

```

1  $(document).ready(function(){
2      $("#rules").modal('show');
3  });
4
5  const card = document.getElementById("card");
6  const status = document.getElementById("status");
7
8  let walletAmount = 1000;
9  let betAmount = 0;
10
11  let winCount = 0;
12  let lossCount = 1;
13
14  let cardsArray = [];
15  let newCardNumber,newSuit,newCard;
16
17  if(!localStorage.getItem("highScore")){
18      localStorage.setItem("highScore", 0);
19  } else {
20  }
21
22
23  function fillCardsArray() {
24      cardsArray[0]= "2C";
25      cardsArray[1]= "2D";
26      cardsArray[2]= "2H";
27      cardsArray[3]= "2S";
28      cardsArray[4]= "3C";
29      cardsArray[5]= "3D";
30      cardsArray[6]= "3H";
31      cardsArray[7]= "3S";
32      cardsArray[8]= "4C";
33      cardsArray[9]= "4D";
34      cardsArray[10]= "4H";
35      cardsArray[11]= "4S";
36      cardsArray[12]= "5C";
37      cardsArray[13]= "5D";
38      cardsArray[14]= "5H";
39      cardsArray[15]= "5S";
40      cardsArray[16]= "6C";
41      cardsArray[17]= "6D";
42      cardsArray[18]= "6H";
43      cardsArray[19]= "6S";
44      cardsArray[20]= "7C";
45      cardsArray[21]= "7D";
46      cardsArray[22]= "7H";
47      cardsArray[23]= "7S";
48      cardsArray[24]= "8C";
49      cardsArray[25]= "8D";
50      cardsArray[26]= "8H";
51      cardsArray[27]= "8S";
52      cardsArray[28]= "9C";
53      cardsArray[29]= "9D";
54      cardsArray[30]= "9H";
55      cardsArray[31]= "9S";
56      cardsArray[32]= "10C";
57      cardsArray[33]= "10D";
58      cardsArray[34]= "10H";
59      cardsArray[35]= "10S";
60      cardsArray[36]= "JC";
61      cardsArray[37]= "JD";
62      cardsArray[38]= "JH";
63      cardsArray[39]= "JS";
64      cardsArray[40]= "QC";
65      cardsArray[41]= "QD";
66      cardsArray[42]= "QH";
67      cardsArray[43]= "QS";
68      cardsArray[44]= "KC";
69      cardsArray[45]= "KD";
70      cardsArray[46]= "KH";
71      cardsArray[47]= "KS";
72      cardsArray[48]= "AC";
73      cardsArray[49]= "AD";
74      cardsArray[50]= "AH";
75      cardsArray[51]= "AS";
76  }
77
78  fillCardsArray();
79  function loadCards() {
80      randomCard = Math.floor(Math.random() * 13 + 2);
81      randomSuit = Math.floor(Math.random() * 4 + 1);
82      randomCardNumber = randomCard;
83      switch (randomCard) {
84          case 14:
85              randomCard = "A";
86              break;
87          case 13:
88              randomCard = "K";
89              break;
90          case 12:
91              randomCard = "Q";
92              break;
93          case 11:

```

CONFIGURE



version 2.13.0

Metrics

There are 14 functions in this file.
Function with the largest signature take 1 arguments, while the median is 0.

Largest function has 52 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 19 while the median is 4.

Blog (/blog)

13 warnings

- 5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 Redefinition of 'status'.
- 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).
- 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).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 15 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 117 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 132 '[highScore]' is better written in dot notation.
- 178 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 246 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

Six undefined variables

- 1 \$
- 2 \$
- 130 \$
- 134 \$
- 135 \$
- 136 \$
- 80 randomCard
- 82 randomCard
- 83 randomCard
- 85 randomCard
- 88 randomCard
- 91 randomCard
- 94 randomCard
- 115 randomCard
- 117 randomCard
- 215 randomCard
- 81 randomSuit
- 99 randomSuit
- 101 randomSuit

```

94     randomCard = "J";
95     break;
96   default:
97     break;
98 }
99 switch (randomSuit) {
100   case 1:
101     randomSuit = "C";
102     break;
103   case 2:
104     randomSuit = "D";
105     break;
106   case 3:
107     randomSuit = "S";
108     break;
109   case 4:
110     randomSuit = "H";
111     break;
112   default:
113     break;
114 }
115 card.style.backgroundImage = "url(assets/img/cards/" + randomCard + randomSuit + ".png";
116
117 let index = cardsArray.indexOf(randomCard + randomSuit);
118 if (index > -1) {
119   cardsArray.splice(index, 1);
120 }
121 console.log(cardsArray);
122 }
123
124 function nCard() {
125
126   if(cardsArray.length > 1){
127   } else {
128     console.log(cardsArray.length);
129
130     $(".buttons").replaceWith("<a href='index.html'><i class='fas fa-sync-alt'></i></a>");
131     if(localStorage.getItem("highScore") < document.getElementById("cash").innerHTML){
132       localStorage["highScore"] = document.getElementById("cash").innerHTML;
133     }
134     $(".bet-amount").replaceWith("High Score: £" + localStorage.getItem("highScore"));
135     $(".next-card").css('visibility', 'hidden');
136     $("#status").html("Game Over");
137     return -1;
138
139   }
140   console.log(cardsArray);
141   while(cardsArray.length > 1){
142     newCard = Math.floor(Math.random() * 13 + 2);
143     newSuit = Math.floor(Math.random() * 4 + 1);
144     newCardNumber = newCard;
145     switch (newCard) {
146       case 14:
147         newCard = "A";
148         break;
149       case 13:
150         newCard = "K";
151         break;
152       case 12:
153         newCard = "Q";
154         break;
155       case 11:
156         newCard = "J";
157         break;
158       default:
159         break;
160     }
161     switch (newSuit) {
162       case 1:
163         newSuit = "C";
164         break;
165       case 2:
166         newSuit = "D";
167         break;
168       case 3:
169         newSuit = "S";
170         break;
171       case 4:
172         newSuit = "H";
173         break;
174       default:
175         break;
176     }
177     if(cardsArray.includes(newCard+newSuit)){
178       let index = cardsArray.indexOf(newCard + newSuit);
179       if (index > -1) {
180         cardsArray.splice(index, 1);
181       }
182       break;
183     }
184   }
185
186   if (option == "h" && newCardNumber > randomCardNumber) {
187     status.innerHTML = "YOU WIN!";
188     document.getElementById("last-bet").innerHTML = betAmount;

