Evan Wells

Professor Lash

12/6/2023

Black Jack Assignment

The point of this assignment was to make a working Black Jack game and have the given rules be exhibited in the javascript code. I will go through each section 1 by 1 of the code and explain what it does

```
let deck = {
   cards: [
       {'img': 'A-C.png', 'value': 11, suit: 'clubs', 'dealt': false, label: 'ace'},
       {'img': 'A-H.png', 'value': 11, suit: 'hearts', 'dealt': false, label: 'ace'},
       {'img': 'A-D.png', 'value': 11, suit: 'diamonds', 'dealt': false, label: 'ace'},
       {'img': 'A-S.png', 'value': 11, suit: 'spades', 'dealt': false, label: 'ace'},

       {'img': '2-C.png', 'value': 2, suit: 'clubs', 'dealt': false, label: 'two'},
       {'img': '2-H.png', 'value': 2, suit: 'hearts', 'dealt': false, label: 'two'},
       {'img': '2-D.png', 'value': 2, suit: 'diamonds', 'dealt': false, label: 'two'},
       {'img': '2-S.png', 'value': 2, suit: 'spades', 'dealt': false, label: 'two'},

       {'img': '3-C.png', 'value': 3, suit: 'clubs', 'dealt': false, label: 'three'},
       {'img': '3-H.png', 'value': 3, suit: 'hearts', 'dealt': false, label: 'three'},
       {'img': '3-D.png', 'value': 3, suit: 'diamonds', 'dealt': false, label: 'three'},
       {'img': '3-S.png', 'value': 3, suit: 'spades', 'dealt': false, label: 'three'},

       {'img': '4-S.png', 'value': 4, suit: 'spades', 'dealt': false, label: 'four'},
       {'img': '4-H.png', 'value': 4, suit: 'hearts', 'dealt': false, label: 'four'},
       {'img': '4-D.png', 'value': 4, suit: 'diamonds', 'dealt': false, label: 'four'},
       {'img': '4-C.png', 'value': 4, suit: 'clubs', 'dealt': false, label: 'four'},

       {'img': '5-S.png', 'value': 5, suit: 'spades', 'dealt': false, label: 'five'},
       {'img': '5-H.png', 'value': 5, suit: 'hearts', 'dealt': false, label: 'five'},
       {'img': '5-D.png', 'value': 5, suit: 'diamonds', 'dealt': false, label: 'five'},
       {'img': '5-C.png', 'value': 5, suit: 'clubs', 'dealt': false, label: 'five'},

       {'img': '6-S.png', 'value': 6, suit: 'spades', 'dealt': false, label: 'six'},
       {'img': '6-H.png', 'value': 6, suit: 'hearts', 'dealt': false, label: 'six'},
       {'img': '6-D.png', 'value': 6, suit: 'diamonds', 'dealt': false, label: 'six'},
       {'img': '6-C.png', 'value': 6, suit: 'clubs', 'dealt': false, label: 'six'},

       {'img': '7-S.png', 'value': 7, suit: 'spades', 'dealt': false, label: 'seven'},
       {'img': '7-H.png', 'value': 7, suit: 'hearts', 'dealt': false, label: 'seven'},
       {'img': '7-D.png', 'value': 7, suit: 'diamonds', 'dealt': false, label: 'seven'},
       {'img': '7-C.png', 'value': 7, suit: 'clubs', 'dealt': false, label: 'seven'},

       {'img': '8-S.png', 'value': 8, suit: 'spades', 'dealt': false, label: 'eight'},
       {'img': '8-H.png', 'value': 8, suit: 'hearts', 'dealt': false, label: 'eight'},
       {'img': '8-D.png', 'value': 8, suit: 'diamonds', 'dealt': false, label: 'eight'},
       {'img': '8-C.png', 'value': 8, suit: 'clubs', 'dealt': false, label: 'eight'},

       {'img': '9-S.png', 'value': 9, suit: 'spades', 'dealt': false, label: 'nine'},
```

