

Final Project

Networked Blackjack

Summary

The goal of this project is to build a networked Blackjack game that can be played by up to two human people against the dealer (a simple AI bot). The networking component can be as simple as using Socket.io and opening multiple browser windows (that's actually how I'm going to test it).

Game Rules

The object of the game is to get as close to the sum total of 21 as possible. If you go over the total of 21 for your hand it is considered a "bust" and you lose the hand.

Each player is dealt two cards by the dealer first and then the dealer deals himself two cards.

Player Rules

A player is dealt a new card by the dealer with each "hit" until he chooses to "stand" (stop drawing) or until he "busts" (goes over 21). The value of each card is calculated by the following:

1. An Ace can be either 1 or 11 depending on the current total (totals start with zero) of the hand. If the current total is less than 11 an Ace would be considered to have the value of 11; otherwise, an Ace would be considered to have the value of 1.
2. Cards of 2 through 10 would have their appropriate value.
3. Face cards (Jack, Queen, King) are treated as though they have the value of 10.
4. If the first two cards dealt to the player make a collective value of 21, it is considered "Blackjack" and control of the game automatically moves to the next player. If there is no next player, control of the game moves to the dealer.

Dealer Rules (AI)

1. Card values for the Dealer are the same as for the Player(s).
2. If the first two cards the Dealer deals to himself have the collective value of 21, then the Dealer is considered to have "Blackjack" and the Player(s) lose automatically and a new hand would begin. This check happens before control of the game is given to the first player.
3. The Dealer must consistently draw until he has at least a collective total of 17. Once a single card puts his collective value at 17 or above he must stop drawing. The Dealer can also bust in doing so.
4. No distinction will be made between a "soft 17" (a set of at least three cards making up a total of 17) and a "hard 17" (a hand containing only a Face Card and a 7) for simplicity's sake.

5. If the Dealer busts, the Player(s) that have not busted all win their hands regardless of whether they have different collective totals. In Blackjack, all players compete against the Dealer, not each other.

When the Dealer is finished drawing and he has not busted, the comparison phase begins between the Dealer and the Player(s).

Comparison Phase Rules

1. If the collective total of the Dealer is greater than the collective total of a Player, the Dealer wins against that Player.
2. If the collective total of the Dealer is the same as the collective total of a Player, the Dealer and the Player "push" (tie).
3. If the collective total of the Dealer is less than the collective total of a Player, the Player wins against the Dealer.

Specification

As usual, I'm more interested in the functionality working than I am about how the UI looks. Just make sure all of the UI pieces are there.

User Interface Components

1. A set of cards that the Player has in his hand. This can be either images or text (like AS for "Ace of Spades", 10C for "Ten of Clubs", etc). You can also choose to show all cards of all players on the screen since Blackjack can be played either face-up or face-down. That's up to you.
2. A set of buttons that allow the Player to choose whether to hit (draw) or stand (stop drawing).
3. A Player's collective total should be shown somewhere on the screen. All players start with a score of 0.
4. One of the Dealer's two initial cards should be shown as face-up, just like the table game. This is what allows the Player(s) to make inferences about what the Dealer's current total might be.

Functionality

1. Player(s) should be able to "hit" or "stand" based upon the Player Rules above.
2. Control of the game moves to the Dealer only when the Player(s) have finished drawing. Control of the game moves to the next Player once the previous Player has either busted or decided to stand.
3. The Comparison Phase only occurs once the Player(s) and the Dealer have all finished drawing.
4. The networking component is intended to be HUMAN players in separate browsers. Multiple browser windows being used on the machine is perfectly fine and qualifies; this is actually how I'm going to test the submission.
5. The networking component needs to be done with Socket.io.

Words of Advice

I strongly suggest you do as much of this assignment in JavaScript as possible. That will make it easier for Socket to communicate with the game logic. This assignment can be done entirely in Node, front-end JavaScript, and HTML.

Grading

This project is worth 20 points, weighted at 30% of the grade.

Extra Credit

Add user registration and login functionality using PHP (can be either vanilla PHP or using something like Laravel). Save the number of hands won and lost for each Player to the database (maybe do this with an AJAX call for simplicity).

The extra credit is worth 6 points.