**CS 1632 - DELIVERABLE 1: Test Plan and Traceability Matrix**

Coffee Maker Quest

Philip Ni

Marcelo d’Almeida

1. **Introduction**

The process of writing test cases started with an exploratory test and Black Box Testing in the whole application to verify how the program worked and what were the possible actions and outcomes. After a great amount of Black Box Testing, some possible defects were found (e.g. when one goes through a door that didn’t exist and the lower case ‘n’ defects).

The process continued with a reading of the requirements and relating it to the experience obtained with the previous test. Some more extensive testing was performed to cover requirement-specific rules that were not covered at that moment. The ‘Help defect’, for example, was found just only after reading it.

In order to determine what the tests should be, we first wanted to determine whether or not our requirements were met by assuming the user played the with only expected behavior; they would simply follow instructions and select options, in order to reach the end of the game and drink coffee. After performing exploratory testing, we knew the rooms with the coffee components in it and the layout of the rooms. Thus, we played assuming the user knew, as well. Using this method, we made test cases that should make our requirements succeed. For example, when we looked as the FUN-WIN and FUN-LOSE cases, we selected test cases that would make these tests pass. These tests make up our base cases.

For our edge and corner cases, we examined the requirements more carefully. By considering each requirement individually and one-by-one, we thought of ways the user may enter incorrect options, and ways in which the game may fail to meet requirements. We also checked a few different possible iterations of the game to see if the requirements were still met.

Another defect that was found came with a more careful reading. We noticed that the furniture at each room was indeed unique, but they weren’t all furniture, as stated by FUN-UNIQ-ROOM-FURNISHING requirement. Something that might pass unnoticed.

1. **Test Cases**

**IDENTIFIER**: 1a

**TEST** **CASE**: Base case; using the Look instruction in the third room

**PRECONDITIONS**: The user is in the refinanced room that displays the following text: "You see a Refinanced room. \ It has a Tight pizza. \ A Dead door leads North. \ A Smart door leads South." This is the third room and the user has not done anything here yet.

**INPUT VALUES**: Type "L" (capital L)

**EXECUTION STEPS**: When you are in the refinanced room (the third room) as specified in the precondition, type "L" (capital L) when prompted for an instruction.

**OUTPUT VALUES**: "There might be something here... \ You found some caffeinated coffee! \ You see a Refinanced room. \ It has a Tight pizza. \ A Dead door leads North. \ A Smart door leads South."

**POSTCONDITIONS**: You have looked in the third room.

**IDENTIFIER**: 1b

**TEST CASE**: Base case; using the Inventory instruction after using the Look instruction in the third room

**PRECONDITIONS**: The user is in the refinanced room that displays the following text: "You see a Refinanced room. \ It has a Tight pizza. \ A Dead door leads North. \ A Smart door leads South." This is the third room. The user has already looked in this room using the "L" (look) instruction and has not done anything else.

**INPUT VALUES**: Type "I" (capital I)

**EXECUTION STEPS**: After navigating to the third room and looking in it as mentioned in the precondition, type "I" (capital I) to check your inventory.

**OUTPUT VALUES**: "You have a cup of delicious coffee. \ YOU HAVE NO CREAM! \ YOU HAVE NO SUGAR! \ You see a Refinanced room. \ It has a Tight pizza. \ A Dead door leads North. \ A Smart door leads South."

**POSTCONDITIONS**: You have looked and checked your inventory in the third room.

**IDENTIFIER:** 2a

**TEST CASE:** Edge case; testing if user enters a character NOT in the accepted list of entries.

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room).

**INPUT VALUES:** type ‘Q’ (capital Q).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘Q’ (capital Q).

**OUTPUT VALUES:** It responds ‘What?’, describe the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’).

**IDENTIFIER:** 2b

**TEST CASE:** Base case; testing if user enters a character in the accepted list of entries.

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room).

**INPUT VALUES:** type ‘L’ (capital L).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘L’ (capital L).

**OUTPUT VALUES:** It describes if you found something (or not), the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’) and will find cream.

**IDENTIFIER:** 2c

**TEST CASE:** Corner case; testing if user enters a character accepted list of entries twice.

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room).

