

Chapter 3 - Arrays\Monster Match\monstermatch.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Monster Match</title>
5          <style>
6              body
7              {
8                  padding: 15px 0;
9                  border: 10px double rgb(255, 0, 0);
10                 text-align: center;
11                 background: linear-gradient(90deg, rgb(0, 238, 255) 0%, rgb(76, 0, 255) 100%)
12                 ,
13                 rgb(0, 26, 255) 44%;
14                 font-family: Verdana, Geneva, Tahoma, sans-serif;
15                 color: rgb(1, 37, 0)
16             }
17             table
18             {
19                 margin-left: auto;
20                 margin-right: auto;
21             }
22         </style>
23         <script>
24             let p1Deck = []
25             let p2Deck = []
26             let deckIndex1
27             let deckIndex2
28             let curCardP1
29             let curCardP2
30             let deckCards = 8
31             let imageArray = ["monster1.svg", "monster2.svg", "monster3.svg", "monster4.svg"]
32             let gameState = "Game Over"
33             let score = 0
34
35             // Description: Return a random integer, n, such that min <= n <= max
36             // Citation: Mozilla Foundation
37             // https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random
38             // Input/Parameters: Two floating point numbers, min and max
39             // Output/Return: Returns a random integer between min and max
40             function randomInt(min, max)
41             {
42                 min = Math.ceil(min);
43                 max = Math.floor(max);
44
45                 return Math.floor(Math.random() * (max - min + 1)) + min;
46             }
47             // Description: Takes a list of images and returns a random one
48             // Citations: None
49             // Input/Parameters: List of images
50             // Output/Return: Random image from the list and displays
51             function formatImage(inputArray, index)
52             {
53                 let fileName = inputArray[index];

```

```

53
54         let output = "<img src='Monster Pics/' + fileName + " ' width=200 height=300>
";
55
56         return output;
57     }
58     // Description: Displays a set of instructions when the button is clicked
59     // Citation: None
60     // Input/Parameters: The instructions
61     // Output/Return: The instructions
62     function viewInstructions()
63     {
64         divIdinstruct.innerHTML = "Full Instructions:" + "<br>" +
65         "Each player has a deck of monster cards. Each round, a " + "<br>" +
66         "player will reveal a random card from their deck. If the " + "<br>" +
67         "cards match, it is a 'Monster Match.' The fastest player " + "<br>" +
68         "can claim the other player's card (it is removed from the " + "<br>" +
69         "deck of the slower player's deck and added to the faster " + "<br>" +
70         "player's deck). Player 1 can claim a match by pressing 'a' " + "<br>" +
71         "before Player 2 can claim the match by pressing 'l'. If a " + "<br>" +
72         "player presses 'a' or 'l' when there is not a match, they " + "<br>" +
73         "will give up their card. If the cards don't match, press 'n' " + "<br>" +
74         "to start the next round, When a player has claimed all the " + "<br>" +
75         "other player's cards, the player wins the game."
76     }
77     // Description: Displays score after every round
78     // Citation: None
79     // Input/Parameters: The length of each player's deck
80     // Output/Return: The number of cards each player has remaining
81     function displayScore()
82     {
83         p1CountId.innerHTML = "Cards Left: " + p1Deck.length;
84         p2CountId.innerHTML = "Cards Left: " + p2Deck.length;
85     }
86     // Description: Picks random index in each player's deck, takes random image
using that index and displays
87     // Citation: None
88     // Input/Parameters: List of images with unique indices
89     // Output/Return: Random image in each player's decl
90     function dealNewCards()
91     {
92         deckIndex1 = randomInt(0, p1Deck.length - 1);
93         deckIndex2 = randomInt(0, p2Deck.length - 1);
94
95         curCardP1 = formatImage(p1Deck, deckIndex1);
96         curCardP2 = formatImage(p2Deck, deckIndex2);
97
98         image1Id.innerHTML = curCardP1
99         image2Id.innerHTML = curCardP2
100
101         gameState = "Play"
102
103         divIdinstruct.innerHTML = "Player 1 press 'a' if matching. Player 2 press 'l'
if matching"
104     }
105     // Description: Starts a new game when the start new game button is clicked,
resets score from previous game

```

