

HELP NOTEPAD ANSWER RESET

CORRECT WRONG GUESS FINAL ACCUSATION

- I envision there being 8 total buttons (same as the ones drawn above) that the user can interact with

- ↳ HELP would show the instructions for the game
- ↳ NOTEPAD would display a notepad the user can take notes on
- ↳ ANSWER would take the user to the end of the game and display the final answer (more for debugging and grading purposes)
- ↳ RESET will start the game from the beginning
- ↳ CORRECT and WRONG would change the color from green and red, respectively, and used to fill the detective's notebook
- ↳ GUESS and FINAL ACCUSATION would be used for the user to make their guesses.

Rooms				
Hall				
Lounge				
Dining Room				
Kitchen				
Ball Room				
Conservatory				
Billiard Room				
Library				
Study				
Suspects				
Col. Mustard				
Prof. Plum				
Mr. Green				
Mrs. Peacock				
Miss. Scarlet				
Mrs. White				
Weapons				
Knife				
Candlestick				
Revolver				
Rope				
Lead Pipe				
Wrench				

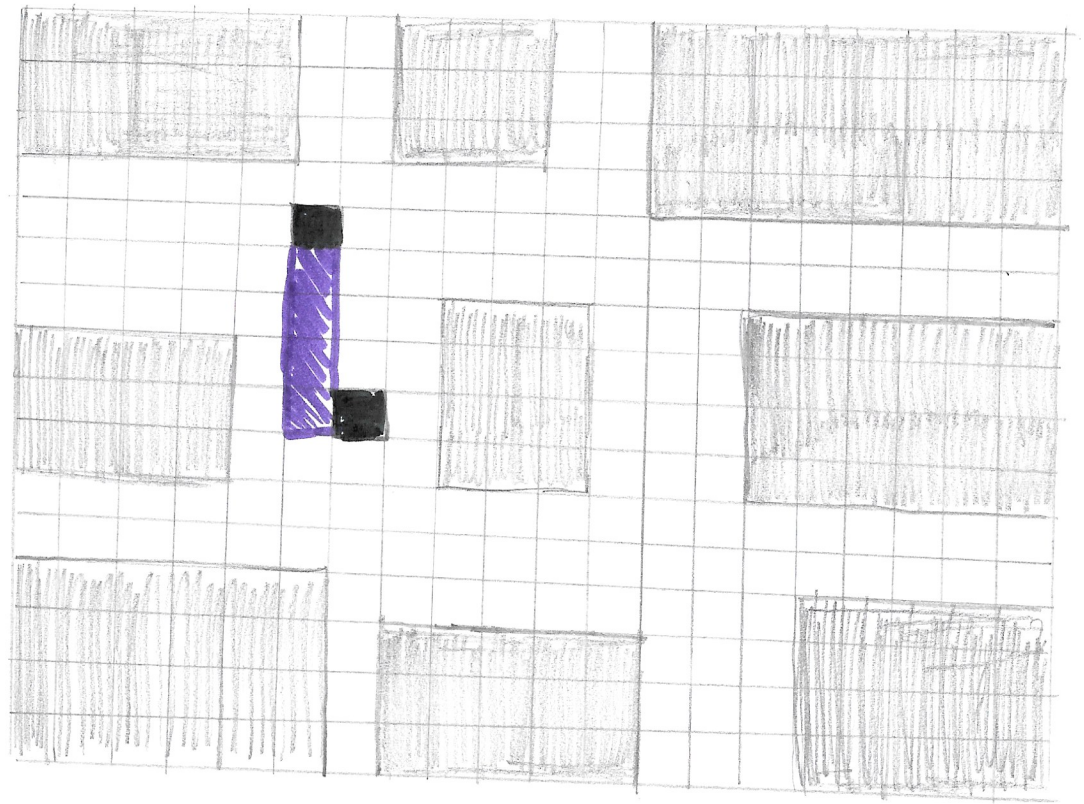
The second part the user can interact with is the detective's notebook. They can use this to keep track of their guesses, answers, and thoughts in general. They can fill out one column of the grid using red or green colors, indicating wrong and right.

