PROMINEO TECH

# Intro to JavaScript Week 6 Coding Assignment

**Points possible:** 70

| Category | Criteria | % of Grade |
|---|---|---|
| **Functionality** | Does the code work? | 25 |
| **Organization** | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| **Creativity** | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| **Completeness** | All requirements of the assignment are complete. | 25 |

**Instructions:** In Visual Studio Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

**Coding Steps:**

For the final project you will be creating an automated version of the classic card game *WAR*. You do not need to accept any user input, when you run your code, the entire game should play out instantly without any user input.

There are many versions of the game *WAR,* but in this version there are only 2 players and you don't need to do anything special when there is a tie on a round.

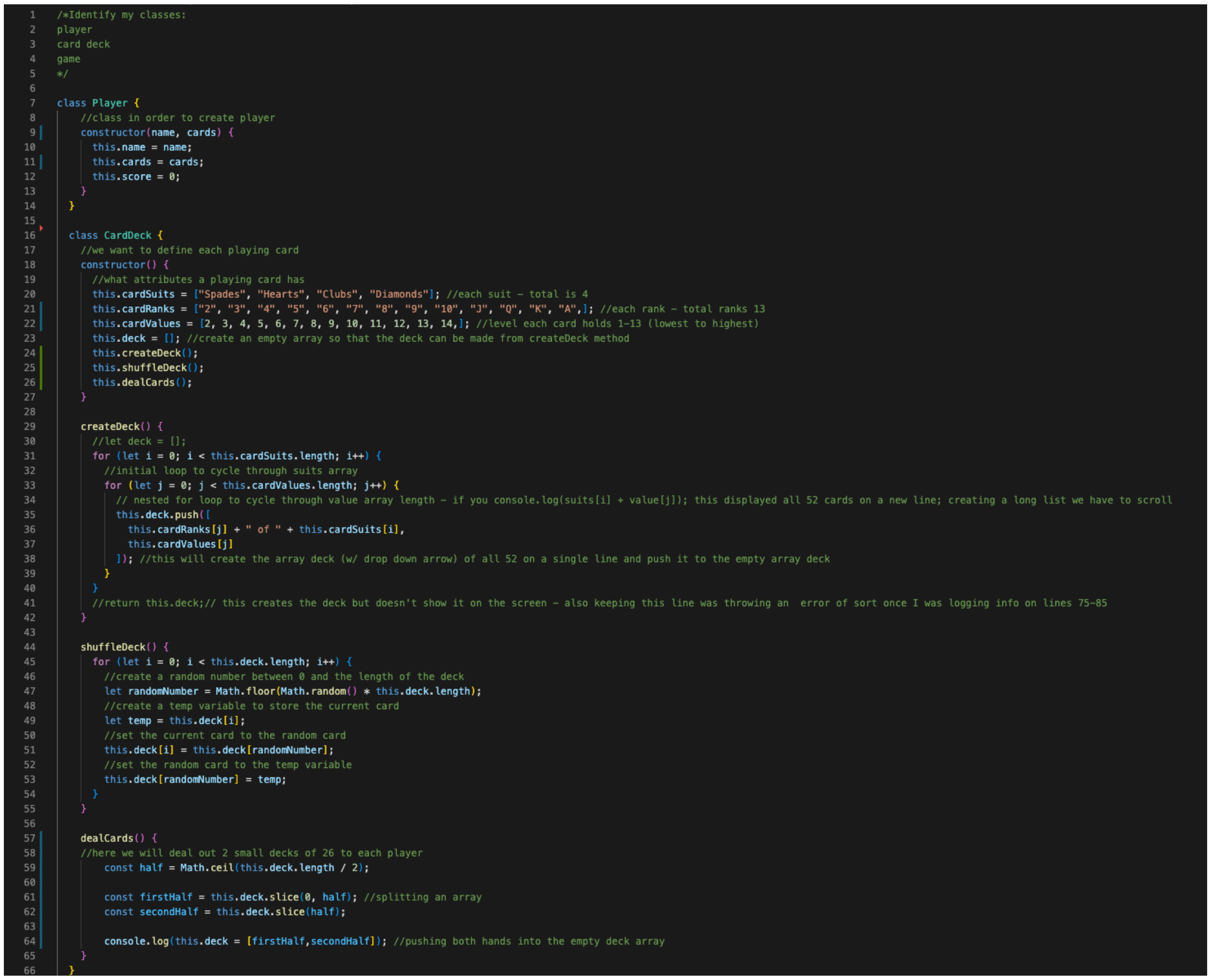
Think about how you would build this project and write your plan down. Consider classes such as Card, Deck, and Player and what fields and methods they might each have. You can implement the game however you'd like (i.e. printing to the console, using alert, or some other way). The completed project should, when run, do the following:

- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card

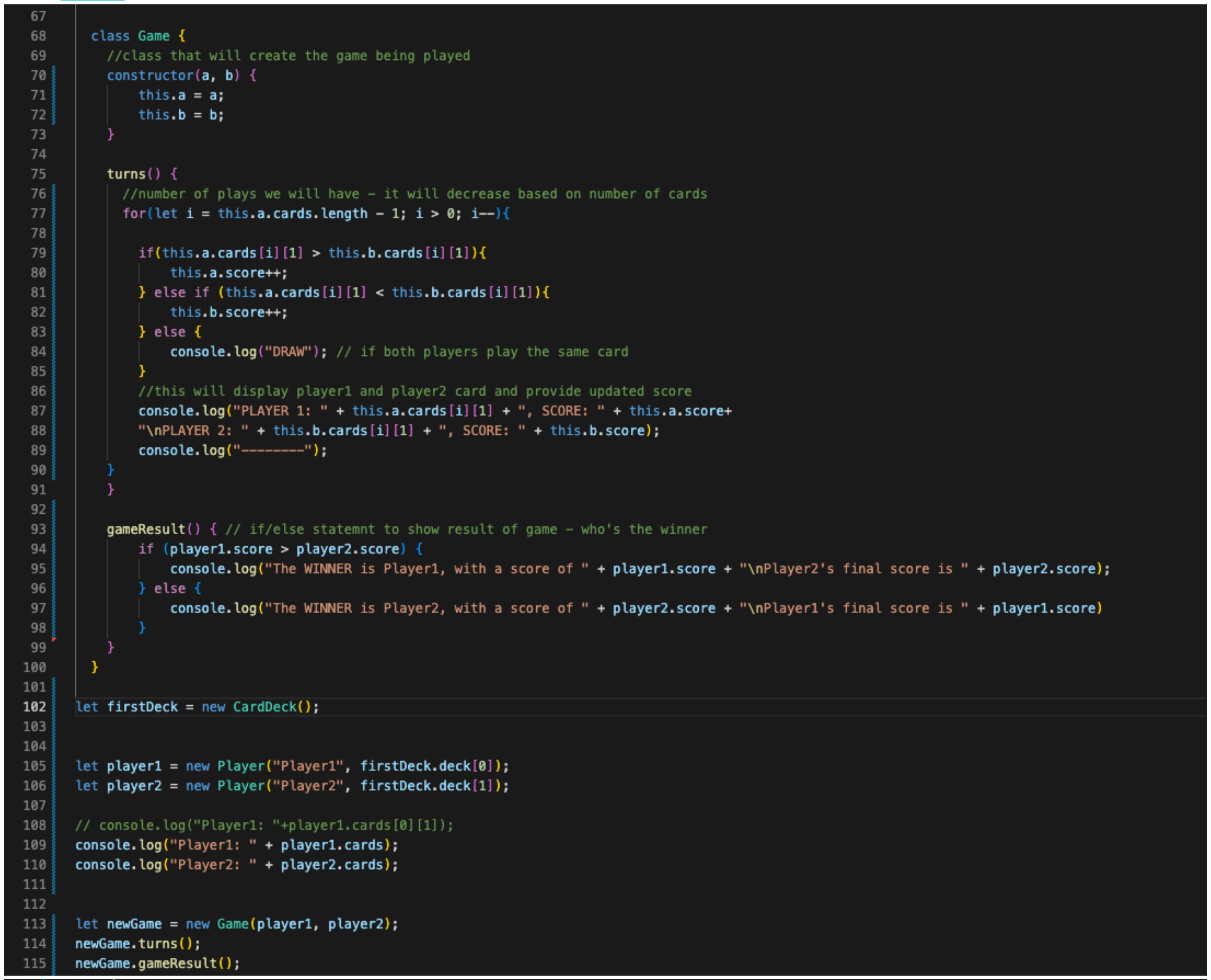
- The Player who played the higher card is awarded a point
    - Ties result in zero points for both Players
- After all cards have been played, display the score and declare the winner.

Write a Unit Test using Mocha and Chai for at least one of the functions you write.

