

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

Ragini Metlapalli

March 2024

## 1 Introduction

This report aims to record the process of teaching GenAI how to play the game Diamonds and asking it to come up with strategies. This is a task given as part of the WE Module 3 GenAI course. The assignment is to teach GenAI the game's rules, help it comprehend them, help it formulate strategies, and help it generate Python code that carries out the strategy.

## 2 Rules Of The Game

- Each player gets a suit of cards other than the diamond suit.
- The diamond cards are then shuffled and put on auction one by one.
- All the players must bid with one of their own cards face down.
- The banker gives the diamond card to the highest bid, i.e. the bid with the most points.
- 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.

## 3 Methodology

I completed this work using Gemini. I started by asking as to whether it was familiar with the game of Diamonds to make sure that it was different from

the other games it had considered so as to prevent any misunderstandings that would arise later. Then I explained to Gemini the rules of the game, corrected the misunderstandings, asked it to come up with strategies to play the game, prompted it to generate Python code to simulate the game with one of the strategies that it came up with, and lastly asked it to refine the strategy.

## 4 Teaching GenAI The Game

At first, Gemini had trouble following the game's instructions. It told me its knowledge of the game's rules once I described them to it, and without my having to ask, it also identified things that were unclear. Thus, this simplified my duty of ensuring that it had comprehended everything. Every time I taught the rules to it, it told me its understanding, so I could correct it if it had picked up on something differently. With the exception of the rule about the number of players, almost all of the other rules were easily understood by it. It foolishly assumed that the game could be played by any number of players at first, even though it was clear about all the other rules (such as the one about each player receiving a single suit, which clearly states that there can only be a maximum of 3 players). It took me five or six iterations to explain the starting hand distribution for two and three players, fixing any errors to make sure it was understood.

## 5 Strategies Discussed

Gemini, in my opinion, provided some excellent gaming strategies. It mentioned "balance bidding," which highlighted the significance of striking a balance between obtaining valuable diamonds and holding onto powerful cards for potential later scoring, in its section on general strategies. Additionally, the two main tactics were "bluffing" and "card counting," which makes players to keep track of played and auctioned cards in order to predict their opponents' hands. In addition to general strategies, it provided methods for two- and three-player games as well as specific plans that adjusted to the way the game progressed (strategies for early game, mid game, end game). It could really understand clearly that situational adaptation is the key and that there is no such thing as a perfect plan.

## 6 Generation And Working Of The Code

Gemini used a strategy that focused on bidding the lowest card in the player's hand that is below half of the revealed diamond card value when I asked it to generate a code to play the game. This code was written in Python and could

only be played for one round (asked it to simulate the entire game later).The player bids the lowest value card in their hand as a final option if no appropriate cards are available.(this was added later after I asked it to include the scenario in which the strategy it was given didn't work.) It even attempted to include anything about a player who would simply pass rather than play, which was against the rules that, only a short while ago, it appeared to grasp. The strategy that it incorporated aims to minimize the bid value while participating in the auction. It adheres to the fundamental principle of bidding cautiously to avoid revealing strong cards unnecessarily. However, the strategy lacked sophistication by not considering advanced tactics like opponent analysis,or card counting which it listed as strategies in just the previous prompts.

There were numerous occasions when it failed to finish writing the code and halted suddenly in the middle of the procedure. It gave me code that required modifications before it could be executed properly; it could not be ran straight. However, it began to recognise my corrected version of the code and made changes to it when I gave it back in the prompt and asked it for some additional modifications (related to the game being played against human and computer with computer using the strategy that it came up with). It totally went off course and began taking into account the player's suit when requested to improve the approach in the code to perform better (which was entirely illogical considering how this game was played). It seemed, in my opinion, to forget the context alone.

## 7 Conclusion And Analysis

The practical outcomes of playing against Gemini's working approach demonstrate that it was not particularly strong and could be readily overcome and won upon. Though it required numerous iterations, GenAI performed rather well given the complexity of the task and the fact that it was unaware of the problem. This process emphasized the importance of clear communication and feedback in educating AI models. Overall, this was a really effective learning experience.