

1. [Standards](#)
2. [Ideas and Pseudocode](#)
3. [Rules of Cluedo](#)
4. [Sprint 1 Breakdown](#)
5. [Sprint 2 Breakdown](#)
6. [Sprint 3 Breakdown](#)

## Standards

- Camel casing
- Curly brackets on same line

```
if(x) {
```

## Ideas, Diagrams and Pseudocode

### Board

- Three Square Types
  - Normal (Traversable)
  - Wall
  - Entry to Room
  - For rooms, we still need visual representation for the UI

### Objects and Properties

- Game Board
  - **Room**
    - playersOn
      - ArrayList
    - weaponOn
      - Object of type "Weapon"
    - entry
      - Pointer to corresponding square(s)
    - secretPassage
      - Pointer to corresponding room
      - Can be null
    - Name
  - **Square**
    - playerOn
    - Pointers to squares up, down, left and right
    - spawnPoint points to player, can be null
  - **NoteCard**
    - Keeps track of all people, weapons and rooms
    - Boolean keeps track of whether a person/weapon/room has been disproven
  - **Hand**
    - Player's cards, consists of 18/(number of players) cards (3, 5, 9)

Rules of Cluedo - <https://imgur.com/a/585r4>

1. 21 cards
  - a. 6 People
    - i. Colonel Mustard
    - ii. Miss Scarlet
    - iii. Professor Plum
    - iv. Mrs. White
    - v. Mrs. Peacock
    - vi. Reverend Green
  - b. 6 Weapons
    - i. Pistol
    - ii. Candlestick
    - iii. Dagger
    - iv. Rope
    - v. Wrench
    - vi. Pipe
  - c. 9 Rooms
    - i. Library
    - ii. Kitchen
    - iii. Ballroom
    - iv. Conservatory
    - v. Ballroom
    - vi. Lounge
    - vii. Study
    - viii. Hall
    - ix. Billiard Room
  - d. Rumour Cards
  - e. Clock Card
  - f. Keeper Card
  - g. Clue Sheet
2. Players go to the cellar/pool, once they think that they have the murderer, and make an accusation
  - a. They secretly check the answer

How chris is doing it:

- An area for the board
- An area for text/events
- An area for user input

We have a checklist for each player

## Scrum 1: Game Board

### Requirements:

#### 1. Classes

##### a. Game

- i. Calls methods to build board, shuffle deck and deal cards, create player list, and run game logic

##### b. Interface: Square

- i. **Every square needs 5 things: A location (as x-y coordinates in an int array) and four pointers to the surrounding squares**

##### ii. FloorSquare (Traversable)

1. playerOn
2. setGeography(): Pointers to squares up, down, left and right
3. location
4. spawnPoint points to player, can be null

##### iii. WallSquare (Non-Traversable)

1. Pointers to squares up, down, left and right
2. location

##### iv. EntrySquare (Room Entry)

1. Pointer to room
  - a. Player enters entry square and is immediately taken into the room
2. Pointers to squares up, down, left and right
  - a. Necessary for movement off of this square as a player is leaving the room
3. location
4. playerOn
  - a. Used when a player exits the room and ends up on this square
  - b. **The player is *only* seen as 'on' upon *exiting* the room. When a player enters the square from a FloorSquare, he is *immediately* taken to the room - playerOn is *not* set.**
5. referenceNumber
  - a. When a player is leaving a multi-exit room through a prompt, they will have to specify which exit they want to leave from. If we give each RoomEntrySquare an integer reference, this will be easier

##### c. Room

##### i. name

1. String representation of the room's name
2. Assigned by constructor **only** from BoardBuilder

##### ii. playersInRoom

1. Linked list of players currently in - as many allowed as there are players in the game

##### iii. weapon

1. Pointer to the weapon located in this room

- a. Can be null - 6 weapons for 9 rooms
- iv. secretPassage
  - 1. Either a pointer to the corresponding room or null
- v. exits
  - 1. Linked list of pointers to the RoomEntrySquare objects related to this room

