The University of Adelaide
School of Computer Science

Software Engineering & Project

Testing Report

Lunar Rover Mapping Robot

Version 1.0

Prepared by Postgraduate Group 28

Tony Gold a1707687

Ren Koh a11931141

Amit Patel a1697700

Hangyue Xu a1692118

Guo Yang a1696877

Irving Yim a1617656

Tianming Zhang a1710509

October 30, 2017

Contents

Revision History

Name	**Date**	**Description**	**Version**
Amit Patel	9/10/2016	Designed the template,Add section 1,2,3	0.1
Hangyue Xu	15/10/2016	Update the content	0.2
Tianming Zhang	18/10/2016	Update the Test Case	0.3
Hangyue Xu	19/10/2016	Update the Test Case,modify Blackbox Testing	0.4
Amit Patel	20/10/2016	Add sections 5 to 8	0.5
Guo Yang	21/10/2016	Update the Test Case and modify section 5	0.6
Irving Yim	22/10/2016	Update section 7	0.7
Ren Koh	24/10/2016	modify section 5 and 6	0.8
Tony Gold	25/10/2016	modify section 3 and find errors in the documents	0.9
Amit Patel	30/10/2016	Release to Client	1.0

Table 1: Revision History

1 Introduction

The testing report outlines the testing methods and outcome of the tests for Lunar Mapping Robot System, designed and built by group PG28.

1.1 Purpose

This document details the testing criteria,test cases and results of tests on the system developed by development team. and the specific details of the implementation of the requirements as set out in the Software Requirements Specification (SRS). The primary intended audience for this documents are the developers of the complete system, who will use this document as a reference on what features are working properly and tested. This document will also be important for the team to develop new features based on current system and find out critical features within the system.

1.2 Scope

The purpose of this project is to analyze the results of different tests which have been done for the project, in order to determine whether the project meets its goals in terms of user requirements or not.

1.3 References

1. Project Description-2017(Sem2)
2. Software Requirement Specification - PG28
3. Software Project Management Plan - PG28
4. Software Design Document - PG28
5. Software Testing Plan - PG28

1.4 Definitions, Acronyms and Abbreviations

API	Application Programming Interface
TA	Test Aborded
TP	Test Pass
TI	Test Incomplete
GUI	Graphical User Interface
I/O	Input and Output
JVM	Virtual Machine
NGZ	No-Go Zone
PC	Personal Computer
robot	The assembled Lego robot, EV3 brick
SDD	Software Design Document
SEP	Software Engineering Project
SPMP	Software Project Management Plan
SRS	Software Requirements Specification
XML	eXtensible Mark-up Language

1.5 Constraints

As of GUI can not be tested fully, some functions were need to be tested manually. Some of the testing are unable to be done by the developers due to the limitations of testing requirements.

2 Test Criteria

The decision is defined relying on the following principles until the test execution completed:
• Test sheet is marked depending on "TP" state when all steps are passed(Test Pass or TP). This indicated that requirements are achieved as described.
• Test sheet is marked relying on "TA" state while all steps have failed(Test Aborted) state or when the result of a step differs from the expected result. Result should be recorded why it makes failure and raise the issue to the issue list.
• Test sheet is marked according to "TPP" state when partial functions are passed and some are failure.

2.1 Testing Method

Manual Testing
Manual testing is a type of testing where robot is been tested on particular scenarios while developing the features.

Black Box Testing
Black box testing is a type of software testing methods that evaluates the unit, integration, system and acceptance of a software.

System Testing
System testing is a testing method that could test software and hardware. This approach can be conducted on evaluation of a system's compliance and its expectations.

Regression Testing
Regression testing is a way to test software that specifies the previous developed software and evaluate its correct performance. The purpose of this method is to ensure there is no new fault occur.

