Testing Report

for

Mesa Mapping Robot

Version 1.0 approved

Prepared by SEP Group 19 - Spark
a1657343 Luo Yawen
a1662541 Yang Jiajun
a1653772 Yun Zhang
a1671836 Wei Jingwen
a1690773 Wang Yuzhu

School of Computer Science,
The University of Adelaide

Nov 01, 2015

Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Wei Jingwen	20/Oct/15	Add section 1	0.1
Wei Jingwen	21/Oct/15	Add section 2	0.2
Wei Jingwen	22/Oct/15	Add section 3.1	0.3
Wei Jingwen	24/Oct/15	Add section 3.2	0.6
Wei Jingwen	26/Oct/15	Add section 3.3	0.8
Wei Jingwen	27/Oct/15	Review all sections	0.9
Luo Yawen	01/Nov/15	Release	1.0

1 Introduction

1.1 Document overview

This document is the test report of the mesa mapping robot testing phase of both software development and robot test in project. It contains the results of tests, which were executed during the testing phase of the robot.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

- CC - Control Centre
- CS - Colour Sensor
- GUI - Graphical User Interface
- IW - Ingkarni Wardli
- NGZ - No Go Zone
- PG19 - Postgraduate Group 19
- SRS - Software Requirement Specification
- USS - UltraSonic Sensor

1.3 References

1.3.1 Project References

#	Document Title
1	Software Requirements Specification for Mesa Mapping Robot PG19 V2.0

1.4 Conventions

In this report, you will find a number of tests for a mapping robot named Spk_EV3 that designed for the mapping mission. The testing will cover the majority function that is programmed for the robot.

2 Overview of Tests Results

2.1 Tests log

The test will cover all the executable functions that were able to meet the requirements according to the SRS document. There is a test plan starting for the basic function to the algorithm which the software engineers design to perfect the robot performance and gain its abilities.

The SparkMesaExplorer (version 4.1.1) was tested on the Spk_EV3 (A mapping robot designed by PG19 and built by LOGO Mindstorm EV3 set) platform by a MacBook Pro which using OS X 10.11.1 located in IW 462, from the 24/10/2015 to the 26/10/2015. The tests of the test phase (ref. software test plan) where executed.

Testers where: Jingwen Wei (a1671836)

2.2 Rationale for decision

After executing a test, the decision is defined according to the following rules:

- **OK:** The test sheet is set to ”OK” state when all steps are in ”OK” state. The real result is compliant to the expected result.

- **NOK:** The test sheet is set to ”NOK” state when all steps of the test are set to ”NOK” state or when the result of a step differs from the expected result.

- **Partial OK:** The test sheet is set to ”Partial OK” state when at least one step of the test is set to ”NOK” state or when the result of a step is partially compliant to the expected result.

- **NOT RUN:** Default state of a test sheet not yet executed.

- **NOT COMPLETED:** The test sheet is set to ”Not Completed” state when at least one step of the test is set ”Not Run” state.

Tests results are listed in section 3.

2.3 Overall assessment of tests

The qualitative overall assessment of tests.

- All the method that are written in the test.java shall have test of its structure and outcome of the robot actions.

- All the testing will be executing for at least 5 times for ensuring the out come is in an acceptable stability.

- All tests with interfaces passed, GUI is not optimised for screens of the test platform.

- All tests program had a theoretical explanation and have no logical mistake.

The quantitative results.

Statistics about tests:

- 66.7% of tests OK
- 28.5% of tests NOK
- 4.8% of tests POK
- 0% of tests NR
- 0% of tests NC

2.4 Impact of test environment

The impact of the test environment is sometimes unavoidable, such as the the signal from the sensor is unstable, the colour is different when dealing the same situation with different when the environmental light. The key issue of the influenced robot movement is the friction of the surface can make the coordinate system broken due to the fact that the there is no such a location feedback from any sensor. The EV3 brick in the robot should also installed LeJOS in order to executed Java program.

The Eclipse Mars Release(V4.5.0) used in this test, the laptop which runs this program should have Bluetooth function to connect with robot, the platform in use in the test is a MacBook Pro runs OS X(V10.11.1). The LeJOS which was installed initially is LeJOS_EV3_0.9.0-beta, as the Plugin demand the version of Java, the actually JRE using is eclipse in Java SE 7 [1.7.0_79].

