Intro to JavaScript Week 6 Coding Assignment

Points possible: 100

URL to GitHub Repository: https://github.com/KatteNoel/Week-6-Coding-Assignment

URL to Your Coding Assignment Video: https://youtu.be/QkoHIuV7xe0

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.

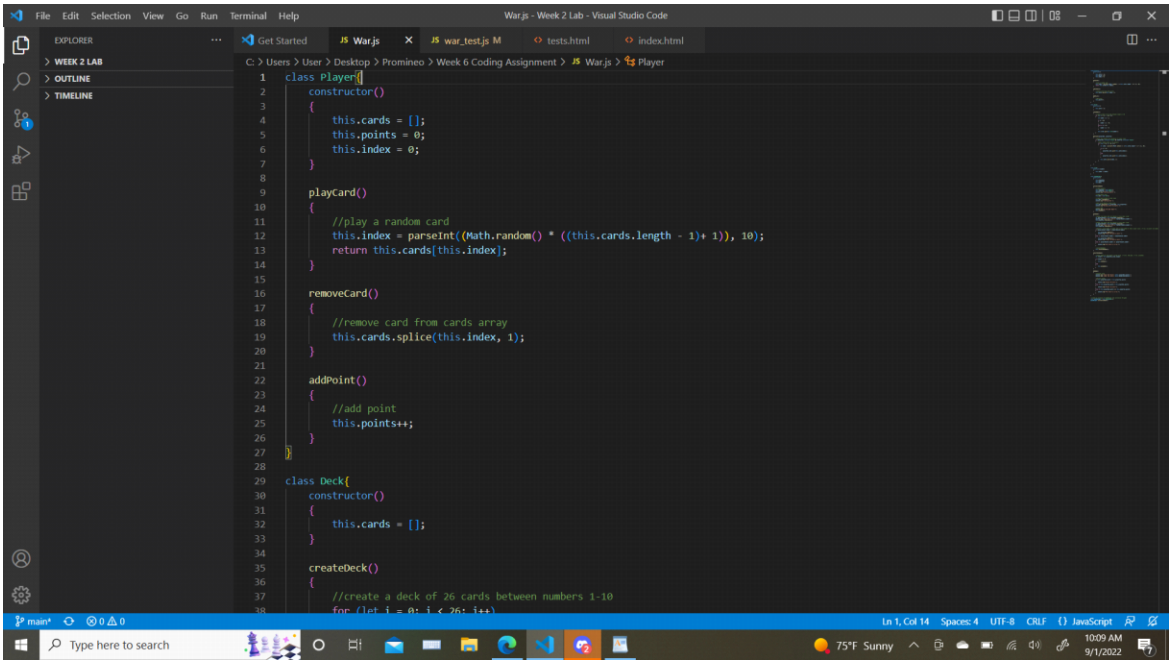
There are many versions of the game *WAR*, but in this version there are only 2 players and you don't need to do anything special when there is a tie on a round.

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:

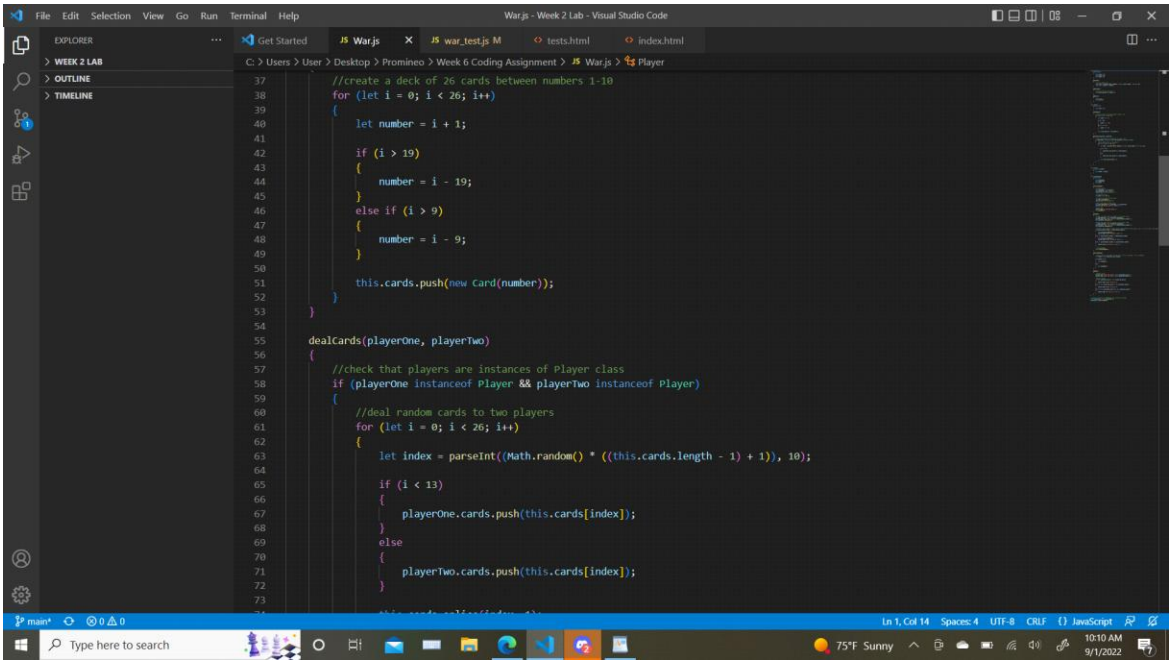
- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card
- The Player who played the higher card is awarded a point
    - Ties result in zero points for both Players
- After all cards have been played, display the score and declare the winner.