This is the deck, this is where all the cards are stored. Each card has it's own value, label and attached image. These are used during the game to represent the player and dealer hand.

```
        {'img': 'K-H.png', 'value': 10, suit: 'hearts', 'dealt': false, label: 'king'},
        {'img': 'K-D.png', 'value': 10, suit: 'diamonds', 'dealt': false, label: 'king'},
        {'img': 'K-C.png', 'value': 10, suit: 'clubs', 'dealt': false, label: 'king'}

    ],

    shuffleDeck: function() {
        for (let i = this.cards.length - 1; i > 0; i--) {
            const j = Math.floor(Math.random() * (i + 1));
            [this.cards[i], this.cards[j]] = [this.cards[j], this.cards[i]];
        }
    },

    pickCard: function() {
        if (this.needsShuffle) {
            this.shuffleDeck();
            this.needsShuffle = false;
        }

        let undealtCards = this.cards.filter(card => !card.dealt);

        if (undealtCards.length === 0) {
            this.needsShuffle = true;
            this.cards.forEach(card => {
                card.dealt = false;
            });
            undealtCards = this.cards.filter(card => !card.dealt);
        }

        const randomIndex = Math.floor(Math.random() * undealtCards.length);
        const selectedCard = undealtCards[randomIndex];
        selectedCard.dealt = true;

        return selectedCard;
    }
}
```

These are the shuffleDeck and pickCard functions. The shuffleDeck function makes it so all the cards that will be picked won't simply be in the order that they are in the code. The cards will be picked using the pickCard function and used for the game

```
function onClick(){
   const betInput = document.getElementById("uBet");
   betAmount = parseInt(betInput.value);

   if (betAmount <= 0 || betAmount % 1 !== 0 || betAmount > 500){
       document.getElementById('hand-header' ).innerHTML = "This bet is
invalid. Please bet a whole number between 1 and 500";
   } else if (betAmount > balance){
       document.getElementById('hand-header' ).innerHTML = "This bet is
invalid. Please bet an amount less than or equal to current balance";
   }
   else{
       document.getElementById('hand-header' ).innerHTML = "Hand Status";
       balance -= betAmount;
       document.getElementById('betInTable' ).innerHTML = betAmount;
       document.getElementById('balance').innerHTML = balance;
       numHits = 0;
       document.getElementById("numHits").innerHTML = numHits;
       UI.playGame(betAmount);
   }

}
```

This is the onClick function. This functions purpose is to take the imputed bet and determine if it is valid. If it isn't, an error pops up. If it is, the bet amount is added to the table and taken out of the balance and the game begins

```
let gamesPlayed = 0;
let playerCard1, playerCard2, dealerCard1, dealerCard2;
let UI = {

   playGame: function (betAmount) {
       gamesPlayed++;
       document.getElementById("gamesPlayed").innerHTML = gamesPlayed;
       document.getElementById("startScreen").style.display = "none";
       deck.shuffleDeck();
       playerCard1 = deck.pickCard();
       playerCard2 = deck.pickCard();
       dealerCard1 = deck.pickCard();
       dealerCard2 = deck.pickCard();

       let playerCardStr =
            `    <div id="userHeader">User Hand: <br/></div>
                 <div id="userHand">
                 <img class='userCardImage' src='images/cards/${playerCard1.img}' alt='Card' />
                 <img class='userCardImage' src='images/cards/${playerCard2.img}' alt='Card' />
             </div>
       <div id="underImageArea"><button type="button" id="hit" onClick="UI.hit();" >Hit</button>
       <button type="button" id="stay" onClick="UI.stay();">Stay</button>
       </div>`;
       let dealerCardStr =
            `  <br/>Dealer Hand: <br/>
                 <div id="dealerHand">
                 <img id ='dealerCardFaceDown' src='images/cards/BACK.png' alt='Card' />
                 <img class ='dealerCardImage' src='images/cards/${dealerCard2.img}' alt='Card' />
            </div>
            <div id="dealerbutton">
            <br/><button type="button" class="dealTurn"  style="display: none;" onClick="UI.dealerTurn();"
>Dealer Play</button>
          </div>`;

       document.getElementById("gameTable").innerHTML = playerCardStr + dealerCardStr;

       let handPoints = playerCard1.value + playerCard2.value;
       if (playerCard1.label === 'ace' && playerCard2.label === 'ace'){
           handPoints = 12;
       }
       if (
           (playerCard1.label === 'ace' && playerCard2.label === 'jack') ||
           (playerCard1.label === 'jack' && playerCard2.label === 'ace')
       ) {
           if (
               (dealerCard1.label === 'ace' && dealerCard2.label === 'jack') ||
               (dealerCard1.label === 'jack' && dealerCard2.label === 'ace')
           ) {
               this.dealerBj()
           } else {

               this.userBj()
           }
       } else if (
           (dealerCard1.label === 'ace' && dealerCard2.label === 'jack') ||
           (dealerCard1.label === 'jack' && dealerCard2.label === 'ace')
       ) {
           this.dealerBj()
       }
       else {
           document.getElementById("userpts").innerHTML = handPoints;
           document.getElementById("dealpts").innerHTML = dealerCard2.value;
       }
   },
```

