

# Game Set Up

Estimated Time: 6-8 hours

## 1. **Mansion Layout:**

- Create a representation of the mansion layout with different rooms (e.g., kitchen, library, ballroom, etc.). This layout can be represented abstractly (e.g., a graph where rooms are nodes and passages are edges) or concretely (e.g., a 2D grid or array). Ensure that the connections between rooms (i.e., which rooms are adjacent and allow movement) are clearly defined. Try to include secret passages between certain rooms as per the original game rules (e.g., Study to Kitchen, Conservatory to Lounge). Define designated starting positions for each character, typically in hallways or outside specific rooms.

## 2. **Character Definition:**

- Define the 6 standard Cluedo characters: Miss Scarlett, Colonel Mustard, Mrs. White, Reverend Green, Mrs. Peacock, and Professor Plum.
- Assign each character a starting position on the board.

## 3. **Weapon Definition:**

- Define the 6 standard Cluedo weapons: Candlestick, Dagger, Lead Pipe, Revolver, Rope, and Wrench.

## 4. **Solution Selection:**

- From the full set of Character, Weapon, and Room cards, randomly select one of each category to form the "murder solution" (the three cards placed in the confidential envelope). These cards should be hidden from all players throughout the game.
- After the solution is selected, the remaining cards (the "deck") must be shuffled and distributed evenly among all active players. If the number of cards doesn't divide evenly, some players may receive one more card than others.

# Player Movement

Estimated Time: 10-12 hours

## 1. **Turn Structure:**

- Implement a turn-based system. Each player takes a turn in sequence (e.g., Miss Scarlett, then Colonel Mustard, etc.).

- A standard Cluedo turn involves:
    - **Dice Roll:** Simulate rolling a standard 6-sided die for player movement. (Note: The core game typically uses one die for movement).
    - **Movement:** Allow the player to move their character token a number of spaces equal to their dice roll. Players can move horizontally or vertically.
    - **Entering Rooms:** If a player enters a room, their movement for that turn ends. They can then make a suggestion.
    - **Exiting Rooms:** A player must roll the dice to exit a room. They cannot simply move out. Secret passages do not require a dice roll for movement.
  - Your system should validate moves (e.g., preventing movement through walls, ensuring movement count is respected).
2. **Suggestions:**
- When a player enters a room (either by moving or via a secret passage), they *must* make a suggestion.
    - A suggestion involves a Character, a Weapon, and the Room the suggesting player *just entered*. For example, "I suggest it was Colonel Mustard, with the Revolver, in the Study."
    - **Character Movement During Suggestion:** If the suggested Character is currently on the board and not in the room where the suggestion is being made, their token *must* be immediately moved into that room. This is a crucial rule for the classic game. The weapon token also moves into that room.
    - The game should prompt the player for their suggestion.

## User Interaction I/O part

- For Part 1, the user interface can be text-based (Command Line Interface - CLI).
- Ensure clear and intuitive prompts for player actions (e.g., 'Player X, press Enter to roll dice', 'Player Y, your roll is 4. Enter your move (e.g., UP 2, RIGHT 2) or type SECRET\_PASSAGE\_TO\_KITCHEN:', 'Player Z, you are in the Study. Make a suggestion (Character, Weapon):').
- The game should output relevant information clearly to the console (e.g., dice roll result, current player position, player's cards, suggestion details and any refutation information relevant only to the suggesting player's view).
- While robust error handling for all invalid inputs is primarily assessed in Part 2 for full reliability, your Part 1 implementation should handle basic expected inputs gracefully.

# Deliverables for Part 1

## 1. Source Code Folder (Studentname\_Project2\_SourceCode)

- Create a folder named "Studentname\_Project2\_SourceCode."
- Organize and structure the complete source code of your Cluedo game application within this folder.
- Ensure clarity in your code organization, including well-named files and directories.
- Your code should be well-commented, especially for complex logic, and adhere to good programming practices for readability and maintainability.

## 2. Readme File (Studentname\_Readme):

- Include a readme file in the same folder.
- Provide detailed instructions on how to run the Cluedo game.
- Specify steps for:
  - Cloning the repository (include if you used version control like Git for your project):.
  - Navigating to the source code directory.
  - Running the game.
- List any dependencies or prerequisites required to execute the code.

## 3. Zip the Folder

- Before uploading to Canvas, zip the folder containing the source code and readme.

**Note on Testing for Part 1:** While comprehensive testing documentation and screenshots are part of the Project 2 Part 2 deliverable, you are expected to thoroughly test all Part 1 functionalities (game setup, player movement, and basic suggestion triggering) to ensure they are robust and correctly implemented before proceeding to Part 2 development.

You will receive a complete/incomplete grade for Project 2, Part 1. Students must demonstrate the core functionalities outlined in Project 2 Part 1 to receive a "Complete"

grade for Part 1, which then allows their Project 2 Part 2 submission to be numerically graded. Failure to complete Project 2 Part 1 effectively will significantly impact the overall score. For more information, please refer to the [Project 2 Guide](#).