3 User Requirements

For each executed test, this document contains:

- Test identification;
- Test title;
- Test decision;
- A comment containing additional information or problems encountered during execution and differences with the test procedure.

3.1 Basic Using Test

- Test 001

Test ID	**Test 001**	**Comment**	**Decision**
Test description	Attempt to connect robot and platform via Bluetooth	The control and command will deliver to robot through Bluetooth	OK
Initial conditions	The robot turned on and the Bluetooth of laptop		OK
Tests inputs	None		
Data collection actions	None		
Tests outputs	None		
Assumptions and constraints	EV3 brick is able to be detected and the connection	Both side ready	OK
Expected results and criteria	Robot can be connected through EV3 CC		OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Start Bluetooth preference	Bluetooth is started	OK
2	Pair the Spk_EV3 device	Spk_EV3 paired and connected	OK
3	Connect Spk_EV3 with EV3 CC	Connection established	OK

- Test 002

Test ID	**Test 002**	**Comment**	**Decision**
Test description	Testing the forward action - straightF	The forward under controlled	OK
Initial conditions	The robot is free and static		OK
Tests inputs	currentAngle - float, legalColor - float, targetX - double, targetY - double	Describing how this forward action works	OK
Data collection actions	None		
Tests outputs	The actions and if robot can perform as the input	Only the movement action and the stop condition	OK
Assumptions and constraints	The towards of the robot is +/- 10 degree to currentAngle		OK
Expected results and criteria	The forward along the input direction, stop when run out legal colour or coordinate	The X and Y cannot both applied	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute straightF(0, Color.WHITE, 20, Double.NaN)	Robot go straight ahead until the colour under CS is not white or reach 20 cm	OK
2	Execute straightF(90, Color.WHITE, Double.NaN, 20)	Same as Step 1, only change the direction manually	OK

- Test 003

Test ID	**Test 003**	**Comment**	**Decision**
Test description	Testing the forward action - straightB	The backward under controlled	OK
Initial conditions	The robot is free and static		OK
Tests inputs	currentAngle - float, targetColor - float, targetX - double, targetY - double	Describing how this backward action works	OK
Data collection actions	None		
Tests outputs	The actions and if robot can perform as the input	Only the movement action and the stop condition	OK
Assumptions and constraints	The towards of the robot is +/- 10 degree to currentAngle		OK
Expected results and criteria	The backward along the input direction, stop when run out legal colour or coordinate	The X and Y cannot both applied	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute straightB(0, Color.RED, 20, Double.NaN)	Robot go straight back until the colour under CS is not red or reach 20 cm	OK
2	Execute straightB(90, Color.RED, Double.NaN, 20)	Same as Step 1, only change the direction manually	OK

- Test 004

Test ID	**Test 004**	**Comment**	**Decision**
Test description	Testing the turning action - r_turn & l_turn	The changing direction method	OK
Initial conditions	The changing direction method		OK
Tests inputs	targetAngle - float (Absolute angle)	Target Angle is based on the initial point	OK
Data collection actions	None		
Tests outputs	The actions and if the robot can turn accurately as the command input		OK
Assumptions and constraints	The target angle is legal e.g. the right turn is large than current angle		OK
Expected results and criteria	The backward along the input direction, stop when run out legal colour or coordinate	The X and Y cannot both applied	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute r_trun(90)	Robot turn right 90 +/- 1	OK
2	Execute r_trun(90)	Robot turn left 90 +/- 1	OK

3.2 Intermediate Test

- Test 005

Test ID	**Test 005**	**Comment**	**Decision**
Test description	Test the ability to go along black boundary - alongLine (float direction)	If the result turns out it is not along, it can only indicate operator to adjust angle	NOK
Initial conditions	The robot placed besides the black boundary		OK
Tests inputs	Direction - float	Target Angle of the line	OK
Data collection actions	None		
Tests outputs	The robot can use its arm to follow the line and visually reflecting the condition	The robot can stop at the end of the black line	OK
Assumptions and constraints	The target angle is legal and the robot is accurately placed besides the line		OK
Expected results and criteria	The robot can turn its arm back to front when the direction has been confirmed stop in the end		OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute alongLine(0)	Robot turn its arm and found black line stop when lost line	NOK
2	Execute alongLine(0)	Robot turn its arm and found black line turn arm back when the direction confirmed stop at the cross boundary	OK

