Brief

Background:

A writer of mysteries lives in seclusion on a deserted island. He finishes his new whodunit and invites editors, publishers and broadcasters to celebrate the accomplishment of the novel. Seven people, gather together for the reciprocal banquet on an isolated island, but the banquet is done in an unusual way … one of them is found in blood! What is the motive for the killing? Is it from a sense of betrayal, or just murderous crime of passion? Will the old saying, "Dead men tell no tales" be the reason of murder? Does the victim hold some hidden secrets? Who knows … but then, that's your job. You pit your wits against people who think they can get away with murder every day. You are an up and coming Detective out to solve crime – specifically, homicide. It's your job to interview the suspects and examine the scene for evidence, discover the murder weapon and where the murder took place … all before the killer can strike again!"

How to play:

The game has 7 people, assigning one to be the victim, one to be the murderer and the other 5 as innocent (but suspicious) persons of interest. Two pairs of suspects will alibi each other, while one will have no alibi and the murderer will say they were with one of the others. The player is the Detective trying to interview the suspects at the scene by moving to different locations and questioning the people found there, examining it to find evidence and/or the crime scene. The player will use simple 1 or 2 word commands to input their actions, such as MAP, GO GARDEN, QUESTION RUSSELL, SEARCH CUP, SEARCH ROOM, etc.

Victory condition:

In order to "win" the game the player must be able to do the following:
Identify the murderer by checking all suspects alibis (to confirm or disprove as true).
Find and collect the bloodied murder weapon (located in a random location of the map).
Find the bloodied crime scene and gather the murderer and all the suspects there.
Accuse the person they think committed the crime.
If they have the correct murder weapon, have found the correct location and have accused the actual murderer, the player wins.
If one, some or all of the above are not correct, the player loses and the murderer escapes!

Help Menu(display when the user type in HELP instruction):

****
Movement:
****

To see the crime scene layout : MAP
To move around the crime scene : GO roomName (for example : GO garden)

To leave the case at any time : QUIT
Rooms and Items:
To search the current room : SEARCH roomName or SEARCH ROOM (for example : SEARCH garden/SEARCH ROOM)
To search an item in the current room : SEARCH itemName (for example : SEARCH knife)
To get an item from the current room : GET itemName (for example : GET knife)
To drop an item in the current room : DROP itemName (for exampke : DROP knife)
To see what you are currently carrying : I
****
Suspects:
****
To question a suspect : QUESTION suspectFirstName (for example : QUESTION Mike)
To accuse a suspect of murder : ACCUSE suspectFirstName (for example : ACCUSE Mike)
To gather all the suspects in the current room : GATHER
To review questioned suspects notes : REVIEW
****
General:
****
To open the help menu(review this note again) : HELP

Development Outline

The Game Setup:

1. Display a splash screen showing the overview of the game to let player know the synopsis of the game, how to play and how to win the game. (text information displayed on splash screen is read in from txt files.)
2. Initialize the game elements:
   I . Load the data about the crime scene from the file and store it in an array(use two dimension array to store the room names)
   II . Create and place the "people", use constructor to initialize their attributes (Name, description, identity, alibi) create and place the potential "weapons" randomly(using rand()/RAND_MAX * array(roomName).size to get a random intger number for the index of those rooms). Create and place the items and store them in a vector(easier for add and remove).
3. Ask the player for their name and set the player's starting location
   I . Add all the extra elements for the player.
   -inventory(a vector that can add and remove elements)
   -current location
   II. use all those information to create a player object of player class

The Player's Turn:

1. Use a while loop to wrap all the code, and use a Boolean variable named endGame to decide when to jump out of the loop. Only when user enter QUIT or accuse the murder, endGame will change to true. Otherwise, the code will go through the loop and ask for

new instructions repeatly.

2. Display suitable guidance to the player so they know what they can do to continue the game. (for example, use “What’s now?” to lead the player enter instructions).
3. After receiving instructions from the player, store it into a string variable. Go to next stage and begin to process the input.
4. After processing the input, clear the screen and be ready to receive the next input.
5. After each player’s turn, a counter used to record the number of player’s turn is incremented by one (for the record of the game displayed at the end of the game and for the convenience of the additional functions)

Processing player input:

1. Guide the player to enter some instructions using specific format. (refer to the help menu above VERB + NOUN)
2. Read in the instructions and take corresponding actions

   I. Use find to get the position of space and use substr to separate VERB and NOUN. If There is no space, just take the instruction as a VERB directly

   II. Use selection structure to find the corresponding function according to that VERB.

   III. If there is a NOUN, use NOUN as parameter to determine the exact file that needs to be opened. (for example, GO bedroom. Use GO to find the goSomewhere(string somewhere) function and use bedroom as parameter of this function to open bedroom.txt file)

   IV . Load the information about the instruction from the file(for example: room description) and display to the player.
3. If the instructions entered by the player is not identified, display an error message and ask for a new instruction.

The end game:

1. The player use instruction to accuse a person as murderer. After accuse instruction is received, begin a selection structure to decide whether the requirements of winning game are met.
2. Judge with a few conditions to decide whether the player win or lose.
   I. Identify is the murderer the right person.
   II. Identify has the correct weapon been found and collected by the player.
   III. Identify is the player in the crime scene now.
3. If all the conditions are met, player win the game. Otherwise, display a “game over” message to the player and show the correct answer for the game.
4. Player can enter QUIT to end the game directly. If the player enter QUIT, a global Boolean variable called endgame will be set to true and therefore the code will jump out of the player’s turn loop. So the game quits.

Additional features:

1. After the player questions the murder, murder wants to get rid of the player. So if the player cannot solve the case in a time limit, player will be the next victim.
   I. After the player question the murder, a variable will hold the current number of

player's turn.

II. If the difference between current number of player's turn and that variable breaks the limits, endGame variable will be set to true. So the code jumps out of the loop and the game ends immediately.

2. All suspect will randomly move around the scene every five player turns
   I. If the current number of player's turn % 5 = 0, location of suspects will be reset.
   II. location variable of those suspects objects will be reset randomly by using array.size() * rand()/RAND_MAX to get a random index of the roomName array.