

# Report on Developing Strategies for the Bidding Card Game 'Diamonds' with GenAI

Priyanka Singh

March 26, 2024

## 1 Problem Statement

In the card game "Diamonds," players compete to win a set of 13 special diamond cards. Each round, a card is drawn randomly, setting the starting bid. Then, each player takes turns offering their own cards to outbid each other. The player offering the highest-value card claims the diamond card. This bidding process continues until all diamond cards are taken. The winner is the player with the diamond cards worth the most points.

## 2 Rules

- **Players:** The game involves two or more players.
- **Card Allocation:** Each player is allocated a suit of cards, excluding diamonds.
- **Auction Phase:** Diamond cards are shuffled and presented for auction.
- **Bidding:** Players bid with one of their own cards during the auction.
- **Winning Bid:** The highest bid wins the diamond card based on card point values.

- **Point Hierarchy:** Cards follow a hierarchy:  $2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K < A$ .
- **Points Allocation:** Points on the diamond card go to the winning bidder. If tied, points are divided.
- **Accumulation:** Players accumulate points from diamond cards in a scoring table.
- **End of Game:** The game ends when all diamond cards are auctioned. The player with the highest total points wins.

### 3 Teaching GenAI the Game

The process of teaching genAI the game of Diamonds was challenging and time-consuming. Initially, I introduced genAI to the rules of the game, explaining how players bid on diamond cards and accumulate points based on their card values. However, when I tasked genAI with generating the algorithm to implement the game, it struggled to grasp the intricacies of the game mechanics. Despite providing clear instructions, genAI initially assigned diamond cards to players and failed to differentiate between diamond cards and other suits during auctioning. As I continued to guide genAI through the development process, it encountered various errors and challenges. For instance, it initially used all cards for auctioning instead of limiting it to diamond cards. Additionally, there were issues with variable names and maintaining a list of available cards for each player. Despite rectifying these issues, genAI's logic remained flawed, and it required repeated corrections to ensure adherence to the game's rules. Furthermore, genAI faced difficulties in implementing features such as displaying the scorecard after each bidding round and handling ties with float division. These issues required additional guidance and refinement of the code to ensure accurate gameplay simulation. Despite the challenges encountered, genAI eventually generated a functional code for the Game of Diamonds. Through persistent guidance and iteration, genAI demonstrated improvements in its understanding of the game mechanics and coding logic. However, the process highlighted the importance of clear communication and thorough testing to ensure the successful implementation of complex game algorithms.

## 4 Iterating upon Strategy

Some strategies that can be kept in mind to increase the chances of winning:

1. **Know Card Values:** Understand card value hierarchy to prioritize bidding.
2. **Observe Opponents:** Adapt strategy based on opponents' card usage.
3. **Strategic Use of High-Value Cards:** Utilize high-value cards wisely.

## 5 Conclusion

Teaching GenAI the game of Diamonds was a learning experience. Clear communication and testing were crucial for success.

## 6 Colab link of the code

[Click here to access the code on Google Colab](#)