## Objective:

This project aims to develop a digital version of the classic murder mystery board game Cluedo, also known as Clue. The game will be implemented using either Prolog or Python programming languages, allowing you to apply your knowledge of logic programming (Prolog) or object-oriented programming (Python) to create a fully functional game application. By engaging in this project, you will enhance your understanding of designing logic-based artificial intelligent systems and gain practical experience implementing game mechanics, reasoning algorithms, and player interactions in a simulated environment. Through developing the Cluedo game, you will strengthen your problem-solving skills, analytical thinking, and ability to apply AI concepts to real-world scenarios.

The "AI" in this project can be demonstrated in one or more of the following ways:

- **Automated Deduction/Knowledge Representation:** How your system internally represents the game state, player knowledge, and uses logic to *assist* in deduction (e.g., tracking refuted cards, possible solutions).
- **AI Player Implementation (Mandatory for Project 2 Part 2):** You must implement at least one fully functional computer-controlled (AI) player that can make logical moves, suggestions, and deductions based on the information it possesses. This AI player will be a core demonstration of your understanding of AI concepts. This can range from a rule-based AI to a more sophisticated reasoning agent. You must clearly define the AI's capabilities and limitations in your documentation."
- **Constraint Satisfaction/Search:** How your system manages the unknown solution (murderer, weapon, room) and potentially uses logical constraints to narrow down possibilities.

## Description of the Game:

Cluedo, also known as Clue in North America, is a classic murder mystery board game where players assume the roles of characters trying to solve a murder. The game is set in a mansion, and players move around to different rooms, gather clues, and make deductions to identify the murderer, the weapon used, and the room where the crime occurred. For more information on how to play this game, please refer to this [video](#).

## Game Scope and UI:

- **Text-Based Interface (CLI) is Acceptable:** For the purpose of this project, a command-line interface (CLI) or text-based interface is fully acceptable for both Part 1 and Part 2. While UI enhancement is an optional component for Part 2, it is not a core requirement for a passing grade, allowing you to focus on the underlying game logic and AI. Your primary focus should be on demonstrating the core game mechanics and the AI/logic aspects.
- **Number of Players:** The game must support a minimum of 3 players (characters) from the start. You are strongly encouraged to implement the full 6 characters, as your design in Part 1 should accommodate multiple players, and in Part 2, at least one of these players must be implemented as an AI. Your game's design should allow for the flexible assignment of players as either human-controlled or AI-controlled.
- **Basic Error handling:** Your implementation should include basic error handling for invalid user inputs (e.g., non-numeric input for dice roll, invalid room names, out-of-bounds moves) to ensure application stability.

## Deliverables:

### Project 2 Part 1 - Due Module 13

1. **Source Code Folder:**
  - Create a folder named "Studentname\_Project2\_SourceCode."
  - Organize and structure the complete source code of your Cluedo game application within this folder.
  - Ensure clarity in your code organization, including well-named files and directories.
2. **Readme File (Studentname\_Readme):**
  - Include a readme file in the same folder.
  - Provide detailed instructions on how to run the Cluedo game.
  - Specify steps for:

- Cloning the repository (if applicable).
  - Navigating to the source code directory.
  - Running the game.
- List any dependencies or prerequisites required to execute the code.
- 3. **Zip the Folder:**
  - Before uploading to Canvas, zip the folder containing the source code and readme.

## **Project 2 Part 2 - Due Module 14**

1. **Presentation (Up to 10 Minutes):**
  - Record a presentation that clearly explains your Cluedo project.
  - Include a demo of the game.
  - Highlight key features, design choices, and implementation details.
2. **Screen Recording:**
  - Record your device screen during the execution of the Cluedo game.
  - Capture the gameplay, interactions, and any relevant aspects.
  - Upload this recording as part of your project submission.
3. **Project Documentation (Up to 10-Page PDF):**
  - Create a comprehensive PDF document describing various aspects of your project:
    - **Abstract:** Briefly summarize the project's purpose and goals.
    - **Introduction:** Explain the context and motivation behind developing the Cluedo game.
    - **Game Rules:** Clearly outline the rules and mechanics of the game.
    - **Thorough Testing:** Include screenshots demonstrating testing scenarios.
    - **Challenges Faced:** Discuss any obstacles encountered during development.
    - **Stability and Reliability:** Evaluate the game's stability and reliability.
    - **Additional Details:** Cover any other relevant information.
  - Ensure the document is well-organized and visually appealing.

Remember to zip the source code folder before uploading it to Canvas. Good luck with your Cluedo project!

---

## Learning Outcomes

This activity addresses the following learning outcomes:

- CLO1. Identify the foundational concepts, theories, and historical developments in artificial intelligence.
- CLO2. Apply various search algorithms and heuristic design techniques independently to solve complex AI problems.
- CLO7. Assess emerging trends and advancements in AI research critically.
- CLO8. Solve and address complex AI-related problems by integrating insights from different disciplines.