# IT 140 Haunted House Game Storyboard

Harrison Labrecque

## Storyboard (Description and Map)

You are an urban explorer who goes into an abandoned house where no has lived for decades. Rumors has it that the house is haunted because the last owner disappeared without a trace. The explorer finds themselves trapped in a haunted mansion and must navigate through a series of rooms to uncover its secrets and find a way to escape. There are several items located throughout the mansion which the explorer must collect in order to leave the mansion safely. The explorer starts from the grand foyer and will need to collect a journal from the kitchen, a rusty key from the attic, a flower from the garden, a flashlight from the dining room, a photograph from a bedroom, and a medallion from the study room, the explorer will then go to the cellar and then to the storage room. The main villain in the mansion is a vengeful ghost named Morgana. She was the former owner of the mansion and seeks revenge on anyone who intrudes. The explorer must overcome her supernatural obstacles and outsmart her to successfully survive and escape the haunted mansion. Below is the layout of the mansion for the explorer to traverse.

East

Attic

Item: Rusty Key

Study room

Item: Medallion

Kitchen

Item: Journal

East

West

West

South

South

North

North

Bedroom

Item: Photograph

Dining room

Item: Flashlight

Grand Foyer

East

West

North

South

South

North

West

Storage Room

Morgana!

Garden

Item: Flower

Cellar

East

## Pseudocode or Flowchart for Code to “Move Between Rooms”

1. Declare the variable currentRoom to keep track of the player's current location.
2. Declare the variable validDirections to store the available directions the player can move.
3. Create a function displayStatus() to show the player's current room and available directions:
4. Display the message "You are in room" followed by the value of currentRoom .
5. Display the message "Available directions:" followed by the values in validDirections .
6. Create a function promptCommand() to ask the player for their command:
7. Display the message "Enter your command:".
8. Read and store the player's input in the variable command .
9. Return the value of command .
10. Create a function validateDirection(direction) to check if the given direction is valid:
11. If the direction exists in validDirections , return true.
12. Otherwise, return false.
13. Create a function movePlayer(direction) to update the player's location based on the chosen direction:
14. If direction is set to "North", update currentRoom to the room north of the current room.
15. If direction is set to "South", update currentRoom to the room south of the current room.
16. If direction is set to "East", update currentRoom to the room east of the current room.
17. If direction is set to "West", update currentRoom to the room west of the current room.
18. Set currentRoom to the starting room's value.
19. Set validDirections to the available directions from the starting room.
20. Repeat the following steps until the value of command is equal to "quit":
21. Call the displayStatus() function to show the player's current location and available directions.
22. Set the value of command to the result of the promptCommand() function.
23. If the value of command starts with "go":
24. Extract the direction from command and assign it to the variable direction.
25. If validateDirection(direction) returns true:
26. Call the movePlayer(direction) function to update the player's location.
27. Display the message "You have entered the [direction] room."
28. Else, display the message "Invalid direction. Please try again."
29. Else, display the message "Invalid command. Please try again."
30. End the program execution when the value of command is equal to "quit".

## Pseudocode or Flowchart for Code to “Get an Item”

1. **Initialize an empty inventory list or array to store the player's collected items.**
2. **Prompt the player for their command by asking "Enter your command:". Read their input and store it in a variable called command .**
3. **Write a function called getItem(item) , which will handle the logic for obtaining an item from the current room:**
4. **Check if the item exists in the player's current room.**
5. **If it does, remove the item from the room and add it to the inventory .**
6. **Display the message "You have obtained [item]."**
7. **If it doesn't exist, display the message "There is no [item] in this room."**
8. **Use a loop to repetitively ask for the player's command until they enter a command to quit:**
9. **Call the promptCommand() function to ask the player for their command and store it in the variable command .**
10. **If the first word in command is "get":**
11. **Extract the item name from command and assign it to the variable item .**
12. **Call the getItem(item) function to obtain the item.**
13. **If the first word in command is not "get", display the message "Invalid command. Please try again."**
14. **If the player enters a command to quit, end the program execution.**