*ecse 211 design project*

Testing Document

Version *1.05*

*13/29/2018*

*ECSE 211 TEAM 11*

VERSION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | Testing Document | | | |
| **Description** | Test the built system to make sure it meets all the requirements | | | |
| **Created By** | Tianyi Zou, Testing leader | | | |
| **Date Created** | 1st March 2018 | | | |
| **Version Number** | **Modified By** | **Modifications Made** | **Date Modified** | **Status** |
| 1.00 | Tianyi Zou | Created the Testing Document Template | 1st March | Preliminary version of the document;  added testing template in the appendix |
| 1.01 | Luka Jurisic | Peer reviewed the document. Fixed some small errors and formatted the document. Added the introduction, 2 appendixes, and the test plan document. Created section 1.1-1.3.2 and section 2 | 2nd March | Preliminary template complete |
| 1.02 | Tianyi Zou, Enan Zaman | Completed section 3.2 and 3.4; Light Sensor and Wheels preference tests | 13th March | All other tests remain |
| 1.03 | Tianyi Zou | Completed section 3.1 and 3.3 | 20th March | All other tests remain |
| 1.04 | Volen | Completed 4.2 and 4.3 | 22nd March | All other tests remain |
| 1.05 | Luka Jurisic | Following criticisms in the 3rd weekly meeting, changes regarding test procedures were updated and corrected. Also Added testing process. Table of contents presentation was improved as well as the Testing Template | 26th March | Outstanding tests remain |

TABLE OF CONTENTS

# 1 INTRODUCTION

1.1 Purpose of the Test Plan Document

1.2 Testing Tools ……………………………………………………………………………

1.3 Quality Objective ……………………………………………………………………

1.3.1 Primary Objective ……………………………………………………………

1.3.2 Secondary Objective…………………………………………………………………

1.4 Testing Process   
 1.4.1 Hardware Components  
 1.4.2 Software Components  
 1.4.3 Integration Testing

# 2 TEST DELIVARABLES

# 3 Test report template

# 4 Hardware Testing

4.1 Ultrasonic Sensor ………………………………………………………………………

4.2 Light Sensor …………………………………………………………………………

4.3 Motors ……………………………………………………………………………

4.4 Wheels…………………………………………………………………………

4.5 Robot Structure…………………………………………………………………………

# 5 Software TESTING

5.1 Localization……………………………………………………………………………

5.2 Back wheel Functionality

5.3 Navigation……………………………………………………………………………

5.4 Obstacle Traversal ………………………………………………………………………

5.5 Search area …………………………………………………………………………

5.6 Capture …………………………………………………………………………

5.7 Obstacle avoidance ………………………………………………………………………

5.8 Wi-fi Integration…………………………………………………………………………

# 6 Integration testing

# Introduction

## Purpose of The Test Plan Document

The document documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project’s product. The document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and testing team. Some portions of this document may on occasion be shared with the client/user.

**1.2 TESTING TOOLS**

The following tools will be used for testing:

|  |  |
| --- | --- |
| PROCESS | TOOLS |
| Test Case Creation | Microsoft Word |
| Test Case Tracking | Microsoft Excel |
| Test Case Execution | Manual |
| Test Case Management | Microsoft Excel |
| Defect Management | Microsoft Excel |

# 1.3 Quality objective

# 

# 1.3.1 Primary Objective

The primary objective of this testing phase is to assure that the system meets the full requirements, including quality requirements, and maintain the metrics for each quality reequipments of the final design. At the end of the project development, the user should find that the project has met or exceeded all of their specifications detailed in the requirements.

# 1.3.2 secondary Objective

The secondary objective of this testing phase is to identify issues and propose solutions to all hardware and/or software issues, and to communicate all this to the project team. This requires careful and methodical testing of the design to ensure all areas of the system are scrutinized appropriately.

# 1.4 TESTING PROCESS

# 1.4.1 Hardware cOMPONENTS

# The hardware components utilized in the final design were separately tested in order to clearly identify weaknesses and allow a swift evaluation of performance. For example, at our disposal were 6 ultrasonic sensors, and tests were run solely on these sensors to determine which one was most accurate. This procedure was also performed for the light sensors and the motors. Motor testing was especially critical as there were two types of motors available, and a decision had to be made as to which type would be implemented into the final design.

