Coffee Maker

Ryan Rahuba;

CS 1632 – DELIVERABLE 1: Test Plan and Traceability Matrix

Identifier: 1

Test Case: To ensure that all programmed commands are excepted

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH; the game has to be started

Input Values: [N, S, L, I, H, D]

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) enter input values and press ”Enter”

Output Values:

“N” – properly move to next room

“S” -- properly move to next room

“L” – looks around room

“I” – adds ingredients to inventory

“H” – “what?”

“D” – drinks ingredients

Postconditions:

Identifier: 2

Test Case: To ensure that unexpected input is handled properly

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH

Input Values: any alpha-numeric, or character, input not specified in FUN-ITERATION

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) enter input values and press ”Enter”

Output Values: “What?”

Postconditions:

Identifier: 3

Test Case: To ensure that input can be case insensitive

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH; the game has to be started

Input Values: [n, s, l, i, h, d]

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) enter input values and press ”Enter”

Output Values:

“n” – “what?”

“s” -- properly move to next room

“l” – looks around room

“i” – adds ingredients to inventory

“h” – “what?”

“d” –drinks ingredients

Postconditions:

Identifier: 4

Test Case: To ensure that the player can move North only if an appropriate door exists

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH; the game has to be started; player must be at Northern most room

Input Values: [N, n]

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) enter input values and press ”Enter”

5) repeat until you reach the northern room

Output Values:

N -- “You are in a magical land! But you are returned to the beginning!”

n – “what?”

Postconditions:

Identifier: 5

Test Case: To ensure that the player can move South only if an appropriate door exists

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH; the game has to be started; player must be at the beginning (southern most room)

Input Values: [S, s]

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) enter input values and press ”Enter”

Output Values: “You are in a magical land! But you are returned to the beginning!”

Postconditions:

Identifier: 6

Test Case: To ensure that the player wins only if he/she has collected coffee, cream, and sugar

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH; coffee, cream, and sugar are in the inventory

Input Values: [D, d]

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) play the game until coffee, sugar and cream are retrieved

5) enter input values and press ”Enter”

Output Values: Player wins and victory dialogue is displayed

Postconditions:

Identifier: 7

Test Case: To ensure that the player loses if he/she drinks without all three ingredients

Preconditions: Any machine with an OS able to run a command shell; Java SDK installed and the appropriate folder added to the PATH; the game has to be started

Input Values: [D, d]

Execution Steps:

1) open the command prompt

2) navigate to the appropriate folder where coffeemaker.jar is stored

3) execute java –jar coffeemaker.jar

4) enter input values and press ”Enter”

Output Values: Player Loses and Losing dialogue is displayed

Postconditions:

Traceability Matrix:

FUN-ITERATION: 1

FUN-UNKNOWN-COMMAND: 2

FUN-INPUT-CAPS: 3

FUN-MOVE: 4, 5

FUN-WIN: 6

FUN-LOSE: 7

Defects:

Description: NORTH-CASE-SENSITIVE

Summary: The command to go North is not case insensitive

Reproduction Steps:

1. Execute java –jar coffeemaker.jar
2. Type “n”
3. Press “Enter”

Expected Behavior: Game transitions to next room

Observed Behavior: “What?” is given as output and player remains in room

Description: HELP-FUNCTION-NOT-VALID

Summary: Help menu does not display when prompted

Reproduction Steps:

1. Execute java –jar coffeemaker.jar
2. Type “H” or “h”
3. Press “Enter”

Expected Behavior: Help menu is displayed

Observed Behavior: “What?” is displayed as output and current room is described again