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

Bhavani Chalasani

March 2024

Problem Statement

Create a report on the teaching and generation of a solution by genAI of the Diamonds card game.

Methodology

1. Prompt AI with these rules, explaining the game.
2. Play against it to ensure it understood the game.
3. Iterate upon strategy.
4. Play against it to ensure it uses the strategies.
5. Prompt it for functions to implement when coding the game as well as corresponding code.

Teaching genAI the game

- The initial prompt for the game should involve the rules of the game which are as follows:
  - Each player receives a set of cards, excluding the diamonds.
  - Diamond cards are shuffled and auctioned one by one.
  - Players bid with one of their own cards face down. These cards must come from the remaining cards the player has in their deck.
  - The highest bid, determined by the card's point value, wins the diamond card.
  - If multiple players bid the same highest card, they equally share the points from the diamond card.

  - Points earned from the diamond card are added to the respective player's score on the scoreboard.

- The next prompt should ask AI to play the game with you and proceed to play the game with the AI, correcting it in case it makes any mistakes.

- In my case, common mistakes made by the AI included not checking if the card I bid was in my deck, incorrectly considering the ordering hierarchy of the cards and incorrectly counting the points obtained by each player.

- Keep playing this game with the AI until you play one whole game during which the AI does not make any mistake, and where there is no need for corrections and the game runs smoothly. This implies that the AI has understood the game well.

- Then prompt the AI for strategies regarding the game, asking it to elaborate on points that may seem more important. Discuss these points with the AI if necessary.

- Play the game again with the AI, asking it to use these strategies. This will help better understand the strategies. In my case, some strategies AI suggested included prioritizing bidding on cards with higher point values, bluffing and taking risks based on your situation.

- Finally, ask AI to generate code for the game and iteratively test and perfect the code using genAI.

Code

```
import random

# Define the hierarchy of cards
card_values = {
    '2': 2, '3': 3, '4': 4, '5': 5, '6': 6, '7': 7, '8': 8, '9': 9, '10': 10,
    'Jack': 11, 'Queen': 12, 'King': 13, 'Ace': 14
}

# Function to generate player decks
def generate_player_decks(num_players, num_cards_per_player):
    deck = [f"{value} of {suit}" for value in card_values for suit in ['Hearts',
    player_decks = {}
    for i in range(num_players):
        player_decks[f"Player {i+1}"] = random.sample(deck, num_cards_per_player
    return player_decks

# Function to generate diamond cards
def generate_diamond_cards(num_diamonds):
    deck = [f"{value} of Diamonds" for value in card_values]
    return random.sample(deck, num_diamonds)

# Function to shuffle the diamond cards
def shuffle_diamonds(diamonds):
    random.shuffle(diamonds)

# Function to conduct the auction for each diamond card
# Function to conduct the auction for each diamond card
def auction_diamonds(players, diamonds):
    bids = {}
    for player, deck in players.items():
        print(f"\n{player}'s turn:")
        print(f"Diamond Card: {diamonds[0]}")
        print(f"Your hand: {deck}")
        valid_bid = False
        while not valid_bid:
            bid = input("Enter your bid: ")
            if bid in deck:
                valid_bid = True
            else:
                print("Invalid bid. Please choose a card from your hand.")
        bids[player] = bid
    return bids

# Function to determine the winner of each diamond card auction
def determine_winner(bids):
    highest_bid = max(bids.values(), key=lambda x: card_values[x.split()[0]])
    winners = [player for player, bid in bids.items() if bid == highest_bid]
    return winners

# Function to update the scores on the scoreboard
def update_scores(scores, winners, points):
    for winner in winners:
        scores[winner] += points // len(winners)

# Function to check if the game should end
def check_end_game(scores):
    # Game ends if any player reaches a certain score threshold or if all diamon
    threshold = 50
    if any(score >= threshold for score in scores.values()):
        return True
    return False


# Main game loop
```

```
def main_game_loop(player_decks, diamonds):
    scores = {player: 0 for player in player_decks}
    display_scores(scores)
    for diamond in diamonds:
        bids = {}
        for player, deck in player_decks.items():
            print(f"\n{player}'s turn:")
            print(f"Diamond Card: {diamond}")
            print(f"Your hand: {deck}")
            bid = input("Enter your bid: ")
            bids[player] = bid
        # Determine winner and update scores
        for player, bid_card in bids.items():
                player_decks[player].remove(bid_card)
        winner = determine_winner(bids)
        update_scores(scores, winner, card_values[diamond.split()[0]])
        display_scores(scores)


# Main function to run the game
def main():
    # Generate player decks
    num_players = 2
    num_cards_per_player = 5
    player_decks = generate_player_decks(num_players, num_cards_per_player)

    # Generate diamond cards
    num_diamonds = num_cards_per_player
    diamonds = generate_diamond_cards(num_diamonds)

    # Shuffle diamond cards
    shuffle_diamonds(diamonds)

    # Start the main game loop
    main_game_loop(player_decks, diamonds)

if __name__ == "__main__":
    main()
```

Analysis and Conclusion

Analysis

1. On giving the initial prompt to genAI, I also asked it to play the game with me, if it had understood the rules properly.

2. We started to play the game, but I noticed an immediate misunderstanding in which the AI did not check whether the card I bid was part of my remaining deck and it also bid cards outside of its own deck.
3. Therefore I had to add another rule, saying that when a card is bid, it must be from the player's remaining card deck to be valid.
4. After the AI understood this rule, we continued playing the game but it started to add diamonds into the card deck that it dealt, so I had to correct that mistake.
5. The AI also made a mistake while counting the points, it did not accumulate the points, but rather took the value of the last win of that player.
6. After explaining how to count the points at the end of the game, I asked the AI to play one more round with me so that I could be confident that it understood the game.
7. The next round went smoothly so I assumed that the AI had understood the game well.
8. I then prompted the AI for strategies and it gave me strategies related to prioritizing bidding on cards with higher point values, bluffing, taking risks based on your situation (bidding low when you have a comfortable enough points and bidding high when you need points) and bidding a card of a certain value based on the value of the diamond card.
9. I then asked genAI to play the game with me using the strategies.
10. The game went smoothly, with the AI explaining where and why it was using these strategies
11. I then prompted it for code, tested the code and iteratively asked it to fix any issues with the code. In my case, it did not remove the card from the player deck after it was bid so I had to fix that.

Conclusion

1. The teaching process proved effective in conveying the rules of the Diamonds card game to the AI. Through iterative play sessions, the AI gradually improved its understanding, ultimately demonstrating proficiency without making errors. This highlights the capability of genAI to quickly grasp game dynamics through interactive learning.
2. The strategies generated by the AI exhibited a mix of conventional and innovative approaches. Prioritizing bidding on high-point-value cards and risk-taking emerged as key tactics.
3. The code generated by AI was devoid of any obvious errors or misunderstandings of the game and on prompting it, it was able to fix any issues and imperfections in the code.

4. In conclusion, the collaboration with AI in teaching and playing the Diamonds card game yielded promising results.

GenAI Transcript

Here is the link: https://chat.openai.com/share/903ff92c-a2d4-4c17-844f-4f9daccb3554