3 System Testing

Testing outcomes are according to defined standard in section 2 (Testing Criteria)

3.1 Basic Movement

Test Case ID: TI01
Test Purpose: Testing the basic movements of the robot
Preconditions: The remote control system is connected with the robot
Expected result: Robot will move according to different buttons pressed respect to their directions and trying automated commands to move robot
Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing and Starting robot on trial map with automated commands for moving around	The robot should able to move in different directions	TP
2	Manually control the robot respectively to do: Forward, Backward, Left, Right	The robot should move in different directions according to button pressed	TP

3.2 Colour Sensor

Test Case ID: TI02
Test Purpose: Testing the color sensor's functionality
Preconditions: manual control system is activated before testing
Expected result: the colours detected by sensor should be presented correctly on the GUI
Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing and Starting robot on trial map by manual control as well with automated commands to move around for searching various colors	The robot should able to move in different directions	TP
2	Testing the sensor with different colors while robot is moving	Dected colour should be reflected on the GUI	TP

3.3 Detect Boundary

Test Case ID: TI03
Test Purpose: Testing the robot for boundary detection based on basic movements and colours identified
Preconditions: manual control system is activated before testing
Expected result: Identifying the specific colored boundary and take 90 degrees turn for identifying whole boundary

Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing and Starting robot on trial map by manual control as well with automated commands to move around for searching for boundary	The robot should able to move in different directions and identify the colors	TP
2	Identifying the boundary without crossing it	Robot sould take turn 90 degree and identify boundary again based on color	TP

3.4 Foot-print detection

Test Case ID: TI04
Test Purpose: Finding the foot print sequence based on color
Preconditions: manual control system is activated before testing
Expected result: Plotting the foot print on the GUI
Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing and Starting robot on trial map by manual control as well with automated commands to move around for searching for boundary	The robot should able to move in different directions and identify the colors sequences	TP
2	Identifying foot-print sequence with the algorithm after finding any separate color part than it starts searching other sequence by rotating it 20 degree ahead left-right vice-verse	showing the path on the GUI	TPP

3.5 Ultrasonic Sensor

Test Case ID: TI05
Test Purpose: Testing whether the ultrasonic sensor can successfully detect obstacles and different colors on the map
Preconditions: Ultrasonic sensor can work well
Expected result: Detect the obstacle and different colors on the map
Testing Outcomes: The ultrasonic sensor can work well under both auto and manual mode

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing an obstacle on the map and starting the robot on both manual control mode, control the robot drive to the obstacle, find out whether the robot would stop before touching the obstacle. In automatic mode, start the robot and find out whether the robot would stop before touching the obstacle	The robot can stop before touching the obstacle under both manual and automatic mode	TP
2	Put different colors on the map and then start the robot to find out whether the ultrasonic sensor can distinguish between different colors	In both manual and automatic mode, the robot can distinguish the different colors on the map	TP

3.6 Mapping under the auto mode

Test Case ID: TI06
Test Purpose: Testing whether the robot can drive successfully under the automatic mode on the map
Preconditions: All components of the robot can work well
Expected result: The robot can successfully drive on the map and describe the map automatically
Testing Outcomes: The robot and other components can work successfully under auto mode

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Starting the robot and switch it on auto mode, find out whether the robot can drive within the boundary, change direction, detect the obstacle, stop before touching the obstacle, distinguish different colors on the map and provide the map details automatically	The robot can mapping successfully under the auto mode	TP
2	While driving, find out whether it can stop after roving around the map, keep all components work and drive inside the boundary	The robot can stop after roving around the map, keep all components work and drive inside the boundary under auto mode	TP

3.7 Mapping under the manual mode

Test Case ID: TI07
Test Purpose: Testing whether the robot can drive successfully under the manual mode on the map
Preconditions: all components of the robot can work well
Expected result: the robot can successfully drive on the map and detect obstacles and colors manually

Testing Outcomes: The robot and other components can work successfully under manual mode

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Starting the robot and switch it on manual mode, find out whether the robot can be controlled to drive within the boundary, change direction, detect the obstacle, stop before touching the obstacle and distinguish different colors on the map manually	The robot can mapping successfully under manual mode	TP
2	While driving, find out whether it can be controlled to stop after roving around the map, keep all components work and drive inside the boundary	The robot can be controlled to stop after roving around the map, keep all components work and drive inside the boundary under manual mode	TP

