Dealing with GenAI

Teaching Gemini a new card game

Dhvani Thakkar, WE Cohort 5

Generative AI Assignment 5

1 Introduction

1.1 About

In this course, starting March 2024, as a part of module 3, we are learning the limits of different GenAI Chat-bots, and what went behind creating them. We aim to experimentally get an idea of when, where and how to use them to help with coding tasks assigned to us.

1.2 Diamonds, The Card Game

A different game than that available on the internet, Asokan taught this game to us in the Summer 2023 Boot-camp.

Setup
Players: 2 or 3 players can play, but we are considering the two player version.
Deal the Decks: Each player receives a full suit (13 cards) of either spades, hearts or clubs. These are the bidding cards.
Diamonds on auction: Set aside the 13 diamond cards, shuffled & face down. These are the cards to bid for.

Auction Rounds
Diamond auctioned: The shuffled diamonds are revealed one by one.
Secret Bidding: Each player secretly chooses one card from their hand and places it face down. This is their hidden bid for the diamond. Once a bidding card has been used, it cannot be bid with again.
Highest Bid Wins: All bids are revealed simultaneously. The player that bid the card with the highest point value wins the diamond.
Ties & Sharing: If players have the same highest value bids, they share the diamond's points equally (even with a decimal value).
Update Score: Winner of the round adds the diamond's point value to their score.

Winning
Thirteen auction rounds are played in the game.
The player with the most points at the game's end wins.

2 Problem Statement

2.1 Expected Format

A technical report in LaTeX, with the result of prompting GenAI to create a computer program that uses some optimizing strategy to play the game of Diamonds against you, including:

- Introduction/Objective/Problem statement
- Prompts given
- Description of the path to "teaching"
- Strategy discussed
- Code for strategy & practical results of playing

Word Count: 500-1000 words.

3 Teaching Gemini the Game

3.1 How it went

I started with giving my rules notes from the boot-camp and asking for a Python code framework. Getting code with incorrect logic, I would ask it what it did not understand, and it would claim to understand perfectly. Correcting each inconsistency was time-consuming. Thus, I started with asking what it did not understand with the rules and it questioned the unclear parts/ nuances in rules & scoring.

Gemini is very clear on the rules once explained, and managed to give almost correct (if inefficient) OOP Python code for the game-play, but soon enough, it forgets the rules it strongly asserted some prompts ago.

I prompted for strategies to play to make sure it understands the game. It could recognise the general flow and idea to suggest bluffing, tracking cards, balancing-bids and different strategies for game phases.

Post minor clarifications on number of rounds to be played[2] & tie-breaking scores, I asked it to simulate the game[3].

As seen in Yahtzee[4], Gemini cannot keep track of scoreboards, losing track of bidding cards already used [5]& cards available. Surprisingly, it perfectly drew the shuffled diamond cards on auction one-by-one, without repeating, and knew which round was being played.

It needed prompts for basic addition[6], but when told it was wrong, changed multiple addition results.

It occasionally confused the order of tasks in a round. Once it understood what was already considered in scoring for a round, it corrected me when I asked it to modify a score[7].

It started forfeiting rounds near the end, even though it had higher cards left than me.

3.2 Main Prompts

Starting a chat: Initially, my prompt was: "I will list the rules of a card game called diamonds. Tell me what you do not understand from the rules...[8]"

Once the game was clear to the GenAI, I prompted "tell me all the rules of the diamonds game in a way you will be able to understand in a new chat", henceforth, starting with this reply[9] for new chats with different purposes.

Framework contents: The object oriented[13] framework[14] contains Classes:

- Card
- Suits: Subclass DiamondSuit that can shuffle
- Player: Subclasses HumanPlayer, RandomPlayer & BotPlayer
- DiamondsGame

BotPlayer has supporting decide_bid & divide_median functions to facilitate choosing bids. DiamondsGame sets up the game, auction round & adds players. A "main" function can add required players to the game, play 13 rounds & declare winner(s).

4 Iterating upon strategy

4.1 Where it started

The response provided when asked to "what are some strategies to play this game?"[10] was very different to when asked "give me a strategy to play this game"[11].

I iterated on this with[12]:

- Tracking player cards, opponent cards, diamonds revealed till now
- Function to decide a bid by the bot
- Considering relative value of the diamond in those remaining
- Trying bring it away from Early, Middle and Last rounds
- Identifying strength of opponent's cards
- Aggressive/Defensive for the highest and lowest diamond among those left

Although I asked for contingencies, returned code threw errors. Many manual fixes were required.

4.2 Strategy chosen

- Highest remaining diamond: Bidding highest card if it can beat the opponent, else lowest card.
- Lowest remaining diamond: Bidding lowers card
- Comparing card averages of opponent and my cards for threshold
- Relative diamond value compared to threshold for a median card in appropriate half of the hand.

