Software Requirements Specification

for

Lunar Rover Mapping Robot Controller

Version 2.3.0

Prepared by Yong Yang
SEP UG-17

School of Computer Science,
The University of Adelaide

2017 October 30

Revision History

Version	Dated	Edited By	Change Summary
0.3.0	18/08/2017	Yong Yang	Added 26 user requirements
0.3.1	19/08/2017	Yong Yang	Changed from US spelling to UK spelling
0.3.2	21/08/2017	Yong Yang	Applied Ben's grammar fixes and Matthew's general modifications
1.0.0	28/09/2017	Yong Yang	Attaching this version to the milestone 1's software
2.0.0	28/09/2017	Yong Yang	Now all requirement items have dependencies and complexity slots. Added about a dozen requirements (there are now 35) elicited from either the client meetings and given documents
2.1.0	18/10/2017	Yong Yang	Filled in the new dependencies and complexity slots for all requirements. Updated the requirements' implementation status. Added more user requirements recorded in client meetings
2.2.0	25/10/2017	Benjamin Schuh	Unification of document style and grammar
2.3.0	30/10/2017	Benjamin Schuh	Final grammar and formatting checks

Related Documentation

ID	**Document Name**	**Version**
1	Software Project Management Plan	2.0.0
2	Software Design Documentation	2.0.0
3	Testing Report	1.0.0
4	User Manual	1.0.0

Contents

1 Introduction **3**
1.1 Purpose . . . 3
1.2 Document Conventions . . . 3
1.3 Intended Audience and Reading Suggestions . . . 3
1.4 Project Scope . . . 4
1.5 References . . . 4

2 Overall Description **5**
2.1 Product Perspective . . . 5
2.2 Product Features . . . 5
2.3 User Classes and Characteristics . . . 5
2.4 Operating Environment . . . 6
2.5 Design and Implementation Constraints . . . 6
2.6 User Documentation . . . 6
2.7 Assumptions and Dependencies . . . 6

3 User Requirements **6**
3.1 Automatic Mode Behaviours . . . 6
3.2 User Interface Elements . . . 10

4 Use Cases **24**
4.1 Automated Use Cases . . . 24
4.2 Manual Use Cases . . . 25

5 External Interface Requirements **26**
5.1 User Interface . . . 26
5.2 Software Interface . . . 26
5.3 Communication Interface . . . 26

6 Other Non-functional Requirements **27**
6.1 User Interface Friendliness Requirements . . . 27
6.2 Safety Requirements . . . 28
6.3 Performance Requirements . . . 28

A Glossary **29**
A.1 Initialisms . . . 29

1 Introduction

1.1 Purpose

This document sets out the requirements for the Lunar Rover Mapping Robot Controller (abbreviated to Controller in the following passages), which is to be used as a candidate for the prototype of the Lunar Rover Mapping Robot (abbreviated to Rover) in the Google Lunar X-Prize by the University of Adelaide. This Software Requirements Specification (SRS) describes the requirements derived from the given client project specification document and expressed wishes of the client recorded during client meetings. The Rover's functionalities to be worked on and the order of achieving them are all based on this document. All the user requirements are also referenced in the Software Design Document (SDD), which will further elaborate the functions implemented.

1.2 Document Conventions

User requirements are each given a unique identifier of the form [Rnnnn]. Any requirements that are deleted shall be marked as Requirement deleted following the identifier. Requirements will be numbered sequentially as they are added to the document. Since the requirements are added in several intervals and grouped into categories, the requirements are not listed in the order of their identifier. User requirements are given a priority of either High, Medium, or Low. High priority requirement are musts explicitly stated in the client project specification and/or musts confirmed with the client during client meetings in week 3 or week 4. Medium priority requirements will be worked on as soon as all high priority requirements are met, but they are not guaranteed to be finished; these are new must have features required by the client after week 4, and features that are desirable but not necessary for the product to be considered successful. Low priority requirements are features deemed hardly possible to implement within the given time, or hardly comparable with other features in terms of importance, but are nevertheless valuable additions to the product; they are, therefore, recorded here so that when the project is restarted at a later date, by the same or a new group, they could serve as suggestions on what to work on.