# Similarly, precise values were obtained of both the wheel radius and wheelbase length, which are key for the software department to have and utilize.

# 1.4.2 software components

# The testing of software components follows on from its respective hardware testing, as an initial or even progressive design is required before any meaningful software testing phase can commence. Once again, key classes that governed the robot’s main behaviors were tested in isolation to fully assess their performance and flaws.

# Regarding software, the primary objective is to realize any flaws within the architecture. This requires that the robot be tested in a varying number of unfamiliar and unorthodox situations to ensure full functionality of the code. For example, localization has to be tested many times because the starting orientation of the robot is unknown, and thus as many cases as possible must be accounted for during this testing phase.

# 1.4.3 Integration testing

# Software and hardware testing ensures that each respective individual component performs accordingly. However, when all the components are working in tandem, this gives rise to problems previously unforeseen. The accumulation of small errors within each subsystem could result in a poor working final design. Thus, integration testing is the key final phase of testing that allows us to ensure that all of the client’s behavioral specifications are met.

# TEST DELIVARABLES

The testing phase will allow a general progression of the project in terms of both hardware and software. The testing phase will provide key deliverables that fall into 3 basic categories: Documents, Test Cases and Reports. The figure below illustrates the dependencies of these 3 categories.



*Diagram Source: https://strongqa.com/qa-portal/testing-docs-templates/test-report*

# 3 Test report template

# 

***Test’s Title:*** *Name of Test*

**Tester:** *Concerned Team Members* **Test Date:** *Month/Day/Year*  **Software Version:** *Software version of Robot* **Hardware Version:** *Hardware version of Robot*

**Objective:** *What is the purpose of the test?*

**Background knowledge (if needed):** *Relevant information disclosed here*

**Procedure:** *Detailed description of the steps performed to carry out the test*

**Expected Results:** *What is the predicted outcome of the test?*

**Test Report:** *Provide the relevant data acquired during the test, along with an analysis of the results. The number of times the test was performed must be included. Tables and/or graphs are key in providing visual information. See appendix for table templates.*

**Conclusion:** *Specify the outcome of the test. Were the requirements met and did they meet the expected outcome?***Action:** *Based on the results, provide a list of what needs to done going forward.*

**Distribution:** *Indicate which sections of the team need to be made aware of the results of the test to allow for necessary modifications.*

# Hardware Testing

# 4.1 Ultrasonic Sensor

***Ultrasonic Sensor Consistency Test***

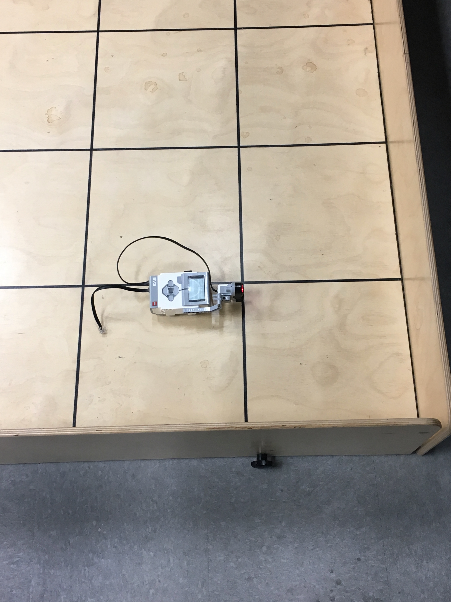
**Tester:** *Tianyi Zou* **Test Date:** *03/15/18* **Software Version:** *N/A\** **Hardware Version:** *N/A\**\**No hardware and software version due to the use of the built-in sensor testing application located in the tools option in the EV3 selection menu*

**Objective:**

Determine the accuracy of ultrasonic sensors available. This test will serve to identify which of the 3 ultrasonic sensors provided to our team have the least incongruities in their distance measurements. Thus, the most accurate and/or consistent ultrasonic sensors will be determined and implemented onto our final design.

**Procedure:**

1. Turn on the EV3 brick. Connect the Port 1 of the brick to the light sensor via a cable.
2. Assemble the ultrasonic sensor vertically in front of the robot. Make sure that the sensor should direct to the front of robot.
3. Use the tools application on the EV3 brick. Select **Tools>Test Sensors >Go> Port 1>EV3 Ultrasonic >Distance**.
4. Record the real distance from each tile and the measured value on the screen of brick. Please refer to the Figure 3.1.1.



