

CS-1810 Final

Understanding Web Development basics

CSS That I'm Proud Of

```
195 .playerform {  
196     display: flex;  
197     flex-direction: column;  
198     align-items: center;  
199     justify-content: center;  
200     max-width: 30%;  
201     margin: 0 auto;  
202     color: rgb(141, 141, 69);  
203 }  
204 .buttonstyle{  
205 width: 30%;  
206 }  
207
```

I added this one solely because it took me a while to get it to look decent.

And more

My cards turned out very well and of course it's about how it looks.

```
9
10 .card {
11     width: 120px;
12     height: 180px;
13     border: 1px solid #ccc;
14     border-radius: 8px;
15     padding: 10px;
16     margin: 5px;
17     display: inline-block;
18     text-align: center;
19     box-shadow: 0 4px 8px rgba(128, 94, 94, 0.1);
20     background-color: #171717;
21     opacity: 1;
22     transition: transform 0.3s ease-in-out;
23 }
24
25 .card:hover {
26     opacity: 60%;
27     transform: scale(1.2);
28 }
29
30 .card p {
31     margin: 0;
32     font-size: 16px;
33 }
34
```


And More

```
<div class="playerform">  
  <form id="addPlayerForm">  
    <label for="playerName">Add Player:</label>  
    <input type="text" id="playerName" name="playerName" class="buttonstyle" required>  
    <button type="submit" class="buttonstyle">Add</button>  
  </form>  
  <button type="reset" id="reset-players">Reset Players</button>  
</div>  
</body>
```

Being able to establish the form
and put others by it in the
javascript.

Java Script

```
38 function sourcedrop(event) {
39     event.preventDefault()
40     // console.log("drop")
41     const cardId = event.dataTransfer.getData("text/plain");
42     // console.log(cardId)
43     const draggedCard = document.getElementById(cardId);
44     if (!draggedCard) return;
45     // Handle dropping the card into the player containers (player1Container, player2Container)
46     const gameboard = document.getElementById("gameboard")
47     //if deck has 1 card if same check if card in battle deck is the same player as dropping card
48     if (canDrop(cardId, battlCards)) {
49         gameboard.appendChild(draggedCard);
50         battlCards.push(cardId)
51         removeCardFromDeck(cardId)
52         if (battlCards.length >= 2) {
53             let winner = compareBattleDeck(battlCards)
54             console.log(winner);
55             addToWinner(battlCards, getOwner(winner))
56             battlCards = []
57             gameboard.innerHTML = ""
58             // console.log(ui.updateScore(deck1,deck2))
59         }
60         let score1 = p1deck.length
61         let score2 = p2deck.length
62         ui.updateScore(score1, score2)
63         ui.updatePlayerNames()
64     }
65     event.currentTarget.classList.remove("dragging");
66     ui.displayCards([p1deck, p2deck])
67     makeCardsDraggable();
68     updateScoresAndCheckWinner(p1deck, p2deck)
```

This drop for the drag and drop just feels like it works so well.

And More

Saving and keeping my players in local storage worked smoothly and helped me later on

```
---
103 function addToPlayers(playerName) {
104     if (PlayersPlayingPlayer1.length === 0) {
105         PlayersPlayingPlayer1.push(playerName);
106         localStorage.setItem("PlayersPlayingPlayer1", JSON.stringify(PlayersPlayingPlayer1));
107     }
108     else if (!PlayersPlayingPlayer1.includes(playerName) && PlayersPlayingPlayer2.length === 0) {
109         PlayersPlayingPlayer2.push(playerName);
110         localStorage.setItem("PlayersPlayingPlayer2", JSON.stringify(PlayersPlayingPlayer2));
111     }
112 }
113
114 document.getElementById('reset-players').addEventListener('click', function () {
115     reset();
116 });
117
118 function reset() {
119     PlayersPlayingPlayer1 = []
120     PlayersPlayingPlayer2 = []
121     localStorage.removeItem("PlayersPlayingPlayer1")
122     localStorage.removeItem("PlayersPlayingPlayer2")
123 }
```

One More

Just being able to add the players to a different javascript file felt really accomplishing to me.

```
91 export function updatePlayerNames() {
92     const playerNames = getPlayerNames();
93
94     let p1 = document.getElementById("p1name");
95     let p2 = document.getElementById("p2name");
96
97     if (playerNames.playersPlayingPlayer1 !== null) {
98         p1.innerHTML = `<td>${playerNames.playersPlayingPlayer1} </td>`;
99     }
100    if (playerNames.playersPlayingPlayer2 !== null) {
101        p2.innerHTML = `<td>${playerNames.playersPlayingPlayer2} </td>`;
102    }
103 }
```


Time for Hard things I implemented:

First one is keeping track of my drag drop what was being added and what was being removed.

Next was getting my player names from registered players into the playgame page to be uploaded.

Finally implementing the API

Summary

- Being Organized
- Having multiple files
- Having a clear plan
- Doing something that's interesting to me