Developing strategies for the bidding card game 'Diamonds' with GenAI.

Nikita Kiran

March 2024

1 Introduction

'Diamonds' is a card bidding game that can be played by two or three players. The rules of the game are as follows -

- Each player has 13 cards each with values ranging from 2 to 14.
- The game starts with a random diamond card generation.
- Every time a diamond card is displayed, the players make a bid using their existing cards, being oblivious to the opponent's bid value.
- The player with the higher bid wins and gets the diamond card.
- If the bids are equal, the points are split.
- The card bid by the players is discarded and cannot be used again.
- This continues until the diamond card set is exhausted.
- The winner is the player with the higher sum value of the diamond cards.

2 Problem Statement

To teach the GenAI assistant the rules of the card bidding game 'Diamonds' and then develop various strategies for the game using its help.

3 Methodology

For this task, I have used ChatGPT as the GenAI assistant. The steps followed to achieve this task were -

- Listing out the rules of the game to ChatGPT
- Asking it to explain its understanding of the game

- Clarifying the rules as necessary
- Playing the game with ChatGPT
- Asking for a code to play this game
- Pointing out errors in the code generated
- Once a working code was generated, asking for different winning strategies

4 Reflections

4.1 Teaching ChatGPT the game

- I found that ChatGPT understood the rules in the first try and could explain it accurately.
- However, on playing the game with it, it made mistakes such as revealing its bid to me before I could place my bid.

4.2 Asking ChatGPT for the code

- On being asked to give a code that was divided into functions, ChatGPT provided a pretty accurate functional decomposition for the problem. It generated the necessary functions such as **initialize_deck, player_bid, computer_bid, play_round, calculate_score and declare_winner**.
- The computer bid was generated through a random function.
- However, ChatGPT made several implementational errors in generating the code.
- Most prominently, it kept on forgetting that bid cards could not be reused.
- On several instances, it mixed up the computer's and players remaining bids.
- Even in the code, ChatGPT repeated the error of displaying the computer bid before asking for user bid.

4.3 Asking ChatGPT for the strategy

- After the code was successfully running, ChatGPT was prompted to replace the random choice of bids with some sort of strategy.
- Initially it provided incorrect strategies like bidding a low value for a high diamond card.
- On being told that the strategy was incorrect, it changed and came up with some interesting strategies.

- It however constantly forgot game rules and generated strategies which did not comply with the rules.
- The strategies generated were a mix of deterministic and random strategies.
- Some sample strategies are given below -

```
def computer_bid(diamond_card, previous_bids):
    """Generate a bid for the computer based on a simple strategy."""
    remaining_bids = list(set(range(2, 15)) - previous_bids)
    remaining_bids.sort()  # Sort the remaining bids

    # Determine bid based on diamond card value
    if diamond_card > 10:
        bid = remaining_bids[-1]  # Bid highest available if diamond card is high
    elif diamond_card < 5:
        bid = remaining_bids[0]  # Bid lowest available if diamond card is low
    else:
        bid = random.choice(remaining_bids)  # Bid randomly if diamond card is mid-range

    return bid

def pattern_based_player_bid(diamond_card, user_previous_bids, opponent_previous_bids):
    """Prompt the user for a bid based on pattern-based strategy."""
    if len(user_previous_bids) % 2 == 0:
        available_bids = sorted(set(range(2, 15)) - user_previous_bids)
        bid = min(available_bids) if available_bids else 2  # Bid the lowest available bid f
    else:
        available_bids = sorted(set(range(2, 15)) - user_previous_bids)
        bid = max(available_bids) if available_bids else 14  # Bid the highest available bid
    return bid

def counter_strategy_player_bid(diamond_card, user_previous_bids, opponent_previous_bids):
    """Prompt the user for a bid based on counter-strategy."""
    if max(opponent_previous_bids) > 10:
        available_bids = sorted(set(range(2, 15)) - user_previous_bids)
        bid = min(available_bids) if available_bids else 2  # Bid the lowest available bid i
    else:
        available_bids = sorted(set(range(2, 15)) - user_previous_bids)
        bid = max(available_bids) if available_bids else 14  # Bid the highest available bid
    return bid
```

5 Conclusion

Though ChatGPT seemed to understand the rules of the game pretty easily, it kept forgetting the rules while generating code and strategies for the game. The strategies generated seem to be derived from other popular bidding games.

6 Appendices

1. Running code - https://colab.research.google.com/drive/10fVlDOYpwWMa2DAp0me8R-eFAntesk3V?usp=sharing
2. Transcript available on github -https://github.com/NikitaKiran/WE_Module3/tree/main/Diamonds