Design Document / User Guide

1. Software Requirements Specification (SRS)
   1. Purpose
      1. The Art Dealer Game is a program designed for students in grades K-8. In this game, the student acts as a seller who submits four cards per round to the dealer. The dealer has different patterns such as selecting cards based on color, rank or prime numbers. The student’s objective is to discover the dealer’s pattern through gameplay and observation.
   2. Scope
      1. This program provides a fun, interactive way for students to practice problem solving, logical reasoning and pattern recognition. The difficulty level adjusts based on the grade group (K-2, 3-5, 6-8), with more complex patterns introduced at higher levels.
   3. Definitions & Abbreviations
      1. Dealer: The computer player that selects cards based on hidden patterns.
      2. Card: A digital representation of an art piece or playing card.
      3. PIL: Python imaging library.
      4. Tkinter: Python’s standard GUI Library.
   4. Overall Description
      1. This is a desktop application developed using python with Tkinter and PIL for card images. The game flow is:
         1. Student selects four cards to submit to the dealer.
         2. The Dealer selects one card to purchase according to the hidden rule.
         3. The student tries to gather the dealer’s rule through repeated rounds.
   5. Specific Requirements
      1. Functional Requirements.
         1. Display cards to students for selection.
         2. Allow students to select and submit four cards per round.
         3. Dealer automatically selects one card based on a hidden pattern.
         4. Game supports multiple difficulty levels:
            1. K-2 Simple visual rules (Color and shape)
            2. 3-5 Moderate rules ( Odd/even numbers)
            3. 6-8 Complex patterns (prime numbers, multi-criteria rules)
         5. Provide visual feedback after each round.
      2. Non- Functional Requirements
         1. Usability - Intuitive and simple interface for children.
         2. Performance - Quick response times, minimal loading delays.
         3. Portability - Runs as .exe on Windows.
         4. Reliability - Robust against invalid inputs.
2. Software Design
   * 1. System Architecture
        1. GUI based, event- driven application.
        2. Standalone desktop game with no external servers.
        3. Assets ( images, rules) bundled with the application.
     2. Modules/ Components
        1. Tkinter
           1. Main game window
           2. Card selection interface
        2. Dealer Logic Module
           1. Contains hidden purchase patterns.
           2. Selects one card per round based on pattern.
        3. Card Management Module
           1. Loads card images using PIL
           2. Displays cards on the GUI
        4. Game Control Module
           1. Manages game flow
           2. Tracks rounds, selected cards and difficulty levels
     3. Data Design
        1. No External database is used.Data are stored in code or files:
           1. Cards: Stored as image files
           2. Dealer Patterns: Defined with Python
     4. Interface Design
        1. User interface:
           1. Visual card display grid
           2. Grade- level selection
3. User’s Guide
   1. System requirements
      1. Operating System: Windows ( for .exe execution).
      2. Python is not required for users; the application runs as an executable.
      3. Sufficient screen resolution for display.
   2. Installation Instructions
      1. Clone or Download Repository:
         1. From Github or provided link
      2. Run the Game:
         1. Locate and double click the provided .exe file.
         2. No installation necessary beyond cloning.
   3. Usage Instructions
      1. Start the game:
         1. Launch the .exe file
         2. Choose grade level to adjust difficulty
      2. Select and submit cards
         1. Click on four cards to select them for submission.
         2. Submit your selection to the dealer.
      3. Observe Dealers Purchase
         1. The dealer will select one of the submitted cards based on its hidden rule.
         2. Analyze the dealer’s choice and try to gather the pattern.
      4. Continue Playing:
         1. Repeat the rounds to further refine your guesses.
         2. Optionally reset the game to try new patterns.
      5. Exit Game:
   4. Support & Contact

Use Cases

**Use Case: Start the Game**

Primary Actor: Student

Description: The Student launches the game from their desktop to start playing.

Preconditions:

* The game .exe is downloaded and available.
* A windows system with sufficient resolution.

Acceptance Criteria:

* The student double clicks the .exe file.
* The game launches and displays the main menu or start screen
* The student selects a grade level
* The game loads the appropriate difficulty level

**Use case: Select and submit cards**

Actor: Student

Description: The program generates 4 random cards to submit for the current round.

Preconditions:

* The game must be running
* The current round must be active.

Acceptance Criteria:

* The game displays a grid of cards
* The student clicks on four cards to select.
* The student clicks the button to send the cards to the dealer.

**Use case : Dealer Selects a Card**

Actor: Dealer ( System)

Description : The dealer automatically selects a card from the students submission based on a hidden pattern.

Preconditions:

* The student has submitted four cards.
* A hidden pattern has been assigned for this round.

Acceptance Criteria:

* The dealer evaluates the submitted cards.
* The dealer applies its hidden pattern (Color, number, prime number)
* The game highlights the dealer's choice.

**Use case: Analyze Dealers Pattern**

Actor: Student

Description: The student observes the dealers selection and tries to guess the hidden pattern.

Preconditions:

* The dealer's choice has been revealed.

Acceptance Criteria:

* The student reviews the chosen card in comparison with the submitted ones.
* The student attempts to guess the dealer's pattern via the game's guess interface.

**Use Case: Reset or Exit the Game**

Actor : Student

Description: The student chooses to reset the game to try new patterns or exit the game.

Preconditions: The game is currently running.

Acceptance Criteria:

* The game will reset if they lose the 3 attempts
* The game will reset if they win the game.
* Student can exit the game at any time by clicking the X.