PLEASE CHOOSE YOUR PLAYER COLOR  
(black, white, green, blue, red, purple)

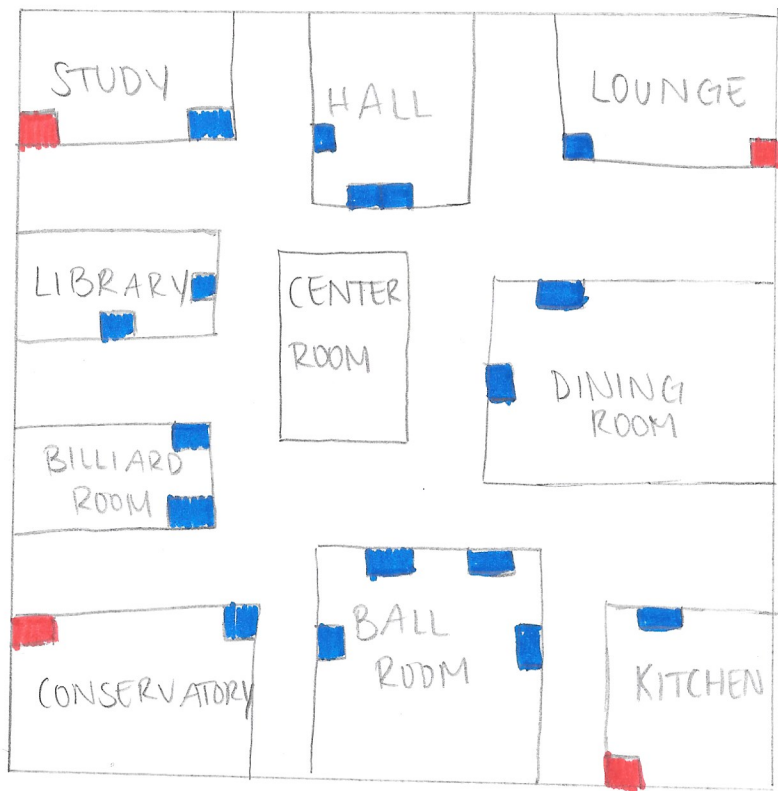
"Input"

CLICK ON ONE OF THE BROWN  
SQUARES TO CHOOSE YOUR STARTING  
LOCATION

At the start of the game the user will receive two messages. The first will allow them to input color of their choice to act as their character. The second will prompt them to choose their starting location on the board.



The next thing the user will be able to do is click on the location they wish to go to. The board will light up the path they must take to their, or return a message saying the move is illegal (ie. player moved too many spaces, is going through a room, etc)



The second aspect of the board the user will interact with is the rooms, doors (blue), and secret passages (red).

They will travel around the board, attempting to enter rooms, however they can only enter using the doors. They also may use the secret passages in the corner rooms to travel between them in one move, rather than across the board in multiple moves.

### MAKE A GUESS!

CLEAR

#### ROOMS

Study

Hall

Lounge

Library

Billiard Room

Conservatory

Ball Room

Kitchen

Dining Room

#### WEAPONS

Knife

Candle Stick

Revolver

Lead Pipe

Rope

Wrench

#### SUSPECTS

Mustard

Plum

Green

Peacock

Scarlet

White

The last part the user will interact with is the above screen which will appear when the user wishes to make a guess. Cards with each of the options will appear and the user will be able to click on only one in each category to make their guess. There will also be a "clear" button in case the user wishes to change their selection.

## Project Proposal

- **Project Description:**

- The name of my term project is Clue, and it will be pretty similar to the current board game version. The idea is for the player to interact with the computer and try to win against it. At the start of the game, the game will automatically choose a suspect (out of 6 total), a weapon (out of 6 total), and a room (out of 9 total). The player and computer will take turns rolling the die and moving around the game board, entering different rooms, and making guesses as to what they believe is the correct answer (the suspect, weapon, and room chosen at the beginning). Rooms can only be entered through certain spots marked with doors and also include back staircases that lead to other rooms. Each of the players will be dealt cards with the rest of the suspects, weapons, and rooms not chosen as the correct answer and they must use that to help them eliminate choices. Once in a room, the player can make a guess and the opposing players will then say if they have any cards that prove your guess to be wrong and tell you which card. To make the final accusation at the end, the player must be in the center room on the board. (The rules are essentially the same as the actual board game, however this game is limited to one player versus the computer).

