BLACKJACK Game - Base

Gameplay Description

The main function runs on each player's turn. The sequence of actions in the game might be the following.

1. Deck is shuffled.
2. User clicks Submit to deal cards.
3. The cards are analysed for game winning conditions, e.g. Blackjack.
4. The cards are displayed to the user.
5. The user decides whether to hit or stand, using the submit button to submit their choice.
6. The user's cards are analysed for winning or losing conditions.
7. The computer decides to hit or stand automatically based on game rules.
8. The game either ends or continues.

**GLOBAL VARIABLES**

playerCards

computerCards

myoutputvalue

**HELPER FUNCTIONS**

1. Make a Deck
   1. Empty array to store cards
   2. Properties of cards(object) = suits, cardName, rankCounter
   3. 1st while loop to increase Suits, 2nd while loop to increase cardName and rankCounter
2. Shuffling of Cards (Deck)
   1. Randomizing function
   2. Card shuffling – using randomizing function with makeDeck

Don’t UDS and cant ‘imagine’:

**IN shuffleCards()**

var randomCard = cardDeck[randomIndex];

var currentCard = cardDeck[currentIndex];

//Swap positions of randomCard and currentCard in the deck

cardDeck[currentIndex] = randomCard;

cardDeck[randomIndex] = currentCard;

1. Make a shuffled Deck
2. Dealing 1 card to Computer and 1 card to player then repeat.

Use **alert(**prompt message for errors/whatever instructions)

What is arrow function?

GAMEFLOW

1. Gamestate = start = what return?
2. Else if gamestate = HitStand = ?
3. var natural = function () {
4. aceConditions(playerHand, computerHand);
5. calculateScores();
6. for (var i = 0; i < playerHand.length || i < computerHand.length; i++) {
7. if (
8. (playerHand[i].name == "Ace" && playerTotalPoints >= 21) ||
9. (computerHand[i].name == "Ace" && computerTotalPoints >= 21)
10. ) {
11. myOutputValue = `NATURAL`;
12. break;
13. }
14. }
15. };

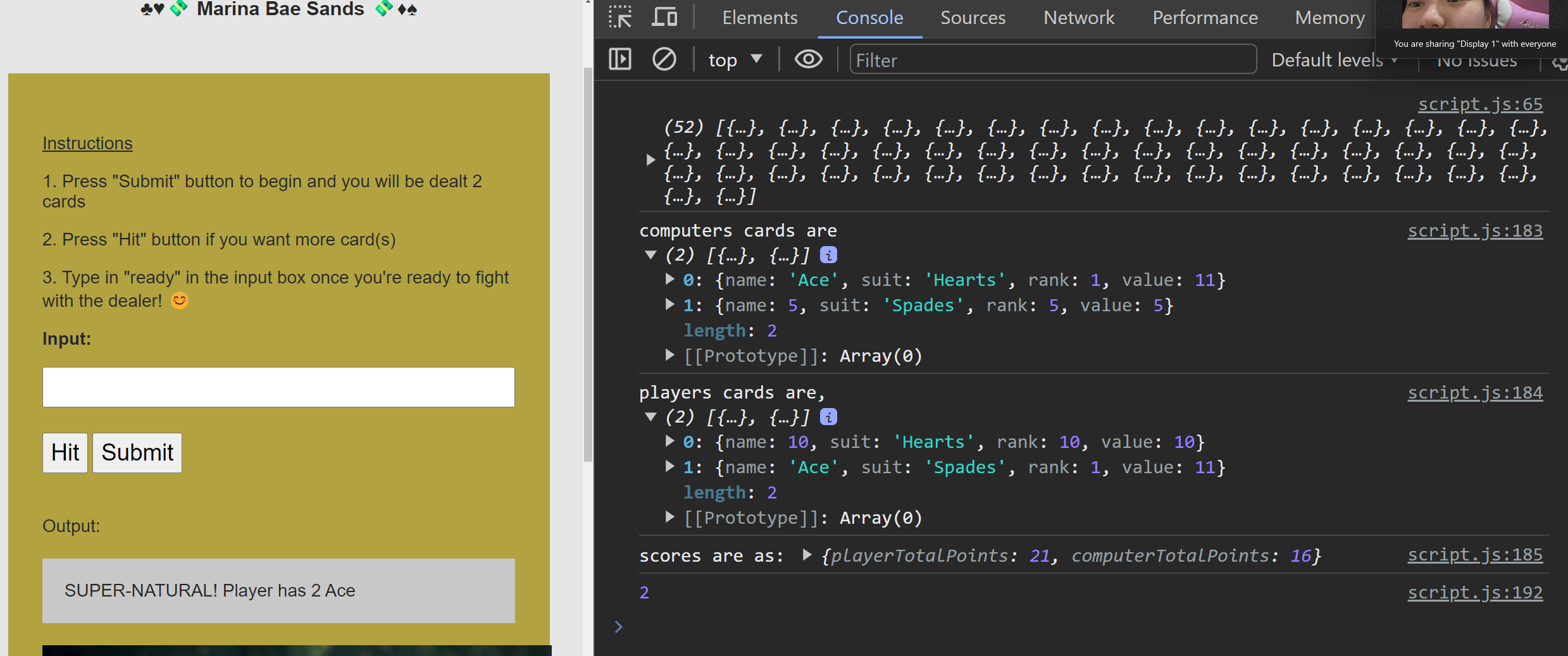
BUGS

1. Comp cards Ace value is 11 when holding 3 cards
2. Show player cards despite compcards = natural (vice versa)
3. For below; aceCondition triggered, enters for loop > 1st IF statement, ace assigned to value of 11 when hand is 2. DOES NOT REENTER LOOP WHEN HIT thereby no reassignment

**0**: {name: 'Ace', suit: 'Clubs', rank: 1, value: 11}

**1**: {name: 4, suit: 'Hearts', rank: 4, value: 4}

**2**: {name: 5, suit: 'Hearts', rank: 5, value: 5}

1. 

MORE COMFORTABLE

1. Conditions for chase-dragon scenario, 5 cards under 21 points