



CS 1632 Software Quality Assurance

Deliverable 1

Member 1 : Xinke Chen

Member 2 : Mouhammadou Dabo

1. Introduction

Starting off with the requirements, we broke the requirements into several different classes of behavior; input, output and functionality. The classes helped to inform what kind of test cases were needed, and also informed the equivalence classes and any potential boundary values to be identified.

The hardest requirements to write test cases for were for the FUN-UNIQ-ROOM and FUN-UNIQ-FURNISHING requirements. In writing these test cases, they required multiple steps with each step generating output that had to be considered in the test case. It was somewhat challenging to find an appropriate phrasing for the post conditions of these test cases.

As for edge cases, the ones that immediately came to mind, since we were dealing with text, were using Unicode characters as input for the program, i.e. emojis. After trying this, the program was largely unphased and didn't create any new defects/errors.

2. Traceability Matrix

FUN-ITERATION

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

FUN-UNKNOWN-COMMAND

21

FUN-INPUT-CAPS

2, 4, 6, 8, 10, 12

FUN-MOVE

1, 2, 3, 4, 13, 14

FUN-WIN

16

FUN-LOSE

15

FUN-INVENTORY

11, 12, 17

FUN-LOOK

7, 8, 16

FUN-HELP

5, 6

FUN-UNIQ-ROOM

19

FUN-UNIQ-ROOM-FURNISHING

20

3. Test Cases

TC-1

- **IDENTIFIER:** ITERATION-NORTH-UPPER
- **TEST CASE:** For the input "N" the player is able to go north
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "N" and press enter
- **POSTCONDITIONS:**
 - o Program should continue to run
 - o Output on the first line should be "You see a Funny room"

TC-2

- **IDENTIFIER:** ITERATION-NORTH-LOWER
- **TEST CASE:** That for the inputs "n" that the player is able to go north
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "n" and press enter
- **POSTCONDITIONS:**
 - o Program should continue to run
 - o Output first line should be "You see a Funny room"

TC-3

- **IDENTIFIER:** ITERATION-SOUTH-UPPER
- **TEST CASE:** That for the inputs "S" that the player is able to go south
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "S" and press enter
- **POSTCONDITIONS:**
 - o Program should continue to run
 - o Output should contain " A door in that direction does not exist."

TC-4

- **IDENTIFIER:** ITERATION-SOUTH -LOWER
- **TEST CASE:** That for the inputs “s” that the player is able to go south
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type “s” and press enter
- **POSTCONDITIONS:**
 - o Program should continue to run
 - o Output should contain “ A door in that direction does not exist.”

TC-5

- **IDENTIFIER:** ITERATION-HELP-UPPER
- **TEST CASE:** That for the inputs “H” that the player is able to see output on how to play
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type “H” and press enter
- **POSTCONDITIONS:**
 - o Program should continue to run
 - o Output should contain available commands to play the game

TC-6

- **IDENTIFIER:** ITERATION-HELP-LOWER
- **TEST CASE:** That for the inputs “h” that the player i able to see output on how to play
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type “h” and press enter 3
- **POSTCONDITIONS:**
 - o Program should continue to run
 - o Output should contain available commands to play the game

TC-7

- **IDENTIFIER:** ITERATION-LOOK-UPPER
- **TEST CASE:** That for the inputs “L” that the player is see the items in the current room

- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “L” and press enter
- **POSTCONDITIONS:**
 - Program should continue to run
 - Output should be the following:
 - There might be something here...
 - You found some creamy cream!
 - You see a Small room.
 - It has a Quaint sofa.
 - A Magenta door leads North.

TC-8

- **IDENTIFIER:** ITERATION-LOOK-UPPER
- **TEST CASE:** That for the inputs “l” that the player is see the items in the current room
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “l” and press enter
- **POSTCONDITIONS:**
 - Program should continue to run
 - Output should be the following:
 - There might be something here...
 - You found some creamy cream!
 - You see a Small room.
 - It has a Quaint sofa.
 - A Magenta door leads North.

TC-9

- **IDENTIFIER:** ITERATION-DRINK-UPPER
- **TEST CASE:** That for the inputs “D” that the player is able to try to drink in the game
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “D” and press enter

- **POSTCONDITIONS:**
 - Program should quit with exit code 1.
 - Output should be the following:
 - 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!

TC-10

- **IDENTIFIER:** ITERATION-DRINK-LOWER
- **TEST CASE:** That for the inputs “d” that the player is able to try to drink in the game
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “d” and press enter
- **POSTCONDITIONS:**
 - Program should quit with error code 1.
 - Output should be the following:
 - 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!

