Design Document - Smart Detective Notes for Clue

Description

Clue is a classic board game where 2-6 players use process of elimination to solve a murder mystery (how to play). The object of the game is to determine who murdered the victim, where the crime took place, and which weapon was used. Each player attempts to deduce the solution by moving around a game board composed of the rooms of a mansion and making suggestions to collect clues about the circumstances of the murder from the other players. Playing the game competitively requires a complex note-taking strategy to maximize the amount of information gathered in every turn.

The online version of the game provides a beginner-level note-taking feature that keeps track of the basic information automatically. This smart note-taking program takes this concept to another level by incorporating advanced strategy automatically. The player inputs what happened each turn, and the program will make all the deductions that could be made given the current information on hand and add them to the basic notes. Once the program has deduced a definite solution, it will alert the player to make an accusation.

As a bonus challenge if time permits, the program will also give the player a recommendation for what to suggest on their next turn.

Features

- To begin taking notes, the player enters their name and the total number of players playing the game in the command line. The player will then be prompted to enter more information to set up the initial state of the board game:
    - The cards in their hand
    - The names of all the players in the order they will be taking turns
- For every turn, the player inputs the suggestion that was made along with who disproved the suggestion.
    - The player can skip taking notes on a turn if they missed the information or if the player taking the turn did not get to make a suggestion
    - The player can indicate if no one disproved the suggestion
    - If it is the player's turn, they can input which specific card was revealed
- After each turn, the program will make automatic deductions for the player:
    - Mark cards that are revealed or eliminated from other players hands during each turn (knowns)
    - If a card was secretly shown to another player, mark those cards as maybes for the disprover
    - Check notes from previous turns for any knowns that can be deduced from old maybes from the new information
    - Keep checking until no new knowns can be found
- After each turn, show the player a summary of all the knowns.
    - If a definite solution has been reached, show just the solution instead
    - Bonus: if the player has a turn next, add a recommendation for what to suggest
- Instead of a turn, the player can type 'quit' to end the program.

Classes

1) **Player** – tracks information related to each player in the game; solution is treated as a player with no turn
   - Attributes: name, num_cards (number of cards in the player's hand), hand (cards in the player's hand)
2) **Card** – represents the cards in the game
   - Attributes: name, type
   - Subclasses: Room, Weapon, Suspect (each a different type)
3) **Turn** – tracks all the information from each turn
   - Attributes: suggestion (suspect, weapon, and room suggested that turn), suggester (player who made the suggestion), disprover (player who revealed a card)
   - Subclass: MyTurn (tracks the player's own turn)
     - Extra attribute: revealed_card (card that was revealed to the player)
4) **Board** – stores all the information about the board game
   - Attributes: turn_order (list of players in order of their turn), turn_history (list of all the turns made in the game), card_list (list of all the cards in the game)
   - Functions:
     - next_turn: given a player who just took a turn, returns the player whose turn is next
5) **Notebook** – stores all the notes made by the Detective in a matrix (card x player)
   - Attributes: view (dictionary that tracks all the entries for viewing the whole matrix), legend (key to decode the printed marks representing each subclass)
   - Subclasses (each represented by a symbol when printing:
     - KnownYes (✓) – tracks cards that are known to be in a player's hand
     - KnownNo (X) – tracks the cards that are known to not be in a player's hand
     - Maybe (? or !) – tracks the cards that might be in a player's hand
     - Unknown ( ) – tracks the cards for which there is no information
6) **Detective** – makes deductions based on information revealed in each turn
   - Functions:
     - take_notes: takes information from the current turn and appends the information to the proper list in Notebook
     - eliminate_cards: when a card is noted as a KnownYes, marks the card as a KnownNo for the rest of the players
     - add_to_hand: adds a card to a Player's hand if a KnownYes
     - deduce: checks Maybe list for any KnownYes that can be deduced; loops through until no new knowns are found
     - recommend: looks through all the notes and follows a set strategy to output a suspect, weapon, and room to suggest
7) **Engine** – moves the program from one state to another
   - Functions:
     - user_prompt: triggered by Board.next_turn, prompts the user to input what happened that turn
     - record_turn: takes user input from a turn and returns an instance of Turn appended to the turn history
     - skip_turn: takes user input to skip and triggers Board.next_turn without recording anything
     - quit_program: takes user input to quit and ends program
     - print_notes: prints out a matrix form of Notebook
       - Bonus: print out a recommendation if the player's turn is next
     - declare_solution: if the solution's hand is full, prints out solution.hand instead of Notebook