Coffee Maker Quest Test Plan

Introduction

This is a test plan for a text-based game called “Coffee Maker Quest” where the player must collect coffee, cream, and sugar from a series of rooms. In order to win, the player must collect all three and drink the coffee so they may study. Failure to do so results in not being able to study and therefore lose the game. The first 12 test cases were based off fulfilling one requirement in the requirements list, in corresponding order. One concern about the testing process is that a lot of the requirements can be done when first starting the game. If these tests pass, then it might be assumed that they always will work, which may or may not be the case. However, if we tested them everywhere, it might become too exhaustive.

Requirements

- FUN-ITERATION - At each iteration of the game, the user shall be able enter one of six commands - "N" to go North, "S" to go South, "L" to Look for items, "I" for Inventory, "H" for Help, or "D" to Drink.
- FUN-UNKNOWN-COMMAND - If a player enters a command not specified by FUN-ITERATION, the system shall respond with the phrase "What?".
- FUN-INPUT-CAPS - The system shall be case-insensitive in regards to input values; that is, it shall accept capital and lower-case letters and treat them as equivalent.
- FUN-MOVE - The system shall allow a player to move North only if a door exists going North, and South only if a door exists going South.
- FUN-WIN - The player shall win the game if and only if Coffee, Sugar, and Cream have been collected by the player and then drunk.
- FUN-LOSE - The player shall lose the game if and only if the player Drinks but has not collected all of the items (Coffee, Sugar, and Cream).
- FUN-INVENTORY - Upon entering "I" for inventory, the player shall be informed of the items that he/she has collected (consisting of Coffee, Sugar, and Cream).
- FUN-LOOK - Upon entering "L" for Look, the player shall collect any items in the room and those items will be added to the player's inventory.
- FUN-HELP - Upon entering "H" for Help, the player shall be shown a listing of possible commands and what their effects are.
- FUN-UNIQ-ROOM - Each room in the house shall have a unique adjective describing it.
- FUN-UNIQ-ROOM-FURNISHING - Each room in the house shall have one and only one unique furnishing visible to the user upon entering the room.

Test Cases

1. Identifier: STARTING_COMMANDS
    a. Description: Ensure that the text of “INSTRUCTIONS (N, S, L , I , D) >” for commands show up when starting the game.
    b. Pre-conditions: Game must be started using java.
    c. Input Values: N/A

   d. Execution Steps: Start the game.
   e. Output Values: N/A
   f. Post-Conditions: Commands are present.

2. Identifier: UNKNOWN_INPUT_TEST
   a. Description: Verify that when a command entered that is not specified, the system responds with the text “What?”
   b. Pre-conditions: Be prompted to enter a command.
   c. Input Values: Type ‘a’
   d. Execution Steps: Type ‘a’
   e. Output Values: The system should respond with the text “What?”
   f. Post-Conditions: The system will ask for another character to be entered.
3. Identifier: CASE_SENSITIVITY_TEST
   a. Description: Checks if system accepts a listed instruction regardless of if that character is upper or lower case.
   b. Pre-conditions: The game should be running, asking for a user to enter a character.
   c. Input Values: ‘n’
   d. Execution Steps: Type ‘n’
   e. Output Values: N/A
   f. Post-Conditions: The game goes North, and goes to the next description.
4. Identifier: MOVE_SOUTH_EXISTENCE_TEST
   a. Description: Try moving south when a path south is not listed as an option in the description of the room.
   b. Pre-conditions: In the description of the room, have no text saying there is a south option.
   c. Input Values: ‘S’
   d. Execution Steps: Type ‘S’
   e. Output Values: Game should tell you that the way doesn’t exist.
   f. Post-Conditions: Game displays the same message as before the user typed the direction.
5. Identifier: WINNING_TEST
   a. Description: Test if you win the game when all the ingredients are collected. You should see the text “You have a cup of delicious coffee. You have some fresh cream. You have some tasty sugar.” When you have all the items.
   b. Pre-conditions: The player should start at the beginning of the game.
   c. Input Values: ‘N’, ‘L’, ‘I’, ‘D’
   d. Execution Steps: Type ‘L’, Type ‘N’, Type ‘N’, Type ‘L’, Type ‘N’, Type ‘N’, Type ‘N’, Type ‘L’, Type ‘I’, Type ‘D’
   e. Output Values: “You drink the beverage and are ready to study!” “You win!”
   f. Post-Conditions: Game exits
6. Identifier: IMMEDIATE_LOSE_TEST
   a. Description: Tests if player loses if player drinks as the first command.
   b. Pre-conditions: Game is started
   c. Input Values: ‘D’
   d. Execution Steps: Run game using java –jar command. Type ‘D’.

