

```

1  /*global $
2  */
3  // $(document).ready(function() {}); removed for jasmine testing
4  // let , var declarations / assignments
5  let masterCardArray = ['card1', 'card1', 'card2', 'card2', 'card3', 'card3', 'card'];
6  let currentCardArray = [];
7  let namePlayer1 = "Player1";
8  let namePlayer2 = "Player2";
9  let scorePlayer1 = 0;
10 let scorePlayer2 = 0;
11 let currentPlayer = "Player1";
12 let firstAttemptDone = 0;
13 let ThisGameOpenedBy = "Player1";
14 let CardRowlength;
15
16 // switches for active states
17 let btnActive = true;
18 let fieldActive = false;
19 let startBtnActive = true;
20 let stopBtnActive = false;
21
22 // implementations of functions
23 function generateCards(num) {
24     let playFieldSize = num;
25     $(".playfield").empty();
26     for (var i = 0; i < playFieldSize; i++) {
27         if (playFieldSize == 9 && i == 4) {
28             $(".playfield").append("<div class='dummycardshell'></div>");
29         }
30         else {
31             $(".playfield").append("<div class='cardshell'></div>");
32         }
33     }
34     $(".cardshell").append("<div class='card front vhaligh'></div>");
35     $(".cardshell").append("<div class='card back vhaligh'></div>");
36 }
37
38 function setCardshellSize(num) {
39     let playFieldSize2 = num;
40     if (playFieldSize2 == 9) {
41         CardRowlength = 3;
42         $(".cardshell").css("width", "31.7%").css("height", "31.7%");
43         $(".dummycardshell").css("width", "31.7%").css("height", "31.7%");
44     }
45     else if (playFieldSize2 == 16) {
46         CardRowlength = 4;
47         $(".cardshell").css("width", "23.7%").css("height", "23.7%");

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint/>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

48     $(".dummycardshell").css("width", "23.7%").css("height", "23.7%");
49   }
50   else if (playFieldSize2 == 36) {
51     CardRowlength = 6;
52     $(".cardshell").css("width", "15.4%").css("height", "15.4%");
53     $(".dummycardshell").css("width", "15.4%").css("height", "15.4%");
54   }
55 }
56
57 // function for generating playfield
58 function fieldInit(num) {
59   let playFieldSize = num;
60   $(".playfield").css('opacity', '0.0');
61   setTimeout(function() {
62     generateCards(playFieldSize);
63     setCardshellSize(playFieldSize);
64     prepAndDeliverCardArray(playFieldSize);
65   }, 800);
66   setTimeout(function() {
67     $(".playfield").css('opacity', '1.0');
68   }, 1200);
69 }
70
71 // function for preparation and delivery of playfield array
72 function prepAndDeliverCardArray(num) {
73   let playFieldSize2 = num;
74   currentCardArray = masterCardArray.concat();
75   let playFieldCardArray = $(".back").toArray();
76   if (playFieldSize2 == 9) {
77     playFieldSize2 = 8;
78     currentCardArray.splice(playFieldSize2, 28);
79   }
80   else if (playFieldSize2 == 16) {
81     currentCardArray.splice(playFieldSize2, 20);
82   }
83
84   currentCardArray.sort(function(a, b) { return 0.5 - Math.random() });
85
86   for (let i = 0; i < playFieldSize2; i++) {
87     $(playFieldCardArray[i]).addClass(currentCardArray[i]);
88   }
89 }
90
91 // function to check for matching cards
92 function checkForMatch() {
93   if ($('.taken .back').length == 2) {
94     fieldActive = false;
95     let takenCard1 = $('.taken .back').eq(0);
96     let takenCard2 = $('.taken .back').eq(1);

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.
The most complex function has a cyclomatic complexity value of 6 while the median is 1.

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS



version 2.10.2

(<https://github.com/jshint/jshint/>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

```

97     let classesCard1 = takenCard1.attr("class");
98     let classesCard2 = takenCard2.attr("class");
99     if (classesCard1 == classesCard2) {
100         matched();
101     }
102     else if (classesCard1 != classesCard2) {
103         notMatched();
104     }
105 }
106 }
107
108 function matched() {
109     setTimeout(function() { // wait until cards have fully turned
110         popupMatch();
111         setTimeout(function() { // delay time to wait until popupMatch is vanished
112             increasePoints();
113             setTimeout(function() { // delay time to wait until players logo is on
114                 $('.taken').addClass('dummyscardshell').removeClass('cardshell');
115                 $('.dummyscardshell').removeClass('taken');
116                 // checkup if all cards have been found by comparing amount of .sh
117                 if ($('.showMe').length == $('.front').length) {
118                     gameCompleted();
119                 }
120                 else {
121                     whoIsNext();
122                     setTimeout(function() { // wait littlebit and make playfield a
123                         fieldActive = true;
124                     }, 300);
125                 }
126             }, 800); // delay time to wait until players logo is on card
127             }, 1000); // delay time to wait until popupMatch is vanished
128         }, 500); // wait until cards have fully turned prior popup 'match'
129     }
130 }
131
132 function notMatched() {
133     setTimeout(function() { // wait until cards have fully turned
134         popupNoMatch();
135         setTimeout(function() { // delay of 1500ms to be able to see shown cards
136             $(".cardshell").removeClass('showMe taken');
137             changePlayer(currentPlayer);
138             setTimeout(function() { // makes whoIsNext checkup little more delayed
139                 whoIsNext();
140                 setTimeout(function() { // wait some time until who is next pop up
141                     fieldActive = true;
142                 }, 1000); // wait some time until who is next pop up is vanished
143             }, 500); // makes whoIsNext checkup little more delayed
144             }, 1500); // delay of 1500ms to be able to see shown cards
145         }, 500); // wait until cards have fully turned
146     }
147 }

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

146
147 //function to assign player's color on card and increase points
148 function increasePoints() {
149     if (currentPlayer == "Player1") {
150         $(' .taken .back').append("<i class='checkmarkPlayer1Big fa fa-check-circle
151         changeFontSizeBigLogo();
152         scorePlayer1++;
153         $(' .scorePlayer1Field').html(scorePlayer1);
154         $(' .checkmarkPlayer1Big').addClass('bubbleIcon');
155     }
156     else if (currentPlayer == "Player2") {
157         $(' .taken .back').append("<i class='checkmarkPlayer2Big fa fa-check-circle
158         changeFontSizeBigLogo();
159         scorePlayer2++;
160         $(' .scorePlayer2Field').html(scorePlayer2);
161         $(' .checkmarkPlayer2Big').addClass('bubbleIcon');
162     }
163 }
164
165 function changeFontSizeBigLogo() {
166     let tmpWidth = $(".playfield").css("width");
167     let tmpWidthInt = parseInt(tmpWidth);
168     let result;
169     let sizeBigLogo;
170     if (CardRowlength == 3) {
171         result = tmpWidthInt / 4; // keep font size smaller than 1/3 of playfields
172         sizeBigLogo = result + "px";
173         $(' .checkmarkPlayer1Big').css('font-size', sizeBigLogo);
174         $(' .checkmarkPlayer2Big').css('font-size', sizeBigLogo);
175     }
176     else if (CardRowlength == 4) {
177         result = tmpWidthInt / 4.5; // keep font size smaller than 1/4 of playfield
178         sizeBigLogo = result + "px";
179         $(' .checkmarkPlayer1Big').css('font-size', sizeBigLogo);
180         $(' .checkmarkPlayer2Big').css('font-size', sizeBigLogo);
181     }
182     else if (CardRowlength == 6) {
183         result = tmpWidthInt / 7; // keep font size smaller than 1/6 of playfields
184         sizeBigLogo = result + "px";
185         $(' .checkmarkPlayer1Big').css('font-size', sizeBigLogo);
186         $(' .checkmarkPlayer2Big').css('font-size', sizeBigLogo);
187     }
188 }
189
190 // function for changing player
191 function changePlayer(str) {
192     let activePlayer = str;
193     if (activePlayer == "Player1") {
194         currentPlayer = "Player2";

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint/>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