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

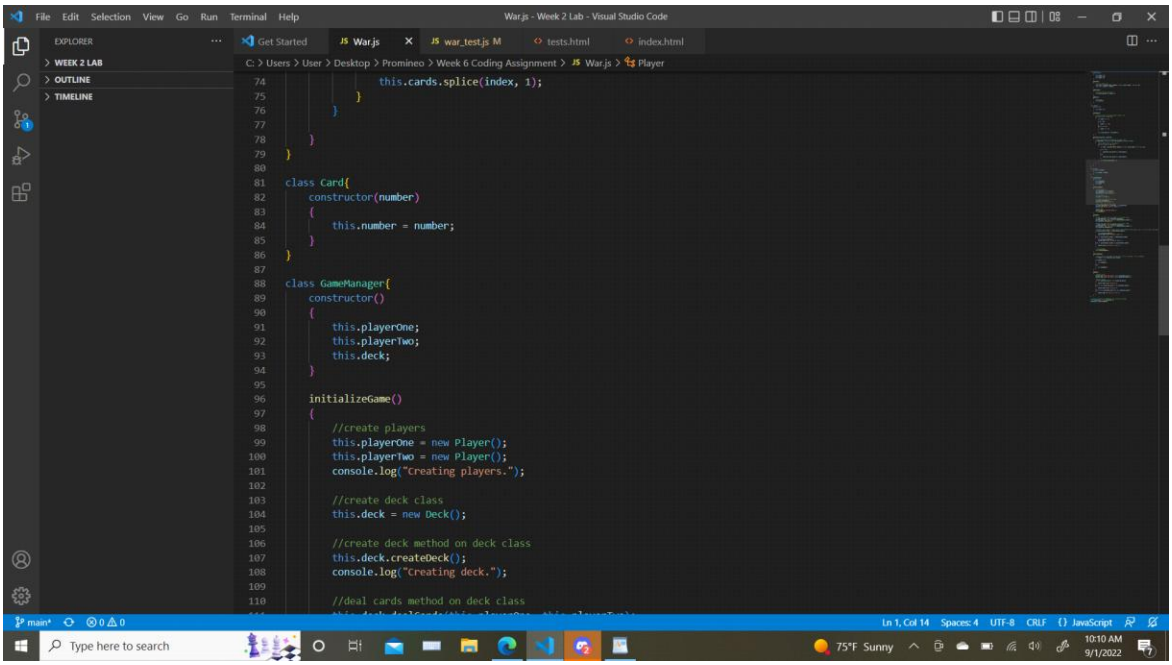
Screenshots of Code:



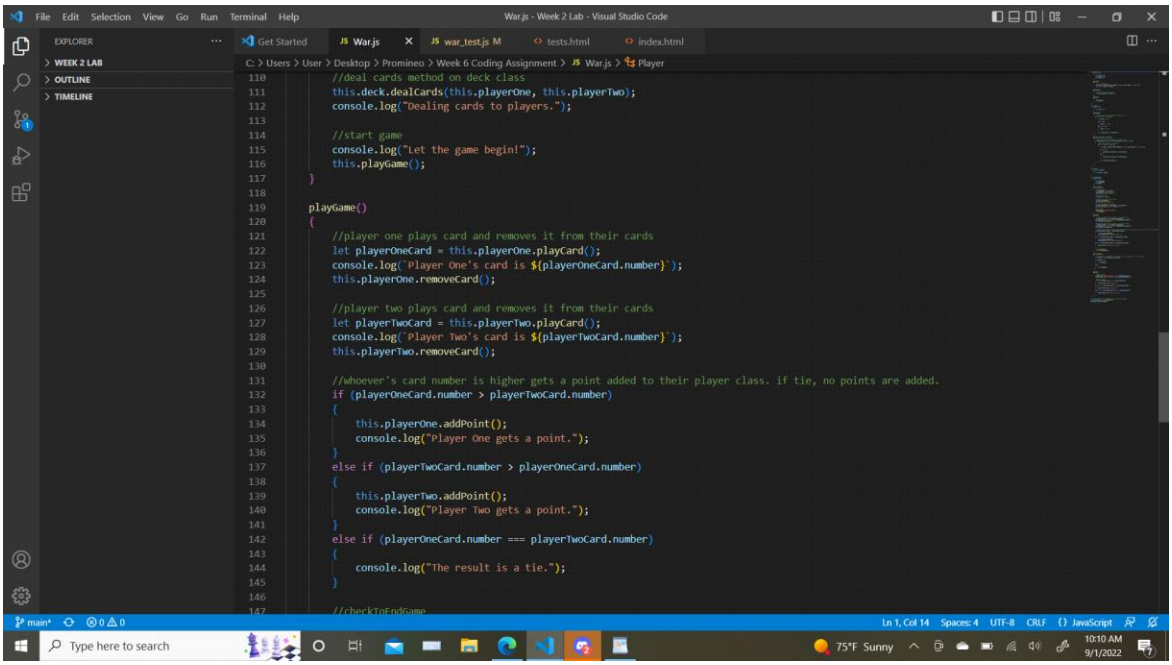
```
class Player{
    constructor()
    {
        this.cards = [];
        this.points = 0;
        this.index = 0;
    }

    playCard()
    {
        //play a random card
        this.index = parseInt((Math.random() * ((this.cards.length - 1)+ 1)), 10);
        return this.cards[this.index];
    }

    removeCard()
    {
        //remove card from cards array
        this.cards.splice(this.index, 1);
    }

    addPoint()
    {
        //add point
        this.points++;
    }
}

class Deck{
    constructor()
    {
        this.cards = [];
    }

    createDeck()
    {
        //create a deck of 26 cards between numbers 1-10
        for (let i = 0; i < 26; i++)
```



