Teaching GenAI a New Card Game and Developing Strategies

Sayali Kadam, WE Cohort 5

March 26, 2024

1 Introduction

This report explains how we can use GenAI to write neat and efficient code for a problem it hasn't encountered before. We'll discuss the right way to ask GenAI for help and how to improve its responses.

1.1 Report Layout

1. Problem Statement
2. Links to Refer
3. Teaching GenAI
4. Getting the Solution
5. Conclusion

2 Problem Statement

We'll start by giving GenAI a new challenge to solve. We have used the Diamond Game as our problem statement. The diamond game has following rules:

- The Diamond Game can be played by either 2 or 3 players.
- Each player receives a set of cards excluding diamonds.
- Players make bids using cards from their respective sets.
- The banker selects a card from the diamond suit, which is awarded to the player with the highest bid. In the event of equal bids, points are shared.
- Points are assigned to cards based on the following hierarchy: $2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < T < J < Q < K < A$.
- After thirteen rounds, the player with the highest accumulated points emerges as the winner.

3 Links to Refer

I have used both Gemini and ChatGPT to solve this problem. I'm attaching the chats with Gemini and ChatGPT below to refer while going through the next sections of the report:

- Gemini: `https://g.co/gemini/share/cc609336dc97`
- ChatGPT: `https://chat.openai.com/share/53d4efb4-b94b-4594-9425-8b492d3508a7`

4 Teaching GenAI

1. **Explaining the Game:** I started by telling GenAI all about the game and then asked it to explain the game back to me. This helped me see if GenAI got the rules right.
2. **Playing the Game:** Next, I played a few rounds of the game with GenAI. This way, GenAI could see how the game is played and how points are earned.
3. **Asking for Strategies:** After GenAI understood the game, I asked it for some tips on how to win. I wanted to see if GenAI could come up with smart ideas for playing.
4. **Writing Code:** Finally, I asked GenAI to write some code for playing the game. I used its understanding and strategies to create code that could play the Diamond Game effectively.

5 Getting the Solution

The code presented several issues. Upon execution, it became evident that while GenAI understood the game rules, it struggled to implement them accurately within the code.

1. **Gemini Attempt:** Initially, I provided Gemini with an explanation of the game and requested it to generate code. Despite producing a clear code structure, the generated code was incomplete. Prompting Gemini to complete it yielded identical incomplete results.
2. **Switch to ChatGPT:** Subsequently, I switched to ChatGPT. After explaining the game, instead of requesting code generation, I provided Gemini's incomplete code to ChatGPT and asked it to fill in the missing parts. However, the resulting code lacked clarity in understanding the rules.
3. **Refining with ChatGPT:** To address this, I employed a step-by-step approach with ChatGPT. I systematically provided each game rule as a prompt and requested specific adjustments to the code accordingly. Through successive iterations of refining prompts and code adjustments, a functional and playable code was eventually obtained.

6 Conclusion

By using both Gemini and ChatGPT, along with specific instructions and making small improvements bit by bit, I managed to get a good solution. Making sure that the GenAI understands the rules and keeps the rules in mind while generating code is important and a difficult task to achieve.