DEVELOPING STRATEGIES FOR THE BIDDING CARD GAME ’DIAMONDS’ WITH GENAI

Introduction

Diamonds is a captivating card game blending luck and strategy. The game involves three players, each assigned a suit except for diamonds, which remain separate initially. A random diamond card is drawn for bidding, conducted secretly, with the highest bidder obtaining the card, valued according to its rank in points. Subsequent diamond cards are bid on until all are acquired, after which players tally points earned from their diamond cards to determine the winner based on the highest total points amassed

With the rise of artificial intelligence, particularly generative models like ChatGPT and Gemini, there’s an opportunity to explore strategies to increase the chances of winning this game. This report aims to discuss the process of teaching GenAI the game, iterating upon strategies, and writing a python code for the same.

Problem Statement

The primary challenge is to teach GenAI the game’s rules comprehensively and ensure its understanding through interactive gameplay and questioning. Subsequently, the aim is to develop strategies competitive against human players and possibly uncover new insights into the game dynamics

Teaching GenAI the Game

To teach GenAI the game, I began by explaining the rules and also introduced the concept of modifying the game for two players.Then I asked it to play the game with me and in this process, corrected its errors until it understood the game. One problem I faced was that it was again and again repeating its bids which is inconsistent with the game’s rules where bids cannot be repeated.

During gameplay, I encouraged GenAI to adopt a more strategic approach and strive to win the game with the remaining cards. After successfully completing a game, I queried GenAI about its strategy, analyzing factors contributing to its losses and exploring alternative strategies.[transcript in appendix]

Following this, I prompted GenAI to guess my strategy and engaged in discussions about additional strategies it could employ. Suggestions included tracking opponents’ cards to predict their bids and monitoring the usage of diamond cards. Additionally, I emphasized the importance of Strategic Prioritization, where GenAI focused on bidding for higher-ranking diamond cards to maximize point accumulation.

I tasked it with coding a Python implementation where the computer plays against a human opponent, simulating the drawing of diamond cards and subsequent bidding. While GenAI provided an initial code, it required numerous adjustments to ensure smooth gameplay. After refinement, a functional code was developed, facilitating gameplay between the computer and human opponent which I validated through testing in Google Colab. The subsequent task involved modifying the code to incorporate the discussed strategies, enabling the computer to strategically win the game. [codes in appendix]

Analyzing Strategies to Win the Game

During the development process, several strategies were explored and iteratively refined to enhance the computer’s chances of winning. The following strategies were sequentially implemented :-

1. Initially, the computer's strategy involved bidding one point higher than the player's bid if the bid was less than or equal to 10, and bidding randomly otherwise. However, this approach was flawed as it assumed visibility of the player's bid, which is not the case during the bidding process.

2. Upon recognizing the limitation of the initial strategy, a new approach was suggested where the computer bids randomly but with a bias towards higher bids, aiming to increase its chances of winning.

3. The Dynamic Bidding Strategy was then implemented, wherein the computer maintained an estimate of the average bid made by the player throughout the game. It adjusted its bidding slightly higher than the average bid, but this strategy consistently led to the player winning the game in multiple trials.

4. Following discussions of the possible stategies that the player could use, This strategy involved an Observation Phase where the computer observed the player's bidding pattern in initial rounds. Subsequently, it employed Adaptive Bidding, adjusting its bids based on the observed pattern. Dynamic Adjustment ensured that the computer adapted its strategy dynamically in response to changes in the player's bidding behavior.

5. After multiple iterations and adjustments, the final code incorporated a sophisticated bidding strategy based on the average rank of the diamond cards drawn so far. Final Bidding Strategy Breakdown:

- Estimating Average Rank: The computer calculates the average rank of the diamond cards drawn, starting with an initial guess of 7.5 if no cards have been drawn yet.

- Adjusting Bidding Strategy: Based on the average rank, the computer adjusts its bidding behavior, bidding conservatively if the average rank is 7.5, and aggressively otherwise.

- Generating Bids: Bids are generated randomly within a specified range, with adjustments based on observed player behavior. The computer bids higher if the player tends to bid aggressively and lower if the player tends to bid conservatively.

- Ensuring Unique Bids: The computer ensures its bid is higher than the player's (if applicable) and unique to avoid repetition.

Overall, the computer's strategy aims to adapt to the player's bidding tendencies by dynamically adjusting its bidding behavior based on the observed average rank of drawn diamond cards

Conclusion

In conclusion, working on strategies for the ”Diamonds” card game with GenAI was a really interesting experience. By teaching GenAI how to play the game and improving its strategies bit by bit, we can learn more about the best ways to play and maybe even help make better AI for other card games.