Use cases are also each given a unique identifier of the form [UCnnnn]. Use cases are defined as series of actions of the same nature in interaction type. Use cases are by no means mutually exclusive, as the user shall have the freedom to switch between action modes all the time. We define use cases so that the user can understand the usual sequence of actions the Rover follows to achieve a specific goal.

And for readability, the terms of "graphical user interface" (GUI) and "user interface" (UI) are interchangeable. So are "user mode" and "manual mode".

1.3 Intended Audience and Reading Suggestions

The audience for this document can be divided into three categories: staff members acting as clients at the University of Adelaide involved in the Google Lunar X-Prize project; developers of the Lunar Rover Mapping Robot Controller; and students in the Schools of Engineering, the School of Computer Science, and the School of Mathematical Sciences at the University of Adelaide.

For clients (staff members), the intention of the document is to clarify the software requirements' details, so that they can be corrected if the group's interpretations differ from the client's ideas, and to elicit more requirements from the client. Clients should read the entire document in order, paying special attention to the user requirements (Section 3), use cases (Section 4) and non-functional requirements (Section 6).

For developers, the intention of the document is to provide details for them to develop the software that could meet the client's expectations. Developers should read the entire document in order, paying special attention to the user requirements (Section 3), use cases (Section 4) and non-functional requirements (Section 6), with a focus on the parts that they're responsible for individually or as a team.

For fellow students, the intention of the document is to share information on the development of the candidate prototype of the Lunar Rover Mapping Robot and the Lunar Rover Mapping Robot Controller for the Google Lunar X-Prize. Our fellow students are welcome to read any part of interest to them, and contact the author Yong Yang (a1695329) or our team's document manager Benjamin Thomas Schuh (a1668396) should they have any questions or comments.

1.4 Project Scope

The software specified in this document is the controller for the Lunar Rover Mapping Robot. The main purpose of the software is to be implemented in Java to control a LEGO Mindstorms EV3 robot remotely, allowing it to automatically navigate through and explore environments with elements represented by lines and areas of different levels of thickness, form, and colour, and to be controlled by an operator manually when required. The major motivation for the development of this software is to produce a LEGO Mindstorms EV3 bot with behaviours desired in the Google Lunar X-Prize, so that it can serve as a prototype for a real Lunar Rover Mapping Robot that will be sent to the Moon.

1.5 References

Table 1: References

Reference Number	**Reference**	**Date**
Ref 01	UG 17 Software Design Document	25/10/2017
Ref 02	Course Group Project Specifications	07/2017
Ref 03	Client Group Project Specifications	07/2017
Ref 04	Documentation- SRS_Example	12/10/2007
Ref 05	Documentation- SRS_template	2002
Ref 06	Documentation- IEEE830	2009
Ref 07	Formal Spec	–/–/–

Note: Referenced documentation 02 through to 07 are all located on the University of Adelaide MyUni website page for Software Engineering & Project

2 Overall Description

2.1 Product Perspective

The product described in this SRS is a remote desktop controller interface for a LEGO Mindstorms EV3 robot and the robot itself, which is also built by the software development group. To participate in the Google Lunar X-Prize, it is required that a vehicle is sent to the Moon to survey the Moon's surface, and, in this project, to also find the remnants of Apollo 17 to win the Apollo Heritage Bonus Prize. And this product is being developed so that it will display similar behaviour to a machine that will win the Google Lunar X-Prize.

2.2 Product Features

The main functions of the Rover include:

- Automatically reach the target and go around it, mapping it on the Rover Map
- Automatically scanning the whole bounded area, marking all objects of interest specified by the client on the Rover Map
- Automatically avoid contacting NGZs and obstacles detected by the sensors or marked by users on the UI's map. This includes rejecting user commands that require it to enter no-go zones (NGZs)
- Automatically return to the landing site after finishing the survey mission, or on the operator's command
- The robot can travel to a point designated by the user automatically

The main features of the controller (user interface) include:

- The Rover Map displaying the robot's position and all objects of interest
- Options to import and export Rover Maps
- Options to choose different colour schemes to both let the robot detect objects in the environment and display them on the map
- An option to mark no-go zones (NGZs) on the Rover Map
- An option to designate a point on the map for the bot to go to
- An option to switch between user control (manual) mode and auto mode
- An option to let the bot return to the landing site
- An emergency stop button that will let the Rover stay where it is immediately
- Four arrow keys (up, down, left, right) to control the robot's movements in manual mode

2.3 User Classes and Characteristics

No professional knowledge is required for using this software, as the elements on the Rover Controller are all common elements similar to those found on maps and modern computers. Although the Rover can finish all of the explorations on its own in automatic mode, it is best that the user be aware of the mission's goals and the Rover's capabilities, so that the user can guide the Rover, to improve its efficiency and research items of interest in detail.