TC-11

- **IDENTIFIER:** ITERATION-INVENTORY-UPPER
- **TEST CASE:** That for the inputs “I” that the player is able to try to drink in the game
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “I” and press enter
- **POSTCONDITIONS:**
 - Program should quit with error code 1.
 - Output should be the following:
 - YOU HAVE NO COFFEE!
 - YOU HAVE NO CREAM!
 - YOU HAVE NO SUGAR!

TC-12

- **IDENTIFIER:** ITERATION-INVENTORY-LOWER
- **TEST CASE:** That for the inputs “i” that the player is able to try to drink in the game
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “i” and press enter
- **POSTCONDITIONS:**
 - Program should quit with error code 1.
 - Output should be the following:
 - YOU HAVE NO COFFEE!
 - YOU HAVE NO CREAM!
 - YOU HAVE NO SUGAR!

TC-13

- **IDENTIFIER:** MOVE-VALID
- **TEST CASE:** That the player is able to continue to move if there is appropriate door in that direction
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type “N” and press enter
- **POSTCONDITIONS:**
 - Program continues to run
 - Output should be:
 - You see a Funny room.
 - It has a Sad record player.
 - A Beige door leads North.
 - A Massive door leads South.

TC-14

- **IDENTIFIER:** MOVE-INVALID
- **TEST CASE:** That for the player is told that the moved in direction is invalid if there is no door in that direction
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell

- **EXECUTION STEPS**
 - Type "S" and press enter
- **POSTCONDITIONS:**
 - Program continues to run
 - Output should contain: "A door in that direction does not exist."

TC-15

- **IDENTIFIER:** LOSE
- **TEST CASE:** That if the inventory does not all have coffee, sugar and cream the game terminates with the player losing the game
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type "D" and press enter
- **POSTCONDITIONS:**
 - Program terminates with error code 1
 - Output should contain the phrase "You lose!"

TC-16

- **IDENTIFIER:** WIN
- **TEST CASE:** That if the inventory does have coffee, sugar and cream the game terminates with the player winning
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - Type "L" and press enter
 - Type "N" and press enter
 - Type "N" and press enter
 - Type "L" and press enter
 - Type "N" and press enter
 - Type "N" and press enter
 - Type "N" and press enter
 - Type "N" and press enter
 - Type "L" and press enter
 - Type "D" and press enter
- **POSTCONDITIONS:**
 - Program terminates with error code 0
 - Output should contain the phrase "You win!"

TC-17

- **IDENTIFIER:** INVENTORY-OUTPUT
- **TEST CASE:** That if the inventory does have coffee, sugar and cream the game terminates with the player winning
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "L" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "L" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "L" and press enter
 - o Type "I" and press enter
- **POSTCONDITIONS:**
 - o Program continues to run
 - o Output should contain the following:
 - You have a cup of delicious coffee.
 - You have some fresh cream.
 - You have some tasty sugar.

TC-18

- **IDENTIFIER:** LOOK
- **TEST CASE:** That looking in a room that contains items should add that the inventory of the player
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "L" and press enter
 - o Type "I" and press enter
- **POSTCONDITIONS:**
 - o Program continues to run
 - o Output should contain the following:
YOU HAVE NO COFFEE!
 - o You have some fresh cream.
 - o YOU HAVE NO SUGAR!

TC-19

- **IDENTIFIER:** UNIQ-ROOM-ADJ
- **TEST CASE:** That the description of each room when entered should contain a unique adjective
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
- **POSTCONDITIONS:**
 - o Program continues to run
 - o For output after each execution step, in all lines places that follow the form "You see <adjectives> room." There should be no duplicated words in the adjective blank

TC-20

- **IDENTIFIER:** UNIQ-ROOM-FURNISHING
- **TEST CASE:** That the description of each room when entered should contain a unique adjective
- **PRECONDITIONS:**
 - o Java 1.8 and above is installed
 - o Current directory has coffeemaker.jar
 - o java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS**
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
 - o Type "N" and press enter
- **POSTCONDITIONS:**
 - o Program continues to run
 - o For output after each execution step, there should be only a single line that follows the form "It has a <furnishing item>"

TC-21

- **IDENTIFIER:** UNKNOWN-COMMAND
- **TEST CASE:**
 - For inputs beyond the set of letters {N, n, H, h, S, s, D, d, L, l, I, i}, the program should output the phrase “What?”
- **PRECONDITIONS:**
 - Java 1.8 and above is installed
 - Current directory has coffeemaker.jar
 - java -jar coffeemaker.jar has been executed in a command line shell
- **EXECUTION STEPS:**
 - Type “J”
- **POSTCONDITIONS:**
 - Program continues to run
 - Program should output “What?”