3.8 Setting NGZ

Test Case ID: TI08
Test Purpose: Testing whether the robot can successfully detect the black line of the NGZ on the map
Preconditions: The robot is connected with the remote control system
Expected result: The robot will stop when it detected the NGZ and then change the current direction
Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing the robot on trial map by auto control and then let it automatically move around	The robot should able to stop in the front of the black line	TP
2	Placing the robot before the NGZ	The robot will recognize the NGZ when it found the black line	TP
3	Placing the robot before the NGZ	The robot will change the direction when it found the NGZ	TP

3.9 Loading the Map

Test Case ID: TI09
Test Purpose: Testing whether the map can be loaded both on the GUI and robot and then robot can automatically go anywhere
Preconditions: The test map is known and the robot can analyze the information of the map
Expected result: The robot can use the map to analyze a way and then go to the target location automatically
Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Input the test map on the GUI and robot	The robot should able to load the information of the map successfully	TP
2	Placing the robot on the test map	After the robot load the information of the map, it can calculate the way to go the target location automatically without detect the real map	TP

3.10 Saving a Map

Test Case ID: TI10
Test Purpose: Testing whether the robot can successfully saving the map
Preconditions: All components of the robot can work well
Expected result: Robot can successfully understand the obstacles from a new map and make the map on GUI
Testing Outcomes:All components of robot are operated successfully, and it can understand the map and save it on GUI

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Starting the robot and switch it on manual mode, give robot a new map with the obstacles, handle the robot drive on the map, then to find out whether the robot can detect the obstacles	Robot can find out the obstacles and other barriers from the new map	TP
2	After robot find out the barriers from the map, it will save both the information and its tracks through the GUI	Robot can successfully save the information of the tracks and obstacles and map it on GUI	TP

3.11 Finding the crater area

Test Case ID: TI11
Test Purpose: Testing whether the robot can successfully detect the crater area on the map
Preconditions: The robot is connected with the remote control system
Expected result: The robot will stop when it detected the crater border and then goes backward, takes 45 degree right turn goes again back takes another 45 degree right turn, goes three time straight than it came back and takes 90 degree left turn again checks crater border and repeats whole step till it comes to starting point where it detected the crater
Testing Outcomes:

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing the robot on trial map by auto control and then let it automatically move around	The robot should able to stop in the front of the green border	TP
2	Placing the robot before the crater	The robot will recognize the craters starting position when it found the green color	TP
3	Placing the robot at random position and on automatic mode letting robot detect the crater area	The robot will stop when it detected the crater border and then goes backward, takes 45 degree right turn goes again back takes another 45 degree right turn, goes three time straight than it came back and takes 90 degree left turn again checks crater border and repeats whole step till it comes to starting point where it detected the crater	TA

4 Black Box Testing

TEST ID	ACTIONS	EXPECTED RESULTS	RESULT
BT01	Staring Lunar Mapping Robot System	The main class should be executed	TP
BT02		GUI should appear	TP
BT03	Press the connect button	Able to connect the robot	TP
BT04		Able to disconnect the robot	TP
BT05	Press the direction button on GUI	The robot can go forward	TP
BT06		The robot can go back	TP
BT07		The robot can turn left	TP
BT08		The robot can turn right	TP
BT09	Press the switch button to auto mode	The robot can search the map automatically	TP
BT010		Able to change to the manual mode	TP
BT11	Press the switch button to manual mode	The robot will stop and waiting for commands	TP
BT12		Able to change to the auto mode	TP
BT13	Press the stop button on GUI	The robot can stop immediately	TP
BT14	Placing the robot on map with different colours	The robot can distinguish different colors	TP
BT15	Placing the robot in front of the NGZ	The robot will not go forward before the NGZ	TP
BT16		The robot will draw the NGZ on the map	TP
BT17		The robot will go back a bit and then turn around	TP
BT18	Placing the robot in front of the obstacle	The robot will not go forward before the obstacle	TP
BT19		The robot will draw the obstacle on the map	TP
BT20		The robot will go back a bit and then turn around	TP
BT21	Drawing the landform on the map part of GUI	Able to show the robot's current location	TP
BT22		Able to draw the NGZ	TP
BT23		Able to draw the obstacle	TP
BT24		Able to view the boundary	TP
BT25		Able to show the target object	TP
BT26	Press the load button	The robot can load the map information from the local directory	TP
BT27		The robot can move automatically	TP
BT28	Robot status	Able to show the status that color detected	TP
BT29		Able to show the status of current location	TP