```

106         // Citation: None
107         // Input/Parameters: A random index from the image array and pushes it into each
player's deck twice
108         // Output/Return: Resets the score and deals new cards
109         function newGame()
110         {
111             p1Deck = []
112             p2Deck = []
113
114             for (i=0; i<=3; i++)
115             {
116                 p1Deck.push(imageArray[i])
117                 p1Deck.push(imageArray[i])
118                 p2Deck.push(imageArray[i])
119                 p2Deck.push(imageArray[i])
120             }
121
122             player1Id.innerHTML = "Player 1"
123             player2Id.innerHTML = "Player 2"
124
125             dealNewCards();
126             displayScore();
127
128             divIdmessages.innerHTML = ""
129         }
130         // Description: Displays winning message once a player steals the other player's
cards
131         // Citation: None
132         // Input/Parameters: Winning message if player 1 wins and vice versa
133         // Output/Return: Displays winning message
134         function winGame()
135         {
136             if (p1Deck.length == 0)
137             {
138                 divIdmessages.innerHTML = "Player 2 has won! Press 'Start New Game' to
play again"
139             }
140             else if (p2Deck.length == 0)
141             {
142                 divIdmessages.innerHTML = "Player 1 has won! Press 'Start New Game' to
play again"
143             }
144             gameState = "Game Over"
145         }
146         // Description: Checks if the cards match based on the key pressed then moves a
card to the other player based on who won the match
147         // Citation: None
148         // Input/Parameters: Keys such as 'a', 'l', and 'n'. Checks if the two cards
displayed are the same or different.
149         // Output/Return: A message based on who won the match, deals new cards, and
changes the score.
150         function checkMatch(event)
151         {
152             pressKey = event.key
153             if (gameState == "Play")
154             {
155                 if (pressKey == "a")
156                 {

```

```
157         if (curCardP1 == curCardP2)
158         {
159             p1Deck.push(p2Deck[deckIndex2])
160             p2Deck.splice(deckIndex2,1)
161             divIdmessages.innerHTML = "Congrats Player 1 has claimed the
match! Press 'n' to deal more cards."
162         }
163         else
164         {
165             p2Deck.push(p1Deck[deckIndex1])
166             p1Deck.splice(deckIndex1,1)
167             divIdmessages.innerHTML = "Player 1 has misfired! Press 'n' to
deal more cards."
168         }
169         gameState = "Pause"
170         displayScore();
171         winGame();
172         dealNewCards();
173     }
174     else if (pressKey == "l")
175     {
176         if (curCardP1 == curCardP2)
177         {
178             p2Deck.push(p1Deck[deckIndex1])
179             p1Deck.splice(deckIndex1,1)
180             divIdmessages.innerHTML = "Congrats Player 2 has claimed the
match! Press 'n' to deal more cards."
181         }
182         else
183         {
184             p1Deck.push(p2Deck[deckIndex2])
185             p2Deck.splice(deckIndex2,1)
186             divIdmessages.innerHTML = "Player 2 has misfired! Press 'n' to
deal more cards."
187         }
188         gameState = "Pause"
189         displayScore();
190         winGame();
191         dealNewCards();
192     }
193     else if (pressKey == "n")
194     {
195         if (curCardP1 == curCardP2)
196         {
197             divIdmessages.innerHTML = "There is a match on the board, someone
must claim it!"
198             gameState = "Play"
199         }
200         else
201         {
202             dealNewCards();
203             gameState = "Play"
204         }
205     }
206     else
207     {
208         divIdmessages.innerHTML = "Oops, you clicked an invalid key. Player 1
please press 'a' if matching and Player 2 please press 'l' if matching otherwise press 'n' to
```

```
209         deal more cards."
210     }
211     else if (gameState == "Pause")
212     {
213         if (pressKey == "n")
214         {
215             gameState = "Play"
216             dealNewCards();
217         }
218     }
219 }
220 </script>
221 </head>
222 <body onkeyup="checkMatch(event)">
223     <h1>Monster Match</h1>
224     <input type="button" value="Start New Game" onClick="newGame()">
225     <input type="button" value="View Game Instructions" onClick="viewInstructions()">
226     <table>
227         <tr>
228             <th><span id="player1Id"></span></th>
229             <th><span id="player2Id"></span></th>
230         </tr>
231         <tr>
232             <th><span id="image1Id"></span></th>
233             <th><span id="image2Id"></span></th>
234         </tr>
235         <tr>
236             <th><span id="p1CountId"></span></th>
237             <th><span id="p2CountId"></span></th>
238         </tr>
239     </table>
240     <div id="divIdmessages"></div>
241     <div id="divIdinstruct"></div>
242     <div id="divIdcurScore"></div>
243 </body>
244 </html>
```