```

CONFIGURE



version 2.13.0

Metrics

There are 14 functions in this file.
Function with the largest signature take 1 arguments, while the median is 0.

Largest function has 52 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 19 while the median is 4.

Blog (/blog)

13 warnings

- 5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 Redefinition of 'status'.
- 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).
- 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).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 15 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 117 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 132 ['highScore'] is better written in dot notation.
- 178 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 246 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

Six undefined variables

- 1 \$
- 2 \$
- 130 \$
- 134 \$
- 135 \$
- 136 \$
- 80 randomCard
- 82 randomCard
- 83 randomCard
- 85 randomCard
- 88 randomCard
- 91 randomCard
- 94 randomCard
- 115 randomCard
- 117 randomCard
- 215 randomCard
- 81 randomSuit
- 99 randomSuit
- 101 randomSuit

```

189     updateResult("win");
190     winCount++;
191     clear();
192 } else if (option == "1" && newCardNumber < randomCardNumber) {
193     status.innerHTML = "YOU WIN!";
194     document.getElementById("last-bet").innerHTML = betAmount;
195     updateResult("win");
196     winCount++;
197     clear();
198 } else if (newCardNumber == randomCardNumber) {
199     status.innerHTML = "DRAW";
200     document.getElementById("last-bet").innerHTML = betAmount;
201     updateResult("draw");
202     clear();
203 } else {
204     status.innerHTML = "YOU LOST!";
205     document.getElementById("last-bet").innerHTML = betAmount;
206     updateResult("lose");
207     lossCount++;
208     clear();
209 }
210
211 function clear() {
212
213     console.log(newCard,newSuit);
214     card.style.backgroundImage = "url(assets/img/cards/" + newCard + newSuit + ".png";
215     randomCard = newCard;
216     randomSuit = newSuit;
217     randomCardNumber = newCardNumber;
218 }
219
220 function updateResult(key){
221     if(key == "win"){
222         walletAmount = walletAmount + betAmount;
223         betAmount = 0;
224     }
225     else if(key == "draw"){
226         betAmount = 0;
227     }
228     else if(key == "lose"){
229         walletAmount = walletAmount - betAmount;
230         betAmount = 0;
231     }
232     document.getElementById("cash").innerHTML = walletAmount;
233     document.getElementById("bet").innerHTML = betAmount;
234 }
235 }
236
237 function hiOption() {
238     option = "h";
239     nCard();
240 }
241 function loOption() {
242     option = "l";
243     nCard();
244 }
245 function playSound(){
246     let sound = new Audio("assets/sound/deal.mp3");
247     sound.play();
248 }
249
250 function updateBet(bet){
251     if(walletAmount >= betAmount + bet && 1001 > betAmount + bet){
252         betAmount = betAmount + bet;
253         document.getElementById("bet").innerHTML = betAmount;
254     }
255     else{
256         swal("ERROR!", "The maximum bet is £1000!", "error");
257     }
258 }
259
260 document.getElementsByClassName("repeat-bet")[0].addEventListener("click",function(){
261     updateBet(parseInt(document.getElementById("last-bet").innerHTML));
262 });
263
264 document.getElementsByClassName("bet-twenty")[0].addEventListener("click",function(){
265     updateBet(20);
266 });
267
268 document.getElementsByClassName("bet-fifty")[0].addEventListener("click",function(){
269     updateBet(50);
270 });
271
272 document.getElementsByClassName("bet-hundred")[0].addEventListener("click",function(){
273     updateBet(100);
274 });

```

CONFIGURE



version 2.13.0

Metrics

There are 14 functions in this file.
Function with the largest signature take 1 arguments, while the median is 0.

Largest function has 52 statements in it, while the median is 2.

The most complex function has a cyclomatic complexity value of 19 while the median is 4.

Blog (/blog)

13 warnings

- 5 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 6 Redefinition of 'status'.
- 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).
- 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).
- 14 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 15 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 117 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 132 '[highScore]' is better written in dot notation.
- 178 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
- 246 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

Six undefined variables

- 1 \$
- 2 \$
- 130 \$
- 134 \$
- 135 \$
- 136 \$
- 80 randomCard
- 82 randomCard
- 83 randomCard
- 85 randomCard
- 88 randomCard
- 91 randomCard
- 94 randomCard
- 115 randomCard
- 117 randomCard
- 215 randomCard
- 81 randomSuit
- 99 randomSuit
- 101 randomSuit