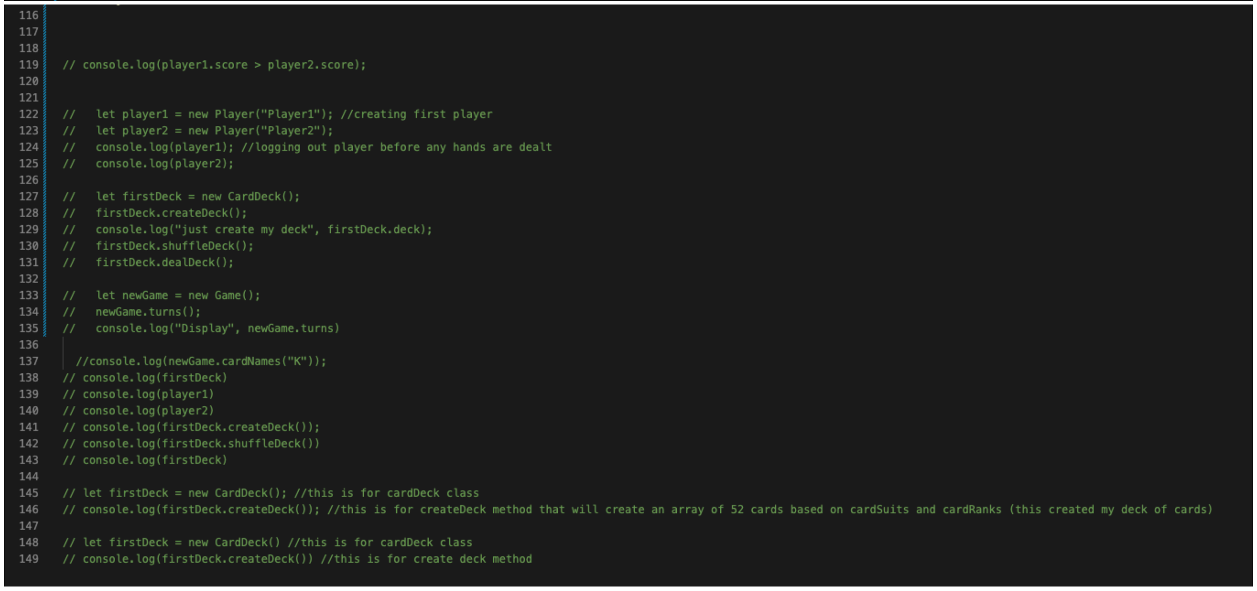
**Screenshots of Code:**



```
/*Identify my classes:
player
card deck
game
*/

class Player {
    //class in order to create player
    constructor(name, cards) {
      this.name = name;
      this.cards = cards;
      this.score = 0;
    }
  }

  class CardDeck {
    //we want to define each playing card
    constructor() {
      //what attributes a playing card has
      this.cardSuits = ["Spades", "Hearts", "Clubs", "Diamonds"]; //each suit - total is 4
      this.cardRanks = ["2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A",]; //each rank - total ranks 13
      this.cardValues = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,]; //level each card holds 1-13 (lowest to highest)
      this.deck = []; //create an empty array so that the deck can be made from createDeck method
      this.createDeck();
      this.shuffleDeck();
      this.dealCards();
    }

    createDeck() {
      //let deck = [];
      for (let i = 0; i < this.cardSuits.length; i++) {
        //initial loop to cycle through suits array
        for (let j = 0; j < this.cardValues.length; j++) {
          // nested for loop to cycle through value array length - if you console.log(suits[i] + value[j]); this displayed all 52 cards on a new line; creating a long list we have to scroll
          this.deck.push([
            this.cardRanks[j] + " of " + this.cardSuits[i],
            this.cardValues[j]
          ]); //this will create the array deck (w/ drop down arrow) of all 52 on a single line and push it to the empty array deck
        }
      }
      //return this.deck;// this creates the deck but doesn't show it on the screen - also keeping this line was throwing an  error of sort once I was logging info on lines 75-85
    }

    shuffleDeck() {
      for (let i = 0; i < this.deck.length; i++) {
        //create a random number between 0 and the length of the deck
        let randomNumber = Math.floor(Math.random() * this.deck.length);
        //create a temp variable to store the current card
        let temp = this.deck[i];
        //set the current card to the random card
        this.deck[i] = this.deck[randomNumber];
        //set the random card to the temp variable
        this.deck[randomNumber] = temp;
      }
    }

    dealCards() {
    //here we will deal out 2 small decks of 26 to each player
        const half = Math.ceil(this.deck.length / 2);

        const firstHalf = this.deck.slice(0, half); //splitting an array
        const secondHalf = this.deck.slice(half);

        console.log(this.deck = [firstHalf,secondHalf]); //pushing both hands into the empty deck array
    }
  }
```