2.4 Operating Environment

As the Controller is implemented in Java, it will run on all machines installed with Java(TM) SE Runtime Environment that is compatible with 1.7 version. (So all versions since 1.7 would do because they are backwards compatible.)

2.5 Design and Implementation Constraints

Although the final product based on the Rover prototype is expected to be able to transmit live video and image (of low quality) and record high quality versions of these, the group is not provided with a camera, so there won't be a camera feed on the UI, and there won't be a place for the camera on the Rover.

2.6 User Documentation

There will be a User Manual document (in PDF format) delivered along with the software package on release, describing all possible actions on the user interface and the automatic behaviours of the Rover. There will also be a help button on the user interface, which will display short explanations of the elements of the user interface.

2.7 Assumptions and Dependencies

We are assuming very little time delay between when the commands are made on the user interface and when the Rover receives it (which will certainly not hold when the Rover is actually receiving commands sent from the Earth to the Moon). We are also assuming the Bluetooth/wifi signal is stable constantly and the Rover is always within reach of the stable signal. Finally, we assume the battery power level is sufficient to let the Rover complete a 20 minutes' survey, no matter what actions within its capabilities were required to perform.

3 User Requirements

3.1 Automatic Mode Behaviours

R0001: Target locating

Description The robot shall be able to locate the single target specified by the client by its radiation signal represented by coloured areas on the map, and mark it on the Rover Map, provided there is indeed a target in the bounded search area and the target is not blocked by NGZs.

Rationale The automated search for the target is a must as manual remote commands are laborious and error-prone. It will also be more efficient and reliable, being supported with tested algorithms.

Acceptance criteria This requirement can be verified by allowing the robot to finish a mock search mission by itself, then comparing the target location on the map with the real location of the target. It will be considered acceptable if it is within a circle of 1cm radius centred at the true position.

Source The client project specification document.

Status Not Implemented.

Dependencies None.

Complexity High.

Priority High.

R0002: Area survey

Description The robot shall be able to search through the whole permitted exploration area, marking all features of interest (decided by the client) on the Rover Map in real time.

Rationale The orderly search covering the whole permitted area is a major goal in the Moon exploration competition, and it is best done systematically with programmed behaviours.

Acceptance criteria This requirement can be verified by allowing the robot to finish a mock search mission by itself, and examine if all objects of interest were recorded on the map to within a 1 cm circle centred on the true position.

Source The client project specification document.

Status Not Implemented.

Dependencies R0001 Target Locating; R0003 Self Protection.

Complexity High.

Priority High.

R0003: Self protection

Description The robot shall not enter NGZs in all situations. So it will avoid going too close to NGZs in its search, and it will reject user commands that will make it enter NGZs. The robot will also detect any obstacles around it and avoid contacting them.

Rationale The safety of the robot is the highest priority, as the survey's progress relies on the robot's normal functionalities.

Acceptance criteria This requirement can be verified by allowing the robot to finish a mock search mission by itself, and check if the robot goes into a NGZ or hits any objects in the process. It would be considered successful if it does not. The Rover shall be considered damaged if it moves an object or is halted by an object with the wheels still rotating.

Source The client project specification document and the second client meeting.

Status Implemented.

Dependencies None.

Complexity Medium.

Priority High.

R0004: Automated return

Description The robot shall be able to return to the landing site (original location) by itself after finding the target and surveying the whole permitted area, or on user command.

Rationale The bot shall be collected after the search, and the most reliable way is to let it go back to the landing site by itself, as in the worst case, it can simply keep doing reversed operations to follow its own trail back to the starting point.

Acceptance criteria This requirement can be verified by allowing the robot to finish a mock search mission, then initiate the return command. It will be considered acceptable if the robot returns to the point where it began the search.

