University of Pittsburgh

Coffee Maker Quest

CS 1699 – Deliverable1: Test Plan and Traceability Matrix

John Abraham & Joseph Hagner
Bill Laboon
1/27/2015

TABLE OF CONTENTS:

1. INTRODUCTION.....1
2. TEST CASES……..……..……..……2
3. TRACEABILITY MATRIX…….……..8
4. DEFECTS FOUND…….……9

1. INTRODUCTION:

"Coffee Maker Quest" is a simple game. The goal is to get coffee, sugar, and cream, and then drink it so that you can stay up and study. In order to do so, you need to visit several rooms in a house and look around. Once you have obtained all the necessary elements, you win. If you decide to drink before getting all of the necessary elements, you lose.

The purpose of this document is to outline the test plan that we developed to test "Coffee Maker Quest" based on the eleven given requirements for the game: iteration, unknown command, input caps, move, win, lose, inventory, look, help, unique room, and unique room furnishing; and to outline the steps that we followed to find the defects in the game.

On the following page we list the test cases for each requirement. A test case is the lowest level of a test plan and it consists of inputs values, pre-conditions, execution steps, output values, and post-conditions. For most requirements we created more than one test case in order to take into account as many inputs as the game requires. For example, the "FUN-INPUT-CAPS" requirement requires that: "the system shall be case-insensitive in regard to input values; that is, it shall accept the capital and lower-case letter and treat them as equivalent." So for this requirement two test cases were made: one to test all the possible input characters as capital letters, and another one to test all the possible input characters as lower-case letters; and the output values were observed to check if they satisfy the expected output values to determine whether to report a defect or not.

For a small program like this, there are not many ways one can act outside of the normal parameters of the game. Therefore, the list of edge cases we tested is very small. For the requirement FUN-UNKNOWN-COMMAND, we first tested input of strings that contained valid characters (N, S, I, L, D, H), but were multiple characters long and thus invalid. This tests whether we could trick the system to accept invalid input by feeding it some valid characters. The other edge case we tested was also for the requirement FUN-UNKNOWN-COMMAND where we tested input with completely irrelevant characters. This was to test how the system would react to input it was not expecting.

2. TEST CASES:

IDENTIFIER: 1
TEST CASE: Press all of the buttons
PRECONDITIONS: The game is running
EXECUTION STEPS: 1) Press the letter "N"
2) Press enter
3) Press the letter "S"
4) Press enter
5) Press the letter "L"
6) Press enter
7) Press the letter "H"
8) Press enter
9) Press the letter "I"
10) Press enter
11) Press the letter "D"
12) Press enter
POSTCONDITIONS: Every time 'enter' is pressed, the game shall present a message that does not start with the text 'What?'

IDENTIFIER: 2
TEST CASE: Invalid strings containing at least one valid character
PRECONDITIONS: The game is running
EXECUTION STEPS: 1) Type the string "Na"
2) Press enter
3) Type the string "SS"
4) Press enter
5) Type the string "LIDH"
6) Press enter
POSTCONDITIONS: Every time 'enter' is pressed, the game shall present the message "What?" and then describe the current room.

IDENTIFIER: 3
TEST CASE: Strings containing only invalid characters
PRECONDITIONS: The game is running
EXECUTION STEPS: 1) Type the string "a"
2) Press enter
3) Type the string "bb"
4) Press enter
5) Type the string "123"
6) Press enter
7) Type nothing
8) Press enter
POSTCONDITIONS: -Every time 'enter' is pressed, the game shall present the message "What?" and then describe the current room.

IDENTIFIER: 4
TEST CASE: Press all uppercase buttons
PRECONDITIONS: The game is running
EXECUTION STEPS: 1) Press the letter "N"
2) Press enter
3) Press the letter "S"
4) Press enter
5) Press the letter "L"
6) Press enter
7) Press the letter "H"
8) Press enter
9) Press the letter "I"
10) Press enter
11) Press the letter "D"
12) Press enter
POSTCONDITIONS: -Every time 'enter' is pressed, the game shall present a message that is does not start with the text 'What?'

IDENTIFIER: 5
TEST CASE: Press all lowercase buttons
PRECONDITIONS: The game is running
EXECUTION STEPS: -Press the letter "n"
-Press enter
-Press the letter "s"
-Press enter
-Press the letter "l"
-Press enter
-Press the letter "h"
-Press enter
-Press the letter "i"
-Press enter
-Press the letter "d"
-Press enter
POSTCONDITIONS: -Every time 'enter' is pressed, the game shall present a message that is does not start with the text 'What?'