195     }
196     else if (activePlayer == "Player2") {
197         currentPlayer = "Player1";
198     }
199 }
200
201 // function to set the activePlayer
202 function setActivePlayer(str) {
203     let player = str;
204     // moving up popup on z axis
205     $('.popupNext').css("transform", "translateZ(400px)").css("z-index", "400");
206     if (player == "Player1") {
207         $('.playerStats1').css('background-color', 'red');
208         $('.playerStats2').css('background-color', 'grey');
209         $('.popupNext').html(namePlayer1 + " is next!");
210     }
211     else if (player == "Player2") {
212         $('.playerStats1').css('background-color', 'grey');
213         $('.playerStats2').css('background-color', 'red');
214         $('.popupNext').html(namePlayer2 + " is next!");
215     }
216     $('.popupNext').css("opacity", "1.0");
217
218     setTimeout(function() {
219         $('.popupNext').css("opacity", "0.0");
220     }, 1000);
221
222     setTimeout(function() {
223         $('.popupNext').css("transform", "translateZ(-10px)").css("z-index", "-1")
224     }, 2000);
225 }
226
227 // function for changing opening player
228 function changeOpeningPlayer(str) {
229     let currentGameOpenedBy = str;
230     if (currentGameOpenedBy == "Player1") {
231         ThisGameOpenedBy = "Player2";
232     }
233     else if (currentGameOpenedBy == "Player2") {
234         ThisGameOpenedBy = "Player1";
235     }
236 }
237
238 // function for displaying final result of who has won the game
239 function gameCompleted() {
240     if (scorePlayer1 > scorePlayer2) {
241         $('.popupGameCompleted').html(namePlayer1 + " has won!");
242     }
243     else if (scorePlayer2 > scorePlayer1) {

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

[\(https://github.com/jshint/jshint/\)](https://github.com/jshint/jshint/)
[About \(/about\)](#)
[Documentation \(/docs\)](#)
[Install \(/install\)](#)
[Contribute \(/contribute\)](#)
[Blog \(/blog\)](#)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

244     $('#popupGameCompleted').html(namePlayer2 + " has won!");
245 }
246 else if (scorePlayer1 == scorePlayer2) {
247     $('#popupGameCompleted').html(namePlayer1 + " and " + namePlayer2 + " have
248 }
249 $('#popupGameCompleted').css("transform", "translateZ(150px)").css("z-index",
changeOpeningPlayer(ThisGameOpenedBy);
250 setTimeout(function() { // have popup for game completion visible for 4 second:
251     $('#popupGameCompleted').css("opacity", "0.0"); // make popup dissappear
252     firstAttemptDone = 0; // resetting indicator for first move of game done.
253     makeBtnActiveButStart();
254     setTimeout(function() { // wait a little until popup for game completion ha
255         $('#popupGameCompleted').css("transform", "translateZ(-10px)").css("z-
256     }, 1200); // wait a little until popup for game completion has vanished
257 }, 4000); // have popup for game completion visible for 4 seconds
258 }
259 }
260
261 // functions to show popups
262 function whoIsNext() {
263     if (firstAttemptDone == 0 && ThisGameOpenedBy == "Player1") { // on 1st move o
264         currentPlayer = "Player1"; // currentPlayer is set to be synchronized accor
265         setActivePlayer(currentPlayer);
266     }
267     else if (firstAttemptDone == 0 && ThisGameOpenedBy == "Player2") {
268         currentPlayer = "Player2";
269         setActivePlayer(currentPlayer);
270     }
271     else if (firstAttemptDone == 1 && currentPlayer == "Player1") {
272         setActivePlayer(currentPlayer);
273     }
274     else if (firstAttemptDone == 1 && currentPlayer == "Player2") {
275         setActivePlayer(currentPlayer);
276     }
277     firstAttemptDone = 1; // game has been started
278 }
279
280 // function to show popup 'match'
281 function popupMatch() {
282     $('#popupMatch').css("transform", "translateZ(400px)").css("z-index", "400");
283     setTimeout(function() {
284         $('#popupMatch').css("opacity", "1.0");
285     }, 300);
286     setTimeout(function() {
287         $('#popupMatch').css("opacity", "0.0");
288     }, 1400);
289     setTimeout(function() {
290         $('#popupMatch').css("transform", "translateZ(-10px)").css("z-index", "-1"
291     }, 2200);
292 }

