }
135
136         /**
137          * Event handler to handle the onsubmit event fired from the playerForm 'Start
138          * @param {*} event DOM Event
139          */
140
141         onStartGameHandler(event) {
142             event.stopImmediatePropagation();
143             event.preventDefault();
144             // Assigns the current player's name to the configuration object to start
145             this.configuration.playerName = event.target[0].value;
```

JSHint

version 2.11.1

(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

```
146             //Starts the game
147             this.startGame();
148         }
149 
150         popSound() {
151             let audio = new Audio("assets/audio/pop.mp3");
152             audio.play();
153         }
154         /**
155          * Renders the card element using the image name passed as a parameter
156          * @param {String} imageName
157          */
158 
159         renderCard(imageName) {
160             return `<div class="card">
161                         <div class="card-back all-cards">
162                             <img class="card-img" src="assets/images/card-back.jpg"  a
163                         </div>
164                         <div class="card-picture all-cards">
165                             <img class="card-value card-img" src="assets/images/${imag
166                         </div>
167                     </div>`;
168         }
169 
170         /**
171          * Appends the cards to the gameboard
172          */
173 
174         appendCards() {
175             // Creates a new array by adding the cardDeck array to itself so there is
176             const allCards = cardDeck.concat(cardDeck);
177 
178             const addCard = document.getElementById("main-gameboard");
179             // insertAdjacentHTML inserts the HTML from the renderCard function for ea
180             allCards.forEach((imageName) => addCard.insertAdjacentHTML("beforeend", th
181 
182 
183             let cards = Array.from(document.getElementsByClassName("card"));
184 
185             cards.forEach((card) => {
186                 card.addEventListener("click", () => {
187                     this.turnCard(card);
188                     this.popSound();
189                 });
190             });
191             this.fullDeck = cards; // Declares a new array of HTML cards for the game
192         }
193 
194         removeCards() {
```

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

version 2.11.1
(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

```
            let cards = Array.from(document.getElementsByClassName("card"));
            cards.forEach((card) => card.remove());
        }


        subscribeButton() {
            document.getElementById("subscribe-submit").addEventListener("click", func
                document.getElementById('email-subscribe').remove();
                document.getElementById('email-text').innerText ="Thank you. You are n
            });
        }

        startCountDown() {
            return setInterval(() => {
                this.timeLeft--;
                this.timer.innerText = this.timeLeft;
                if (this.timeLeft === 0)
                    this.gameOver(); // Ends game when countdown reaches 0
            }, 1000);
        }

        gameFinished() {
            clearInterval(this.countDown);
            // Removes remaining cards from the board
            this.removeCards();
            this.showPlayerPanel();
        }

        currentScore() {
            let currentScore = document.getElementById('current-score');
            currentScore.innerText = this.totalTurns;
        }

        gameOver() {
            this.renderScores();
            this.boardPanel.classList.toggle("d-none", true);
            this.playerPanel.classList.toggle("d-none", false);
            this.gameFinished();
        }

        gameWin() {
            this.currentScore();
            this.updateScores();
            this.gameFinished();
        }
        /**
           * Updates the scores based on the last played game after game has been won
           */

```

JS Hint

version 2.11.1

(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

```
    updateScores() {
        // Disable the previous current played game
        let index = this.configuration.scores.findIndex((score) => score.currentPl
        if (index !== -1) {
            this.configuration.scores[index].currentPlayer = false;
        }

        // Adds the new score to the scoreboard
        this.configuration.scores.push({
            playerName: this.configuration.playerName,
            flips: this.totalTurns,
            totalTime: this.totalTime - this.timeLeft,
            currentPlayer: true
        });

        /**
           * Sorts the scores by comparing values to include the new score - the s
           */

        this.configuration.scores.sort((a, b) => {
            if (a.flips < b.flips) {
                return -1;
            }
            if (a.flips > b.flips) {
                return 1;
            }
            if (a.totalTime < b.totalTime) {
                return -1;
            }
            if (a.totalTime > b.totalTime) {
                return 1;
            }
            return 0;
        });

        // Pop removes the last player from the list of top players if there is an
        if (this.configuration.scores.length > maxTopScores) {
            this.configuration.scores.pop();
        }

        // Converts the configuration into a string and updates it in localStorage
        localStorage.setItem(gameId, JSON.stringify(this.configuration));
    }

    hideCards() {
        this.fullDeck.forEach((card) => {
            card.classList.remove('visible');
        });
    }
```

version 2.11.1

(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

```
    /**
        *
        * @param {Element} card The card element
        */

    turnCard(card) {
        if (this.isCardFacedDown(card)) {
            // Increases the number of turns
            this.totalTurns++;
            // Increases the number of turns on screen
            this.turns.innerText = this.totalTurns;
            // Shows the card
            card.classList.add('visible');
            if (this.checkCard) {
                this.checkForMatch(card);
            } else {
                this.checkCard = card;
            }
        }
    }

    /**
     * Checks if the card is a match with the previously selected card
     * @param {Element} card
     */

    checkForMatch(card) {
        if (this.checkCardType(card) === this.checkCardType(this.checkCard)) {
            this.cardMatcher(card, this.checkCard);
        } else {
            this.notAMatch(card, this.checkCard);
            // Clears the card selection
            this.checkCard = null;
        }
    }

    /**
        *
        * @param {*} card1 First card selected
        * @param {*} card2 Second card selected
        */

    cardMatcher(card1, card2) {
        // Adds the cards to the matchedCards array to track progress
        this.matchedCards.push(card1);
        this.matchedCards.push(card2);
        setTimeout(() => {
            card1.classList.add("invisible");
```

JSHint

version 2.11.1

(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla

```
342                 card2.classList.add("invisible");
343             }, delayBeforeRemovingCards);
344             this.checkCard = null;
345             // Ends the game when all cards have been matched
346             if (this.matchedCards.length === this.fullDeck.length) {
347                 this.gameWin();
348             }
349         }
350 
351         notAMatch(card1, card2) {
352             this.busy = true;
353             setTimeout(() => {
354                 card1.classList.remove("visible");
355                 card2.classList.remove("visible");
356                 this.busy = false;
357             }, 500);
358         }
359 
360         checkCardType(card) {
361             return card.getElementsByClassName("card-value")[0].src;
362         }
363 
364         /**
365            * Fisher-Yates algorithm shuffles through the card array swapping
366            * the last element with a random element from the array
367            */
368 
369         shuffleDeck() {
370             for (let i = this.fullDeck.length - 1; i > 0; i--) {
371                 let randomIndex = Math.floor(Math.random() * (i + 1));
372                 this.fullDeck[randomIndex].style.order = i;
373                 this.fullDeck[1].style.order = randomIndex;
374             }
375         }
376 
377         isCardFacedDown(card) {
378             return (
379                 !this.busy && !this.matchedCards.includes(card) && card !== this.check
380             );
381         }
382     }
383 
384     const game = new BoardGame(gameTime);
385     game.start();
386 
387 
```

version 2.11.1

(https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 41 functions in this file.

Function with the largest signature take 2 arguments, while the median is 0.

Largest function has 12 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 5 while the median is 1.

38 warnings

5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

16 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

17 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

18 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

19 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

46 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

109 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

113 'let' is available in ES6 (use 'esversion: 6') or Mozilla