4. DEFECTS

DEFECT-1

- **IDENTIFIER:** DEFECT-LOWER-N
- **SUMMARY:**
 - For input “n” does not move the player north, the player instead gets “what?” which should not be the case as it is one of enumerated commands
- **DESCRIPTION:**
 - Violates ITERATION-NORTH-LOWER (2)
 - When user types in “n” instead of N the displayed output is incorrect. Instead of moving into the next room, it displays What?
 - Regardless of which room you’re in , i.e. navigating by typing N, it still displays “what?”
 - Output:

```
xinkechen@Xinkes-MacBook-Pro deliverable_1 % java -jar coffemaker.jar
Coffee Maker Quest 1.0

You see a Small room.
It has a Quaint sofa.
A Magenta door leads North.

INSTRUCTIONS (N,S,L,I,D) >
n
What?

You see a Small room.
It has a Quaint sofa.
A Magenta door leads North.

INSTRUCTIONS (N,S,L,I,D) >
N

You see a Funny room.
It has a Sad record player.
A Beige door leads North.
A Massive door leads South.
```
- **REPRODUCTION STEPS:**
 - Execute in the terminal `java -jar coffemaker.jar`
 - Type in “n”
 - EXPECTED BEHAVIOR: Player is moved into the next room with output describing the newly entered room and doors
 - OBSERVED BEHAVIOR: Player instead sees an erroneous “What?”

DEFECT-2

- **IDENTIFIER:** DEFECT-HELP
- **SUMMARY:**
 - For input “H” or “h” does not display how to use the program but rather just displays “what?”

- **DESCRIPTION**

- Violates ITERATION-HELP-LOWER (6), Violates ITERATION-HELP-UPPER (5)
- When user types in “h” or “H” the displayed output is incorrect. Instead of displaying how to use the program it just displays “what?” with no explanation on how to play the game.
- Output:

```
xinkechen@Xinkes-MacBook-Pro deliverable_1 % java -jar coffemaker.jar
Coffee Maker Quest 1.0

You see a Small room.
It has a Quaint sofa.
A Magenta door leads North.

INSTRUCTIONS (N,S,L,I,D) >
H
What?

You see a Small room.
It has a Quaint sofa.
A Magenta door leads North.

INSTRUCTIONS (N,S,L,I,D) >
h
What?

You see a Small room.
It has a Quaint sofa.
A Magenta door leads North.
```

- **REPRODUCTION STEPS**
 - Execute in the terminal `java -jar coffemaker.jar`
 - Type in “H” and press enter
 - Type in “h” and press enter
- **EXPECTED BEHAVIOR:**
 - Player is shown all the commands as enumerated in the requirements with their effects
- **OBSERVED BEHAVIOR:**
 - Player instead sees an erroneous “What?”

DEFECT-3

- **IDENTIFIER:** DEFECT-INVALID MOVE

- **SUMMARY:**

- In cases where there is an invalid move, i.e. no door in desired direction, it does not display the correct output

- **DESCRIPTION:**

- Violates MOVE-INVALID (14)
- When the user tries to move in a direction, either north or south, instead of displaying, "a door in that direction does not exist", is instead outputs "You are in a magical land! But you are returned to the beginning!"
- Reproduction outlines just going south but the same is true for the north command as well.
- Additionally, the regardless of lower or uppercase input the same behavior is observed with the exception of "n" which is a separate defect (1)
- Output:

```
xinkechen@Xinkes-MacBook-Pro deliverable_1 % java -jar coffeemaker.jar  
Coffee Maker Quest 1.0
```

```
You see a Small room.  
It has a Quaint sofa.  
A Magenta door leads North.
```

```
INSTRUCTIONS (N,S,L,I,D) >
```

```
S
```

```
You are in a magical land! But you are returned to the beginning!
```

```
INSTRUCTIONS (N,S,L,I,D) >
```

```
s
```

```
You are in a magical land! But you are returned to the beginning!
```

```
INSTRUCTIONS (N,S,L,I,D) >
```

- **REPRODUCTION STEPS:**

- Execute in the terminal `java -jar coffemake.jar`
- Type in "S" and press enter
- Type in "s" and press enter

- **EXPECTED BEHAVIOR:**

- Game outputs "a door does not exist in that direction"

- **OBSERVED BEHAVIOR:**

- Player instead sees "You are in a magical land! But you are returned to the beginning!"