#### d. Board-Builder\*

- i. A 2d array that contains all the objects that we need to make the room
  - 1. We will hard-code in the 'impassable' terrain, and entrance ways
    - a. Then we just loop the rest of the terrain as walkable

#### e. GUI

- i. This is what is actually going to show the board
      - 1. [More details below](#)
  - f. Token
    - i. Another word for player, since they all do the same thing we really don't need to have a different class for different player
  - g. Weapon
    - i. We don't have to worry about having different weapon types, because they all literally end up doing the same thing
2. Board-Builder
    - a. Correct geographical locations for square
    - b. Position of tokens and weapons on board
    - c. Graphical interface\*
  3. GUI
    - a. JFrame with 3 JPanels as sections
      - i. Board
        - 1. Can import jpg and overlay a grid?
      - ii. Text-Based display of moves/actions
        - 1. Commands are echoed to the messages on here
      - iii. Input panel for user commands
        - 1. Text field for commands and an action button to press when command is entered
          - a. When pressed:
            - i. Check that command is valid
            - ii. Check what command is:
              - 1. Movement
                - a. Check that the move is valid
                - b. Move token to location (visually and within the logical grid)
              - 2. Enter Room
                - a. Place token in room (visually and on the room playerIn pointer)
              - 3. Guess
                - a. Check guesses against appropriate lists for validity

- b. Then check for correctness
- 4. Solve
  - a. Check against solve deck
- iii. If command is invalid, prompt again
- iv. If command is valid, echo to output box
  - 1. Then, if guessing or solving, check that

What do we need to do -- who is doing it?

- 3 Jpanels
  - Board graphic (1 person)
    - George
    - Import the JPEG and set a grid over it
      - Take the image and load it into a JPanel
      - Find out how we overlay a grid layout onto the image
  - Input panel (2 people)
    - Kelsey
    - Josh
    - Set up the action listener, and use it to send info to the correct places
  - Text Output
- Surrounding Jpanel that makes sure that the other Jpanels are all together

Suggestions: (Sorry if I'm fuckin up your layout)

- Sort classes in directories?

**Sprint 2 -- REF Git Issues / Project Board for more detail**

Task	Notes	Working On	Issues
Connect user entry in UI to game engine, link to the graphical portion of the UI so it all updates from the same action listeners	<p>"Connecting the backend to the frontend"</p> <p>Connect movement to frontend</p>	<p><b>Kelsey - I/O DONE</b></p> <p><b>Josh - Graphical</b></p>	
Write a method to prompt the user for entry of players, then initialize the player list with those players	<p>JPanel with drop-down options for players would be a good idea. Could have a drop down for the 6 characters, and let you pick human, AI or off. Names taken.</p> <p>The first thing that is added to the frame -- before anything else</p> <p>Look into K's "startgame" button</p> <p>Change PlayerArray to a linked list</p>	<p><b>George - UI And list population DONE</b></p> <p>Kelsey: I set up a linked list called 'Tokens'</p>	
Construct the deck of cards and write a method to shuffle and deal	<p>Characters in this are independent of the player list - don't need Token objects, just strings.</p> <p>Each player gets <math>n/(\text{num-players})</math> rounded down cards, 3 are picked for the culprit.</p>	<p>Kelsey</p> <p>Not due in Sprint 2 per se, but should be quick</p>	
Movement	Let player move UP TO dice roll number of steps.	<p><b>Kelsey: Full movement is functional, dice aren't in the picture yet though. Since dice is on here separately I'm saying DONE.</b></p>	
User Entry	Single-letter movement commands.	Kelsey: Need to allow for	