- **Competitive Analysis:**

- I have seen two projects similar to mine in the 112 gallery that have also done Clue as their term project. The first one was by Riya Shrivastava and there are certain aspects of her project that I would like to include something similar to, as well as aspects that I will do differently. To start, I prefer a user interface where the player can see most of the parts of the game on the screen at once instead of having to click around the main menu to reach each different component. Additionally, I would like to make it more similar to the actual board game, where the player will have a “detective’s notebook” they can use to cross items off their list instead of listing the names as she had. It also appeared that the player did not take turns going against an AI or computer and I would like to try to implement an AI to play against the user. A few things that will be similar in my project, however, is her system for making a guess. While I can’t and haven’t seen her code, I am also planning to automatically store the room that the user is currently in, thus not allowing them to choose a different room when making a guess. I also want to have buttons that the user can click on to make their guess. It also appears that she uses some sort of pathfinding algorithm to show the player all their possible moves. I also want to use a pathfinding algorithm, but for the purpose of showing the player the shortest possible path to their desired destination. The second project that did Clue as their term project was by Elias Prieto and I am envisioning my user interface to be more similar to his. One main difference is that it appears that he made the game board himself, but I am planning to use the

image of the real game board, but store all the user's moves and pathfinding on a grid functioning on the back-end of the code. He also used pathfinding and an AI system to play against the computer which is something I will be implementing as well. My design to get the user's guess will be different as I do not plan to use an input box for this, but rather a button system the user can click on to make their guess. Lastly, as I mentioned above, I plan to store each player's location, something I do not believe this project did. The purpose of storing the location will be to make sure that the player does not make guesses they are not allowed to make as per the rules of the game.

- **Structural Plan:**

- I want to separate my final project into multiple files (Pathfinding, AI, different modes of the animation)
- I have begun this process but am running into an issue on how to call the different functions from different files. I plan to discuss this with my mentor.
- For the images, I have placed them in a separate folder organized by room, weapon, suspect, and dice.
- Each component of my project will have multiple functions to avoid repetitive code
  - Animation functions - app started, mouse pressed, key pressed, timer fired, redraw all
  - Pathfinding function
  - AI functions - make move, make guess, make final accusation



















- **Algorithmic Plan:**

- The two parts that stand out as most algorithmically complex to me are the pathfinding algorithm and the AI that must be implemented to play against the player
- Pathfinding
  - I am using A\* pathfinding to find the shortest path from the user's/computer's current position to the position they wish to go to
  - The algorithm will consist of multiple lists (an open list of all possible squares, a closed list of the squares already considered, the final path, and score list of each square that corresponds with its index in the open list) and a dictionary (mapping the current location to its parent)
  - It will calculate three different scores for each potential move - the g-score (number of squares from the starting point), the h-score (distance from the starting point to the end point), and the f-score (sum of g-score and h-score)
  - The goal is to find the squares with the lowest f-scores and add them to the closed list and the dictionary



- At the end, the algorithm will go through the dictionary and work from the end square and find its parent until it reaches the start square, adding all of the parent squares to the path list
  - Finally it should return the path list
- AI
  - The AI must be intelligent enough to do everything the human player will do - move around the board, enter rooms, make guesses, eliminate wrong answers, show their cards, and make the final accusation
  - While this algorithm is not as detailed as the previous one, my current plan is as follows:
    - All cards will be stored in the computer's dictionary (the keys will be the rooms, weapons, and suspects)
      - allCards = {suspects: {'Col. Mustard', 'Prof. Plum', 'Mr. Green', 'Mrs. Peacock', 'Miss Scarlett', 'Mrs. White'}, weapons: {'Knife', 'Candlestick', 'Revolver', 'Rope', 'Lead Pipe', 'Wrench'}, rooms: {'Hall', 'Lounge', 'Dining Room', 'Kitchen', 'Ball Room', 'Conservatory', 'Billiard Room', 'Library', 'Study'}}
    - The cards dealt to the computer at the beginning of the game will be removed from the dictionary
    - To make guesses, the computer will randomly choose a weapon and suspect from its dictionary (the room will be whichever room it is currently in). If the guess is proven wrong, that weapon/suspect/room will be removed from its dictionary
    - The computer will be able to make its final accusation when there is only one suspect, weapon, and room left in each key of the dictionary
    - To move around the board, the code will first choose the room the computer wishes to go to. This will be chosen by picking the closest room from the rooms still left in the computer's dictionary with the key "room". Then, using the pathfinding algorithm above, the code will find the shortest path from the computer's current location to the doors of the room it wishes to go to.
- **Timeline Plan:**
  - Sat Nov 28th: I plan to finish the project proposal and start implementing the pathfinding algorithm.
  - Sun Nov 29th: I plan to finish the pathfinding and make sure it works correctly, finish up the storyboard, scan and upload pictures of all the game cards, and distribute them randomly between the player and the computer