Source The client project specification document.

Status Not Implemented.

Dependencies None.

Complexity Low.

Priority High.

R0005: Moving to designated location

Description The robot shall be able to travel to a point on the Rover Map designated by the user in the same way it goes after the target, unless that point is not reachable due to the obstruction of NGZs.

Rationale The robot should be able to travel to any parts of the map that is of interest to the user, or in case the automated survey did not suffice so manual search was required.

Acceptance criteria This requirement can be verified by designating multiple points in multiple experiments (different Rover Maps) to let the robot go to. It will be considered acceptable if the robot reaches all possible points (those that can be reached without entering NGZs).

Source The client project specification.

Status Implemented.

Dependencies None.

Complexity Low.

Priority High.

R0028: Transmitting data to the GUI

Description The robot shall be able to send data such as its current position and heading to the GUI wirelessly.

Rationale Both the Rover's position and the surveyed environment can only be displayed on the GUI if the Rover can successfully send data to the GUI in real time remotely.

Acceptance criteria This requirement can be verified by letting the Rover start an automatic survey, and see if the information of the area it'd searched through can be updated on the GUI correctly. It will be considered acceptable if the real map was updated on the GUI, and the colours were detected correctly at the right places.

Source The client project specification.

Status Implemented.

Dependencies R0002 Area Survey; R0006 Rover Map.

Complexity Medium.

Priority High.

R0029: Receiving data from the GUI

Description The robot shall be able to receive data such as movement commands, the area of a new user drawn NGZ, or elements of an imported map from the GUI wirelessly.

Rationale All commands from the GUI can have effects on the Rover only if the data can be passed to it successfully.

Acceptance criteria This requirement can be verified by sending any commands from the GUI to the Rover and observe it's reaction. It will be considered acceptable if the Rover answered to the commands with expected behaviours.

Source The client project specification.

Status Implemented.

Dependencies R0002 Area Survey; R0006 Rover Map.

Complexity Medium.

Priority High.

3.2 User Interface Elements

R0006: Rover Map

Description There shall be a Rover Map to display the robot's current position, elements of interest on the map, and unmapped area.

Rationale The Rover Map is the major display to show the robot's current surroundings and the progress of the search, so it is a must in the graphical user interface.

Acceptance criteria This requirement can be verified by letting the robot finish a mock search mission, while user observes whether the bot's location and elements of interest were correctly displayed on the map.

Source The client project specification.

Status Implemented.

Dependencies None.

Complexity High.

Priority High.

R0007: Options to import and export Rover Maps

Description There shall be options to save (export) the current map and load (import) existing maps.

Rationale The search is time consuming and not without risks, and naturally the results are expected to be saved and shared. So it is necessary to export the map to save current progress, and to import maps to continue searches.

Acceptance criteria This requirement can be verified by using the import and export options and checking if the resulting map files and map work normally.

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity High.

Priority High.

R0008: Option to choose different colour schemes

Description There shall be options to choose different colour schemes so that the robot can recognise objects with multiple colour representation settings.

Rationale The objects can be represented by any colour, and changes to these relations can be resulted in different sensors or image conversion programmes, so the matching relations shouldn't be assumed to be fixed. Therefore, the ability to redefine colour representations of objects is necessary for long term usage and development.

Acceptance criteria This requirement can be verified by presenting the robot various environments with different colours, after setting the corresponding configurations (colour scheme).

Source The first client meeting.

Status Implemented.

Dependencies R0006 Rover Map; R0007 Options to import and export Rover Maps; R0016 Legends.

Complexity Medium.

Priority High.

R0009: Option to mark NGZs on the Rover Map

Description There shall be an option to allow the user to mark NGZs on the Rover map on the user interface. After selecting that option, the user can draw any shapes they want by clicking and dragging the left mouse button. Once the mouse is released, the beginning point (the point where the mouse clicked) and the ending point (the point on which the mouse was released) will be connected automatically, and the enclosed area formed by the whole line will be marked as NGZ on the GUI.

Rationale There may be certain areas that are not recognised as known NGZs by the robot's sensor, but could nevertheless be hazardous to enter. The operator will therefore mark these areas as NGZs to prevent the robot from entering them.