    e. Output Values: “You drink the air, as you have no coffee, sugar, or cream. The air is invigorating, but not invigorating enough. You cannot study. You lose!”
    f. Post-Conditions: Game exits.

7. Identifier: INVENTORY_TEST
    a. Description: Tests if the inventory is displayed when the appropriate key is entered.
    b. Pre-conditions: Game asks for input.
    c. Input Values: ‘I’
    d. Execution Steps: Type ‘I’
    e. Output Values: System displays the current inventory above the room description.
    f. Post-Conditions: Game asks for input.
8. Identifier: LOOK_TEST
    a. Description: By using to look command, the game with search for items in the room and inform the user if an item is found, and what it is; if an item exists in the room.
    b. Pre-conditions: Game asks for an input.
    c. Input Values: ‘L’
    d. Execution Steps: Type ‘L’
    e. Output Values: “There might be something here... You found some creamy cream!”
    f. Post-Conditions: Game asks for input.
9. Identifier: DRINK_TEST
    a. Description: You should observe that the game declares you loser if you fail to collect all the items in your drink and would display the items that you are missing.
    b. Pre-conditions: Game asks for input
    c. Input Values: ’D’
    d. Execution Steps: Type ‘L’, ‘I’, and then ‘D’
    e. Output Values: You drink the cream, but without caffeine, you cannot study. You lose!
    f. Post-Conditions: Game exits
10. Identifier: HELP_TEST
    a. Description: When entering ‘H’ a list of the possible commands should be displayed as text.
    b. Pre-conditions: Game asks for input.
    c. Input Values: ‘H’
    d. Execution Steps: Type ‘H’
    e. Output Values: Shows list of possible commands.
    f. Post-Conditions: Game asks for input.
11. Identifier: UNIQUE_ADJECTIVE_TEST
    a. Description: Check if each room has a unique adjective describing it in the description of the room.
    b. Pre-conditions: Game asks for input at the beginning of the game.
    c. Input Values: ‘N’
    d. Execution Steps: Check for adjective. Type ‘N’. Repeat until reaching the last room.
    e. Output Values: Unique adjectives for each room.
    f. Post-Conditions: Game asks for input.
12. Identifier: UNIQUE_FURNISHING_TEST

    a. Description: Check if there is test about one furnishing in the room that the user is currently in.
    b. Pre-conditions: Game asks for input at the beginning of the game.
    c. Input Values: ‘N’
    d. Execution Steps: Check for furnishing. Type ‘N’. Repeat until reaching the last room.
    e. Output Values: One and only one furnishing for each room.
    f. Post-Conditions: Game asks for input.

13. Identifier: KEEP_ITEMS_TEST
    a. Description: Ensure that the game allows players to keep their items collected after entering a “magical land.”
    b. Pre-conditions: Game starts and asks for input.
    c. Input Values: 'L’, ‘N’, ‘S’, ‘I’
    d. Execution Steps: Type ‘I’, Type ‘L’, Type ‘S’, Type ‘I’
    e. Output Values: YOU HAVE NO COFFEE! You have some fresh cream. YOU HAVE NO SUGAR!
    f. Post-Conditions: Game asks for input.

