Intro to AI: Final Practicum Report

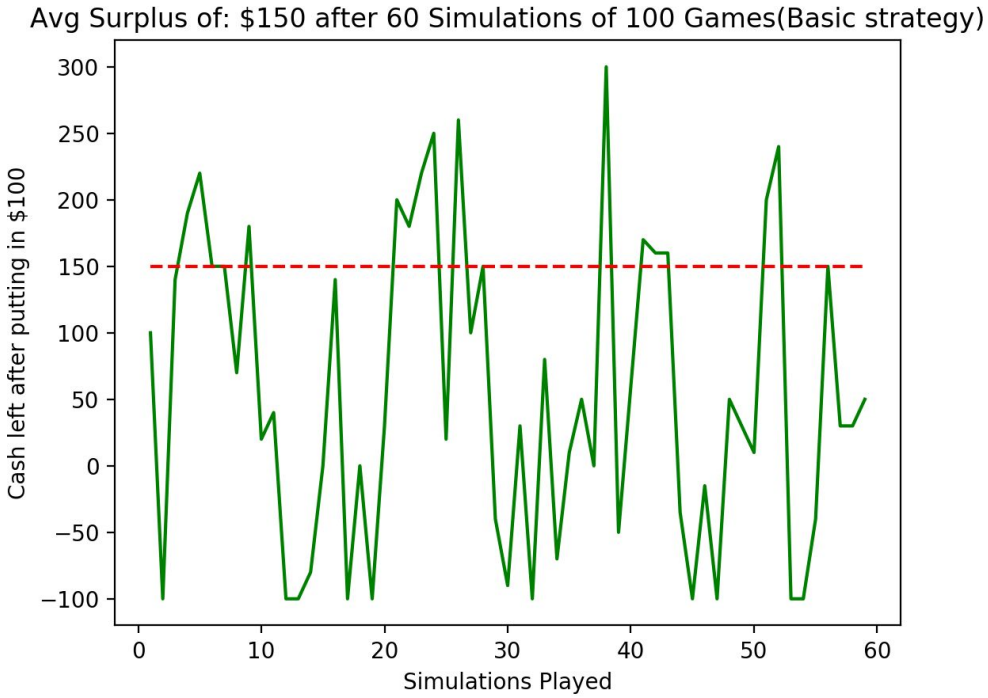
Parker Eischen

Project: Implement a BlackJack AI to play and potentially win with a surplus of cash

Implementation of BlackJack: I first had to create the game of BlackJack from scratch in python. This step was more fun than tedious. Since I had never created a game before, this was quite the learning curve. I had to keep implementing more and more functions into my classes in order for the game to work correctly. My Deck Class, created the card elements and shuffled them to form the array that consisted of the deck. I created a Player Class to keep track of the Dealer's and Player's hands, actions, and had a method to add a card from the Deck Class into their respective hands. A GameBoard Class, which keep track off all the decks, players, sum of hands, card values, and the ability to check if a player's hand had busted. Finally, I added a State class which implemented most of the rules of BlackJack and kept track if the game ended or not. The entire game is then run in my PlayGame method. Which allows the player to place a bet amount per hand and choose an action based off the tradition "HIT" ,"Stand" and "DoubleDown". After countless hours of bug testing and making it look pretty in the terminal, I had a working(for the most part) replica BlackJack. I feel like if I had more time to work on it, I would have been able to adjust the classes to be more fluid, like put all game rules in GameBoard rather than State, but for now, I'm satisfied with the result.

