

**CS 1632 Software Quality Assurance**

**Deliverable 1**

Member 1 Name: Maxwell Benko

Member 2 Name: Zohaib Wasim

1. **Introduction**

Write your introduction here.

1. **Traceability Matrix**

**FUN-ITERATION** - SIX\_COMMANDS\_TEST,

**FUN-UNKNOWN** - COMMAND – UNKNOWN\_COMMAND\_MESSAGE,

**FUN-INPUT- CAPS** - IGNORE\_CASE\_TEST,

**FUN-MOVE** - DOOR\_OR\_NO\_DOOR,

**FUN-WIN** - GOOD\_ENDING\_ACHIEVED,

**FUN-LOSE** - BAD\_ENDING\_ACHIEVED,

**FUN-INVENTORY** - INVENTORY\_CONTENTS\_MESSAGE,

**FUN-LOOK** - PLAYER\_LOOK\_AND\_COLLECT,

**FUN-HELP** - PLAYER\_NEEDS\_HELP,

**FUN-UNIQ-ROOM** - ROOM\_DESCRIPTION\_TEST,

**FUN-UNIQ-ROOM-FURNISHING** - LONE\_FURNISHING\_TEST

1. **Test Cases**

**FUN-ITERATION**:

Test Case: Testing to see if at each iteration of the game, the user is 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.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"

Execution Steps:

1. Type “N” then press enter.
2. Type “S” then press enter.
3. Type “L” then press enter.
4. Type “I” then press enter.
5. Type “H” then press enter.
6. Type “D” then press enter.

Postconditions: After each press of enter the player is greeted with a message and a new iteration of the game begins unless they win or lose in which the system exits.

**FUN-UNKNOWN**:

Test Case: Testing to see if a player enters a command not specified by FUN-ITERATION, and the system responds with the phrase "What?".

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"

Execution Steps:

1. Type “Z” then press enter.

Postconditions: The “What?” message is printed to the console and a new iteration of the game starts.

**FUN-INPUT- CAPS**:

Test Case: Testing to see if the system is case-insensitive in regard to input values; that is, "n", "s", "l", "i", "h", "d" is equivalent to their uppercase counterparts "N", "S", "L", "I", "H", "D" respectively.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"

Execution Steps:

1. Type “N” then press enter.
2. Type “S” then press enter.
3. Type “L” then press enter.
4. Type “I” then press enter.
5. Type “H” then press enter.
6. Type “D” then press enter.
7. Launch program by: "java -jar coffeemaker.jar"
8. Type “n” then press enter.
9. Type “s” then press enter.
10. Type “l” then press enter.
11. Type “i” then press enter.
12. Type “h” then press enter.
13. Type “d” then press enter.

Postconditions: Each command is recognized by the game and the “What?” message never prints.

**FUN-MOVE**:

Test Case: Testing to see if the system allows a player to move North only if a door exists going North, and South only if a door exists going South. If a door does not exist, the move shall be disallowed with the message: "A door in that direction does not exist."

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"
2. Type “N” then press Enter. Complete this step a total of five times in a row.

Execution Steps:

1.Type “N” then press Enter.

Postconditions:

The user is greeted with the Message "A door in that direction does not exist." and a new iteration of the game will begin.

**FUN-WIN**:

Test Case: Testing to see if the player wins the game if and only if Coffee, Sugar, and Cream have been collected by the player and then drunk.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

Execution Steps:

Postconditions:

**FUN-LOSE**:

Test Case: Testing to see if the player loses the game if and only if the player drinks but has not collected all of the items (Coffee, Sugar, and Cream).

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

Execution Steps:

Postconditions:

**FUN-INVENTORY**:

Test Case: Testing to see if upon entering "I" for inventory, the player is informed of the items that he/she has collected (consisting of Coffee, Sugar, and Cream).

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"

Execution Steps:

1. Type “I” then press enter.

Postconditions: A message prints to the console consisting of the player’s inventory.

**FUN-LOOK**:

Test Case: Testing to see if upon entering "L" for Look, the player collects any items in the room and those items are added to the player's inventory.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

Execution Steps:

Postconditions:

**FUN-HELP**:

Test Case: Testing to see if upon entering "H" for Help, the player is shown a listing of possible commands and what their effects are.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

Execution Steps:

Postconditions:

**FUN-UNIQ-ROOM**:

Test Case: Testing to see if each room in the house a unique adjective describing it.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"

Execution Steps:

Postconditions:

**FUN-UNIQ-ROOM-FURNISHING**:

Test Case: Testing to see if each room in the house has one and only one unique furnishing visible to the user upon entering the room.

Preconditions: Java 8 is installed and is in the OS Path environment variable. The game has started. Steps:

1. Launch program by: "java -jar coffeemaker.jar"

Execution Steps:

1. Type “N” then press enter. Complete this step a total of five times in a row.

Postconditions: There are 6 unique pieces of furniture shown in the history of the console’s text.

1. **Defects** Write your defects here.

IDENTIFIER:

SUMMARY:

DESCRIPTION:

REPRODUCTION STEPS:

EXPECTED BEHAVIOR:

OBSERVED BEHAVIOR:

IDENTIFIER:

SUMMARY:

DESCRIPTION:

REPRODUCTION STEPS:

EXPECTED BEHAVIOR:

OBSERVED BEHAVIOR:

IDENTIFIER:

SUMMARY:

DESCRIPTION:

REPRODUCTION STEPS:

EXPECTED BEHAVIOR:

OBSERVED BEHAVIOR: