



# **CS 1632 Software Quality Assurance**

## **Deliverable 2**

Member 1 Name: Yuliang Xiao (yux37)

Member 2 Name: Yunjie Zhang (yuz173)

Team Name: Yuliang-and-Yunjie

## **1. Introduction**

In our deliverable 2, we divided work into two parts. Yuliang works on writing the JUnit test class for CoffeeMakerQuest and Yunjie works on implementing the CoffeeMakerQuest class. Our difficulties while using JUnit are setting things up. First, we need to implement the actual CoffeeMakerQuest class. Our difficulties in implementing the CoffeeMakerQuest class are creating member variables for the class. We need to figure out a way to store all the rooms because the actual game requires going between all the rooms. Once variables are figured out, implementing methods are straightforward. There are six rooms, a CoffeeMakerQuest, and a player. We need to set up correctly so that we focus on only testing the CoffeeMakerQuest. Mockito plays an important role in our JUnit Testing because Player and Room are different classes. By using Mockito, we can now do unit testing that focuses on CoffeeMakerQuest.

## 2. Code Coverage

At below, paste a screenshot of code coverage stats **after** having completed the coding.

Problems @ Javadoc Declaration Console Coverage				
TestRunner (Jun 21, 2021 9:09:02 PM)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
☐ Coffee Maker JUnit	71.9 %	1,219	477	1,696
☐ src	71.9 %	1,219	477	1,696
☐ (default package)	71.9 %	1,219	477	1,696
> ☐ Player.java	0.0 %	0	144	144
> ☐ Room.java	0.0 %	0	130	130
> ☐ Game.java	0.0 %	0	124	124
☐ CoffeeMakerQuestImpl.java	91.4 %	352	33	385
☐ CoffeeMakerQuestImpl	91.4 %	352	33	385
● processCommand(String)	89.5 %	154	18	172
■ addFormat(Player)	90.0 %	81	9	90
● addRoomAtNorth(Room, String,...)	93.5 %	58	4	62
● setCurrentRoom(Room)	89.5 %	17	2	19
▲ CoffeeMakerQuestImpl()	100.0 %	17	0	17
● addFirstRoom(Room)	100.0 %	13	0	13
● getCurrentRoom()	100.0 %	3	0	3
● getInstructionsString()	100.0 %	2	0	2
● isGameOver()	100.0 %	3	0	3
● setPlayer(Player)	100.0 %	4	0	4
> ☐ TestRunner.java	60.5 %	49	32	81
> ☐ Config.java	60.0 %	15	10	25
> ☐ CoffeeMakerQuest.java	60.0 %	6	4	10
> ☐ CoffeeMakerQuestTest.java	100.0 %	753	0	753
> ☐ Item.java	100.0 %	44	0	44