**INPUT VALUES:** type ‘LL’ (capital LL).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘LL’ (capital LL).

**OUTPUT VALUES:** It responds ‘What?’, describe the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’).

**IDENTIFIER**: 3a

**TEST CASE**: Base case; testing upper-case valid entry

**PRECONDITIONS**: User is in the first room that reads, "You see a Small room. \ It has a Quaint sofa. \ A Magenta door leads North" at the start of the game. This is the small room. The game should be waiting for instructions.

**INPUT VALUES**: Type "N" (capital N)

**EXECUTION STEPS**: 1. At the start of the game, the user should see the name of the game: "Coffee Maker Quest 1.0" and then the room description as specified in the precondition (3a).  
2. Enter a "N" (capital N) through the keyboard input, when prompted for instructions.

**OUTPUT VALUES**: The player should see the text, "You see a Funny room. \ It has a Sad record player. \ A Beige door leads North. \ A Massive door leads South."

**POSTCONDITIONS**: The user should now find themselves in a new room with a description that is different from the small room as mentioned in the precondition.

**IDENTIFIER**: 3b

**TEST CASE**: Base case; testing lower-case valid entry

**PRECONDITIONS**: User is in the first room that reads, "You see a Small room. \ It has a Quaint sofa. \ A Magenta door leads North" at the start of the game. This is the small room. The game should be waiting for instructions.

**INPUT VALUES**: Type "n" (lower-case n)

**EXECUTION STEPS**: 1. At the start of the game, the user should see the name of the game: "Coffee Maker Quest 1.0" and then the room description as specified in the precondition (3b).  
2. Enter a "n" (lower-case n) through the keyboard input, when prompted for instructions.

**OUTPUT VALUES**: The player should see the text, "You see a Funny room. \ It has a Sad record player. \ A Beige door leads North. \ A Massive door leads South."

**POSTCONDITIONS**: The user should now find themselves in a new room with a description that is different from the small room as mentioned in the precondition.

**IDENTIFIER**: 4a

**TEST CASE**: Base case; testing moving North

**PRECONDITIONS**: User is in the first room that reads, "You see a Small room. \ It has a Quaint sofa. \ A Magenta door leads North" at the start of the game. This is the small room. The game should be waiting for instructions.

**INPUT VALUES**: Type "N" (capital N)

**EXECUTION STEPS**: 1. At the start of the game, the user should see the name of the game: "Coffee Maker Quest 1.0" and then the room description as specified in the precondition (4a).  
2. Enter a "N" (capital N) through the keyboard input, when prompted for instructions.

**OUTPUT VALUES**: The player should see the text, "You see a Funny room. \ It has a Sad record player. \ A Beige door leads North. \ A Massive door leads South."

**POSTCONDITIONS**: The user should now find themselves in the funny room that is North of the small room mentioned in the precondition.

**IDENTIFIER**: 4b

**TEST CASE**: Edge case; testing moving South when a southern door does not exist.

**PRECONDITIONS**: User is in the first room that reads, "You see a Small room. \ It has a Quaint sofa. \ A Magenta door leads North" at the start of the game. This is the small room. The game should be waiting for instructions.

**INPUT VALUES**: Type "S" (capital S)

**EXECUTION STEPS**: 1. At the start of the game, the user should see the name of the game: "Coffee Maker Quest 1.0" and then the room description as specified in the precondition (4b).  
2. Enter a "S" (capital S) through the keyboard input, when prompted for instructions.

**OUTPUT VALUES**: The player should see the text reflecting that they have not moved or changed state from the precondition.

**POSTCONDITIONS**: The user should find themselves still in the small room as stated in the precondition.

**IDENTIFIER**: 4c

**TEST CASE**: Edge case; testing moving North when a northern door does not exist.

**PRECONDITIONS**: You are in the rough room and it says, “You see a Rough room. \ It has a Perfect air hockey table. \ A Minimalist door leads South.” It is the last room. You have already travelled through all the other rooms to the south of this room.

**INPUT VALUES**: Type "N" (capital N)

**EXECUTION STEPS**: 1. Navigate to the rough room -- which is the northern most room -- if you aren't there already.  
2. While you are in the rough room, type "N" when prompted for an instruction.

**OUTPUT VALUES**: The player should see the text reflecting that they have not moved or changed state in the precondition.

