

Matt McGauley, Bea Casey, Jack Otto, Sami Wurm

CSCI205 — King

### BlackJack with AI User Manual

We wanted to be able to play blackjack on our computers and also simulate both a random player of blackjack and an artificially intelligent player who learns how to play the game with a better strategy, thus increasing the player's success rate. In a single round of blackjack in our game, both the player (you; the client) and the dealer are dealt 2 cards from a single deck. Then, you can choose to either hit or stand based on the current cards in your hand and bet money on yourself winning. Then, once bets have been placed, whoever's hand is closer to 21 without going over 21 wins. After we were able to create an interface to play a game of blackjack, we wanted to compare our ability to win with that of a random agent to see how someone would do if they randomly chose to hit or stand each round. This agent randomly chooses a number between 0 and 1 and if the number is  $< 0.5$  it hits, otherwise it stands. This agent does very poorly at the game and does not improve over time. Finally, we implemented a neural net, which acts as a player who plays many hands of blackjack and learns the result of hitting or standing with each hand that it plays. This agent stores the results of each of its hands in its memory, and based on the probability of success in previous results, it decides to hit or stand. This agent's success rate goes from 0 to about 1-2%, which is quite good considering the Random player's success rate never steadily increases, and human card-counters usually only gain a 0.5-1% increase in their success rate (<https://wizardofodds.com/games/blackjack/card-counting/introduction/>). Overall, we found that humans do better at blackjack than random agents, but worse than artificially intelligent agents, and we were able to confront the problem of

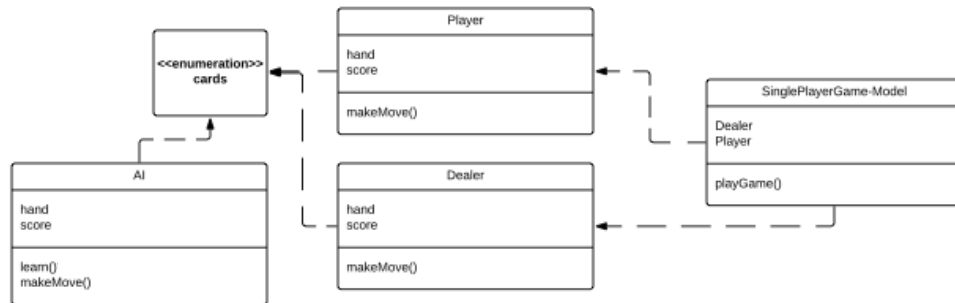
seeing if we could build a machine learning agent who could learn how to play the game of blackjack well (we could!).

The problem we wanted to solve was understanding how one can master a game that is partially based on luck. Our main goal was to demonstrate that as you get better at the game of blackjack, your success only improves to a point. To do so, we first simulated the game with an AI that made completely random moves for a specified number of hands of the game. Doing so would demonstrate how someone who is new to the game might play. It assumes that you know nothing about the strategy of the game and that you know nothing about how to win the game. You simply hit or stand until you happen to win or lose. We then wanted to see how an AI could learn how to play the game. The AI plays the game in a similar way to someone who has been playing the game for some amount of time and is able to predict when their hit or stand will help them win the game. But above all else, we wanted to show the limitations of learning how to play. You can develop strategies for hitting or standing based on the strength of the cards in your hand, but your success is based on the luck of the draw from the deck. Therefore, your intelligence about the game is limited and doesn't work in every situation because of the random aspect of the game.

We also wanted to make it so that you could play the game with a computer. We started by making a console-based version of the game that would hit and stand with input from the keyboard. We then transitioned the game to a graphical user interface. We created buttons that would allow you to hit or stand and you can play the game with money. You can play the game as much as you want until you run out of money. The game is smart enough to know when to switch the value of aces. The dealer aspect of the game is completely independent of the player. This made it so that the AI can implement the same code that we used to play blackjack.

## Blackjack UML

Jack Otto | December 8, 2019



Run the file “LandingPageMain” and it will take you to your home screen, where a GUI will pop up, and music will start playing. If you want to play single-player Blackjack, select that option and it will take you to the interface to play the game. To start a new hand, select the deck (the red rectangle in the top right corner). You will be prompted to put in a dollar amount to bet (your starting balance is \$100.00). Type the amount you want to bet (in number form, eg ‘10.00’ not ‘ten’) and the cards will be dealt. You will see both of your cards and beneath them will be the current score of your cards. You will also see one of the dealer’s cards. If you want to get another card, click the hit button and if you want to remain with your cards, then hit stand. You can hit as many times as you would like, but if you go over 21, you have busted and will automatically lose. If you have an ace in your hand, then based on your other cards it will switch from a value of 11 to 1. Once you decide to stand, the dealer will reveal your cards and the game will tell you whether you won, lost, or tied and the amount of money you have in your balance will change accordingly. To deal a new hand, simply click on the deck again and place a new bet! If you are finished playing the game and want to use the other functionalities of the game, click the button ‘return to main menu.’ If you would like to see two examples of AI play the game, click the button that says ‘Play the game with AI’ and you will be taken to a different page. From the drop down menu, you can choose whether you want to simulate the game with a

random solver or a neural net solver. Then, you can type the number of hands you would like to simulate (again, enter ONLY numeric values, eg '1000', not 'one thousand'). The program will then load a graph showing the success rate of the solver as it played. WARNING: the higher the number used for the neural net solver, the longer it takes for the program to load the graph and you will get the spinning beach ball of death. Additionally, if you over-train the neural net, its success rate will decrease. If you decide that you want to return to the main menu, you can click on the 'return to main menu' button and decide what you would like to do from there.