

Developing GUI for diamond cards game using PyGame

Radhika Amar Desai

April 8, 2024

1 Introduction

The report aims to document the project "Developing GUI for Diamond Cards Game using PyGame". Pygame is a library which the user is not familiar with. In this project, we explore the capabilities of Pygame, a powerful Python library primarily used for developing video games and multimedia applications, with a focus on creating a graphical user interface (GUI) for the Diamond Cards game.

This project goes beyond the surface goal of enhancing the Diamond Cards game; it serves as an opportunity for exploration and learning. The primary learning objectives encompass a range of technical skills, problem-solving abilities, and software development methodologies.

First and foremost, the project aims to familiarize the user with Pygame, introducing them to its features, syntax, and best practices. By working hands-on with Pygame, participants will gain a deep understanding of how to utilize this versatile library for creating interactive applications.

Moreover, this project offers a rich learning experience encompassing technical skills, problem-solving abilities, and agile development methodologies. By embarking on this journey of exploration and learning, participants will not only enhance the Diamond Cards game but also acquire valuable skills and insights that will serve them well in their future endeavors in software development.

2 Problem Statement

The objective of this project is to develop a Graphical User Interface (GUI) for the Diamonds game and integrate it with the bidding strategy. Additionally, the project aims to analyze the response of the GenAI Tool, particularly the Gemini model, to various prompts, evaluating its ability to retain context in discussions, its learning capabilities, and the quality of generated code.

The primary focus is on understanding Large Language Models (LLMs) and mastering prompt engineering.

3 Developing the GUI using Pygame

This section details the development process of the GUI for the Diamonds game using Gemini as a coding assistant. Initially, Gemini was introduced to the project's objective without specific instructions regarding the GUI's appearance. It suggested the basic layout of the project structure and provided guidance on how to approach the project, recommending a review of the Pygame documentation.

Subsequently, Gemini was tasked with designing a basic interface featuring four players bidding random cards. It generated code for a rudimentary GUI in Pygame, which required minor corrections through user intervention. Additionally, Gemini provided suggestions for enhancing the GUI's functionality.

The iterative development process involved incorporating additional features incrementally. Initially, Gemini was instructed to render red and yellow rectangles adjacent to each other and display the player's name along with the bidded card at appropriate locations. Subsequently, it was asked to integrate images into the rectangles. Throughout this process, user intervention was necessary to address various issues such as adjusting the code structure, rectifying bugs in the generated code, and refining the appearance of the GUI.

The integration of the bidding strategy code and the Pygame GUI was solely performed by the user, ensuring seamless coordination between the game mechanics and the visual interface.

Some notable characteristics of Gemini's response were:

- **Incomplete code** : Gemini often generated incomplete code or provided comments like "rest of the code" above and below the new code generated for achieving the latest modification requested by the user.
- **Explanation** : Gemini provided an explanation about the functioning of the generated code.
- **Additional suggestions** : Gemini provided additional suggestions like adding other features.

4 Conclusion: Developing an Effective Prompting Strategy

To leverage GenAI effectively for code generation, following a structured, step-by-step process is paramount. Initially, it's imperative to introduce our project's objectives clearly to GenAI. Users should attentively comprehend GenAI's responses and carefully consider the suggestions provided. This collaborative approach aids in refining the project vision and provides a clear direction for execution.

Subsequently, requesting a basic version of the project and iteratively modifying it is crucial. This approach enables GenAI to focus on one task at a time, facilitating the generation of error-free code. Each modification should

be thoroughly tested and analyzed by the user to ensure the robustness and functionality of the generated code.

By adhering to this methodical process, users can maximize the potential of GenAI for code generation, leading to more efficient and reliable outcomes in software development projects.