

Developing Strategies for the bidding card game 'Diamonds' with GenAI

GenAI Tool used: ChatGPT

March 26, 2024

1 Introduction

This report documents the reflections/observations made while teaching GenAI the card game "Diamonds", helping it write code for the it and come up with some winning strategies for the same. The main reason for choosing this version of the game is that GenAI seems to lacks any prior knowledge of it, leading to unbiased responses.

2 Problem Statement

The Game of Diamonds has the following rules:

- There can be 3 or 2 players.
- Each player gets a single suit of cards other than the diamond suit.
- The diamond cards are then shuffled and put on auction one by one.
- All the players must bid with one of their own cards; it is a closed bidding system, i.e., the other players don't know which card you bid until the bids are revealed.
- The banker gives the diamond card to the highest bid, i.e., the bid with the most points.
- The ranking system is: $2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < T < J < Q < K < A$
- The highest bidder gets the points of the diamond card added to their score. If there are multiple players that have the highest bid with the same card, the points from the diamond card are divided equally among them.
- The player with the most points at the end of the game wins.

3 Teaching genAI the game

Before delving into the conversation with GenAI, I attempted to understand its comprehension of "The Game of Diamonds." Ofcourse, what it understood differed significantly from what we intended to explain. I initiated a new conversation thread, to prevent its prior knowledge from influencing the discussion, which is why the initial prompt clearly stated that this game was different from what it might already know.

The primary objective of the initial conversation was to explain the game's rules to GenAI and obtain a code capable of running the game for two players.

The subsequent prompts explained the rules of the game, and requested GenAI to explain its understanding, with necessary corrections provided as needed.

To assess GenAI's comprehension of the game, I asked it to provide three examples covering various scenarios that could arise during gameplay, including instances where two players bid cards of the same rank, resulting in point division, or when all three players bid the same ranked card.

While GenAI consistently covered all points outlined in the prompts and provided coherent explanations in text, its ability to implement its understanding was lacking significantly. The initial examples it provided demonstrated minimal to no understanding of the game and it only managed to provide at max two correct examples in each response.

Though through multiple test cases and examples, along with subsequent corrections, GenAI eventually reached a point where I believed it understood the game.

4 Iterating upon strategy

When asked for strategies, GenAI replied with ones that completely disregarded the rules of the game and simply provided a semblance of strategy.

Then, I attempted to get it to write code for a game where the computer randomly selects any card to bid. This task was easily accomplished by using the random function.

After that, I tried to have it write code for a relatively simpler strategy by merely mapping the cards bid to the cards auctioned.

However, the second task was much more complex, and it clearly demonstrated that GenAI struggles to focus on multiple data points simultaneously. If it made a recurring mistake and was prompted multiple times, it would correct it, but it would then neglect some other point that it had gotten right the first time.

In the end, numerous corrections were needed in the code to make it run, and GenAI was also unable to determine why the computer was choosing the wrong cards to bid. It would simply respond with prompts like "I apologize" and provide the code without making any changes."

5 Reflection on Code Generated

While prompting GenAI to write code I realised that in some cases, where GenAI could make assumptions, like assuming that there should be 13 rounds because a standard diamond suit has 13 cards or for a two-player game, where it could have simply removed any suit except diamonds from the list, it instead, chooses to generalise the code by saying rounds should equal the number of diamond cards in the deck, rather than putting in an actual value or using list functions to remove a certain suit that wouldn't be used.

But for some details that were explicitly explained, like shuffling only the diamonds deck and giving the players a single suit each, it somehow assumes that entire deck needs shuffling and distribution, possibly due to its knowledge of solitaire or other card games.

GenAI also tries its best to stick to prior ideas for example, even after giving multiple prompts in order to not make it shuffle the entire deck, it still tried to do so. Even in a new conversation thread where it was given a clear prompt and working code for a two-player game, it still attempted to change the `deal_cards` function code twice, despite the prompt stating that the current code was correct and only the bidding strategy needed changing. On further prompting, it also couldn't replace the modified code with the previous version until explicitly specified what code to replace.

Analysis and Conclusions

Explaining a card game to GenAI in theory was relatively simpler, possibly because much of the data it provided in responses was directly derived from the prompts given. However, getting it to implement this understanding, even with some simple examples, was really challenging. It was tough to get GenAI to disregard some very common assumptions about card games and to get it to focus on multiple rules and facts of the game simultaneously.