

Mini Poker Game With AI

Miguel de la Cruz Cabello

Computer Science Department
North Central College
Naperville, Illinois
mdelcruzcabello@noctrl.edu

Even Myren Nybø

Computer Science Department
North Central College
Naperville, Illinois
emnybo@noctrl.edu

Abstract

A cheating poker game is one of the most popular variations of poker in online gambling. Nevertheless, it is one of the benchmark challenges for AI because players do not have a common idea of the game being played. In this project, we present a “mini” version of how AI can benefit online gambling companies to generate profit by building a program that allows the computer to choose intelligibly when to play a hand in a way that it does not get caught by the user. The architecture of the program contains the main module that precomputes the solution of both the user’s and the computer’s hand and gives them scores. Then, it compares which hand is greater and through certain conditions determines a “play strategy”. If the computer score is less than the user score, then it will replace the last card of both the user and computer to get a higher chance for the computer to win. This module is only activated when the computer is losing money and randomly every 0 to 4 cycles of the game.

I. INTRODUCTION

Recreational games have been used in AI as benchmarks to analyze the progress of the field, and poker is one of the examples. In this game, the user does not know the exact state of the game at every point because there is another player that is not sharing some information. Therefore, knowing that information beforehand requires AI and is crucial to the future of the game. Poker is a very interesting game with certain advantages in its online version such as convenience (no need to drive to the nearest casino and is available 24/7) or the ability to get more hands over time than in standard poker. Nevertheless, there are some potential dangers where the user can be cheated while playing online poker. The purpose of this project is to teach the user not to gamble against the machine by building a program that generates positive revenue. Therefore, the program comes with a little twist (explained later) to trick the user in a way that he or she does not notice.

Our program consists of seven classes (Card, StandardDeck, Scores, CheckHand, Player, Computer, and Game) that build together to create the game. Cards in the deck were ranked from 2 to 14 (being 14 the strongest card). The Ace was counted as the strongest card without the possibility of changing to its lowest value. We gave each of the viable hands a score and compare them to determine the highest hand. When two hands have the same highest card, the computer will return the money back to the user and call it a tie. The AI method acts as the “computer’s brain” activating every certain cycle and every time it is in negative revenue.

II. ALGORITHM/METHODS

Our AI method consists of a counter that **activates randomly** when the counter hits 0 or **activates when the computer is losing revenue**. The program is **keeping track of the computer’s wallet** all the time, but it does not show it to the user. The method **is built on different conditional statements** that will guide the computer in making decisions. Each hand will have a score. The computer will know the score of each hand before playing. It will compare both hands and decide whether fold, raise, or match a bet. If the computer’s score is smaller than the user’s score, the computer will fold, just losing the initial bet. In case the computer’s score is greater than the user’s, it will raise or match the bet depending on the user’s bet. If the bet of the user is greater than 50% of his wallet, then the computer will match the user’s bet, winning a big amount of the user’s wallet. In addition, if the user’s match is less than 40% of the wallet, the computer will raise the bet by 40% of the initial bet (since it knows it will win), putting the player under pressure. Usually, players start betting low amounts, that is why we are just raising the bet by 40% as it is something affordable for the user and he can be tricked.

III. EXPECTED PERFORMANCE

The AI will generate money over time because the odds are always on the computer’s side if the algorithm is used. If the computer wallet is negative, the algorithm we created will be used in every game until it has made a profit. Therefore, instead of having a 50/50 chance of winning, the computer will have a 70/30 advantage because it already knows the outcomes. The only way for the computer to lose money using this algorithm is by folding it every time. If the computer folds, it knows that it will lose that round, and instead of losing money from matching the user’s new bet or raise, it will instead fold to lose the least amount of money that round. In other cases, the computer knows it will win. Therefore, it can raise the bet to trick the user into matching; every time the computer raises or matches in this algorithm, the computer is going to win money since it will know the card values of both hands. Since the algorithm will be used in every case when the company wallet is negative, it will gradually cause the computer wallet to be positive. When the computer wallet is positive and the algorithm is not used every time, we create a random cycle that chooses a cycle in-between zero and four and counts down every game, and when the variable is equal to zero, it will use the algorithm that compares the hands before deciding what the computer should do. The cycle will gradually generate money for the company since even

though the company wallet is positive, it will still use the algorithm to increase the company wallet using this cycle. We chose a random generator for this cycle because the user will not be able to see a pattern for when they lose, which will make it difficult as a player to see any suspicious algorithms happening behind the user interface. The cycle is there, so the user will be able to win some money from time to time, so they will have a feeling that they can win money.

IV. REFLECTION

We believe the algorithm we created is good in terms of not making it obvious that the program has code in the algorithm to manipulate the game's outcome. We created our game to increase the chances of the computer winning and use the information from the cards for the benefit of the computer's decision. The algorithm is made to gradually earn money without it being too obvious that code behind the scenes will make the computer win money most of the time. The idea of manipulating the last card of the user's hand and the computer's hand by changing it to a new card if the user's hand was better than the computer's hand guaranteed a higher chance of winning for the computer.

Overall, we believe that we have created a good balance between using the algorithm to get money and making the player feel that they can win from time to time.

V. CONCLUSION

In conclusion, using the algorithms that take the value of each hand and compare it to find the winner before the computer decides the next move is working well. How the AI is activated randomly if the computer wallet is positive is important for the growth of the profits to the computer and the user interaction since it allows the user to win some money. Having the user able to win some money instead of losing all the time is essential because you want the customers to stay playing at your casino. The algorithm is laying weight on creating a discrete way of implementing the AI into the gameplay, which decreases the chances of suspicion by the users.