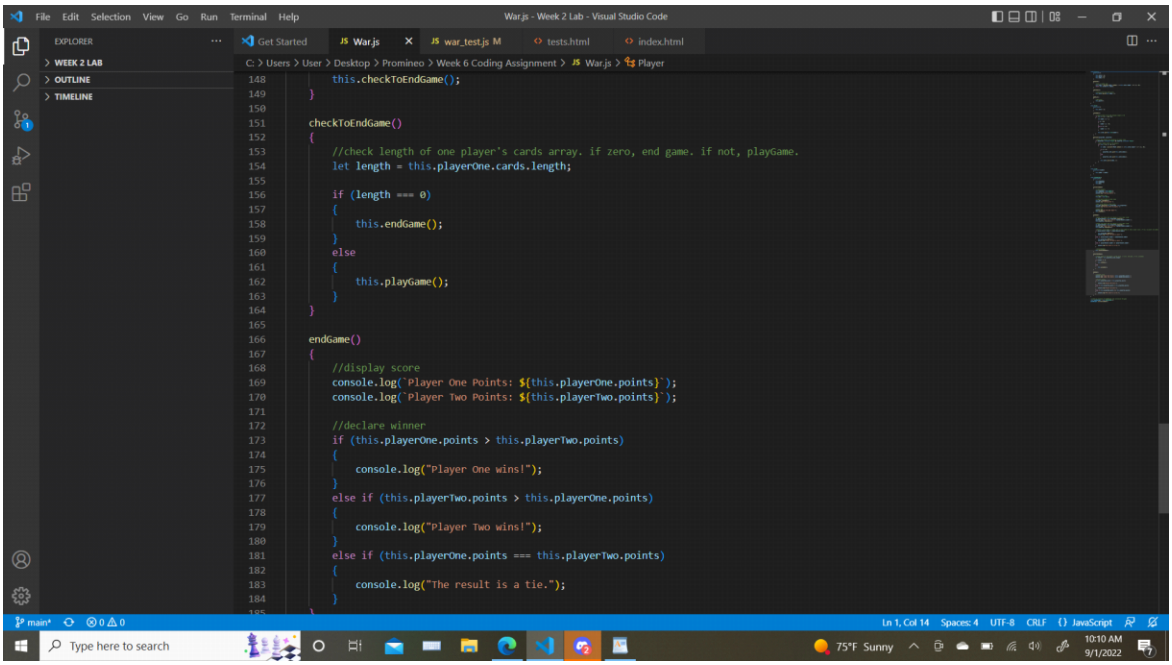
```
        //create a deck of 26 cards between numbers 1-10
        for (let i = 0; i < 26; i++)
        {
            let number = i + 1;

            if (i > 19)
            {
                number = i - 19;
            }
            else if (i > 9)
            {
                number = i - 9;
            }

            this.cards.push(new Card(number));
        }
    }

    dealCards(playerOne, playerTwo)
    {
        //check that players are instances of Player class
        if (playerOne instanceof Player && playerTwo instanceof Player)
        {
            //deal random cards to two players
            for (let i = 0; i < 26; i++)
            {
                let index = parseInt((Math.random() * ((this.cards.length - 1) + 1)), 10);

                if (i < 13)
                {
                    playerOne.cards.push(this.cards[index]);
                }
                else
                {
                    playerTwo.cards.push(this.cards[index]);
                }
```

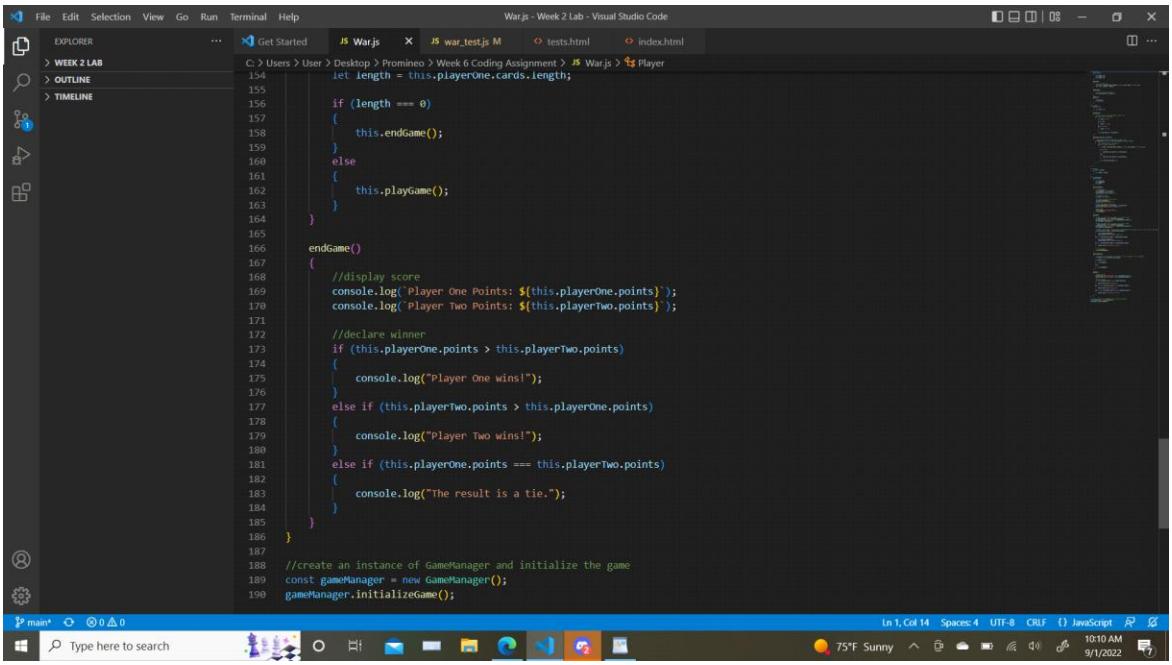


```
                this.cards.splice(index, 1);
            }
        }

    }
}

class Card{
    constructor(number)
    {
        this.number = number;
    }
}

class GameManager{
    constructor()
    {
        this.playerOne;
        this.playerTwo;
        this.deck;
    }

    initializeGame()
    {
        //create players
        this.playerOne = new Player();
        this.playerTwo = new Player();
        console.log("Creating players.");

        //create deck class
        this.deck = new Deck();

        //create deck method on deck class
        this.deck.createDeck();
        console.log("Creating deck.");

        //deal cards method on deck class
```



```
        //deal cards method on deck class
        this.deck.dealCards(this.playerOne, this.playerTwo);
        console.log("Dealing cards to players.");

        //start game
        console.log("Let the game begin!");
        this.playGame();
    }

    playGame()
    {
        //player one plays card and removes it from their cards
        let playerOneCard = this.playerOne.playCard();
        console.log(`Player One's card is ${playerOneCard.number}`);
        this.playerOne.removeCard();

        //player two plays card and removes it from their cards
        let playerTwoCard = this.playerTwo.playCard();
        console.log(`Player Two's card is ${playerTwoCard.number}`);
        this.playerTwo.removeCard();

        //whoever's card number is higher gets a point added to their player class. if tie, no points are added.
        if (playerOneCard.number > playerTwoCard.number)
        {
            this.playerOne.addPoint();
            console.log("Player One gets a point.");
        }
        else if (playerTwoCard.number > playerOneCard.number)
        {
            this.playerTwo.addPoint();
            console.log("Player Two gets a point.");
        }
        else if (playerOneCard.number === playerTwoCard.number)
        {
            console.log("The result is a tie.");
        }

        //checkToEndGame
```