*Figure 3.1.1 Placement of ultrasonic sensor during the test*

5. Calculate the average value and standard deviation of error that between real distance and measured distance.

6. Test other two ultrasonic sensors by using the same procedure. Compare the result from all different ultrasonic sensors.

**Expected result:**

We expect all the sensors would provide small inconsistencies when measuring distances and different ultrasonic sensor would perform differently. Some sensors would perform better than others, which give us a sense of which sensor to choose for object avoidance and localization.

**Test Report**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **US Sensor 1** | | **US Sensor 2** | | **US Sensor 3** | |
| **Real distance (cm)** | **Measured distance (cm)** | **Error (cm)** | **Measured distance**  **(cm)** | **Error (cm)** | **Measured distance**  **(cm)** | **Error (cm)** |
| 29.8 | 30.4 | 0.6 | 30.4 | 0.6 | 30.2 | 0.4 |
| 60.1 | 62.2 | 2.1 | 67.0 | 6.9 | 64.3 | 4.2 |
| 90.4 | 91.6 | 1.2 | 91.7 | 1.3 | 91.7 | 1.3 |
| 120.7 | 121.8 | 1.1 | 122.3 | 1.6 | 121.7 | 1.0 |
| 151.3 | 152.9 | 1.6 | 152.9 | 1.6 | 152.1 | 0.8 |
| 181.6 | 182.9 | 1.3 | 182.9 | 1.3 | 182.4 | 0.8 |
| 211.9 | 213.6 | 1.7 | 213.0 | 1.1 | 213.2 | 1.3 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sensor 1** | **Sensor 2** | **Sensor 3** |
| **Average Error (cm)** | 1.37 | 2.06 | 1.40 |
| **Standard Deviation (cm)** | 0.48 | 2.16 | 1.27 |

**Conclusion**:

The average error of US sensor 1 is the lowest and it has the lowest standard deviation among three sensors. This means sensor 1 provides more accurate values in distances.

**Action**:

We use ultrasonic sensor 1 as the priority choice. If more than one ultrasonic sensor needs to be used, sensor 3 would be the second-best choice.

**Distribution**: Hardware team

# 3.2 LIGHT Sensor

***Light Sensor Consistency Test***

**Tester’s names:** *Enan Zaman, Tianyi Zou* **Test Date:** *03/12/18* **Software Version:** N/A  
**Hardware Version:** N/A

**Objective:**

Determine the best performing light sensors available. This test will serve to identify which of the 3 light sensors provided has the clearest detection of an object from different heights.

**Procedure:**

1. Put a blue paper on the table. Use a ruler to measure the distance by putting a block adjacent to the ruler so that the ruler can be stabilized and placed perpendicular to the table surface.
2. Connect the Port 1 on the brick to the sensor via a cable.
3. Use the tools application on the EV3 brick. Select Tools>Test Sensors >Go> Port 1>EV3 Color >Color ID.
4. Place the light sensor next to the ruler and on the table surface. Make sure the direction of light should be to the blue paper on the table.
5. Move up the light sensor along the ruler slowly.
6. Record the distance *d1* between the light screen of light sensor and the table surface when the value of color ID shown on the screen of EV3 brick becomes 2.0.
7. Repeat step 5 and record the distance *d2* when the value of color ID becomes 7.0.
8. Repeat step 5 and record the distance *d3* when the value of color ID becomes -1.0.
9. Do the same procedure to test other two light sensors.

**Expected result:**

The value of *d1* and *d2* are respectively the closest and farthest distance that the light sensor is able to precisely detect an object, which means light sensor can both detect the object in front of it and identify the color of the object. Value of *d3* is the farthest distance that the light sensor can detect an object, but not able to identify the color.

**Test reports:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sensor # | *d1(cm)* | *d2(cm)* | *d3(cm)* |
| *Light Sensor 1* | 0.3 | 1.8 | 4.5 |
| *Light Sensor 2* | 0.6 | 1.0 | 3.5 |
| *Light Sensor 3* | 0.5 | 1.5 | 3.7 |

**Conclusion:**

Comparing the average value differences of all 3 sensors for any given height, it can easily be seen that the differences between the sensors are minimal and cannot be detected with tests of this quality. However, there is no need for more precise testing, as this test shows that all 3 color sensors would get the job done equally well as they can each detect the black lines with at least one data point for each black line.