**POSTCONDITIONS**: The user should find themselves still in the rough room as stated in the precondition.

**IDENTIFIER**: 5a

**TEST CASE**: Base case; you have all ingredients (full inventory) and you take a drink.

**PRECONDITIONS**: You are in the rough room and it says, “You see a Rough room. \ It has a Perfect air hockey table. \ A Minimalist door leads South.” It is the last room. You have already looked in all the other rooms to the south and your inventory should have all possible items (coffee, cream, and sugar).

**INPUT VALUES**: Type "D" (capital D)

**EXECUTION STEPS**: 1. After looking in all other rooms (as mentioned in the precondition), navigate to the rough room -- which is the northern most room -- if you aren't there already.  
2. While you are in the rough room, type "D" when prompted for an instruction.

**OUTPUT VALUES**: The game lists the items in your inventory again: "You have a cup of delicious coffee. \ You have some fresh cream. \ You have some tasty sugar," and then says, "You drink the beverage and are ready to study! \ You win!"

**POSTCONDITIONS**: You have won. The game closes with no error.

**IDENTIFIER:** 5b

**TEST CASE:** Edge case; you have all ingredients (full inventory) and you double-back to check all rooms again, only to take a drink in the first room.

**PRECONDITIONS:** You are in the rough room and it says, “You see a Rough room. \ It has a Perfect air hockey table. \ A Minimalist door leads South.” It is the last room. You have already looked in all the other rooms to the south and your inventory should have all possible items (coffee, cream, and sugar).

**INPUT VALUES:** Using "S" (capital S) and "L" (capital L) to double back to the first room (small room). Type "D" (capital D)

**EXECUTION STEPS:** 1. After looking in all other rooms (as mentioned in the precondition), you should be in the rough room with the items mentioned in the precondition.  
2. Select look by typing "L" (capital L) when prompted for an instruction in the rough room again.  
3. Select move South by typing "S" (capital) when prompted for an instruction in the rough room.  
4. Continue steps 2-3 until you have looked in all the rooms again and find yourself in the small room at the very beginning of the game (the room you started in). Essentially, you want to double-back and navigate to the room at the beginning, picking up the items into your inventory again as you look through the rooms a second time.  
5. While you are in the small room as specified in step 4, type "D" when prompted for an instruction.

**OUTPUT VALUES:** The game lists the items in your inventory again: "You have a cup of delicious coffee. \ You have some fresh cream. \ You have some tasty sugar," and then says, "You drink the beverage and are ready to study! \ You win!"

**POSTCONDITIONS:** You have won. The game closes with no error.

**IDENTIFIER**: 6a

**TEST CASE**: Edge case; the player drinks without coffee, cream, or sugar in their inventory.

**PRECONDITIONS**: User is in the first room that reads, "You see a Small room. \ It has a Quaint sofa. \ A Magenta door leads North" at the start of the game. This is the small room. The game should be waiting for instructions.

**INPUT VALUES**: Type "D" (capital D)

**EXECUTION STEPS**: Once the game starts, and you find yourself in the small room as specified in the precondition, type "D" (capital D) when prompted for an instruction.

**OUTPUT VALUES**: The game should tell you that you have no items in your inventory and you lose: "YOU HAVE NO COFFEE! \ YOU HAVE NO CREAM! \ YOU HAVE NO SUGAR! \ \ 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!"

**POSTCONDITIONS**: The game ends with the user in a lose state.

**IDENTIFIER**: 6b

**TEST CASE**: Base case; the player drinks without coffee or sugar in their inventory. They only have cream.

**PRECONDITIONS**: User is in the first room that reads, "You see a Small room. \ It has a Quaint sofa. \ A Magenta door leads North" at the start of the game. This is the small room. The player should have already looked in this room and found the cream. The game should be waiting for another instruction.

**INPUT VALUES**: Type "D" (capital D)

**EXECUTION STEPS**: 1. Once the game starts, and you find yourself in the small room as specified in the precondition, look in this room first.  
2. After looking in this small room and finding cream, type "D" (capital D) when prompted for an instruction.

**OUTPUT VALUES**: The game should tell you that you have no items in your inventory and you lose: "YOU HAVE NO COFFEE! \ You have some fresh cream. \ YOU HAVE NO SUGAR! \ \ You drink the cream, but without caffeine, you cannot study. \ You lose!"

