

JavaScript Code Validation Test for Kids Bible Memory Game

The screenshot displays a web browser with two main panels. The left panel shows a JavaScript file named 'jshint.com' with the following code:

```

47 }
48 gameOver() {
49   this.stopMusic();
50   this.gameOverSound.play();
51 }
52 }
53
54 //For the Game Activities
55 class FindAMatch {
56   constructor(totalTime, cards) {
57     this.cardsArray = cards;
58     this.totalTime = totalTime;
59     this.countDown = totalTime;
60     this.timer = document.getElementById('time-count');
61     this.ticktok = document.getElementById('flips');
62     this.soundControl = new SoundController();
63   }
64
65   //Starts game when called
66   playGame() {
67     this.totalClicks = 0;
68     this.countDown = this.totalTime;
69     this.verifyCard = null;
70     this.matchedCards = [];
71     this.busy = true;
72
73     //Delays game start time
74     setTimeout(() => {
75       this.soundControl.playMusic();
76       this.cardsShuffle(this.cardsArray);
77       this.timecount = this.startCountDown();
78       this.busy = false;
79     }, 500);
80     this.hideCards();
81     this.timer.innerText = this.countDown;
82     this.ticktok.innerText = this.totalClicks;
83   }
84   //Makes the card close after being flipped
85   hideCards() {
86     this.cardsArray.forEach(card => {
87       card.classList.remove('visible');
88     });
89   }
90
91   //Allows player to flip cards
92   cardFlip(card) {
93     if (this.allowedToFlip(card)) {
94       this.soundControl.flip();
95       this.totalClicks++;
96       this.ticktok.innerText = this.totalClicks;
97       card.classList.add('visible');

```

The right panel shows the JSHint configuration page. The 'CONFIGURE' tab is active, displaying a list of 15 warnings. The first warning is 'class' is available in ES6 (use 'esversion: 6' or Mozilla JS extensions (use moz)). The last warning is 'let' is available in ES6 (use 'esversion: 6' or Mozilla JS extensions (use moz)). The page also includes a sidebar with links for About, Documentation, Install, and Contribute.

JavaScript Code Validation Test for Kids Bible Memory Game

jslint.com

```
90
91 //Allows player to flip cards
92 cardFlip(card) {
93   if (this.allowedToFlip(card)) {
94     this.soundControl.flip();
95     this.totalClicks++;
96     this.ticktock.innerText = this.totalClicks;
97     card.classList.add('visible');
98
99     if(this.verifyCard) {
100      this.checkForCardMatch(card);
101    } else {
102      this.verifyCard = card;
103    }
104  }
105 }
106 //Checks if the cards match or not
107 checkForCardMatch(card) {
108   if(this.getCardType(card) === this.getCardType(this.verifyCard))
109     this.cardMatch(card, this.verifyCard);
110   else
111     this.cardNotMatch(card, this.verifyCard);
112 }
113 this.verifyCard = null;
114
115 //Stores matched cards and called when they are a match
116 cardMatch(card1, card2) {
117   this.matchedCards.push(card1);
118   this.matchedCards.push(card2);
119   card1.classList.add('matched');
120   card2.classList.add('matched');
121   this.soundControl.matched();
122   if(this.matchedCards.length === this.cardsArray.length)
123     this.winner();
124 }
125
126 //Stores unmatched cards and called when they are not a match
127 cardNotMatch(card1, card2) {
128   this.busy = true;
129   setTimeout(() => {
130     card1.classList.remove('visible');
131     card2.classList.remove('visible');
132     this.busy = false;
133   }, 1000);
134 }
135 //called when the value of a card is required
136 getCardType(card) {
137   return card.getAttributeByClassName('card-value')[0].src;
138 }
139
```

CONFIGURE

Metrics

There are 30 functions in this file.
Function with the largest signature take 2 arguments, while the median is 0.
Largest function has 9 statements in it, while the median is 2.
The most complex function has a cyclomatic complexity value of 3 while the median is 1.