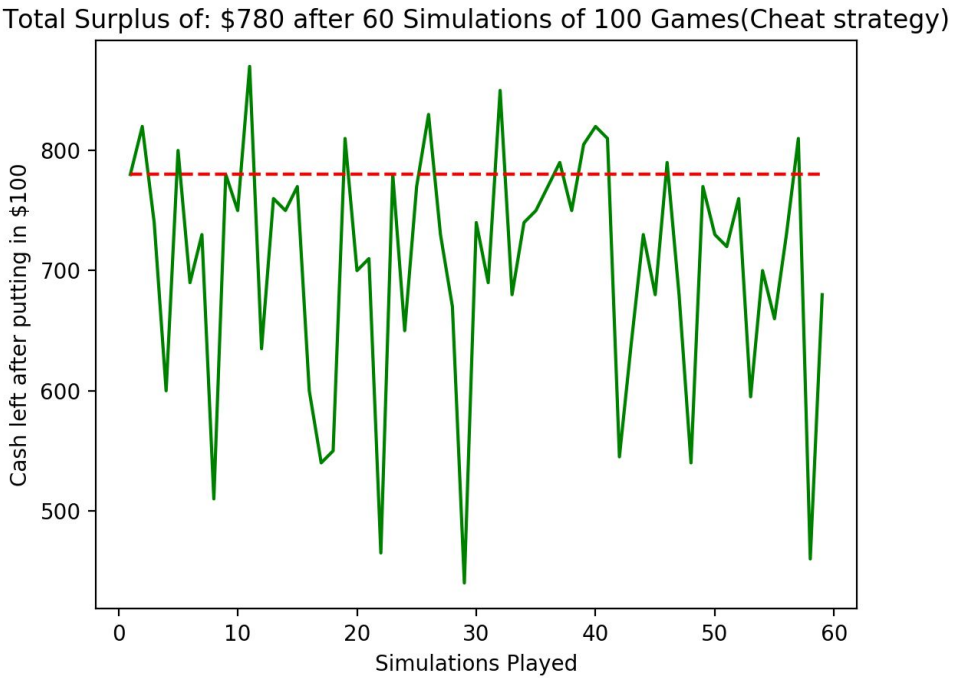
AI Implementation: It was a struggle for me to find the right algorithm from class to implement into this game. I was originally going to do a miniMax however the Dealer doesn't make any actions in that aren't hard programmed in the rules. The dealer does the same thing based on their hand. It hits on each card until it reaches a total value of 17 or busts. So there wouldn't be in variation in it's moves that would make it play it's best, like a miniMax is based off of. Instead I implemented a basic AI to originally play on the best strategy that professional BlackJack players use. Card counting and solid normal play. Card counting works based on assigning value for each card drawn during play. +1 for low cards [2-6], med cards are 0 [7-9] and high cards are -1 [10-Ace]. This is as an overall tally called The Running Count. The running count is then divided by the total decks remaining to get the Total Count. The higher the total count, the more you want to bet in the next hand. I also, created a function to decide the best possible move on the current card values based of the player hand and dealer hand to account for

free hits and times when to stand. The results of this implementation are as follows.



The green line represents the total cash surplus (out of 100) after 60 simulations of 100 games played. The red line is the avg surplus of about $150 per game.

I decided to have fun and implement a bot that could calculate the best action based if it new the order of the deck. Although this is unrealistic in modern day situations of the game, I wanted to see how much the bot would win if it could calculate how when it would bust, or if the dealer would bust anyway. This bot used a recursive depth search of the upcoming cards and calculated when to hit or stand. After the same amount of games and simulations, the results were



This means, if you possibly could calculate the order of the deck this bot could win you and average of $780 after each 100 games. The dips in the graph are based off of situations when neither you or the dealer would bust and your cards would not beat out the dealers. The more drastic dips are most likely if the CardCount function bets a lot more than normal and loses it. Although this is highly unrealistic and %100 cheating, it is still fun to know.

Conclusion: Although the correct algorithm for this type of game is most likely Machine Learning based, implementing an AI, even to win on average, wasn't too much of a stretch. Winning in this game is all knowledge and data based. The basic strategy with card counting is simple for an AI to win successfully, near impossible for a human to compute in the moment. This is a game that the house is going to win the majority of the time. This project has taught me how difficult basic game development is and all the specific traits a game has to have for an AI to work within it. Of course it's easy if all the unknown information is given to you. I also developed an interest in the game. I learned more about the strategy and culture than expected. I think I even picked up a gambling addiction on the side. I downloaded a BlackJack app and played it more than I'd like to admit.

If I had more time with this project, I would try see if other in class algorithms would be possible to simulate in BlackJack and add "Splitting" into account, which I left out of the game

to make it simpler. I'll most likely come back to this project as my progress through my undergrad and gain more information Artificial Intelligence and Machine Learning techniques.