Different light sensor performs differently. Generally, all sensors can detect objects well within the distance between 0.6 cm and 1.0 cm. Sensors can still detect objects but fail to identify the color at distance from 1.8 cm to 3.5cm.

**Action:**

We must place the sensor at distance between 0.6 cm and 1.0 cm from the block in order to detect the block and identify its color.

**Distribution:** Hardware team

# 3.4 Wheels

***Wheel type: Treads vs Regular Wheels***

**Tester:** *Enan Ashaduzzaman & Tianyi Zou* **Test Date:** *03/12/18* **Software Version:** *N/A* **Hardware Version:** *N/A*

**Objective:**

The objective of this test is to determine whether to use the treads or the regular wheels. Treads can be very useful at overcoming the bumps and not implementing a variable track. Fears include treads not being accurate during navigation. This test serves to provide the necessary *qualitative* data so that the best form of transportation can be implemented on the final robot.

**Background knowledge:**

From past knowledge from the labs, it was seen that the robot completes the navigation of the square accurately. Please note that design 1 and design 2 referred to in the test report below refer to the initial 2 preliminary designs discussed in the hardware document.

**Procedure:**

1. Build a robot using either the treads or regular wheels.
2. Check how the robot completes its navigation through the bridge.
3. Make the robot complete the square navigation to see the accuracy.

**Expected Results:**

It is expected both the regular wheels and treads will overcome the bumps on the bridge as they are relatively small. It is expected that the treads will be less accurate than the regular wheels during navigation. These little discrepancies can accumulate at the end of the day.

**Test Report:**

*Treads*

*Design 1:* The treads were loose on the robot since the wheels didn’t span the entire length of the tread. The robot completed the navigation through the bumps. The robot had a lot of accuracy issues during the square navigation.

*Design 2:* The front wheel of the tread was lifted slightly, tightening the tread on the robot. The robot had no difficulty traversing the bumps on the bridge. While the navigation got better from the first implementation, there were still some problems in the navigation of the robot.

Regular Wheels

*Design 1:* Having two wheels on each motor helped with the traction of the robot. The single marble holding the back end of the robot caused many issues when going through the bumps. It caused the robot to never travel straight. At the end, the robot never made it through the bridge during the tests. The navigation of the robot was more accurate than the navigation using treads.

*Design 2:* Having two marbles on the back end instead of one slightly helped the robot when traversing the bridge. It still encountered a lot of problems. It was concluded that the marble was not going to be a viable option for the robot.

**Conclusion:**It was concluded that the final robot would have the regular wheels implemented over the treads. Even though the treads were better at traversing the bridge, it was only by a slight margin. Considering that navigation is a key component during the final project, it is important to use the hardware that completes the navigation the bests. Therefore, the regular wheels were chosen considering they were a ton better at navigation.  **Action:**The next step in the hardware process is to implement a lazy wheel. This wheel will have similar abilities to a marble but will traverse the bumps on the bridge with ease.   
**Distribution:**Hardware team

# SOFTWARE tESTING

# 5.2 LANDING GEAR TEST

***Test’s Title: Back Wheel Functionality***

**Tester’s names:** *Volen Mihaylov* **Test Date:** *03/22/18*

**Software Version:***00.00.00*

**Hardware Version:***1.03*

**Objective:**

The objective of this test is to determine the optimal angle at which to turn the variable back wheels of the robot. This test will serve to highlight the angles at which the back wheels rotate from a stored to lowered position and vice versa that cause no overturning of the motor. Please refer to the procedure for a definition of overturning.

**Background knowledge:** None

**Procedure:**

1. Place the robot parallel to any one black line on the grid.  
2. Ensure the back wheels are lowered.

3.Run the relevant version of the software code.  
4. Observe the robot’s performance, keeping note of how the back wheels overturn. Overturning in this case refers to the back wheels coming too much into contact with other hardware components, producing a “clicking” sound.

**Expected Results:** The robot is not expected to overturn/under turn at all during this test run. The back wheels should lock smoothly back into place.