This is the first function of the game, playGame. This function deals the picked cards and updates the user and dealer points. The dealer only has one card up. If the dealer or user has a blackjack (jack and ace) the game ends and the person with the blackjack wins. If that is the user, they win 2x their bet.

```
hit: function () {
    numHits++;
    document.getElementById("numHits").innerHTML = numHits;
    let newCard = deck.pickCard();

    let userPoints = parseInt(document.getElementById("userpts").innerHTML);
    userPoints += newCard.value;

    let newCardImage = document.createElement("img");
    newCardImage.className = "userCardImage";
    newCardImage.src = `images/cards/${newCard.img}`;

    document.getElementById("userHand").innerHTML += newCardImage.outerHTML;
    document.getElementById("userpts").innerHTML = userPoints;
    let displayedUserPoints = userPoints;
    let aces = document.querySelectorAll(".userCardImage[src*='A']");
    let numAces = aces.length;

    while (displayedUserPoints > 21 && numAces > 0) {
        userPoints -= 10;
        numAces--;
    }
    if (userPoints > 21) {
        this.userBust()
    }

},
```

This is the hit function. This function adds a card to the users hand. If that card value makes the users hand > 21, the user busts and loses.

```
stay : function(dealerPoints){
    document.querySelector('#hit').style.display = 'none';
    document.querySelector('#stay').style.display = 'none';
    document.querySelector('.dealTurn').style.display = 'block';
},
```

This is the stay function. Clicking this will remove the hit and stay option and allow the deakerTurn button to appear. Click it to start the dealer's turn.

```
dealerTurn: function () {
   let playerPoints = parseInt(document.getElementById("userpts").innerHTML);
   let dealerPoints = parseInt(document.getElementById("dealpts").innerHTML);

   document.getElementById("dealerCardFaceDown").src =
`images/cards/${dealerCard1.img}`;
   dealerPoints += dealerCard1.value;
   document.getElementById("dealpts").innerHTML = dealerPoints;
   if (dealerCard1.label === 'ace' && dealerCard2.label === 'ace'){
       dealerPoints = 12;
   }
   if (playerPoints > dealerPoints && dealerPoints > 17) {
       this.userWin();
   }
   while (dealerPoints < 17){
       numHits++;
       document.getElementById("numHitsDeal").innerHTML = numHits;
       let newCard = deck.pickCard();
       let newCardImage = document.createElement("img");
       newCardImage.className = "userCardImage";
       newCardImage.src = `images/cards/${newCard.img}`;

       document.getElementById("dealerHand").innerHTML += newCardImage.outerHTML;
       dealerPoints += newCard.value;
       document.getElementById("dealpts").innerHTML = dealerPoints;
       document.getElementById("startScreen").style.display = "block";
       console.log("dealerpts 2:", dealerPoints)
   }
   if (dealerPoints > 21){
       this.dealerBust()
   } else if (playerPoints > dealerPoints && playerPoints <= 21){
       this.userWin()
   } else if (dealerPoints > playerPoints && dealerPoints <= 21){
       this.dealerWin()
   } else if (playerPoints === dealerPoints){
       this.dealerWin()
   }

}
```

This is the dealerTurn function, the function that is primarily responsible for determining wins and losses. Both the dealers cards will be revealed, if the combined value is less than 17, then the dealer will keep hitting until the value is over 17. If this makes them bust, they lose. If this gives them a value > the user's, they win.

```
userWin : function(){
   userWins++
   let userPoints = parseInt(document.getElementById("userpts").innerHTML);
   document.getElementById("userWins").innerHTML = userWins;
   document.getElementById("userpts").innerHTML = userPoints;
   document.querySelector('#underImageArea').innerHTML = "<img
class='resultImage' src='images/winner.png' alt='none'>";
   document.querySelector('#dealerbutton').innerHTML = "<img
class='resultImage' src='images/loser_black_back.png' alt='none'>";
   this.winBet(betAmount)
   document.getElementById('balance').innerHTML = balance;
   document.getElementById("startScreen").style.display = "block";
},
dealerWin : function(){
   userLosses++
   document.getElementById("userLosses").innerHTML = userLosses;
   document.querySelector('#dealerbutton').innerHTML = "<img
class='resultImage' src='images/winner.png' alt='none'>";
   document.querySelector('#underImageArea').innerHTML = "<img
class='resultImage' src='images/loser_black_back.png' alt='none'>";
   document.getElementById("startScreen").style.display = "block";
},
```