Traceability Matrix

	Reqs. Tested	ITERATION	UNKNOWN	INPUT_CAPS	MOVE	WIN	LOSE	INVENTORY	LOOK	HELP	UNIQ-ROOM	UNIQ-FURNISH
Test Cases												
1	1	x										
2	2	x	x									
3	2	x		x								
4	3	x		x	x							
5	4	x			x	x		x				
6	2	x					x					
7	2	x						x				
8	2	x							x			
9	4	x				x	x		x			
10	2	x								x		
11	3	x			x						x	
12	3	x			x							x
13	4	x			x			x	x			

Defects Found

1. Identifier: LOWERCASE_NORTH
    a. Description: The system incorrectly interprets the lowercase ‘n’ as an invalid input value.

   b. Reproduction Steps: When prompted to enter a character, type ‘n’.
   c. Expected Behavior: Game should accept the value and move north.
   d. Observed Behavior: Game displays the message “What?”
2. Identifier: Only Having Cream and Sugar in your beverage without coffee is a ‘win’
   a. Description: The game will tell you that you have won even you don’t have caffeinated coffee in your beverage.
   b. Reproduction Steps: Collect creamy cream from magical land, collect sugar from the farthest room.
   c. Expected Behavior: Game should display ‘You Lose!’
   d. Observed Behavior: The game outputs that you have won the game by drinking the beverage with cream and sugar but no coffee.
3. Identifier: Door names are not correct in any connections between the rooms.
   a. Description: For Example, if beige door leads north from the funny room, same door I.e. beige door be used to get back south towards funny room, but the game says it’s the smart door that’s leading the gamer to south.
   b. Reproduction Steps: Enter ‘N’ to go through all rooms
   c. Expected Behavior: Game should use one name for each door.
   d. Observed Behavior: Game displays many names for doors which is not logical.
4. Identifier: NO_HELP
   a. Description: There is no command that will show the list of possible commands.
   b. Reproduction Steps: Type ‘H’ when asked for an input.
   c. Expected Behavior: A list of possible commands should be shown, as described in the FUN-HELP requirement.
   d. Observed Behavior: The game returns “What?”
5. Identifier: NORTHERN_MAGICAL_LAND
   a. Description: Upon reaching the last room in the game and going through the non-existent door going north, the user is transported to the beginning of the game.
   b. Reproduction Steps: Type ‘N’ six times.
   c. Expected Behavior: Game should inform the user that they cannot go that direction or bring them back to that same room.
   d. Observed Behavior: Game informs user they are in a magical land and sends them back to the beginning of the game.
6. Identifier: AUTOMATIC_COLLECTING_CREAM
   a. Description: When you type ‘L’ to list the things in the small room, it would automatically add the cream in your inventory.
   b. Reproduction Steps: Type ‘L’ in the small room and go to rough room to collect sugar, you will find the sugar alongside cream too in the inventory.
   c. Expected Behavior: Game should not include items into the inventory with ‘L’ command.
   d. Observed Behavior: Game will add the cream with command ‘L’ in the small room.

7. Identifier: NO COFFEE?

a. Description: However, you need all three items in your beverage to win the game, you can win this game by only collecting the creamy cream and sugar.

b. Reproduction: Type ‘L’ then ‘I’ to collect cream from the funny room and keep Typing ‘N’ till you get to the Rough room where you type ‘L’ then ‘I’ to collect the sugar and then type ‘D’.

c. Expected Behavior: After drinking the beverage with only cream and sugar, the game should display ‘You Lose!’ because you should have coffee in your drink to win the game.

d. Observed behavior: The game will declare you winner even without coffee in your drink.

Who Did What..!

- Thomas
  - Wrote the introduction
  - Wrote test cases 1-8, 10-13
  - Created the traceability matrix for tests 1-13
  - Identified defects 1, 4, 5
- Adarsh
  - Wrote test cases 9
  - Identified defects 2, 3, 6, 7