- Mon Nov 30th: Make sure my design proposal, storyboard, and preliminary code meet the TP1 criteria and submit those to autolab, finishing up the parts that did not work over the weekend. I also need to add constraints to the player's moves (not moving more than the die roll, starting only on start spaces)
- Tue Dec 1st: get started with the AI to allow the computer to play against the player (in particular, understand how I can incorporate algorithmic complexity into this part of the project)
- Wed Dec 2nd: finish the AI and debug any issues that may arise
- Thurs Dec 3rd: implement the interface for the player to make guesses and for the computer to display its guesses. Also, implement the interface for both the player and computer to show which card they have from the guess
- Fri Dec 4th: spend the day debugging and working out an issues in the code
- Sat Dec 5th: any final touches for a fully functioning game and continue improving the user experience and UI
- Mon Dec 7th: finish the video explaining my project and edit it.
- **Version Control Plan:**
  - I am using GitHub to back up my code. I have created a GitHub repository and backed up my current version of the code to it. As I continue to add features, I will keep pushing my code to GitHub.

 <b>Aanya Gupta and Aanya Gupta</b> add comments and beginning to structure files         9651175 3 minutes ago 2 commits		
	Design Proposal	add comments and beginning to structure files 3 minutes ago
	Images	add comments and beginning to structure files 3 minutes ago
	__pycache__	add comments and beginning to structure files 3 minutes ago
	.DS_Store	add comments and beginning to structure files 3 minutes ago
	TermProject.py	add comments and beginning to structure files 3 minutes ago
	answer.py	add comments and beginning to structure files 3 minutes ago
	appStarted.py	add comments and beginning to structure files 3 minutes ago
	cmu_112_graphics.py	initial commit - board and pathfinding 3 hours ago
	gameBoardGrid.py	add comments and beginning to structure files 3 minutes ago
	instructions.py	add comments and beginning to structure files 3 minutes ago
	main.py	add comments and beginning to structure files 3 minutes ago
	mousePressed_timerFired.py	add comments and beginning to structure files 3 minutes ago
	notepadGrid.py	add comments and beginning to structure files 3 minutes ago
	pathfinding.py	add comments and beginning to structure files 3 minutes ago
	pathfindingBoard.py	add comments and beginning to structure files 3 minutes ago
	redrawAll.py	add comments and beginning to structure files 3 minutes ago
	title_buttons.py	add comments and beginning to structure files 3 minutes ago

- 
- **Module List:**

- The only additional modules I am planning on using are cmu-112-graphics and PIL but it is within the course notes and what we learned in the optional lecture.
- **Storyboard:**
  - Separate doc
- **TP2 Update:**
  - The only things I changed were a few design things
    - Instead of a correct and wrong button to toggle the color from red to green, I decided that the color gets automatically toggled when you click on the notebook itself
    - I got rid of the extra notebook button and instead changed it so that it displayed the player's cards and the notebook remained on the screen
    - I also added a changing room layout button and feature to allow the user to change the room placement as they want
- **TP3 Update:**
  - Since the game can be confusing to understand, I added step by step instructions on the side that change depending on the part of the game the player is on
  - I added a "Ready to Accuse" button so the player cannot enter the center room until they want to make their final accusation of the game
  - I animated the die a bit so it is not so stagnant