

Developing Strategies for the Bidding Card Game 'Diamonds' with GenAI

Shreya Sinha

March 26, 2024

1 Introduction

Diamonds is a two-player, trick-taking card game with a unique twist. Players are dealt a single suit (Spades, Hearts, or Clubs) and compete for un-shuffled Diamonds with point values (could be anything but took 5 in the code for simplicity) that act as trump cards. Bidding and winning tricks are based solely on card rank within a player's hand, making strategic card selection crucial. This report explores the development of strategies for Diamonds using Generative Artificial Intelligence (GenAI).

2 Methodology

2.1 Teaching GenAI the Game

GenAI was provided with a detailed description of the Diamonds game rules, including:

- Deck setup (single suit per player, separate un-shuffled Diamond deck)
- Dealing and bidding process (players take turns bidding a card, highest rank wins)
- Trick resolution (winner takes revealed Diamond, discards played cards)
- Winning condition (player with the most points from collected Diamonds wins)

Initial Challenges in Conversation

1. **Misunderstanding of Key Rules:** GenAI initially did not understand key rules such as the scoring rules (for eg. if the player cards will be included in the points), if the cards will be shuffled or not, which cards will be kept face down and which face up, etc.
2. **Explaining Game Setup:** Explaining the overall structure of the game, such as the number of cards each player has, the fact that the suit of the cards did not matter and all non-diamond cards need to be treated similarly, etc was a challenge.

Addressing The Issue To address this issue, GenAI was asked to generate a clearer description of the game itself after a significant amount of conversation and clarifications. This description was then used in a new chat altogether, to generate better, less buggy code.

2.2 Prompting and Iterating on Strategy

GenAI was prompted to develop bidding strategies for the computer player, focusing on:

- High-value Diamonds: Play the highest card when possible.
- Low-value Diamonds: Play a low card to potentially lose gracefully.
- Mid-range Diamonds (2-4): Explore bluffing or strategic play.

GenAI generated several strategies, including prioritizing high cards for consistent wins and bluffing with mid-range cards when the revealed Diamond is high-value. These strategies were iterated, and prompts refined to achieve a balance between aggressive and safe play.

2.3 Code Implementation

Three strategies were implemented in code: "Prioritize High Cards" being a safe and effective option. "Bluff with Mid-Range Diamonds" more aggressive which could lead to quick wins but also carried a higher risk of losing tricks and finally "Calculated Risk" which offered a balance between playing it safe and taking calculated risks.

3 Reflections

3.1 Reflections on Conversation with GenAI

1. While initial difficulties exposed reliability on human intervention while trying to code using GenAI, it also gave insights on the potential of AI to learn and generate code based on clear instructions. Thus, detailed explanations and debugging are crucial for successful collaboration.
2. Prompting GenAI with specific scenarios and desired behaviors (e.g., prioritizing high cards, bluffing with mid-range) proved effective in guiding strategy development.
3. It was easier to generate desired output when every requirement was laid out step-by-step (such as first focusing on game structure and later on strategy after fully debugging initial code)

3.2 Reflections on Code that was Generated

The generated Python code showcased GenAI's capability to translate strategies into partially functional code snippets which worked well after debugging. The link to the colab file can be found in Appendix B.

4 Conclusion

This project demonstrates the potential of GenAI in developing game-playing AI with strategic capabilities. The "Calculated Risk" strategy proved effective in winning against human opponents in some cases. However, further exploration is needed:

- More Complex Strategies: Implement strategies like card counting and risk estimation to estimate remaining cards and adjust bidding accordingly.
- More Complex Rules: With introduction of more complexities into the game, it will be interesting to explore how much more challenging it gets to prompt GenAI to learn them.
- Self-Learning AI: Develop an AI that learns and adapts its strategy through gameplay experience, potentially using reinforcement learning techniques.

By leveraging GenAI's ability to generate and refine code, we can create increasingly sophisticated opponents for games like Diamonds, leading to more engaging and closer to real game experiences.

5 Appendices

5.1 Appendix A (Transcript of conversations with GenAI)

- Part 1 : [Gemini Chat 1](#)
- Part 2 : [Gemini Chat 2](#)

5.2 Appendix B (Colab file of generated code)

- Diamonds Game - [Diamonds Game Colab](#)