These are the user and dealer win functions. If the user wins, a win will be added to their tally, they will win their money, and they will be given the chance to play again. If they lose, the dealerWin function takes place and they will have a loss added to their record, and still be given the chance to play again.

```
userBust : function(){
   let userPoints = parseInt(document.getElementById("userpts").innerHTML);
   document.getElementById("userpts").innerHTML = userPoints;
   userLosses++;
   document.getElementById("userLosses").innerHTML = userLosses;
   document.querySelector('#dealerbutton').innerHTML = "<img
class='resultImage' src='images/winner.png' alt='none'>";
   document.querySelector('#underImageArea').innerHTML = "<img
class='resultImage' id='bust' src='images/busted.png' alt='none'>";
   document.getElementById("startScreen").style.display = "block";
},
dealerBust : function(){
   userWins++
   document.getElementById("userWins").innerHTML = userWins;
   document.querySelector('#underImageArea').innerHTML = "<img
class='resultImage' src='images/winner.png' alt='none'>";
   document.querySelector('#dealerbutton').innerHTML = "<img
class='resultImage' id='bust' src='images/busted.png' alt='none'>";
   this.winBet(betAmount)
   document.getElementById('balance').innerHTML = balance;
   document.getElementById("startScreen").style.display = "block";
},
```

These are the user and dealer bust functions. They work similar to the userWin and dealerWin functions. If the user busts, they lose. If the dealer busts, the user wins.

```
userBj : function(){
   let handPoints = playerCard1.value + playerCard2.value;
   let dealerPoints = dealerCard1.value + dealerCard2.value;
   document.getElementById('userpts').innerHTML = handPoints;
   document.getElementById('dealpts').innerHTML = dealerPoints;
   document.getElementById("hit").style.display = "none";
   document.getElementById("stay").style.display = "none";
   document.querySelector('#underImageArea').innerHTML = "<img class='resultImage'
src='images/winner.png' alt='none'> <img class='bjImg' src='images/blackJack.png'
alt='none'>";
   document.querySelector('#dealerHand').innerHTML = "<img class='resultImage'
src='images/loser_black_back.png' alt='none'>";
   userWins++
   document.getElementById("userWins").innerHTML = userWins;
   this.winBetBj(betAmount);
   document.getElementById('balance').innerHTML = balance;
   document.getElementById("startScreen").style.display = "block";
},
dealerBj : function(){
   let handPoints = playerCard1.value + playerCard2.value;
   let dealerPoints = dealerCard1.value + dealerCard2.value;
   document.getElementById('userpts').innerHTML = handPoints;
   document.getElementById('dealpts').innerHTML = dealerPoints;
   document.getElementById("hit").style.display = "none";
   document.getElementById("stay").style.display = "none";
   document.querySelector('#dealerHand').innerHTML = ` <img id ='dealerCardFaceDown'
src='images/cards/${dealerCard1.img}' alt='Card' />
                                                      <img class
='dealerCardImage' src='images/cards/${dealerCard2.img}' alt='Card' />`;
   document.querySelector('#userHand').innerHTML = "<img class='resultImage'
src='images/loser_black_back.png' alt='none'>";
   document.querySelector('#dealerbutton').innerHTML =  "<img class='resultImage'
src='images/winner.png' alt='none'> <img class='bjImg' src='images/blackJack.png'
alt='none'>";
   userLosses++
   document.getElementById("userLosses").innerHTML = userLosses;
   document.getElementById("startScreen").style.display = "block";
},
```

These are the blackjack functions. They work slightly different from the win functions. If the user gets a blackjack, they win 2x their bet. If the dealer gets a blackjack (even if the user does too), the dealer wins. These functions happen automatically and dont allow the user to hit or stay.

```
winBet : function(betAmount){
   balance = balance + (betAmount * 2);
},
winBetBj : function(betAmount){
   balance = balance + (betAmount * 3);
},
```

Lastly, these are the winning bet functions. If the user wins, they win their bet back. If they win with Blackjack, they get 2x their bet back.

This code took a very long time to make and I am very proud of it. It works very well for the blackjack game and I am proud of myself for problem solving and getting through it.