5 Analysis and Conclusion

Results The Colaboratory notebook resulting from this project[14], complete with manual testing, debugging and additional conditions, as well as cleaning up is able to simulate the game with:

- Two/three human players
- A human/random choice player and a bot
- A random choice players and human bots

with condition of total 2-3 players for the game to start.

When tested the bot-player with a random choice player for 5,00,000 two-player games, the bot won 78.6% of the games.

Conclusions

- Re-prompting to change certain part of the result may be very useful when Gemini allows it.
- Editing the last prompt given to word it clearer is a great learning opportunity.
- Gemini does a great job cross-checking and referencing its sources from Google search.
- Even when self written correct code is referenced for a simulation, GenAI is likely to hallucinate.
- However many contingencies are asked for, testing generated code and doing careful manual editing is necessary.

References

1. Link to the chat: Explaining rules, initial python code, basic strategies and game simulation.

2. Getting vague answers for number of rounds to be played in the rules, I prompt: "number of bidding rounds should be 13 right?" to get"
   You're absolutely right! I apologize for the mistake in my previous explanation. There are indeed 13 bidding rounds in the Diamonds game, one for each of the hidden diamond cards. Each round follows the bidding process where players secretly choose a card from their hand and the highest bidder wins the diamond.

3. Prompt: "simulate playing the diamonds game with me" to get: Absolutely! Let's play a simulated game of Diamonds. Here's how it will work:

   Card Dealing: I'll randomly assign you a suit (spades or hearts) and keep the other for myself. We won't reveal them yet.
   Bidding Rounds: I'll announce each diamond card being revealed. You'll then tell me which card from your hand (spade or heart) you want to bid with (e.g., "7 of Spades"). I'll choose a card from my hand secretly. **Reveal and Scoring:** We'll

reveal our bids, and the player with the highest card value (based on Ace being the highest) wins the diamond and its points. Ties will be split equally.
Keeping Track: I'll keep track of the score and remaining cards in each hand (simulated).
Ready to begin? Let me know if you'd like to proceed, and I'll deal the cards!

4. Yahtzee with ChatGPT: Learning and playing a simulation of the game.
5. I had used the 9 of Spades before, not the 8 of Spades:
   However, there's a small catch in Diamonds! You can't bid a card with the same value you've already used. Since you used the 8 of Spades in a previous round, it's no longer available for bidding.
   Would you like to choose another spade card from your remaining hand, or would you like to forfeit this round (meaning you wouldn't bid any card and I would automatically win the 8 of Diamonds)?
   It did not let me regenerate that part of the response with the prompt: i have used 8 of spades before but 9 of spades is available so i can bid with it in round 7. I had to correct in the next prompt.
6. In a game round, it had miscalculated my score. I had to explicitly tell it 36+9=45. It forfeited the next round. My updated score was incorrectly found to be 62, and when asked to retry the round, it claimed 62+8=80. When I prompted: "62+8 is not 80", I got:
   You: 45 points (Queen of Diamonds + Ace of Diamonds + 9 of Diamonds + 8 of Diamonds) + 8 points (winning the 8 of Diamonds in round 7 replay) = 53 points
   Me: 22 points (7 of Diamonds + 5 of Diamonds) = 22 points
   Here, I then prompted the part of the response "7+5=12" to get the right answer.
7. I made the mistake of forgetting it already updated my score, and prompted "53+8=61. Update the scores." and got:
   I cannot update the score calculation. There might be a mistake in the formula, as 53 + 8 equals 61, and you've already won 8 diamonds.
   Here's the score update without the formula change:
   You: 53 points (Queen of Diamonds + Ace of Diamonds + 9 of Diamonds + 8 of Diamonds) = 53 points
   Me: 12 points (7 of Diamonds + 5 of Diamonds) = 12 points
   The lead is significant in your favor with only 6 more diamonds to reveal. Are you ready for the next round?
8. Complete prompt:
   I will list the rules of a card game called diamonds. Tell me what you do not understand from the rules:
   player 1 gets a suit of 13 spade cards a-k, player 2 gets a suit of 13 heart cards a-k, and the clubs suit is put aside for a two player version. The diamond cards are randomly put on auction one by one. All the players must bid with one of their own cards without knowing other's current bid.
   The banker gives the diamond card to the highest bid, i.e. the bid with the most points. card points: 2<3<4<5<6<7<8<9<T<J<Q<K<A.
   The winning player gets the points of the diamond card to their column in the table. 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 wins at the end of the game. 13 rounds are played.

