Build a UI for Diamonds Game using Pygame with GenAI

ANVESHA SINGH
MANPREET BAWEJA
SAHANA BALAJEE
SANYA VASHIST

April 8, 2024

Contents

1 Introduction

This report details the process and experiences of generating code for a playable Diamonds game using the Pygame library with the help of GenAI.

Results: The final playable Diamonds game code will include functionalities like card graphics, player hands, scorekeeping, basic gameplay logic and also implement some strategies that the AI came up with to win the game.

Experiences with GenAI: Discussion on the interaction with GenAI throughout the development process. This will cover aspects like:

- How GenAI's capabilities were leveraged to understand and implement the Diamonds game rules.
- The effectiveness of Gemini's code generation and guidance in creating the Pygame functionalities.
- Any challenges faced while working with AI and how they were addressed.
- Coming up with strategies to win the game with the help of GenAI.

2 Rules of the Game

Objective: Be the player with the most points at the end of the game.

Players: 2 or more

Deck: A standard deck of cards. Remove the diamonds suit entirely. Shuffle the remaining cards (hearts, clubs, spades) and deal them all face down to the players, ensuring everyone gets an equal number of cards.

Diamond Pile: Shuffle the diamond suit deck and place it face down in the center of the table. This is the auction pile.

Bidding:

- Reveal the top card from the diamond pile. This is the diamond being auctioned.
- Each player, in turn order, secretly chooses one card from their hand and places it face down in front of them as their bid.
- Once everyone has placed a bid, reveal all the bids simultaneously.
- The player who played the card with the highest point value wins the bidding. Use the following ranking system (from lowest to highest):
- 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack (J), Queen (Q), King (K), Ace (A)
- If multiple players have the same highest bid, the diamond's point value is divided equally amongst them (rounded down if necessary).

Winning the Diamond:

The winning bidder takes the revealed diamond card and adds its point value (based on the ranking system above) to their personal score. Discard the winning player's bid card face down into a separate discard pile.

Next Round:

Repeat steps 1-5 above to auction the next diamond card from the pile.

Ending the Game:

The game ends when all the diamonds have been auctioned. Winning:
The player with the most points at the end of the game wins!

3 Generating Code for Pygame

The AI seems to understand the game perfectly however when it comes to implementing and writing code for it, it seems to forget established rules and give code with many errors.

There were a lot of difficulties like conceptual misunderstandings and syntax errors faced while generating the code:

- It had problems generating correct decks with randomly shuffled cards.
- Since the game is called Diamonds, GenAI emphasized it a little too much and overlooked the fact that the players don't receive Diamond cards. This reflected in the UI as each player received cards from different suits, not Diamonds.
- When trying to update a section of the code, the AI would often forget its previously written classes and functions. It would sometimes declare entirely new functions which don't exist or have a different name in the rest of the code.
- The AI had difficulty recognising the flow of the game and displayed it's bid before displaying the auctioned card.
- Gemini often provides only half of the code and stops running mid response however ChatGPT is able to provide the full code at once.
- Although ChatGPT seems more prone to forgetting previous responses than Gemini.
- The errors generated by the AI code are very difficult to resolve even for the AI.
- There were also problems with the scaling of the images used in the UI of the Pygame however those were easy to fix.

4 Conclusion

In conclusion, although GenAI is able to write basic code, it struggles with implementing the logic of the game properly and also making sure the code is error free. However, there weren't many difficulties working with images making it easy to implement Pygame.

Git Repo Link

5 References

- Gemini Chat Link
- ChatGPT Link
- ChatGPT Link 2