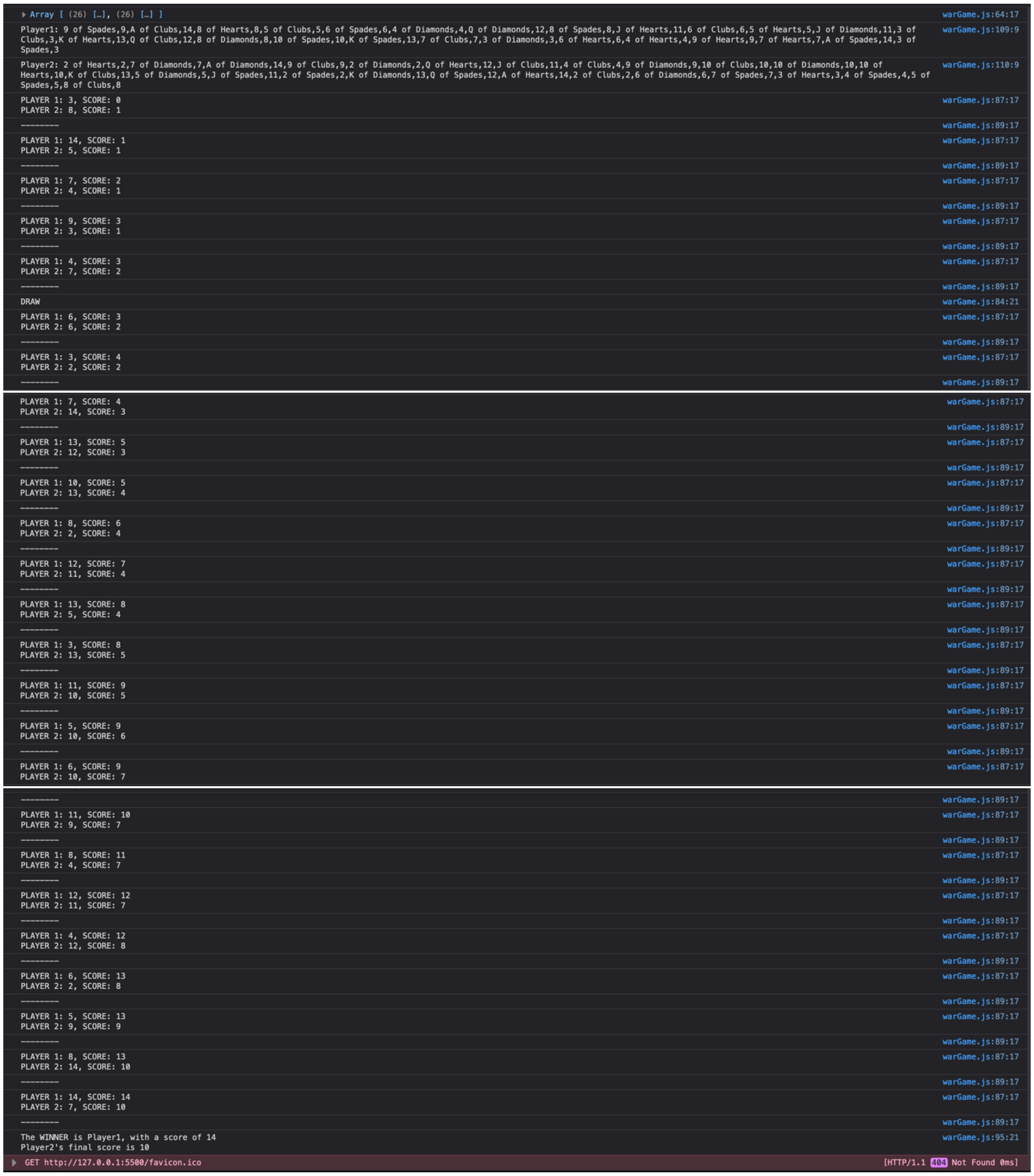


```
  class Game {
    //class that will create the game being played
    constructor(a, b) {
        this.a = a;
        this.b = b;
    }

    turns() {
      //number of plays we will have - it will decrease based on number of cards
      for(let i = this.a.cards.length - 1; i > 0; i--){

        if(this.a.cards[i][1] > this.b.cards[i][1]){
            this.a.score++;
        } else if (this.a.cards[i][1] < this.b.cards[i][1]){
            this.b.score++;
        } else {
            console.log("DRAW"); // if both players play the same card
        }
        //this will display player1 and player2 card and provide updated score
        console.log("PLAYER 1: " + this.a.cards[i][1] + ", SCORE: " + this.a.score+
        "\nPLAYER 2: " + this.b.cards[i][1] + ", SCORE: " + this.b.score);
        console.log("--------");
    }
    }

    gameResult() { // if/else statmnt to show result of game - who's the winner
        if (player1.score > player2.score) {
            console.log("The WINNER is Player1, with a score of " + player1.score + "\nPlayer2's final score is " + player2.score);
        } else {
            console.log("The WINNER is Player2, with a score of " + player2.score + "\nPlayer1's final score is " + player1.score)
        }
    }
  }

let firstDeck = new CardDeck();


let player1 = new Player("Player1", firstDeck.deck[0]);