**Results obtained:**

*Note: A positive angle refers to the back wheels moving from an up position(stored) to a lowered position. A negative angle refers to the back wheels moving from a lowered position to a stored position.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tests** | **Variables** | | | | **Passed?** | **Comments:** |
| Angle down | Angle Up |  |  |  |
| **1** | 270 | -270 |  |  | Yes | Overturning, clicking sound can be heard |
| **2** | 270 | -270 |  |  | Yes | Overturning, clicking sound can be heard |
| **3** | 270 | -270 |  |  | Yes | Overturning, clicking sound can be heard |
| **4** | 250 | -250 |  |  | Yes | No overturning sound, wheels well placed |
| **5** | 250 | -250 |  |  | Yes | No overturning sound, wheels well placed |
| **6** | 250 | -250 |  |  | Yes | No overturning sound, wheels well placed |

**Conclusion:** Utilizing a 250 degree angle within the software proves to be superior to an angle of 270 degrees. This is highlighted in the above test results since at 270 degrees the robot’s back wheels overturned during each test run. However, the issue could also lie within the hardware design of the robot. A

**Action:** As of now, implement 250-degree angle within the software. Following this test, a discussion needs to take place between the software and hardware teams to discuss the possibility of a hardware improvement.

**Distribution:** Hardware/Software Team

***Test’s Title: Back Wheel Functionality -with stopper***

**Tester’s names:** *Volen Mihaylov* **Test Date:** *03/22/18*

**Software Version:\_***00.00.00*

**Hardware Version:** *1.03 with stopper*

**Objective:** The objective of this test is to determine the optimal angle at which to turn the variable back wheels of the robot. This test will serve to highlight the angles at which the back wheels rotate from a stored to lowered position and vice versa that cause no slight overturning/underturning of the motor.

**Background knowledge:** This test is identical to the “Back Wheel Functionality test” except that in this case a “stopper has been implementing within the hardware. The stopper serves to fully ensure that the motors don’t overturn. Also, the stopper acts as a support so that the robot can stand up on its back wheels.

**Procedure:**

1. Place the robot parallel to any one black line on the grid.  
2. Ensure the back wheels are lowered.

3.Run the relevant version of the software code.  
4. Observe the robot’s performance, keeping note of how the back wheels overturn. Overturning in this case refers to the back wheels coming too much into contact with other hardware components, producing a “clicking” sound.

**Expected Results: :** The robot is not expected to overturn/under turn at all during this test run. The back wheels should lock smoothly back into place.

**Results obtained:***Note: A positive angle refers to the back wheels moving from an up position(stored) to a lowered position. A negative angle refers to the back wheels moving from a lowered position to a stored position.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tests** | **Variables** | | | | **Passed?** | **Comments:** |
| Angle down | Angle Up |  |  |  |
| **1** | 200 | -200 |  |  | No | Too much Rotation |
| **2** | 195 | -195 |  |  | No | Too little rotation, does not go all the way |
| **3** | 192 | -192 |  |  | No | Too little rotation, almost goes all the way but is missing a couple of degrees |
| **4** | 190 | -190 |  |  | No | Not enough rotation |
| **5** | 185 | -185 |  |  | Yes | Looks good |
| **6** | 180 | -180 |  |  | No | Too little rotation, almost goes all the way but is missing a couple of degrees |
| **7** | 180 | -180 |  |  | No | Too little rotation, almost goes all the way but is missing a couple of degrees |
| **8** | 180 | -180 |  |  | Yes | Perfect |

**Conclusion:** When implementing a stopper, a wheel rotation of 195 degrees proves to be the most smooth and well-done action. The rotation is qualitatively “perfect”. The test also shows that the addition of a stopper to deal with overturning/underturning and to provide support is proving effective.

**Action:** Implement 195 degree of the wheel rotation within the software.

**Distribution:** Software Team

# Navigation

**Test’s Title: Navigation**

**Tester:** *Volen Mihaylov* **Test Date: 03/22/18**

**Software Version:** *01.00.07 to 1.01.00*

**Hardware Version:** *1.03 with stopper*

**Objective:** The objective of this test is to ensure that the robot can successfully and reliablty localize using both ultrasonic and light localization. The client’s requirement is ibitial localization must be performed under 30 seconds.

**Background knowledge (if needed):** None

**Procedure:**

1. Place the robot at its original position; that is, (0,0) as defined by you co-ordinate orientation.
2. Make the robot turn at certain wanted degrees. These are: (±45, ±90, ±270). Verify the offset, if there is any.
3. Make the robot move by a set wanted distance and verify the offset ( 1 tile, 2 tiles, 6 tiles).
4. Finally, make the robot move to a certain co-ordinate and and verify the distance covered. In this case, our starting co-ordinate is (0,0) and our end co-ordinate is (5,3).