Acceptance criteria This requirement can be verified by letting the operator mark certain areas on the map as NGZs, and checking if the robot's NGZ avoiding programme functioned as intended when asked to traverse those areas.

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity High.

Priority High.

R0010: Option to designate a point to move to

Description There shall be an option to let the operator choose a point on the Rover Map for the robot to move to.

Rationale The robot should be able to travel to any parts of the map that is of interest to the user, in case the automated survey did not suffice, so manual search was required.

Acceptance criteria This requirement can be verified by designating multiple points in multiple experiments (different Rover Maps) to let the robot go to. It will be considered acceptable if the robot reaches all possible points (those that can be reached without entering NGZs).

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Medium.

Priority High.

R0011: Four arrow keys to control the robot's movement

Description There shall be four arrow keys (up, down, left, right) to allow the operator to control the robot's movement directly in manual mode. The up arrow key will make the Rover go forward as long as there are no obstacles or NGZs right in front of it. The left and right arrow keys will make the Rover rotate around in the respective directions. And the down arrow key will let the Rover go backwards.

Rationale The operator will need direct control of the robot to perform certain desirable tasks in the X-Prize contest, such as viewing an object at a specific angle, or following a special movement pattern, etc.

Acceptance criteria This requirement can be verified by letting an operator control the robot in manual mode, and checking if the robot behaves as expected when the arrow keys are pressed.

Source The First Client Meeting.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Low.

Priority High.

R0012: Option to switch between manual mode and automated mode

Description There shall be an option to switch between auto mode, which will let the robot explore the bounded search area automatically, and return to the landing site upon completion of survey, and manual mode, which will allow the operator to control the robot directly with arrow keys.

Rationale As the robot can either move by itself or be controlled by the user, there must an option to switch between these two modes, as they are mutually exclusive.

Acceptance criteria This requirement can be verified by observing if the robot will respond to arrow keys in auto mode, or if it will move on its own in manual mode and ensuring these do not occur.

Source The first client meeting.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Low.

Priority High.

R0013: Option to let the robot return to landing site

Description There shall be an option to let the robot return to the starting point by itself.

Rationale There could be many occasions that require the robot to return to the landing site without completing the survey, such as the Moon Rocket landed on an undesirable location, the target was proven to be too far to reach, or the remaining power supply cannot support it to finish a thorough search, etc.

Acceptance criteria This requirement can be verified by ordering the robot to return to the landing site with the return option under different circumstances (interrupting all sorts of behaviours) and checking if the robot reacts immediately and proceeds to return to the beginning point.

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Low.

Priority High.

R0014: Emergency stop button

Description There shall be an emergency stop button to order the robot to cease all actions immediately. The emergency mode will prevent all future movements until it is disengaged by double clicking the emergency button.

Rationale This function is for unexpected occasions in which the operator would not want the robot to behave as programmed, or the user regrets the previous command and want the robot to stop as soon as possible.

Acceptance criteria This requirement can be verified by pressing the emergency stop button while the robot is doing all other procedures. It will be considered successful if the robot indeed stopped immediately after receiving the command under all circumstances.

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Medium.

Priority High.

R0015: Status bar

Description There will be a message bar displaying the Rover's current action.

Rationale It is very difficult to tell purely from the Rover's location on the Rover Map what the Rover is currently doing, and the few possible cases are not always obvious to users, especially those not quite familiar with the product. Therefore, a status bar will help the user to keep track of the Rover's behaviour and decide when to intervene (such as telling the Rover to return to landing site when the status is target blocked by NGZs).

Acceptance criteria This requirement can be verified by comparing the Rover's behaviours which are defined in the User Manual and checking if they match the description on the status bar.

Source The second client meeting.

Status Implemented.

Dependencies Any requirement from R0001 to R0005 or R0007 to R0014.

Complexity Medium.

Priority High.

R0021: Option to remove user drawn NGZs

Description There should be an option to select and remove NGZs defined by the user. Once the button is clicked and therefore the removing NGZ mode is triggered, when the user clicks on a user defined NGZ with the left mouse button, it will be removed from the GUI, and the area under it available to be visited by the Rover again.