15 warnings

- 21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 55 'class' 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').
- 86 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 129 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 142 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 165 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 166 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 186 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 187 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

JSHint version 2.11.1

- About
- Documentation
- Install
- Contribute
- Blog

jslint.com

```
140
141 //Starts Timer
142 startCountdown() {
143   return setInterval(() => {
144     this.countDown--;
145     this.timer.innerText = this.countDown;
146     if(this.countDown === 0)
147       this.gameOver();
148   }, 1000);
149 }
150 //This is called when the game ends
151 gameOver() {
152   clearInterval(this.timeCount);
153   this.soundControl.gameOver();
154   document.getElementById('game-over-msg').classList.add('visible');
155 }
156 //This is called when the player finds all matching cards
157 winner() {
158   clearInterval(this.timeCount);
159   this.soundControl.winner();
160   document.getElementById('winner-msg').classList.add('visible');
161   this.hideCards();
162 }
163 // Fisher-Yates Shuffle Algorithm - Allow for cards shuffle
164 cardShuffle() {
165   for(let i = this.cardsArray.length - 1; i > 0; i--) {
166     let randomInt = Math.floor(Math.random() * (i + 1));
167     this.cardsArray[randomInt].style.order = i;
168     this.cardsArray[i].style.order = randomInt;
169   }
170 }
171 //This stops or allows a player to flip a card when there is an ongoing activity or no
172 allowedToFlip(card) {
173   return !this.busy && !this.matchedCards.includes(card) && card !== this.verifyCard;
174 }
175 }
176
177 //Allows HTML page to load first
178 if (document.readyState === 'loading') {
179   document.addEventListener('DOMContentLoaded', ready);
180 } else {
181   ready();
182 }
183
184 //To initialize game and rules

```

CONFIGURE

Metrics

There are 30 functions in this file.
Function with the largest signature take 2 arguments, while the median is 0.
Largest function has 9 statements in it, while the median is 2.
The most complex function has a cyclomatic complexity value of 3 while the median is 1.

15 warnings

- 21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 55 'class' 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').
- 86 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 129 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 142 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 165 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 166 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 186 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 187 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

JSHint version 2.11.1

- About
- Documentation
- Install
- Contribute
- Blog

JavaScript Code Validation Test for Kids Bible Memory Game

The screenshot displays the JSHint online code validator interface. The left pane contains the JavaScript code for a Kids Bible Memory Game, and the right pane shows the validation results.

JavaScript Code (Left Pane):

```
156 winner() {
157   clearInterval(this.timeCount);
158   this.soundControl.winner();
159   document.getElementById('winner-msg').classList.add('visible');
160   this.hideCards();
161 }
162
163 // Fisher-Yates Shuffle Algorithm - Allow for cards shuffle
164 cardsShuffle() {
165   for(let i = this.cardsArray.length - 1; i > 0; i--) {
166     let randomInt = Math.floor(Math.random() * (i + 1));
167     this.cardsArray[randomInt].style.order = i;
168     this.cardsArray[i].style.order = randomInt;
169   }
170 }
171 //This stops or allows a player to flip a card when there is an ongoing activity or no
172 allowedToFlip(card) {
173   return !this.busy && !this.matchedCards.includes(card) && card !== this.verifyCard
174 }
175
176
177 //Allows HTML page to load first
178 if (document.readyState == 'loading') {
179   document.addEventListener('DOMContentLoaded', ready);
180 } else {
181   ready();
182 }
183
184 //To initialise game and overlays
185 function ready() {
186   let allOverlays = Array.from(document.getElementsByClassName('overlay-msg'));
187   let cards = Array.from(document.getElementsByClassName('card'));
188   let game = new FindAMatch(30, cards);
189
190   allOverlays.forEach(overlay => {
191     overlay.addEventListener('click', () => {
192       overlay.classList.remove('visible');
193       game.playGame();
194     });
195   });
196
197   cards.forEach(card => {
198     card.addEventListener('click', () => {
199       game.cardFlip(card);
200     });
201   });
202 }
203
204
```

Validation Results (Right Pane):

CONFIGURE

Metrics

There are 30 functions in this file.
Function with the largest signature take 2 arguments, while the median is 0.
Largest function has 9 statements in it, while the median is 2.
The most complex function has a cyclomatic complexity value of 3 while the median is 1.

15 warnings

- 21 'class' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 55 'class' 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').
- 86 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 129 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 142 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').
- 165 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 166 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 186 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 187 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

JSHint
version 2.11.1

[About](#)
[Documentation](#)
[Install](#)
[Contribute](#)
[Blog](#)