	<p>Player can enter 'done' to end turn prematurely, or 'quit' to exit Allow for mixed case and leading/trailing zeroes</p> <p>Only accept a set of instructions (look at AcceptedUserEntry code)</p>	<p>edge cases and mixed case, and add 'quit', but normal entry is done</p>	
Room Logic	<p>When player enters a room, his token is placed in that room. Multiple tokens in a room are placed side-by-side.</p> <p>When leaving a room with multiple exits, player is asked which exit to take.</p> <p>A player landing on an EntrySquare from a FloorSquare is asked if they want to enter the room <b>as long as they have at least one more move</b>. A player entering an EntrySquare from a Room is not - they move from that square like normal (prompt if the user wants to enter the room)</p>	<p>Kelsey: Players currently enter room immediately upon stepping on square, but the square is kinda in the room so I figure that's good enough.</p> <p>Josh: Need graphical portion of this</p>	
Dice	<p>rollDice() method</p> <ul style="list-style-type: none"> <li>- Currently have a limited inner class that George made -- doesn't connect to anything</li> </ul> <p>Josh said he had written</p>	Kelsey	

	a class for the dice already, with some animations. So feel free to use that if ya want		
Error Messages	Player trying to enter a wall space  Invalid Input  Anything else we deem necessary	<b>Kelsey: I can't think of a situation where the appropriate error doesn't show up. DONE.</b>	
Bug Fixes and Cleaning	I assume the BoardImage, BoardBuilder and UserInterface classes will have to be changed a bit to look and work better and more efficiently. Pick one you want to look at and put your name on it.	Kelsey - UserInterface	
Documentation and Code Cleaning	Go through the classes and methods that you wrote and clarify, clean and modularize it so it's easier to work with from the perspective of others.	<b>Kelsey:</b> <b>UI</b> <b>BoardBuilder</b> <b>BoardSquare</b> <b>FloorSquare</b> <b>EntrySquare</b> <b>WallSquare</b> <b>DONE</b> <hr/> Josh: BoardImage	

Relevant Class Notes:

- Can make BoardPanel extend JPanel
- Override paintComponent() to paintComponent(BoardImage img)

## Sprint 3 Breakdown

New Tasks		
Task	Comments	Assigned
<ul style="list-style-type: none"> <li>- Roll for who goes first</li> </ul>	<p>Expand Dice implementation</p> <p>Highest roll first, order of entry from then on</p> <p>So set that player to the head of the circularly linked list</p>	<p>George</p> <p>DONE</p>
<ul style="list-style-type: none"> <li>- Populate Deck with 21 cards</li> </ul>	<ul style="list-style-type: none"> <li>- Assign 3 cards to "murder envelope"</li> <li>- Deal the rest to the players</li> <li>- Any extras are publicly visible</li> </ul> <p>Added all functionality, and notecard is currently viewable as text in output display. Hand is not currently viewable.</p>	<p>Kelsey</p> <p>DONE</p> <p>+ Josh (displaying the deck on the JFrame)</p>
<p>'notes'</p> <ul style="list-style-type: none"> <li>- List all tokens, weapons and rooms and have marks if player has checked them off</li> <li>- X for just this player, A for publicly visible</li> </ul>	<p>Will need another class for Notes</p> <ul style="list-style-type: none"> <li>- Prints all cards and keeps track of whether or not one has been checked off</li> <li>- Update at the beginning with any publicly viewable cards</li> </ul> <p>Think this is good unless we want to add graphical representation over the board</p>	<p>Kelsey</p> <p>DONE</p>
<p>'cheat'</p> <ul style="list-style-type: none"> <li>- Allow inspection of murder envelope</li> </ul>	<p>This is literally for debugging, it doesn't even have to be its own JPanel -- can literally just be a JOptionpane popup</p> <p>I added it as a JPanel in the output area. But it's done!</p>	<p>Kelsey</p> <p>DONE</p>
<p>'help'</p> <ul style="list-style-type: none"> <li>- Provide instructions on how to play</li> </ul>	<p>Opens in a separate window</p>	<p>George</p> <p>DONE</p>
Error messages	Put in valid error messages for invalid user input, etc	All
Fixes		
Fix exiting a room (graphically)		Josh

		DONE
<p>Allow user entry of names as they pick their character</p> <p>Fix issue where user is allowed to create the playerArray before choosing all the players</p>	<p>Done / bugs still exist with the playerCreator menu</p>	<p>George</p>
<p>Make dice rolls explicit instead of implicit</p>		<p>Kelsey</p> <p>NEXT TIME</p>