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

([https://github.com/jshint/j](https://github.com/jshint/jshint)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS


```

293
294 // function to show popup 'no match'
295 function popupNoMatch() {
296     $('.popupNoMatch').css("transform", "translateZ(400px)").css("z-index", "400")
297     setTimeout(function() {
298         $('.popupNoMatch').css("opacity", "1.0");
299     }, 300);
300     setTimeout(function() {
301         $('.popupNoMatch').css("opacity", "0.0");
302     }, 1400);
303     setTimeout(function() {
304         $('.popupNoMatch').css("transform", "translateZ(-10px)").css("z-index", "-1");
305     }, 2200);
306 }
307
308 // setting indicator on button for playfield size
309 function make_field8BtnVisActive() {
310     $(".field8Btn").addClass("selectedSize").removeClass("bg-fieldSizeBtn"); // in
311     $(".field16Btn").removeClass("selectedSize").addClass("bg-fieldSizeBtn"); // r
312     $(".field36Btn").removeClass("selectedSize").addClass("bg-fieldSizeBtn"); // r
313 }
314
315 function make_field16BtnVisActive() {
316     $(".field8Btn").removeClass("selectedSize").addClass("bg-fieldSizeBtn"); // rer
317     $(".field16Btn").addClass("selectedSize").removeClass("bg-fieldSizeBtn"); // i
318     $(".field36Btn").removeClass("selectedSize").addClass("bg-fieldSizeBtn"); // r
319 }
320
321 function make_field36BtnVisActive() {
322     $(".field8Btn").removeClass("selectedSize").addClass("bg-fieldSizeBtn"); // rer
323     $(".field16Btn").removeClass("selectedSize").addClass("bg-fieldSizeBtn"); // r
324     $(".field36Btn").addClass("selectedSize").removeClass("bg-fieldSizeBtn"); // i
325 }
326
327 // disabling buttons
328 function makeBtnInactive() {
329     // set dimmed state to all buttons except 'how to' and 'stopbutton'
330     $(".enterPlayersBtn").addClass("btnlocked");
331     $(".field8Btn").addClass("btnlocked");
332     $(".field16Btn").addClass("btnlocked");
333     $(".field36Btn").addClass("btnlocked");
334     $(".startBtn").addClass("btnlocked");
335     $(".stopBtn").removeClass("btnlocked"); // stopbutton will become visually act
336     // button states
337     stopBtnActive = true;
338     fieldActive = true;
339     btnActive = false;
340     startBtnActive = false;
341 }

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

342
343 // function for removing dimmed button state and to make buttons active ...
344 function makeBtnActive() {
345     // ... remove dimmed state
346     $(".enterPlayersBtn").removeClass("btnlocked");
347     $(".field8Btn").removeClass("btnlocked");
348     $(".field16Btn").removeClass("btnlocked");
349     $(".field36Btn").removeClass("btnlocked");
350     // start button
351     $(".startBtn").removeClass("btnlocked");
352     $(".stopBtn").addClass("btnlocked"); // dimmed state applied when game stopped
353     // button states
354     btnActive = true;
355     startBtnActive = true;
356     stopBtnActive = false;
357 }
358
359 function makeBtnActiveButStart() {
360     // ... remove dimmed state
361     $(".enterPlayersBtn").removeClass("btnlocked");
362     $(".field8Btn").removeClass("btnlocked");
363     $(".field16Btn").removeClass("btnlocked");
364     $(".field36Btn").removeClass("btnlocked");
365     $(".stopBtn").addClass("btnlocked"); // dimmed state applied when game stopped
366     // button states
367     btnActive = true;
368     fieldActive = false;
369     stopBtnActive = false;
370 }
371
372 // function for counter reset on game startup
373 function resetCounters() {
374     scorePlayer1 = 0;
375     $(".scorePlayer1Field").html(scorePlayer1);
376     scorePlayer2 = 0;
377     $(".scorePlayer2Field").html(scorePlayer2);
378 }
379
380 function showRegistrationPopup() {
381     $('#enterPlayersModal').css("transform", "translateZ(400px)").css("z-index", "400");
382 }
383
384 function checkNames() {
385     //should either field have string length of 0, or...
386     if ($('#nameFieldPlayer1Form').val().length == 0 || $('#nameFieldPlayer2Form').val().length == 0) {
387         $('#popupCheckNames').html('Please fill in names<br> in both fields.');
```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).


```

391     else if ($('#nameFieldPlayer1Form').val().length > 8 || ($('#nameFieldPlayer2Form').val().length > 8)) {
392         $('#popupCheckNames').html('Please fill in names with no more than 8 characters');
393         popupCheckNames();
394     }
395     //should both fields have same name, the user will be informed with alert popup
396     else if ($('#nameFieldPlayer1Form').val() == ($('#nameFieldPlayer2Form').val())) {
397         $('#popupCheckNames').html('Please provide different names for each player');
398         popupCheckNames();
399     }
400     // in any other case
401     else {
402         processNames();
403     }
404 }
405
406 function popupCheckNames() {
407     $('#popupCheckNames').css("transform", "translateZ(500px)").css("z-index", "500");
408     setTimeout(function() {
409         $('#popupCheckNames').css("opacity", "1.0");
410     }, 300);
411     setTimeout(function() {
412         $('#popupCheckNames').css("opacity", "0.0");
413     }, 2200);
414     setTimeout(function() {
415         $('#popupCheckNames').css("transform", "translateZ(-10px)").css("z-index", "500");
416     }, 2800);
417 }
418
419 // string values of textinput fields written to HTML elements and internal variables
420 function processNames() {
421     namePlayer1 = ($('#nameFieldPlayer1Form').val());
422     $('#namePlayer1Field').html(namePlayer1 + ": ");
423     namePlayer2 = ($('#nameFieldPlayer2Form').val());
424     $('#namePlayer2Field').html(namePlayer2 + ": ");
425     $('#enterPlayersModal').css("opacity", "0.0");
426     setTimeout(function() {
427         $('#enterPlayersModal').css("transform", "translateZ(-10px)").css("z-index", "500");
428     }, 1000);
429 }
430
431 function showHowToPopup() {
432     $('#howToModal').css("transform", "translateZ(400px)").css("z-index", "400").css("opacity", "1.0");
433 }
434
435 // code executed on startup:
436 window.onresize = function() { changeFontSizeBigLogo(); };
437 fieldInit(9); // generating playfield of 3x3 per default on startup
438 make_field8BtnVisActive();
439 resetCounters();

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

440 setTimeout(function() { // show registration when playfield is generated
441     showRegistrationPopup();
442 }, 2000);
443
444 // click events for registration popup button, ...
445 $(".enterPlayersBtn").on('click', function() {
446     if (btnActive == true) {
447         showRegistrationPopup();
448     }
449 });
450
451 // ... how-to popup button,
452 $(".howToBtn").on('click', function() {
453     // Howto button is accessible all the time
454     showHowToPopup();
455 });
456
457 // ... 8-card playfield button,
458 $(".field8Btn").on('touchstart click', function() {
459     if (btnActive == true) {
460         make_field8BtnVisActive();
461         fieldInit(9); // initializing fieldsize 3x3 cards / 4 pairs with one free c
462         resetCounters();
463         startBtnActive = true;
464         $(".startBtn").removeClass("btnlocked");
465     }
466 });
467
468 // ... 16-card playfield button,
469 $(".field16Btn").on('touchstart click', function() {
470     if (btnActive == true) {
471         make_field16BtnVisActive();
472         fieldInit(16); // initializing fieldsize 4x4 cards / 8 pairs
473         resetCounters();
474         startBtnActive = true;
475         $(".startBtn").removeClass("btnlocked");
476     }
477 });
478
479 // ... 36-card playfield button,
480 $(".field36Btn").on('touchstart click', function() {
481     if (btnActive == true) {
482         make_field36BtnVisActive();
483         fieldInit(36); // initializing fieldsize 6x6 cards / 18 pairs
484         resetCounters();
485         startBtnActive = true;
486         $(".startBtn").removeClass("btnlocked");
487     }
488 });

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint/>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