- Test 006

Test ID	**Test 006**	**Comment**	**Decision**
Test description	Test the ability of recognise the obstacle - obstacle()	The robot is able to identify the obstacle when right arm encounter one	OK
Initial conditions	The robot is under forward condition		OK
Tests inputs	None		
Data collection actions	None		
Tests outputs	The robot can use its right arm to identify obstacle and avoid it	As the size of the robot, the robot will backward a little	OK
Assumptions and constraints	The obstacle is in front of robot right arm and flat	It may miss the obstacle when the USS not function well	POK
Expected results and criteria	The robot is able to identify the obstacle and back turn left and go straight until scan over the line then turn back to original direction		OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute scan(0); then put an obstacle in the path	Robot crash on the obstacle due to the surface of the obstacle can not detected by robot	NOK
2	Execute scan(0); then put an obstacle in the path	Robot meet obstacle, back a little with light turn red, then turn left scan a path then turn back to right with red light off	OK

- Test 007

Test ID	**Test 007**	**Comment**	**Decision**
Test description	Test the ability of recognise the NGZ - ngzDectected()	The robot is able to identify the red line as NGZ	OK
Initial conditions	The robot is under scan condition		OK
Tests inputs	None		
Data collection actions	None		
Tests outputs	The robot can use its right arm to identify NGZ and avoid it	The robot ought to backward a little but it does not functioned	POK
Assumptions and constraints	The obstacle is in front of robot right arm and flat	It may miss the NGZ	POK
Expected results and criteria	The robot is able to identify the NGZ and back turn left and go straight until scan over the line then turn back to original direction	Indicate light orange	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute scan(0); then make a red line in front of robot	Robot across the NGZ due to the edge of NGZ is in the right side of robot	NOK
2	Execute scan(0); then make a red line in front of robot	Robot meet obstacle, back a little with light turn orange, then turn left scan a path then turn back to right with orange light off	OK

- Test 008

Test ID	**Test 008**	**Comment**	**Decision**
Test description	Test the ability of recognise the deposit - scan()	The robot is able to identify the deposit only if CS through the deposit	POK
Initial conditions	The robot is under scan condition		OK
Tests inputs	Colourful deposit		OK
Data collection actions	The coordinate and the colour of the deposit		OK
Tests outputs	The robot can stop and reflect in green light, storage the data of the deposit		OK
Assumptions and constraints	The deposit is placed in the path of robot's left arm	It may miss the deposit due to the CC is limited	POK
Expected results and criteria	The robot will record the deposit store the data and stop to indicate its discovery	Indicate light green	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute scan(0); then make a colourful deposit in path of robot's left arm	Robot across the deposit, stopped and light turned green then continue move and stop and light turned off after CC move out deposit	OK

3.3 Mission Test

- Test 009

Test ID	**Test 009**	**Comment**	**Decision**
Test description	Test the ability of recognise the map - initialMap()	The robot is able to identify the boundary	OK
Initial conditions	The robot is under scan condition		OK
Tests inputs	None		
Data collection actions	The size of the map		OK
Tests outputs	The robot can circle round the map and appropriate detecting the boundary of the map		OK
Assumptions and constraints	The robot is accurately placed besides the Boundary		NOK
Expected results and criteria	The robot is able to go alone each line of the boundary and stop at the end	The robot can not self adjust its direction when lose line	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute initialMap(); then make adjustment when every time it lose line	Robot can barely automatically accomplish the mission but with the manual assist	POK

- Test 010

Test ID	**Test 010**	**Comment**	**Decision**
Test description	Test the ability scanning the map - lineScan()		OK
Initial conditions	At the origin point after recognise the boundary		OK
Tests inputs	None		
Data collection actions	The size of the map, the NGZ points, the deposit information		OK
Tests outputs	Information from the map		OK
Assumptions and constraints	The coordinate of the robot is accurate and is able to run under the program controlled and the obstacle, NGZ and deposit is detectable	It can accomplish with the assumption of every action is accomplished accurately	OK
Expected results and criteria	The NGZ points, the deposit information	Indicate light orange	OK
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Execute lineScan(); repeatly	Robot across the NGZ due to the edge of NGZ is in the right side of robot	NOK
2	Execute lineScan(); repeatly	Robot lost the direction after two turns and the coordinate	NOK
3	Execute lineScan(); repeatly	Robot crash on the obstacle which not appear on the right side of the robot	NOK
4	Execute lineScan(); repeatly	Robot can finally accomplish all the map	OK