let player2 = new Player("Player2", firstDeck.deck[1]);

// console.log("Player1: "+player1.cards[0][1]);
console.log("Player1: " + player1.cards);
console.log("Player2: " + player2.cards);


let newGame = new Game(player1, player2);
newGame.turns();
newGame.gameResult();
```



```



// console.log(player1.score > player2.score);


//   let player1 = new Player("Player1"); //creating first player
//   let player2 = new Player("Player2");
//   console.log(player1); //logging out player before any hands are dealt
//   console.log(player2);

//   let firstDeck = new CardDeck();
//   firstDeck.createDeck();
//   console.log("just create my deck", firstDeck.deck);
//   firstDeck.shuffleDeck();
//   firstDeck.dealDeck();

//   let newGame = new Game();
//   newGame.turns();
//   console.log("Display", newGame.turns)

  //console.log(newGame.cardNames("K"));
// console.log(firstDeck)
// console.log(player1)
// console.log(player2)
// console.log(firstDeck.createDeck());
// console.log(firstDeck.shuffleDeck())
// console.log(firstDeck)

// let firstDeck = new CardDeck(); //this is for cardDeck class
// console.log(firstDeck.createDeck()); //this is for createDeck method that will create an array of 52 cards based on cardSuits and cardRanks (this created my deck of cards)

// let firstDeck = new CardDeck() //this is for cardDeck class
// console.log(firstDeck.createDeck()) //this is for create deck method
```