**Expected Results:**

Have the robot move and rotate by the wanted amount

**Results obtained:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tests** | **Variables** | | | | **Passed?** | **Comments:** |
| Degrees tested at | Distance tested at | Coordinates tested at | Code Constant changed: |  |
| **1** | 45 |  |  | 0 | No | Left turn is off by 2 degrees to the left |
| **2** | -45 |  |  | 0 | No | Right turn is perfect |
| **3** | 90 |  |  | 0 | No | Left turn is off by 2 degrees to the left |
| **4** | -90 |  |  | 0 | No | Right turn is perfect |
| **5** | +270 |  |  | 0 | No | Right turn is perfect |
| **6** | -270 |  |  | 0 | No | Left turn is off by 2 degrees to the left |
| **1b** | 45 |  |  | Left: +4 | No | Left turn is off by 2 degrees to the right. |
| **2b** | -45 |  |  | Left: +4 | No | Right is perfect |
| **3b** | 90 |  |  | Left: +4 | No | Left turn is off by 2 degrees to the right. |
| **4b** | -90 |  |  | Left: +4 | No | Right is perfect |
| **5b** | +270 |  |  | Left: +4 | No | Right is perfect. |
| **6b** | -270 |  |  | Left: +4 | No | Left turn is off by 2 degrees to the right |
| **1b** | 45 |  |  | Left: +2 | No | Left Turn is perfect. |
| **2b** | -45 |  |  | Left: +2 | No | Right Turn is perfect |
| **3b** | 90 |  |  | Left: +2 | No | Left Turn is perfect. |
| **4b** | -90 |  |  | Left: +2 | No | Right Turn is perfect |
| **5b** | +270 |  |  | Left: +2 | No | Right Turn is off by 2 degree not enough. |
| **6b** | -270 |  |  | Left: +2 | No | Left is perfect |
| **7** |  | 1 tile |  |  | No | 0.5cm too short |
| **8** |  | 1 tile |  |  | No | 0.5cm too short |
| **9** |  | 2 tiles |  |  | No | 0.9cm too short |
| **10** |  | 2 tiles |  |  | No | 0.9cm too short |
| **11** |  | 6 tiles |  |  | No | 2cm too |
| **12** |  | 6 tiles |  |  | No | 2cm too short |
| **7b** |  | 1 tile |  | Rotation multiplied by: (*TILESIZE*+0.4)/*TILESIZE* | Yes | Perfect |
| **8b** |  | 1 tile |  | Rotation multiplied by: (*TILESIZE*+0.4)/*TILESIZE* | Yes | Perfect |
| **9b** |  | 2 tiles |  | Rotation multiplied by: (*TILESIZE*+0.4)/*TILESIZE* | Yes | Perfect |
| **10b** |  | 2 tiles |  | Rotation multiplied by: (*TILESIZE*+0.4)/*TILESIZE* | Yes | Perfect |
| **11b** |  | 6 tiles |  | Rotation multiplied by: (*TILESIZE*+0.4)/*TILESIZE* | Satisfactory | Perfect |
| **12b** |  | 6 tiles |  | Rotation multiplied by: (*TILESIZE*+0.4)/*TILESIZE* | Satisfactory | Perfect |
| **13** |  |  | (5,3) |  |  | Off by 3cm |
| **14** |  |  | (5,3) |  |  | Perfect |
| **15** |  |  | (5,3) |  |  | Perfect |

**Conclusion:** Navigation is functional. Left constant of +2 is satisfactory and so is the rotateByDistance constant of (*TILESIZE*+0.4)/*TILESIZE*

**Action:** Acceleration was changed to 500 instead of 2000 despite the code saying otherwise.

**Distribution:** Software

# Approval

The undersigned acknowledge they have reviewed the **Test Plan** document and agree with the approach it presents. Any changes to this document will be coordinated with and approved by the undersigned.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

Appendix A: References

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
|  |  |  |

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tests** | **Variables** | | | | **Passed?** | **Comments:** |
|  |  |  |  |  |
| **1** | v |  |  |  |  |  |
| **2** |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |
| **6** |  |  |  |  |  |  |

# Test template ^