Helen Sun (hs667), Julian Massarani (jm2249)

Project Proposal: Blackjack AI

We decided to create an AI to play Blackjack that will count cards, make an informed decision to bet or not, how much to bet, and whether to hit, stay, split, or fold. The AI will keep track of the count of face cards that appear and calculate the probability of each hand that its opponent(s) may have. It will also evaluate what the dealer may have, the probability that he busts, and the probability that he will have a greater value than the AI.

We will use an evaluation function that takes in all the data that a player would have access to (including the count) and determine the best move for the AI to make (ie. hit, split, etc.). The AI will take into consideration the general "dealer's strategy". We will implement a minimax-style tree and evaluate each possible hand the dealer and the other players have, as well as the possible next card the AI will get if it "hits", and based on each possibility, determine the best move (whichever has the best value) to make for each case. It will then calculate the probability of each scenario, and determine which move to make based on the probability and value of each case. We will also implement an algorithm to determine how much the AI will bet. We may also implement the Monte Carlo method, to gather information about trends in the game (for example, realizing that there are no more of a certain card, and how that affects the outcome of the game).

While a GUI is not entirely necessary for this implementation, we plan to make one using PyQt. PyQt is a cross platform GUI that is very well documented and easy to use. We will attempt to emulate a UI that one would find on an online gambling website.

We will have a variety of people test our Blackjack AI by having them play against the AI, trying to win against it. They will go through one deck of cards, and whoever has more money at the end is the winner. We may also implement a less sophisticated AI that decides a random amount to bet, with a simpler method of deciding whether to hit, stay, split, or fold, and determine to basically not bust. We will calculate three running statistics: the percentage of the time the AI wins, the percentage of the time the AI beats the dealer, and the average amount of money the AI makes per game.

We currently don't have a set timeline, but we live close to each other so finding times to do work should be easy. Also, we will use github for version control.