IDENTIFIER: 6
TEST CASE: Move North only if a door exists going North.
PRECONDITIONS: Be in a room where there does not exist a door going North.
INPUT VALUES: the character "N"
EXECUTION STEPS: 1) Run the program
2) Keep pressing "N" and Enter until you find a room with no door going North.
3) Once you’re there press 'N"
POSTCONDITIONS: A message should appear telling the user that the move is not possible
OUTPUT VALUES: Back to beginning (FAILURE TO MEET REQUIREMENT)

IDENTIFIER: 7
TEST CASE: Move South only if a door exists going South.
PRECONDITIONS: Be in a room where there does not exist a door going South.
INPUT VALUES: the character "S"
EXECUTION STEPS: 1) Run the program
2) Initially there's only one door and it goes North, so press "S"
POSTCONDITIONS: A message should appear telling the user that the move is not possible and keeps user in the same room
OUTPUT VALUES: User stays in the same room (SUCCESS: REQUIREMENT MET)

IDENTIFIER: 8
TEST CASE: Test winning conditions
PRECONDITIONS: The player has coffee, cream, and sugar in their inventory
EXECUTION STEPS: 1) Press the letter "D"
2) Press enter
OUTPUT: Error code of 0
POSTCONDITIONS: -The console shall print out that the player has won and the game exits.

IDENTIFIER: 9
TEST CASE: Drink with zero items.
PRECONDITIONS: The game is running and no commands have been entered
INPUT VALUES:
EXECUTION STEPS: 1) Press the character "D"
2) Press enter
OUTPUT VALUES: Error code of 1
POSTCONDITIONS: The game shall display the message "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!"

IDENTIFIER: 10
TEST CASE: Drink with one item
PRECONDITIONS: The game is running and the player has cream in their inventory
INPUT VALUES:
EXECUTION STEPS: 1) Press the character "D"
2) Press enter
OUTPUT VALUES: Error code of 1
POSTCONDITIONS: The game shall display the message "You drink the cream, but without caffeine, you cannot study. You lose!"

IDENTIFIER: 11
TEST CASE: Drink with two items
PRECONDITIONS: The game is running and the character has coffee and cream in their inventory
INPUT VALUES:
EXECUTION STEPS: 1) Press the character "D"
2) Press enter
OUTPUT VALUES: Error code of 1

POSTCONDITIONS: The game shall display the message "Without sugar, the coffee is too bitter. You cannot study. You lose!"

IDENTIFIER: 12

TEST CASE: Inventory check.
PRECONDITION: The game is running and the player has cream in their inventory
INPUT VALUES: the character "I"
EXECUTION STEPS: 1) Press I
2) Press enter
POSTCONDITIONS: -The console shall print out NO SUGAR NO COFFE You have cream

IDENTIFIER: 13

TEST CASE: Get cream
PRECONDITIONS: The game is running, but no commands have been entered yet
EXECUTION STEPS:
1) Press the character "L"
2) Press enter
3) Press the character "I"
4) Press enter
POSTCONDITIONS: -The game shall say "There might be something here... You found some creamy cream!" upon pressing "L"
-The game shall say "YOU HAVE NO COFFEE! You have some fresh cream. YOU HAVE NO SUGAR!" upon entering "I"

IDENTIFIER: 14

TEST CASE: Get coffee
PRECONDITIONS: The game is running, but no commands have been entered yet
EXECUTION STEPS:
1) Press the character "N"
2) Press enter
3) Press the character "N"
4) Press enter
5) Press the character "L"
6) Press enter
7) Press the character "I"
8) Press enter
POSTCONDITIONS: -The game shall say "There might be something here... You found some caffeinated coffee!" upon pressing "L"
-The game shall say "You have a cup of delicious coffee. YOU HAVE NO CREAM! YOU HAVE NO SUGAR!" upon entering "I"

IDENTIFIER: 15

TEST CASE: Get sugar
PRECONDITIONS: The game is running, but no commands have been entered yet
EXECUTION STEPS:
1) Press the character "N"

2) Press enter
3) Repeat steps 1-2 until the player is in a room with no door to the north
4) Press the character "L"
5) Press enter
6) Press the character "I"
7) Press enter

POSTCONDITIONS: -The game shall say "There might be something here... You found some sweet sugar!" upon pressing "L"

-The game shall say "YOU HAVE NO COFFEE! YOU HAVE NO CREAM! You have some tasty sugar." upon entering "I"

IDENTIFIER: 16

TEST CASE: The player shall be shown a listing of possible commands and what their effects are when entering "H"

PRECONDITIONS: None

INPUT VALUES: The character "H"

EXECUTION STEPS: 1) Run the program

2) Type the character "H" into the command line

POSTCONDITIONS: None

OUTPUT VALUES: The player shall be shown a listing of possible commands and what their effects are (FAILURE: REQUIREMENT NOT MET)

IDENTIFIER: 17

TEST CASE: The player shall be shown a listing of possible commands and what their effects are when entering "h"

PRECONDITIONS: None

INPUT VALUES: The character "h"

EXECUTION STEPS: 1) Run the program

2) Type the character "h" into the command line

POSTCONDITIONS: None

OUTPUT VALUES: The player shall be shown a listing of possible commands and what their effects are (FAILURE: REQUIREMENT NOT MET)

IDENTIFIER: 18

TEST CASE: Check every room for distinct room adjectives.