File Edit Selection View Go Run Terminal Help
War.js - Week 2 Lab - Visual Studio Code
EXPLORER
WEEK 2 LAB
OUTLINE
TIMELINE
Get Started
War.js
war_test.js M
tests.html
index.html
C: > Users > User > Desktop > Promineo > Week 6 Coding Assignment > War.js > Player
this.checkToEndGame();
}
checkToEndGame()
{
//check length of one player's cards array. if zero, end game. if not, playGame.
let length = this.playerOne.cards.length;
if (length === 0)
{
this.endGame();
}
else
{
this.playGame();
}
}
endGame()
{
//display score
console.log(`Player One Points: ${this.playerOne.points}`);
console.log(`Player Two Points: ${this.playerTwo.points}`);
//declare winner
if (this.playerOne.points > this.playerTwo.points)
{
console.log("Player One wins!");
}
else if (this.playerTwo.points > this.playerOne.points)
{
console.log("Player Two wins!");
}
else if (this.playerOne.points === this.playerTwo.points)
{
console.log("The result is a tie.");
}
main*
0 0
Ln 1, Col 14
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
75°F Sunny
10:10 AM
9/1/2022


File Edit Selection View Go Run Terminal Help
War.js - Week 2 Lab - Visual Studio Code
EXPLORER
WEEK 2 LAB
OUTLINE
TIMELINE
Get Started
War.js
war_test.js M
tests.html
index.html
C: > Users > User > Desktop > Promineo > Week 6 Coding Assignment > War.js > Player
let length = this.playerOne.cards.length;
if (length === 0)
{
this.endGame();
}
else
{
this.playGame();
}
}
endGame()
{
//display score
console.log(`Player One Points: ${this.playerOne.points}`);
console.log(`Player Two Points: ${this.playerTwo.points}`);
//declare winner
if (this.playerOne.points > this.playerTwo.points)
{
console.log("Player One wins!");
}
else if (this.playerTwo.points > this.playerOne.points)
{
console.log("Player Two wins!");
}
else if (this.playerOne.points === this.playerTwo.points)
{
console.log("The result is a tie.");
}
}
}
//create an instance of GameManager and initialize the game
const gameManager = new GameManager();
gameManager.initializeGame();
main*
0 0
Ln 1, Col 14
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
75°F Sunny
10:10 AM
9/1/2022

Screenshots of Running Application: