**Report要求，正文请看下一页**

Overview of project: 5

Code design process: 15

Pros/cons, discussion of extending it: 10

Results / output: 10

Reflection / acknowledgements / references: 5

Report

Your report for this project is intended as a portfolio of your work. Please explain your design process, describe your results (including any output), and explain the pros and cons of your choices. Include a discussion of how you might extend the project or make it more inclusive given more sophisticated tools. Any results / output should be accompanied by an explanation. Please finish your report with a short reflection on your work this semester and acknowledgements.

**Blackjack Game in Python: Project Report**

**1. Overview of the Project**

The goal of this project is to create a simple text-based Blackjack game using Python. The game simulates a typical 1-on-1 Blackjack experience, in which a player competes against the bot dealer. The game includes basic functionalities such as creating a deck of poker cards, card dealing, hand value calculation, and winner determination based on Blackjack rules.

**2. Code Design Process**

**Conceptualization**:

**Objective:** To create an interactive game of Blackjack that is user-friendly and resembles real-life play.

**Gameplay Mechanism:** The player and dealer each receive two cards initially. The player will have the option to "hit" (receive another card) or "stand" (stop receiving cards). On the other hand, the dealer must hit until their hand value is at least 17 (according to casino rules).

**Implementation**:

Deck Creation: A deck of cards is created with suits and ranks and shuffled to ensure randomness. We used the random module to assist our function.

Hand Management: Functions to deal cards and calculate hand values. We made sure to distribute one card in each round to each player without replacement to the deck.

Game Logic: Conditions for winning, losing, and ties based on hand values. If hand value is greater than 22, the player busts and loses the game.

User Interaction: Prompts for the player to decide whether to hit or stand.

**Testing and Refinement**:

Continuous testing was done to ensure that game mechanism such as dealing cards and calculating hand values worked correctly.

**3. Pros, Cons, and Potential Extensions**

**Pros**:

Simplicity: Easy for users to understand and interact with.

Educational Value: Helps players learn the basic rules of Blackjack.

**Cons**:

Limited Features: Lacks advanced features like betting, multiple players, or graphical interface.

Randomness: The randomness of shuffling might not perfectly mimic a real deck.

**Extensions**:

1. Graphical User Interface (GUI): Implementing a GUI would make the game more engaging and visually appealing.
2. Multiplayer Functionality: Allowing multiple players to play against the dealer.
3. Betting System: Introduce a virtual currency system for betting.
4. Strategy Hints: Offering suggestions or tips based on Blackjack strategy.
5. Difficulty Level: Use probability calculation to enhance the bot’s winning strategy.
6. Sound Effects and Animations: Incorporate sound effects for card shuffling, dealing, and other game events. Add animations for card movements and interactions.

**4. Results and Output**

The game successfully simulates a basic version of Blackjack. It handles card dealing, hand value calculation, and decision-making processes effectively. During gameplay, players receive clear prompts and updates about their hand and the dealer's hand, which guides them through the game. However, the game is strictly text-based and lacks visual elements, limiting the overall gaming experience.

**5. Reflection, Acknowledgements, and References**

Reflecting on this project, the simplicity of the game made it an excellent learning experience in programming logic and user interaction in Python. We were able to incorporate the materials from the course and code planning methods while designing the game. The project could have been more challenging and instructive with the integration of more advanced features like a GUI or multiplayer options.

**Acknowledgements**:

Thanks to the Python community for their extensive resources and documentation which were invaluable throughout the development process.

Credit to classic Blackjack rules and gameplay mechanics for guiding the structure of the game.

**References**:

Python Documentation: For syntax and function references.

Online forums and communities: For troubleshooting and code optimization tips.

In conclusion, this project was a rewarding venture into game development using Python. While the game is basic, it lays a solid foundation for more complex and interactive game development in the future.