

[Return to "Front-End Web Developer Nanodegree" in the classroom](#)

Memory Game

REVIEW

CODE REVIEW 7

HISTORY

▼ js/app.js 5

```
1 // * Create a list that holds all of your cards
2 let allCards = ["fa-anchor", "fa-bolt", "fa-bicycle", "fa-bomb", "fa-cube", "fa-diamond", "fa-paper-pla
```

AWESOME

ES6 introduces the following variable declaration syntax –
Using the let.
Using the const.
Well done on using ES6 variables 👍

```
3
4 // http://albert-gonzalez.github.io/easytimer.js/
5 let timer = new Timer();
6 timer.addEventListener('secondsUpdated', function (e) {
7   $('#timer').html(timer.getTimeValues().toString());
8 });
9
10 // set counters
11 let moves=0, match=0, trackCard=[], oneStar=22, twoStar=16, time=false;
12
13 $('#repeat').click(repeat);
14
15 function displayDeck(card) {
16   $('#deck').append(`<li class="card animated"><i class="fa ${card}"></i></li> `);
17 }
18
19 function displayCards(n) {
```

AWESOME

Great that you append cards and dynamically in the grid 🙌 .

```
20 for (let i = 0; i < n; i++) {
21   allCards = shuffle(allCards);
22   allCards.forEach(displayDeck);
23 }
24 }
25
26 function displayStars(n) {
27   let i=0;
28   while (i < n) {
29     $('#stars').append(`<li><i class="fa fa-star"></i></li>`);
30     i++;
31   }
32 }
33 // Shuffle function from http://stackoverflow.com/a/2450976
34 function shuffle(array) {
35   var currentIndex = array.length, temporaryValue, randomIndex;
36
37   while (currentIndex !== 0) {
```

SUGGESTION

Try making your own shuffle logic after deleting this given starter code. It'll be more fun 😊

Rate this review



```

38     randomIndex = Math.floor(Math.random() * currentIndex);
39     currentIndex -= 1;
40     temporaryValue = array[currentIndex];
41     array[currentIndex] = array[randomIndex];
42     array[randomIndex] = temporaryValue;
43 }
44
45 return array;
46 }
47
48 function toggleCard() {
49     if (time == false) {
50         time = true;
51         timer.start();
52     }
53     if (trackCard.length === 0) {
54         $(this).toggleClass("show open").animateCss('flipInY');
55         trackCard.push($(this));
56         disableClick(true);
57     }
58     else if (trackCard.length === 1) {
59         countMoves();
60         $(this).toggleClass("show open").animateCss('flipInY');
61         trackCard.push($(this));
62         setTimeout(matchOpenCards, 1100);
63     }
64 }
65
66 function countMoves() {
67     moves += 1;
68     $('#moves').html(` ${moves} Moves `);
69     if (moves == oneStar || moves == twoStar) {
70         $('#stars').children()[0].remove();
71         $('#stars').append('<li><i class="fa fa-star-o"></i></li>');
72     }
73 }
74
75 function disableClick(f) {
76     if (f==true){
77         trackCard.forEach(function (card) {
78             card.off('click');
79         });
80     } else {
81         trackCard[0].click(toggleCard);
82     }
83 }
84
85
86 function matchOpenCards() {
87     if (trackCard[0][0].firstChild.className == trackCard[1][0].firstChild.className) {
88         trackCard[0].addClass("match").animateCss('pulse');
89         trackCard[1].addClass("match").animateCss('pulse');
90         disableClick(true);
91         trackCard = [];
92         setTimeout(displayResult, 1000);
93     }
94     else {
95         trackCard[0].toggleClass("show open").animateCss('flipInY');
96         trackCard[1].toggleClass("show open").animateCss('flipInY');
97         disableClick(false);
98         trackCard = [];
99     }
100 }
101
102
103 function displayResult(){

```

SUGGESTION

Well done with using functions. But now you can use javascript for making functions called arrow functions. C
[@https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions)

```

104     match += 1;
105     if (match == allCards.length) {
106         timer.pause();
107         let playTime = timer.getTimeValues().toString();
108         $('#deck-container')[0].style.display = "none";
109         $('#sucess-result')[0].style.display = "block";
110         $('#sucess-result').append(endGame(moves,calStar(moves),playTime));
111     }
112 }
113
114 function calStar(m){
115     let star=0;
116     if (m >= oneStar) {
117         star=1;
118     }else if (m >= twoStar) {
119         star=2;
120     }else {

```

```

121     star=3;
122   }
123   return star;
124 }
125
126 function endGame(moves,star,time) {
127   swal({
128     allowEscapeKey: false,
129     allowOutsideClick: false,
130     title: 'Congratulations! You Won!',
131     text: 'With ' + moves + ' Moves and ' + star + ' ' + ((star > 1) ? "Stars" : "Star") + ' in ' + time + '.\r
132     type: 'success',
133     confirmButtonColor: '#02ccba',
134     confirmButtonText: 'Play again!'
135   }).then(function (isConfirm) {
136     if (isConfirm) {
137       repeat();
138     }
139   })
140 }
141
142 function repeat() {
143   moves = 0;
144   match = 0;
145   $('#deck').empty();
146   $('#stars').empty();
147   $('#deck-container')[0].style.display = "";
148   time=false;
149   timer.stop();
150   $('#timer').html("00:00:00");
151   runGame();
152 }
153
154 $.fn.extend({
155   animateCss: function(animationName) {
156     var animationEnd = 'animationend oAnimationEnd mozAnimationEnd webkitAnimationEnd';
157     this.addClass(animationName).one(animationEnd, function() {
158       $(this).removeClass(animationName);
159     });
160     return this;
161   }
162 });
163 function runGame() {
164   displayCards(2);
165   displayStars(3);
166   $('.card').click(toggleCard);
167   $('#moves').html("0 Moves");
168 }
169

```

AWESOME

Overall, I loved your project. I appreciate your hard work. Good Luck 🍀

```

170 runGame();
171

```

► README.md 1

► js/easytimer.min.js 1

► css/app.css

► index.html

RETURN TO PATH