Rationale Not all users are familiar with the rules and effects of marking NGZs. And while some NGZs may be drawn with careful considerations, they might prove to be unnecessary eventually, or the user may have made a mistake while drawing it, which could happen very often when it is done with a mouse. In the end, the function to remove NGZs will be helpful at least, and might even affect the success of the exploration, as the Rover will be forced to abort the mission if the paths to the target are blocked by user marked NGZs.

Source The second client meeting.

Status Implemented.

Dependencies R0006 Rover Map; R0009 Option to mark NGZs on the Rover Map.

Complexity High.

Priority Medium.

R0022: Set destination to travel to by entering coordinates

Description Beside the option to designate a point to travel to on the map, there will also be a slot to enter x and y coordinates to set as destination for the Rover to go to.

Rationale Destinations set with mouse clicks lack both accuracy and portability (a destination on maps of different sizes cannot be defined with mouse clicks), and it can be fixed with this option to enter coordinates. However, this option is more likely to be favoured by users who are comfortable working with coordinate systems in scientific explorations only, so it is not a high priority as this product is meant for the general public.

Source The second client meeting.

Status Not Implemented.

Dependencies R0006 Rover Map.

Complexity Low.

Priority Low.

R0023: Rover log

Description There should be a time log integrated with the status bar that can be navigated with a scrolling bar, so that the user can not only check the current Rover behaviour, but also view all records of the past actions performed, and all discoveries with time stamps.

Rationale While the time log is not required for a prototype, it is a must for almost all automated research robots. This also provides a convenient means to see if the Rover has identified objects of interest successfully on the map.

Source The second client meeting.

Status Implemented.

Dependencies R0015 Status Bar.

Complexity Medium.

Priority Medium.

R0024: Options to show and hide travelled route and future path

Description There will be options to turn on and off the route the Rover has travelled and the calculated path it is going to take on the Rover Map.

Rationale The history of the Rover's trail is very valuable in analysing the Rover's behaviour under various circumstances and exploration process. The calculated route can help the user in deciding when to take over control, but might also distract the user if the path is recalculated too often.

Source The second client meeting.

Status Not Implemented.

Dependencies R0005 Moving to designated location; R0006 Rover Map.

Complexity Medium.

Priority Low.

R0025: Speed options

Description There will be a slot on the UI to allow the user to set the Rover's speed ranging from 1 cm/s to 5 cm/s .

Rationale As the final Lunar Rover based on this prototype is expected to take part in a competition, there will be certain terrains that should be traversed slowly, and some that

should be traversed quickly. The speed option is therefore desirable, though it is not required by the client for the prototype.

Source The second client meeting.

Status Not Implemented.

Dependencies None.

Complexity Low.

Priority Low.

R0026: Power level display

Description There will be an icon displaying the current power level in percentage.

Rationale The vehicle's remaining power level is always a figure of great importance in remote scientific explorations. Although the group's allowed to assume enough power for a 20 minutes' search in all experiments, it is safer to keep monitoring the power level to be aware of various possible emergency cases. This is also expected in the final Rover version based on this prototype.

Source The second client meeting.

Status Not Implemented.

Dependencies None.

Complexity Medium.

Priority Low.

R0027: Mini Map

Description There will be a second smaller map displayed on the UI, which will show the Rover's immediate surroundings with all the elements of interest as in the main Rover map.

Rationale The smaller map will show the user the Rover's immediate surroundings and allow the user to react to emergency situations much quicker.

Acceptance criteria This requirement can be verified by checking the position of elements on the smaller map against those on the larger map.

Source The first client meeting.

Status Implemented.

Dependencies R0006 Rover Map; R0030 Zooming.

Complexity Medium.

Priority Medium.

R0030: Zooming

Description The user will be able to zoom in and zoom out of the main map by pressing the plus and minus keys on the number pad.

Rationale Beside showing the Rover's current position, the main use of the Rover map is to display the elements of interest in the area. And to inspect the elements closely, there must be a zooming function to focus on a part of the map and enlarge it.

Acceptance criteria This requirement can be verified by zooming in or out of the map and see if the elements' sizes were changed with consistent ratio.

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Medium.

Priority Medium.

R0031: Panning

Description The user will be able to pan the Rover Map by clicking and dragging it with left mouse button when the user is not in destination setting mode, NGZ creation mode, or NGZ removal mode.