Table 2

5 Manual Testing

Manual testing is done through developing the some scenarios related to final presentation of the Lunar Robot System. As below map was used for testing our developed Lunar Robot system where different small scenario's were taken to test the robots outcome

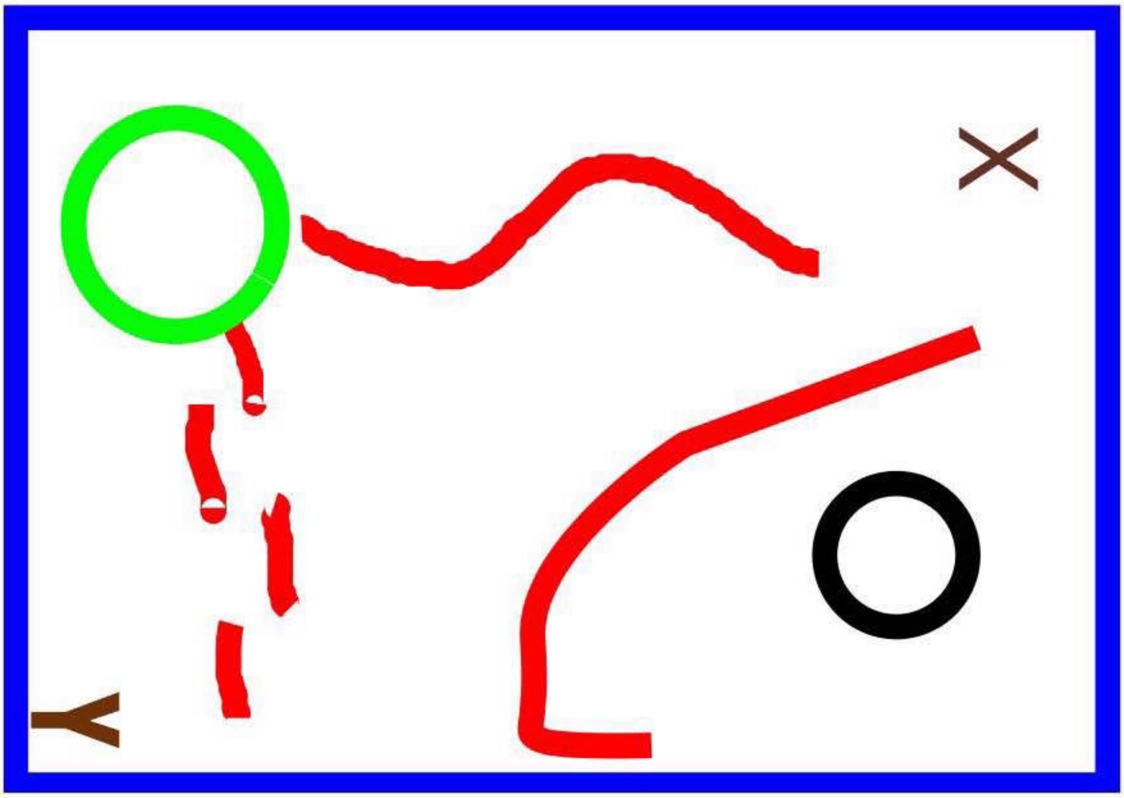

Figure 1: Testing Scenario

No:	Scenario with small activities	Expected Test results	Actual Outcomes
SC1	Robot is placed on the random position on the map	The robot should able to manually control by operator's commands	TP
SC2	It is drawn till X point and added as starting point in gui	the robot should start to map itself by X,Y coordinates and starts based on XML map	TP
SC3	It rotates 360 degree to find out the obstacles on the map	It should add the positions of the obstacles in the form of X and Y coordinates on map	TP
SC3	It starts moving straight and checks the distances between obstacles and different colors detected by robot	If it finds the red line , it should follow that red line and draw it on the map	TP

