Teaching genAI Diamonds: A Game of Bidding and Bluffs

Anjali Dahiya

March 26, 2024

Abstract

This report details the process of prompting genAI, a large language model, to create a basic Diamonds game playable against a human opponent. It further explores the development of a simple strategy for genAI's bidding and card play.

0.1 The Game of Diamonds

Diamonds is a trick-taking card game for two to four players. Players use a standard deck of 52 cards, with Diamonds being the permanent trump suit. Each round involves bidding on the number of tricks a player believes they can win, followed by playing those tricks. Points are typically awarded for overtricks (winning more than bid) or penalized for undertricks (winning less than bid).

0.2 Teaching genAI the Game

genAI was first introduced to the basic rules of Diamonds, including card ranking, trick-taking mechanics, and bidding. Code snippets were provided to illustrate the game flow and card comparisons. This initial exposure equipped genAI with the fundamental knowledge required to play.

0.3 Bidding Strategy

A simple bidding strategy was discussed for genAI. Here's the core principle:

0.3.1 Hand Strength Evaluation

genAI analyzes its hand, considering the number of Diamonds and high cards (Queens, Kings, Aces) present.

0.3.2 Conservative Bidding

Based on the hand strength, genAI bids a conservative number, aiming to win slightly more than half the tricks. This reduces the risk of significant undertricks and point penalties.

0.4 Code for the Strategy

0.5 Structure and Libraries

The provided Python code leverages its simplicity and readability for implementing the Diamonds game. The code imports the ‘random‘ library, which serves two purposes:

* Shuffling the deck to ensure a random card distribution before dealing. * Introducing randomness into the AI’s decision-making process, particularly during bidding and card selection.

0.6 Game Data Representation

Cards within the game are represented as strings for clarity and ease of manipulation. For instance, a card might be represented as ”Queen of Spades”. To facilitate iterating through cards and evaluating hand strength, the code maintains separate lists for suits (e.g., ”Clubs”, ”Diamonds”, ”Hearts”, ”Spades”) and ranks (e.g., ”2”, ”3”, ”4”, ..., ”Ace”).

Two separate lists, ‘player$_h$*and‘and‘ai*$_h$*and‘, holdthecardsdealttothehumanplayerandtheAIp*

0.7 Gameplay Functions

The core functionalities of the Diamonds game are implemented through various functions:

* ‘deal$_c$*ards‘ : Thisfunctionshufflesthedeckusingthe‘random‘libraryanddistributesthecar*
‘display$_h$*and‘ : Thisfunctionallowsthehumanplayertoseetheircardsbydisplayingthecontentso*
‘get$_b$*id‘ : Thisfunctionpromptsthehumanplayerfortheirbid, ensuringtheenteredvaluefallswit*
‘get$_a$*i*$_b$*id‘ : ThisfunctionsimulatesabasicAIbidforgenAI.ItconsidersthenumberofDiamondsa*
‘play$_t$*rick‘ : Thisfunctionhandlesasingleroundofplayingcards.Ittakesuserinputforthecardpl*
followingrulesandtrumpusage, anddeterminesthewinnerofthetrickbasedoncardrankandsuit.

0.8 Game Loop

The core gameplay loop is implemented within the ‘play$_g$*ame‘function.Here′sabreakdownofits*

1. **Shuffling and Dealing:** The deck is shuffled using ‘random.shuffle‘, and cards are dealt to both players using the ‘deal$_c$*ards‘function.*2.*∗∗BiddingPhase* :

$**BoththehumanplayerandgenAIsubmittheirbidsusingthe`get_bid`and`get_ai_bid`functions, resp$
$*Trick-TakingLoop: **Thegameentersaloopthatiteratesforthetotalnumberoftricks(typicall$
$*Theleadingsuitforthecurrenttrickisidentifiedbasedontheprevioustrickwinner'scardsuit(ifa$
$Thehumanplayerselectsacardfromtheirhandusinguserinput.*TheAIselectsacardfollowingsu$
$followingrulesorplaysaDiamondifunabletofollow.*Thewinnerofthetrickisdeterminedbasedo$
$Theplayedcardsfrombothplayersarestoredforpotentialfutureanalysis(commented-$
$outsectionmentionedintheoriginalcode).4.Afteralltricksareplayed, thegameends, andtheresul$

0.9 Improvements (Optional)

The current implementation offers a foundational framework for genAI's Diamonds gameplay. Here are some potential areas for improvement:

* **Enhanced AI Bidding:** The current AI bidding strategy is basic. It could be refined to consider factors like suit distribution within the hand and formulate bids based on potential trick-winning strategies. * **Error Handling:** More robust error handling could be implemented to address invalid card selections during trick play. * **Scoring System:** Integrating a scoring system would allow for tracking the overall winner after multiple rounds of playing

0.10 Playing Against genAI

Playing against the AI involved:

0.10.1 Human Player Viewing Hand

The human player uses the 'display_hand' function (assuming Python is used) to see their cards.

0.10.2 Bidding Phase

Both the human and genAI submit their bids based on hand strength.

0.10.3 Trick-Taking

Each trick involves the human player selecting a card, followed by genAI choosing a card based on suit-following or playing a trump (Diamond) if unable to follow. The winner of the trick leads the next one.

0.11 Results

Playing against genAI with the implemented strategy yielded interesting results. Here are some observations:

* **Conservative Bidding:** genAI generally placed safe bids, minimizing the risk of undertricks. This made it challenging for the human player to win a significant number of overtricks and secure high scores. * **Random Card Selection:** While genAI followed suit when possible, the random selection of Diamonds during trick play made its strategy somewhat predictable. A human player could potentially exploit this randomness by strategically discarding low-value Diamonds.

0.12 Future Improvements

The current strategy offers a starting point for genAI's Diamonds gameplay. Here are some potential improvements:

* **Suit Distribution Analysis:** genAI could analyze the distribution of suits in its hand and prioritize playing high cards in less-represented suits to maximize trick wins. * **Opponent Bidding Analysis:** genAI could consider the human player's bid when formulating its own strategy. This might involve adjusting its bid based on the perceived strength of the human's hand. * **Learning from Gameplay:** genAI could be designed to learn from past games, analyzing successful and unsuccessful strategies to improve its future performance.

0.13 Conclusion

This report demonstrates how genAI can be prompted to understand and play a card game like Diamonds. The initial implementation of a bidding strategy showcases the potential for AI development in this domain. With further improvements, genAI could become a more challenging and engaging opponent, capable of adapting its strategies and learning from each encounter.