**POSTCONDITIONS**: The game ends with the user in a lose state.

**IDENTIFIER:** 7a

**TEST CASE:** Base case; testing when user checks an empty inventory

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room).

**INPUT VALUES:** type ‘I’ (capital I).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘I’ (capital I).

**OUTPUT VALUES:** It describes your inventory, the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’) and the inventory has no coffee, no cream and no sugar.

**IDENTIFIER:** 7b

**TEST CASE:** Base case; testing when user checks an inventory with cream.

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room) and you have cream in your inventory.

**INPUT VALUES:** type ‘I’ (capital I).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘I’ (capital I).

**OUTPUT VALUES:** It describes your inventory, the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’) and the inventory has cream, but has no coffee and no sugar.

**IDENTIFIER:** 7c

**TEST CASE:** Base case; testing when user checks a full inventory.

**PRECONDITIONS:** You are in the room that says “You see a Rough room. It has a Perfect air hockey table. A Minimalist door leads South.” (It is the last room) and you have cream, coffee and sugar.

**INPUT VALUES:** type ‘I’ (capital I).

**EXECUTION STEPS:** When you are in the ‘Rough room’ (defined in the precondition), when prompted type ‘I’ (capital I).

**OUTPUT VALUES:** It describes your inventory, the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Rough room’) and the inventory has cream, coffee and sugar.

**IDENTIFIER:** 8a

**TEST CASE:** Base case; testing if user looks at ‘Small room’ and gets cream.

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room) and you don’t have cream in your inventory.

**INPUT VALUES:** type ‘L’ (capital L).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘L’ (capital L).

**OUTPUT VALUES:** It responds “There might be something here... You found some creamy cream!” It describes that you found cream, the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’) and just cream was added to your inventory.

**IDENTIFIER:** 8b

**TEST CASE:** Base case; testing if user looks at place that has no item and gets nothing to his/her inventory.

**PRECONDITIONS:** You are in the room that says “You see a Funny room. It has a Sad record player. A Beige door leads North. A Massive door leads South.” (It is the second room).

**INPUT VALUES:** type ‘L’ (capital L).

**EXECUTION STEPS:** When you are in the ‘Funny room’ (defined in the precondition), when prompted type ‘L’ (capital L).

**OUTPUT VALUES:** It responds “You don't see anything out of the ordinary.” It describes that you found nothing, the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Funny room’) and the inventory didn’t change.

**IDENTIFIER:** 9a

**TEST CASE:** Base case; testing when user asks for help

**PRECONDITIONS:** You are in the room that says “You see a Small room. It has a Quaint sofa. A Magenta door leads North” (It is the first room).

**INPUT VALUES:** type ‘H’ (capital H).

**EXECUTION STEPS:** When you are in the ‘Small room’ (defined in the precondition), when prompted type ‘H’ (capital H).

**OUTPUT VALUES:** It describes each one of the instructions commands ("N" to go North, "S" to go South, "L" to Look for items, "I" for Inventory, "H" for Help, or "D" to Drink.), the current room and it prompts user for another instruction.

**POSTCONDITIONS:** You still are at the same room (‘Small room’).

**IDENTIFIER:** 10a

**TEST CASE:** Base case; testing if all rooms have different names.

**PRECONDITIONS:** You are in the room named “Small room” (It is the first room).

**INPUT VALUES:** type ‘N’ (capital N) several times.

**EXECUTION STEPS:** Walk North through all 6 rooms (Small room, Funny room, Refinanced room, Dumb room, Bloodthirsty room and Rough room) and check their names to verify they are unique. When prompted, if there is any North door, type ‘N’ (capital N).

**OUTPUT VALUES:** It describes the last room and it prompts user for another instruction.

**POSTCONDITIONS:** You are at the last room (‘Rough room’).

**IDENTIFIER:** 11a

**TEST CASE:** Base case; testing if all rooms have one and only one unique furnish.

PRECONDITIONS: You are in the room named “Small room”, that has a Quaint sofa. (It is the first room).

**INPUT VALUES:** type ‘N’ (capital N) several times.

**EXECUTION STEPS:** Walk North through all 6 rooms (Small room, Funny room, Refinanced room, Dumb room, Bloodthirsty room and Rough room) and check their furnish (Quaint sofa, Sad record player, Tight pizza, Flat energy drink, Beautiful bag of money and Perfect air hockey table, respectively) to verify they are unique. When prompted, if there is any North door, type ‘N’ (capital N).