No:	Testing Actions	Expected Test results	Actual Outcomes
SC4	It starts moving straight and checks the distances between obstacles and different colors detected by robot	If it finds the red line in broken pieces , it should follow that red line and draw the footprint path on the GUI map	TP
SC5	It starts moving straight and checks the distances between obstacles and different colors detected by robot	If it finds the green color, robot should identify the green color as crater border and stop their. After that, it should go backward, takes 45 degree right turn goes again back takes another 45 degree right turn, goes three time straight than it came back and takes 90 degree left turn again checks crater border and repeats whole step till it comes to starting point where it detected the crater.	TPP
SC6	It starts moving straight and checks the distances between obstacles and different colors detected by robot	If it noted the crater or paths position in the map or it finds the border which is blue line it should take 60 degree left turn each time and go straight.	TP
SC7	It starts moving straight and checks the distances between obstacles and different colors detected by robot	If robot finds the black color border then it should stop and identify it as starting point of the NGZ on Map. After that it should go backward, takes 45 degree right turn goes again back takes another 45 degree right turn, goes three time straight than it came back and takes 90 degree left turn again checks crater border and repeats whole step till it comes to starting point where it detected the NGZ or If it is straight line than it will simply takes opposite turn	TPP
SC8	It follows the straight path after the last detection depending on the actions defined	If robot finds obstacle then it should identify as obstacle and should avoid to go near near obstacle	TP
SC9	Placing the robot randomly on the map after finding all the obstacles,paths,NGZ and crater area	It should notify X and Y coordinates on the map and find current position on the map after taking 360 turn	TA

6 Regression Testing

No:	Testing Actions	Expected Test results	Actual Outcomes
1	Placing and Starting robot on trial map by manual control as well with automated commands to move around	The robot starts to move around	TP
2	the colours detected by sensor should be presented correctly on the GUI	The robot detects the color and shows the same colour on GUI	TP
3	Placing and Starting robot on trial map by manual control as well with automated commands to move around	The robot starts to move around	TP
4	Identifying the boundary without crossing it	The robot detects the boundary and turns 90 degrees and goes forward for 5cm and randomly turns left	TP
5	Placing and Starting robot on trial map by manual control as well with automated commands to move around	The robot starts to move around	TP
6	Robot will move according to different buttons pressed respect to their directions and trying automated commands to move robot	The robot should move in different directions according to button pressed	TP
7	sketch a No-Go-Zone in the current map	The No-Go-Zone can be sketched on the GUI	TP
8	sketch a different paths in the current map	The paths and shapes(circle/triangle) can be sketched on the GUI	TP
9	In sketch mode try to put different shapes as No-Go-Zone	The map will identify as the NGZ	TP
10	Using the motors and color sensors by manually, try to identify the NGZ and robot takes turn	The robot can identify the NGZ and bypass it	TA
11	sketch a No-Go-Zone in the current map	The No-Go-Zone can be sketched on the GUI	TP
12	In sketch mode try to put different shapes as No-Go-Zone	The map will identify as the NGZ	TP
13	Using the motors and color sensors by manually, try to identify the NGZ and robot takes turn	The robot can identify the NGZ and more than 2/3 robot bypass it	TA

