Developing strategies for the bidding game ’Diamonds’ with genAI - Gemini

1 Introduction

’Diamonds’ is a card bidding game. It is a 2-3 player game. This report involves the analysis and experience of using genAI, here, Gemini, to set the ground rules, play the game with the genAI and write a working code with it.

2 Problem Statement

To develop strategies and code for the game with genAI.

3 Teaching genAI the game

Since the Gemini couldn’t find the game, I had to give it a prompt detailing the game rules. It seemed to understand quite well when it repeated the rules in a more presentable way. When I asked it to play a game with me, it wanted to clarify few things such as the number of players, the point system, and the bidding limits. I clarified that we are trying it out for 2 players, myself and the genAI. Clarified the point system and that there are no bidding limits.

We started playing. It picked a random card on my behalf and I needed to clarify that I play my own game. When I wanted to start afresh, it changed one of the rules that we each get 5 random cards from which we need to bid. I clarified and we used ’Spade’ as my deck of cards, and ’Heart’ as genAI’s deck of cards. Then on, it changed a couple of rules, one of them that we shuffle our decks, which was harmless but unnecessary, so I forbid that. Another rule that we get some amount of access (knowing the highest valued and least valued card or my bid or we randomly pick the bid) to the bid. I had to clarify that it is not the case and we strategically bid and reveal the bid simultaneously.

We played 3-4 rounds. The working was perfect except minor issues. After one round, it calculated score incorrectly, I corrected it and after every round it asks me for the score since it thinks I am keeping track of the score. When I asked it to keep the track of the score itself, it made mistakes here and there (as Aruvi says, it’s not a calculator). One more issue is that after I reveal my bid, it says I won’t reveal my bid since it’s my strategy unless I have to use another prompt to force it to reveal it’s bid. Also, sometimes, after I reveal my bid, it asks to type ’reveal’ so that we can reveal our bids simultaneously. I don’t know what that was.

I was quite satisfied with the working. Although, I felt there was no strategy the genAI is using.

4 Iterating upon strategy

When asked to suggest some strategies, it gives the follow:

1. Considering the Revealed Diamond:
   (a) High Diamonds: For high-value diamonds (Queens, Kings, Aces), you might want to be more aggressive with your bids, especially if you have strong cards (Jacks, Queens, Kings, Aces) in your hand.
   (b) Low Diamonds: For low-value diamonds (2s-7s), you can be more strategic with your bids. You might use lower value cards (2s-5s) to win these diamonds if they fit your overall strategy.
2. Knowing Your Hand Strength:
   (a) Balanced Hand: If you have a balanced hand with a mix of high and low cards, you'll need to adapt your bids based on the revealed diamond and the potential actions of your opponent.
   (b) Strong Hand: With a hand heavy in high cards, you can be more confident in using them to win valuable diamonds. However, be mindful of not overspending on every diamond.
   (c) Weak Hand: If your hand is weak (mostly low cards), you might want to be more conservative with your bids. You can use lower value cards to try and win some diamonds but also save higher cards for later rounds with potentially more valuable diamonds.
3. Bluffing and Mind Games:
   (a) Overbidding: Sometimes, a well-timed high bid can scare your opponent away from using a strong card, even if your own bid isn't that strong. However, use this tactic sparingly to avoid being predictable.
   (b) Underbidding: Occasionally, a low bid can entice your opponent to use a higher card unnecessarily, especially for lower value diamonds. But be cautious, as your opponent might see through this if done too often.
   (c) Reading Your Opponent: Pay attention to your opponent's bidding patterns. Are they aggressive with high bids, or do they tend to be more conservative? This can help you predict their moves and adjust your strategy accordingly.
4. General Tips:
   (a) Track the Unplayed Diamonds: As the game progresses, keep track of which diamonds have already been revealed. This can help you estimate the likelihood of certain diamonds being revealed later and how valuable they might be.
   (b) Don't Be Predictable: Mix up your bidding strategy throughout the game. Don't always use your highest card, and be willing to use lower cards for some diamonds.
   (c) Consider the Overall Point Value: While winning high-value diamonds is important, don't neglect the lower value diamonds entirely. Every point counts towards the final score.

I raised a point that at any point of time, you know all the cards other player's have (by simply keeping track of their bids).
When asked Gemini to refine the bidding strategy based on this is:

(a) Early Rounds (1-3): Bid mid-range cards to win some diamonds while also revealing some of your own hand. Simultaneously observe the opponents' bidding patterns.

(b) Mid Rounds (4-6): Exploit opponent's weaknesses: if opponent is conservative, you might win lower value diamonds with lower bids; if opponent is aggressive: consider strategically using some of your own high cards to secure valuable diamonds.

(c) Late Rounds (7-13): As the game progresses and high-value diamonds (Qs, Ks, As) remain, prioritize using your own high cards (Js, Qs, Ks, As) to win them; In some cases, a well-timed high bid, even with a weaker card, might scare your opponent away from using a strong card they might be saving; Keep a mental note of the diamonds that haven't been revealed yet. This can help you estimate the likelihood of high-value diamonds remaining and prioritize bids accordingly.

Python code with strategy for the diamonds game:

```
import random

class DiamondsPlayer:
  def __init__(self):
    self.hand = list(range(2, 15))  # Spades (2-14)
    self.opponent_discards = []  # Track opponent's used cards

  def bid(self, revealed_diamond):
    # Early Rounds (1-3): Gather Information
    if len(self.opponent_discards) < 3:
      # Use mid-range cards to reveal some of our hand
      return random.choice(self.hand[5:10])  # Choose from 7s-Qs

    # Mid Rounds (4-6): Exploit Opponent's Weaknesses
    aggressive_opponent = len(self.opponent_discards) > 6  #
       Assumes frequent high bids

    if aggressive_opponent:
      # Prioritize high cards to compete for valuable diamonds
      return self.play_high_card(revealed_diamond)
    else:
      # Use lower cards for some diamonds if opponent seems
         conservative
      return self.play_low_card(revealed_diamond) if
         revealed_diamond < 8 else
```

```
            self.play_high_card(revealed_diamond)

    def play_high_card(self, revealed_diamond):
        # Prioritize high cards for valuable diamonds
        for card in self.hand[::-1]:  # Iterate in descending order
            (highest cards first)
            if card > revealed_diamond:
                return card
        # No high card above revealed diamond, use lowest remaining card
        return self.hand.pop(0)

    def play_low_card(self, revealed_diamond):
        # Use lower cards for less valuable diamonds
        for card in self.hand:
            if card <= revealed_diamond and card not in
                self.opponent_discards:
                return card
        # No usable low card, use highest remaining card
        return self.hand.pop()

    def update_opponent_discard(self, bid):
        self.opponent_discards.append(bid)
        self.hand.remove(bid)  # Assume opponent reveals the used card
            (like your deduction strategy)

# Example Usage (assuming you reveal the diamond card as a variable)
opponent = DiamondsPlayer()
opponent_bid = opponent.bid(revealed_diamond)
# Compare your bid with opponent_bid to determine the winner

# After each round, update opponent's discards with your bid
opponent.update_opponent_discard(your_bid)
```

5 Analysis and Conclusion

The Gemini took a while and a couple of incorrect prompts to understand the game and play correctly. The strategies were reasonable. The code generated was also reasonable.