**OUTPUT VALUES:** It describes the last room and it prompts user for another instruction.

**POSTCONDITIONS:** You are at the last room (‘Rough room’).

1. **Traceability Matrix**
   1. FUN-ITERATION: 1a, 1b
   2. FUN-UNKNOWN-COMMAND: 2a, 2b, 2c
   3. FUN-INPUT-CAPS: 3a, 3b
   4. FUN-MOVE: 4a, 4b, 4c
   5. FUN-WIN: 5a, 5b
   6. FUN-LOSE: 6a, 6b
   7. FUN-INVENTORY: 7a, 7b, 7c
   8. FUN-LOOK: 8a, 8b
   9. FUN-HELP: 9a
   10. FUN-UNIQ-ROOM: 10a
   11. FUN-UNIQ-ROOM-FURNISHING: 11a
2. **Defects**

**SUMMARY**: Lower-case "n" is not a valid entry.

**DESCRIPTION**: The game does not allow for lower-case "n" instead of upper-case "N" to move North, when the user has the option to move North.

**REPRODUCTION** **STEPS**: Start the game in the Small room and try to enter lower-case "n" to move North.

**EXPECTED BEHAVIOR**: The user should have moved North into the Funny room exactly as if they had entered upper-case "N".

**OBSERVED BEHAVIOR**: The game returns with the text "What?" as if the instruction (lower-case "n") was invalid.

**SUMMARY**: The game allows the user to go North from the Rough room even though a northern door does not exist.

**DESCRIPTION**: You can select the 'move North' ("N") instruction while you are in the last room, even though a northern door does not exist and is not mentioned in the room's descriptions.

**REPRODUCTION STEPS**: In the last room (the Rough room), type "N" (capital N) to go North. You will find that the room's description only lists a door to the South as the only door in the room.

**EXPECTED BEHAVIOR**: Typing "N" in the Rough room should be an invalid instruction. There should be no door or movement to any room, dimension, magical land, etc. The user remains in the Rough room and does not move.

**OBSERVED BEHAVIOR**: The game informs the user that "You are in a magical land! But you are returned to the beginning!" and sends them back to the Small room where the user originated at the start of the game.

**SUMMARY**: The game allows the user to go South from the Small room even though a southern door does not exist.

**DESCRIPTION**: You can select the 'move South' instruction while you are in the first room, even though a southern door does not exist and is not mentioned in the room's descriptions.

**REPRODUCTION STEPS**: In the first room (the Small room), type "S" (capital S) to go South. You will find that the room's description only lists a door to the North as the only door in the room.

**EXPECTED BEHAVIOR**: Typing "S" in the Small room should be an invalid instruction. There should be no door or movement to any room, dimension, magical land, etc. The user remains in the Small room and does not move.

**OBSERVED BEHAVIOR**: The game informs the user that "You are in a magical land! But you are returned to the beginning!" and sends them back to the Small room.

**SUMMARY:** No help is showed to the user **DESCRIPTION:** No help is showed to the user when he types ‘H’ (uppercase and lowercase) **REPRODUCTION STEPS:** In the first room (‘Small Room’), type ‘H’ when prompted (capital H). **EXPECTED BEHAVIOR:** A help message appears when the user types ‘H’ **OBSERVED BEHAVIOR:** ‘H’ (uppercase and lowercase) are not recognized as a valid input and therefore no help information is showed

**SUMMARY:** Some ‘furnishing items’ are not furnishing (e.g. ‘Tight pizza’, ‘Flat energy drink’ and ‘Beautiful bag of money’) **DESCRIPTION:** As you go through the rooms, some of the items located in the room doesn’t meet the requirement that they should be furnishing **REPRODUCTION STEPS:** Walk North through all 6 rooms (Small room, Funny room, Refinanced room, Dumb room, Bloodthirsty room and Rough room) and check if their ‘furnish’ are actually a real furnish. When prompted, if there is any North door, type ‘N’ (capital N). **EXPECTED BEHAVIOR:** Each room shall have one and only one unique furnishing **OBSERVED BEHAVIOR:** There are rooms that have items other than furnishing