```

489
490 // ... start button
491 $(".startBtn").on('touchstart click', function() {
492     if (startBtnActive == true) {
493         makeBtnInactive();
494         whoIsNext();
495         setTimeout(function() {
496             fieldActive = true;
497             }, 1200);
498     }
499 });
500
501 // ... stop button
502 $(".stopBtn").on('touchstart click', function() {
503     if (stopBtnActive == true) {
504         fieldActive = false;
505         makeBtnActive();
506     }
507 });
508
509 // ... cards on playfield
510 $(document).on('touchstart click', '.cardshell', function() {
511     if (fieldActive == true) {
512         $(this).addClass("showMe taken");
513         checkForMatch();
514     }
515 });
516
517 // ack button howTo modal
518 $('#gotItBtn').on('touchstart click', function() {
519     $('#howToModal').css("opacity", "0.0");
520     setTimeout(function() {
521         $('#howToModal').css("transform", "translateZ(-10px)").css("z-index", "-1");
522     }, 1000);
523 });
524
525 // ... for save button button on registration modal
526 $('#saveBtn').on('touchstart click', function() {
527     checkNames();
528 });

```

CONFIGURE

Metrics

There are 67 functions in this file.

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

Largest function has 17 statements in it, while the median is 3.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.



version 2.10.2

(<https://github.com/jshint/jshint>)

[About \(/about\)](#)

[Documentation \(/docs\)](#)

[Install \(/install\)](#)

[Contribute \(/contribute\)](#)

[Blog \(/blog\)](#)

35 warnings

- 5 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 7 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 8 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 9 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 10 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 11 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 12 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 13 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).