**Screenshots of Running Application:**



```
▸ Array [ (26) […], (26) […] ]                                                                warGame.js:64:17
Player1: 9 of Spades,9,A of Clubs,14,8 of Hearts,8,5 of Clubs,5,6 of Spades,6,4 of Diamonds,4,Q of Diamonds,12,8 of Spades,8,J of Hearts,11,6 of Clubs,6,5 of Hearts,5,J of Diamonds,11,3 of Clubs,3,K of Hearts,13,Q of Clubs,12,8 of Diamonds,8,10 of Spades,10,K of Spades,13,7 of Clubs,7,3 of Diamonds,3,6 of Hearts,6,4 of Hearts,4,9 of Hearts,9,7 of Hearts,7,A of Spades,14,3 of Spades,3     warGame.js:109:9
Player2: 2 of Hearts,2,7 of Diamonds,7,A of Diamonds,14,9 of Clubs,9,2 of Diamonds,2,Q of Hearts,12,J of Clubs,11,4 of Clubs,4,9 of Diamonds,9,10 of Clubs,10,10 of Diamonds,10,10 of Hearts,10,K of Clubs,13,5 of Diamonds,5,J of Spades,11,2 of Spades,2,K of Diamonds,13,Q of Spades,12,A of Hearts,14,2 of Clubs,2,6 of Diamonds,6,7 of Spades,7,3 of Hearts,3,4 of Spades,4,5 of Spades,5,8 of Clubs,8     warGame.js:110:9
PLAYER 1: 3, SCORE: 0                                                                         warGame.js:87:17
PLAYER 2: 8, SCORE: 1
--------                                                                                      warGame.js:89:17
PLAYER 1: 14, SCORE: 1                                                                        warGame.js:87:17
PLAYER 2: 5, SCORE: 1
--------                                                                                      warGame.js:89:17
PLAYER 1: 7, SCORE: 2                                                                         warGame.js:87:17
PLAYER 2: 4, SCORE: 1
--------                                                                                      warGame.js:89:17
PLAYER 1: 9, SCORE: 3                                                                         warGame.js:87:17
PLAYER 2: 3, SCORE: 1
--------                                                                                      warGame.js:89:17
PLAYER 1: 4, SCORE: 3                                                                         warGame.js:87:17
PLAYER 2: 7, SCORE: 2
--------                                                                                      warGame.js:89:17
DRAW                                                                                          warGame.js:84:21
PLAYER 1: 6, SCORE: 3                                                                         warGame.js:87:17
PLAYER 2: 6, SCORE: 2
--------                                                                                      warGame.js:89:17
PLAYER 1: 3, SCORE: 4                                                                         warGame.js:87:17
PLAYER 2: 2, SCORE: 2
--------                                                                                      warGame.js:89:17
PLAYER 1: 7, SCORE: 4                                                                         warGame.js:87:17
PLAYER 2: 14, SCORE: 3
--------                                                                                      warGame.js:89:17
PLAYER 1: 13, SCORE: 5                                                                        warGame.js:87:17
PLAYER 2: 12, SCORE: 3
--------                                                                                      warGame.js:89:17
PLAYER 1: 10, SCORE: 5                                                                        warGame.js:87:17
PLAYER 2: 13, SCORE: 4
--------                                                                                      warGame.js:89:17
PLAYER 1: 8, SCORE: 6                                                                         warGame.js:87:17
PLAYER 2: 2, SCORE: 4
--------                                                                                      warGame.js:89:17
PLAYER 1: 12, SCORE: 7                                                                        warGame.js:87:17
PLAYER 2: 11, SCORE: 4
--------                                                                                      warGame.js:89:17
PLAYER 1: 13, SCORE: 8                                                                        warGame.js:87:17
PLAYER 2: 5, SCORE: 4
--------                                                                                      warGame.js:89:17
PLAYER 1: 3, SCORE: 8                                                                         warGame.js:87:17
PLAYER 2: 13, SCORE: 5
--------                                                                                      warGame.js:89:17
PLAYER 1: 11, SCORE: 9                                                                        warGame.js:87:17
PLAYER 2: 10, SCORE: 5
--------                                                                                      warGame.js:89:17
PLAYER 1: 5, SCORE: 9                                                                         warGame.js:87:17
PLAYER 2: 10, SCORE: 6
--------                                                                                      warGame.js:89:17
PLAYER 1: 6, SCORE: 9                                                                         warGame.js:87:17
PLAYER 2: 10, SCORE: 7
--------                                                                                      warGame.js:89:17
PLAYER 1: 11, SCORE: 10                                                                       warGame.js:87:17
PLAYER 2: 9, SCORE: 7
--------                                                                                      warGame.js:89:17
PLAYER 1: 8, SCORE: 11                                                                        warGame.js:87:17
PLAYER 2: 4, SCORE: 7
--------                                                                                      warGame.js:89:17
PLAYER 1: 12, SCORE: 12                                                                       warGame.js:87:17
PLAYER 2: 11, SCORE: 7
--------                                                                                      warGame.js:89:17
PLAYER 1: 4, SCORE: 12                                                                        warGame.js:87:17
PLAYER 2: 12, SCORE: 8
--------                                                                                      warGame.js:89:17
PLAYER 1: 6, SCORE: 13                                                                        warGame.js:87:17
PLAYER 2: 2, SCORE: 8
--------                                                                                      warGame.js:89:17
PLAYER 1: 5, SCORE: 13                                                                        warGame.js:87:17
PLAYER 2: 9, SCORE: 9
--------                                                                                      warGame.js:89:17
PLAYER 1: 8, SCORE: 13                                                                        warGame.js:87:17
PLAYER 2: 14, SCORE: 10
--------                                                                                      warGame.js:89:17
PLAYER 1: 14, SCORE: 14                                                                       warGame.js:87:17
PLAYER 2: 7, SCORE: 10
--------                                                                                      warGame.js:89:17
The WINNER is Player1, with a score of 14                                                     warGame.js:95:21
Player2's final score is 10
▸ GET http://127.0.0.1:5500/favicon.ico                                  [HTTP/1.1 404 Not Found 0ms]
```

**URL to GitHub Repository:**

[https://github.com/Denaag/Week-6/blob/main/warGame.js](https://github.com/Denaag/Week-6/blob/main/warGame.js)