9. **Diamonds Card Game:**
A fun and strategic card game for two players. Here are the rules: Tell me what you do not understand:
Preparation:
Deal the Decks: Each player receives a full deck (13 cards) of either spades or hearts (chosen randomly). These are your bidding tools.
The Hidden Treasure: Set aside the 13 diamonds cards face down. These are the diamonds you'll be bidding for
The Bidding Frenzy:
One Diamond at a Time: We'll reveal the diamonds one by one, just like uncovering a single gem from the treasure chest.
Secret Bidding: Here's the exciting part! Each player secretly chooses one card from their hand (spades or hearts) and places it face down. This is your hidden bid for the diamond.
Highest Bid Wins: We reveal everyone's bids. The player who played the card with the highest point value (based on a pre-defined system, like Ace being the highest) wins the diamond! They claim the treasure! **Ties & Sharing:** If players have the same highest value card, they share the diamond's points equally (even if it's a decimal value). It's like splitting the treasure among friends!
Update Your Score: The winner of each round adds the diamond's point value to their score. The player with the most points at the game's end wins!
Special Considerations:
Blind Bidding: You don't know what cards others are bidding with, adding an exciting element of risk and strategy.
Unique Cards Per Bid: You can't use the same card you bid with again in future rounds. Choose wisely!
Bidding Resources: The unused cards in your hand (spades or hearts) are what you use throughout the game to win diamonds. Manage them strategically!
Key Points to Remember: There are 13 bidding rounds, one for each diamond. You use cards from your dealt suit (spades or hearts) for bidding. Used cards cannot be used again for bidding.

10. Here are some strategies you can employ to increase your chances of winning at Diamonds:
Bidding Strength:
Balance Offense and Defense: While aiming for high-value diamonds, consider keeping some lower-value cards (especially higher spades/hearts) for later rounds with potentially more valuable diamonds. Read Your Hand: Analyze your dealt suit (spades or hearts) and identify strong and weak points. Prioritize bidding with strong cards for high-value diamonds you anticipate.
Playing the Odds:

Track Revealed Diamonds: As diamonds are revealed, keep track of which have already been won and the remaining values. This can help you predict the potential value of upcoming diamonds and adjust your bidding strategy.
Observe Your Opponent: Pay attention to the cards your opponent bids, particularly if they consistently win with high or low values. This might indicate their remaining high cards and influence your bids. **Bluffing and Information War-**

fare: Sacrifice Bids: Occasionally, bid a lower-value card to "win" a less valuable diamond. This can free up a higher-value card for a future, potentially more valuable diamond.
Calculated Risk: Sometimes, bidding a high card for a less valuable diamond can be strategic. It might force your opponent to use a high card they were saving, putting them at a disadvantage later. **Resource Management:**
Prioritize Early Rounds: Diamonds revealed early in the game tend to have lower point values. Use your weaker cards for these rounds to preserve your high-value cards for potentially more valuable diamonds later.
Calculated Discard: If you have a very weak hand (mostly low spades/hearts), consider discarding a high card early on. This frees up space to "win" a low-value diamond you might not otherwise win, allowing you to keep your remaining high cards for later.
Remember: Diamonds is a game of both chance and strategy. While these strategies can give you an edge, there's always an element of luck involved.
Good luck playing Diamonds!

11. Here's a combined strategy for Diamonds, incorporating both resource management and playing the odds:
Early Rounds (Diamonds 1-5):
Prioritize Discarding Low Spades/Hearts: Aim to get rid of your weakest cards (2-5) in the first few rounds. This frees up space in your hand for potentially valuable diamonds later.
Low-Ball Bidding: Use your remaining low-value cards (6-8) to "win" the lower-value diamonds (1-5). This preserves your high cards for later rounds.
Mid Rounds (Diamonds 6-9):
Shift Focus: As the revealed diamonds increase in value, start prioritizing bids with your mid-range cards (9-11).
Track Revealed Diamonds: Keep a mental note of the diamonds already won. This can help you estimate the potential value of upcoming diamonds. Late Rounds (Diamonds 10-13):
Go Big or Go Home: Save your highest value cards (Jacks, Queens, Kings, Ace) for the final rounds. These diamonds are likely to be the most valuable.
Calculated Risk: Consider using a high card to "win" a mid-value diamond if you suspect your opponent might use a similarly high card. This can give you an advantage later. General Tips:
Observe Your Opponent: Pay attention to the cards your opponent bids, particularly if they consistently win with high or low values. This might indicate their remaining high cards and influence your bids. Adapt and Adjust: This strategy is a guideline. Be flexible and adjust your approach based on the revealed diamonds, your remaining hand, and your opponent's plays.
Remember, luck plays a part in Diamonds. However, by employing these strategies and adapting as the game progresses, you can increase your chances of outsmarting your opponent and winning the most valuable diamonds!
12. Link for the Gemini chat for strategies to play the game and their code
13. Link to Gemini chat providing OOP Python Framework for the game
14. Link to the Colab Notebook containing the final code