No:	Testing Actions	Expected Test results	Actual Outcomes
14	sketch a No-Go-Zone in the current map	The No-Go-Zone can be sketched on the GUI	TP
15	Switch the sketch mode by drawing the NGZ lines to make a shape	The NGZ can switched between Irregular polygon and circle shape	TP
16	Using the motors and color sensors by manually, try to identify the NGZ and robot takes turn	The robot can identify the NGZ and more than 2/3 robot bypass it	TPP
17	sketch a No-Go-Zone in the current map	The No-Go-Zone can be sketched on the GUI	TP
18	Switch the sketch mode by drawing the NGZ lines to make a shape	The NGZ can switched between Irregular polygon and circle shape	TP
19	Using the motors and color sensors by manually, try to identify the NGZ and robot takes turn	The robot can identify the NGZ and more than 2/3 robot bypass it	TPP
20	sketch a No-Go-Zone in the current map	The No-Go-Zone can be sketched on the GUI	TP
21	Switch the sketch mode by drawing the NGZ lines to make a shape	The NGZ can switched between Irregular polygon and circle shape	TP
22	Using the motors and color sensors by manually, try to identify the NGZ and robot takes turn	The robot can identify the NGZ and more than 1/3 robot bypass it	TPP
23	sketch a No-Go-Zone in the current map	The No-Go-Zone can be sketched on the GUI	TP
24	Switch the sketch mode by drawing the NGZ lines to make a shape	The NGZ can switched between Irregular polygon and circle shape	TP
25	Using the motors and color sensors by manually, try to identify the NGZ and robot takes turn	The robot can identify the NGZ and more than 1/3 robot bypass it	TPP
26	Open (Click the open file button) and upload the map in the auto system by uploading map	The map is loaded into the program	TP

No:	Testing Actions	Expected Test results	Actual Outcomes
27	Set the robot to automatic mode by ticking the "Auto" check-box	The robot is set to automatic mode and start running	TP
28	Robot starts to find out the position on map and scan for obstacles by rotating 360 degree and find the paths by color sensors	Robot can find the obstacles and search the path without any manual control commands	TP
29	Putting robot on the random place on the map	Robot starts mapping the whole map	TA
30	Open (Click the open file button) and upload the map in the auto system by uploading map	The map is loaded into the program	TP
31	Set the robot to automatic mode by ticking the "Auto" check-box	The robot is set to automatic mode and start running	TP
32	Robot starts to find out the position on map and scan for obstacles by rotating 360 degree and find the paths by color sensors	Robot can find the obstacles and search the path without any manual control commands	TP
33	Putting robot on the random place on the map	Robot starts mapping the whole map	TA

7 Issue Lists

The issue list significantly identifies all issues that occurred during the whole test process. The whole test process includes the functional, regression and manual tests. The issue list will specifically include all tests those aborted (TA) and partially passed (TPP).

7.1 Issue List 1

Issue List			
Issue ID	**Tests with unexpected outcome**	**Severity**	**Test ID**
FT01	Lunar robot should find the NGZ and try not to cross it and should take turn but if NGZ is like U shape and robot is between two black line and one straight line then It can not escape the NGZ and bypass it in last.	Moderate	egression Test No: 10
FT02	it finds the green color, robot should identify the green color as a crater border and stop their. After that, it should go backward,takes 45 degree right turn and again it goes back, takes another 45 degree right turn, goes three time straight than it came back and takes 90 degree left turn again checks crater border and repeats whole step till it comes to starting point where it detected the crater. Here, robot is able to find half of the crater and notifying it as whole crater and need to start robot manually from half crater.	High	SC5 and 3.11/3
FT03	Placing the Lunar robot on the random place on loaded map and it should able to find out its position, where robot was trying to find the noted position with some mark.	Low	SC9 and regression test no: 33

7.2 Solutions

Identified solutions		
Issue ID	**Planned solutions to solve failed test Cases:**	**Issue ID**
PA01	It tries not to go in such situations where NGZ border is three sides around the robot. Moreover, it assumes there will be no trap of NGZ on the map.	FT01
PA02	Before the end date it will come up with better way to find out whole crater area with accurate algorithm.	FT02
PA03	It was out of scope to find out its position on the loaded map when it is placed randomly. Robot will start from starting point.	FT03

8 Conclusion

Overall, there were different testing methods were used to find out the issues before delivering final Lunar robot system. It is notified after this many testing trial with different test methods, some issues were identified. Those Issues were solved with the defined actions/solutions. At the fist regression testing, there are 9 individual test cases executed,while 3 issues located through the whole regression testing. Through the next regression testing, issues are reduced to 0. All the tested functions are running smoothly,which gives the group confidence to deliver the system with smooth functioning. There were 2 issues where system did compromise with it as described in the action list. So, it can be concluded that testing is very crucial before delivering final system.