PRECONDITIONS: The game is running, but no commands have been entered yet

EXECUTION STEPS:

1) Record the room's description (pencil and paper?)
2) Press the character "N"
3) Press enter
4) Record the room's description (pencil and paper?)
5) Repeat steps 2-4 until the player is in a room with no door to the north

POSTCONDITIONS: The record of room descriptions shall have no duplicate items on it.

IDENTIFIER: 19
TEST CASE: Check every room for distinct furniture.
PRECONDITIONS: The game is running, but no commands have been entered yet
EXECUTION STEPS:

1) Record the room's items (pencil and paper?)
2) Press the character "N"
3) Press enter
4) Record the room's items (pencil and paper?)
5) Repeat steps 2-4 until the player is in a room with no door to the north

POSTCONDITIONS: The record of room items shall have no duplicate items and every room shall have exactly one item in it.

3. TRACEABILITY MATRIX:

FUN-ITERATION : 1
FUN-UNKNOWN-COMMAND : 2, 3
FUN-INPUT-CAPS : 4, 5
FUN-MOVE : 6, 7
FUN-WIN : 8
FUN-LOSE : 9, 10, 11
FUN-INVENTORY : 12
FUN-LOOK : 13, 14, 15
FUN-HELP : 16, 17
FUN-UNIQ-ROOM : 18
FUN-UNIQ-ROOM-FURNISHING : 19

Table: list of requirements (left column) matched with the identifier numbers listed in each test case.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ITERATION	•																		
UNKNOWN-COMMAND		•	•																
INPUT-CAPS				•	•														
MOVE						•	•												
WIN								•											
LOSE									•	•	•								
INVENTORY												•							
LOOK													•	•	•				
HELP																•	•		
UNIQ-ROOM																		•	
UNIQ-RM-FURNISHING																			•

5. DEFECTS FOUND:

DEFECT #1:
DESCRIPTION: 'N' not case insensitive
SUMMARY: The letter 'N' is not case insensitive. Only the uppercase, 'N', produces the expected results. 'n' is recognized as an invalid character. This violates the requirement FUN-INPUT-CAPS
REPRODUCTION STEPS:
1) Start the game
2) Confirm that you are in the 'Small Room'
3) Type the character 'n'
4) Press enter
EXPECTED BEHAVIOR: The player should be shown a description of the 'Funny Room.'
OBSERVED BEHAVIOR: The game shows the message "What?" and shows a description of the 'Small Room.'

DEFECT #2:
DESCRIPTION: 'H' is not recognized
SUMMARY: The character 'H' is not recognized as a valid input command. This violates the requirements FUN-ITERATION and FUN-HELP.
REPRODUCTION STEPS:
1) Start the game
2) Type the character 'H'
3) Press enter
EXPECTED BEHAVIOR: The player should be shown a listing of possible commands and what their effects are.
OBSERVED BEHAVIOR: The game shows the message "What?"

DEFECT #3:
DESCRIPTION: Can move north with no north door
SUMMARY: The game allows the player to move north even when no door exists to the north. This violates the requirement FUN-MOVE.
REPRODUCTION STEPS:
1) Start the game
2) Keep pressing "N" and Enter until you find a room with no door going North
3) Type the character "N"
4) Press enter
EXPECTED BEHAVIOR: The game shall indicate in some way that the player is unable to move north.
OBSERVED BEHAVIOR: The game presents the message "You are in a magical land! But you are returned to the beginning!" which indicates that movement occurred without a door existing to the north.

DEFECT #4:
DESCRIPTION: Can move south with no south door
SUMMARY: The game allows the player to move south even when no door exists to the south. This violates the requirement FUN-MOVE.
REPRODUCTION STEPS:

1) Start the game
2) Confirm that no door exists to the south.
2) Type the character "S"
3) Press enter

EXPECTED BEHAVIOR: The game shall indicate in some way that the player is unable to move south.
OBSERVED BEHAVIOR: The game presents the message "You are in a magical land! But you are returned to the beginning!" which indicates that movement occurred without a door existing to the south.