Upon clicking the button on the front page (*index.html, FronPage.js*)

* Call the function OpenNewWindow
  + Open a new window called “GameWindow” with a width of 1000 pixels by 800 pixels and open Game.html
* End the function

Upon loading the game page (*Game.html, GamePage.js*)

* Call the function InitialisePage
  + Pass the description of room 2 to the Feedback area
  + Execute the function RedrawMap
  + Execute the function RedrawSuspectArea and pass the room number to it
* End the function

Upon clicking one of the rooms on the map (*Game.html, GamePage.js*)

* Call the function CheckRoom and accept the value passed to it as the variable RoomNum
  + If the visited flag of the room is equal to 1 then
    - Execute the function GoToRoom and pass the value of RoomNum to it
  + End If
* End the function

Upon clicking a green room link or a lit room on the map (*Game.html, GamePage.js*)

* Call the function GoToRoom and accept the value passed to it as variable RoomNum
  + Define the variable RoomComplete as 0
  + Define the variable RoomStatus as a new array of 2 indices (0 to 1)
  + Set index 0 of RoomStatus to a blank string
  + Set index 1 of RoomStatus to "<p class = 'Inform'>Bear has investigated everything here. There is nothing more to see.</p>"
  + If the total uninvestigated clues for the room is equal to 0 then
    - Set RoomComplete to 1
  + End if
  + Pass the description of the room designated by RoomNum plus the contents of the index of RoomStatus designated by RoomComplete to the inner HTML of the feedback area
  + Set the room’s visited flag to 1
  + Execute the function RedrawMap
  + Execute the function RedrawSuspectArea and pass the room number to it
* End the function

Upon clicking a blue clue link (*Game.html, GamePage.js*)

* Call the function GoToClue and accept the value passed to it as the variable ClueNum
  + Define the variable StepAwayLink as "<p class = 'RoomLink' onClick = 'GoToRoom(", the clue’s original room number and ")'>Back away.</p>";
  + If the clue’s investigated flag is equal to 0 then
    - If there is a question for this clue then
      * Pass the room description, the question, its three answer choices and StepAwayLink to the inner HTML of the feedback area
    - Otherwise
      * Pass the room description and StepAwayLink to the inner HTML of the feedback area
    - End if
  + Otherwise
    - Pass a warning that the player has already looked at the clue to the inner HTML of the feedback area
  + End if
  + Execute the function RedrawSuspectArea and pass the room number of the clue to it
* End the function

Upon selecting an answer (*Game.html, GamePage.js*)

* Call the function GoToAnswer and accept the two values passed to it as ClueNum and Response
  + Define the variable StepAwayLink as "<p class = 'RoomLink' onClick = 'GoToRoom(", the clue’s original room number and ")'>Okay.</p>";
  + If Response is equal to the correct answer number then
    - Pass a line to inform the player that they are correct followed by the explanation for the clue question to the inner HTML of the feedback area
    - Increase the player’s score by 10
  + Otherwise
    - Pass a line to inform the player that they are wrong followed by the explanation for the clue question to the inner HTML of the feedback area
  + End if
  + Execute the function FlagClue and pass the clue number and the clue’s room number to it
  + Execute the function RedrawSuspectArea and pass the clue’s room number to it
* End the function

Upon starting the game or moving to a new room (*Game.html, GamePage.js*)

* Call the function RedrawMap
  + Define the variable MapCode as a blank string
  + Define the variable RoomStatus as a blank string
  + For each time the variable a is less than 7
    - If the visited flag of the room designated by a is equal to 1 then
      * Set RoomStatus to “Visited”
    - Otherwise
      * Set RoomStatus to “Unvisited”
    - End if
    - Append a new div using a class of MapRoomVisited or MapRoomVisited (using RoomStatus to determine the class) based on the room’s name, co-ordinates and size and include an onClick that executes the function CheckMap and passes a to it
  + Loop and increase a by 1
  + Pass MapCode to the inner HTML of the map area
* End the function

When fully investigating a clue (*Game.html, GamePage.js*)

* Call the function FlagClue and accept the two values passed to it as ClueNum and RoomNum
  + Set the investigated flag of the clue designated by ClueNum to 1
  + Reduce the total uninvestigated clues of the room designated by RoomNum by one
* End the function

When moving to a different room or investigating a clue (*Game.html, GamePage.js*)

* Call the function RedrawSuspectArea and accept the value passed to it as RoomNum
  + Define the variable DisabledState as “ disabled” (with the preceding space)
  + Define the variable GameComplete as 0
  + Define the variable ClueCount as 0
  + Define the variable ClueStatus as a new array of two indices (0 to 1)
  + Set index 0 of ClueStatus to "<p>Clues in room:", the total remaining clues in the room and "</p>"
  + Set index 1 of ClueStatus to "<p class = 'Inform'>You've found all you can.<br/>Time to point the finger.</p>"
  + For each time the variable a is less than the number of rooms
    - Increase ClueCount by the total remaining clues for the room selected by a
  + Loop and increase a by 1
  + If ClueCount is equal to 0 then
    - Set DisabledState to “”
    - Set GameComplete to 1
  + End if
  + Pass the player’s score, the index of ClueStatus selected by GameComplete, and a form with two checkboxes for the suspects and a button to make the finalise the player’s selection to the inner HTML of the suspect area
* End the function

When clicking the “Accuse!” button (*Game.html, GamePage.js*)

* Call the function MakeAccusation
  + Define the variable ChosenSuspect and pass the total of the first checkbox’s checked value, and the second checkbox’s checked value multiplied by two to it
  + Go to EndPage.html and pass ChosenSuspect and the player’s score to that page
* End the function

Upon loading the Ending Page (*EndPage.html*)

* Call the function GetEnding
* Define the variable IncomingData and pass the whole substring of the page’s address to it
* Define the variable Len and pass the character length of IncomingData to it
* Define the variable SeparatorPosition, find the location of an underscore in IncomingData and pass it to the variable
* Define the variable EndingNumber and pass the value before the separator to it
* Define the variable Score and pass the value after the separator to it
* Pass the ending designated by EndingNumber to the inner HTML of the top content pane
* Pass the Score to the inner HTML of the score counter
* End the function