Rationale The zooming function alone won't allow the user to inspect any place of the map, as the map window's size is limited. Therefore, the panning function is necessary.

Acceptance criteria This requirement can be verified by left clicking and dragging the main map in normal mode and see if all the elements are moving by the same distance.

Source The client project specification.

Status Implemented.

Dependencies R0006 Rover Map.

Complexity Medium.

Priority Medium.

R0032: Option to focus on the centre of the Rover Map

Description There will be a button to allow the Rover Map to be put to the centre of the map window after it had been dragged off the centre.

Rationale The main map may be dragged off the window. It would be much more convenient to allow the user to return the map to the centre with a single click than to let the user drag the map back to the centre again.

Acceptance criteria This requirement can be verified by dragging the Rover Map away from its original place, then click on the focus button and see if the Rover Map did return to the centre of the map window.

Source The client project specification.

Status Not Implemented.

Dependencies R0006 Rover Map.

Complexity Medium.

Priority Medium.

R0033: Alternative way to mark NGZs: single line style

Description Other than the default drawing NGZ shapes function, there will be another way to mark NGZsby drawing a single line, which won't form an area.

Rationale Sometimes it's not easy to enclose an area as NGZ precisely with mouse moves. A reasonable alternative would then be allowing the user to draw lines that the Rover cannot pass through.

Acceptance criteria This requirement can be verified by choosing the drawing NGZ line mode and click and drag with the left mouse button and see if the traversed cells become NGZs on the GUI.

Source Derived from R0009 Option to mark NGZs on the Rover Map.

Status Not Implemented.

Dependencies R0006 Rover Map; R0009 Option to mark NGZs on the Rover Map.

Complexity Low.

Priority Low.

R0034: Alternative way to mark NGZs: paint brush style

Description Other than the default drawing NGZ shapes function, there will be another option to mark NGZs by drawing areas with thick lines as with a paint brush.

Rationale An intuitive way to mark areas is painting them with mouse moves. This function allows users to draw areas of any shape by painting them directly instead of defining their borders.

Acceptance criteria This requirement can be verified by choosing the drawing NGZ brush mode and click and drag with the left mouse button and see if the traversed cells and their adjacent cells became NGZs on the GUI.

Source Derived from R0009 Option to mark NGZs on the Rover Map.

Status Not Implemented.

Dependencies R0006 Rover Map; R0009 Option to mark NGZs on the Rover Map.

Complexity Low.

Priority Low.

R0035: Alternative way to mark NGZs: circles

Description Other than the default drawing NGZ shapes function, there will be another option to mark NGZs by drawing circle NGZ areas of various sizes.

Rationale There are circle NGZs on the DTD map example, so it would be reasonable to allow users to draw in such a way.

Acceptance criteria This requirement can be verified by choosing the drawing NGZ circle mode and clicking and dragging with the left mouse button to see if a circle NGZ appears on the GUI map.

Source The DTD map example.

Status Not Implemented.

Dependencies R0006 Rover Map; R0009 Option to mark NGZs on the Rover Map.

Complexity Low.

Priority Low.

R0036: Adjust the screen resolution automatically

Description The program will detect the user's computer screen resolution and resize itself

accordingly and automatically.

Rationale There are many possible sizes for computer screens, and many different preferences for resolution. The GUI would be very inconsistent if it always presented itself with a default resolution and width-height ratio. Therefore, a function to suit itself to every screen's resolution ratio is desirable.

Acceptance criteria This requirement can be verified by running the UI program on computers with different screen sizes and resolution levels. It will be considered acceptable if the GUI is automatically changed to fit to screen so that it's neither too small nor too large to use.

Source The first client meeting.

Status Implemented.

Dependencies None.

Complexity Low.

Priority Medium.

R0037: Rotate Rover Map

Description There will be an option to rotate the main Rover Map by 90 degrees to the left and another option 90 degrees to the right.

Rationale There isn't a default coordinate system with directions specified, so upon initialization of the Rover map, North, East, South, and West may not be defined as expected. This problem can be solved by allowing rotations of the map.

Acceptance criteria This requirement can be verified by running the UI program, doing some random operations on the map, then clicking on either of the rotation buttons once, and seeing if the Rover Map's response to those operations was correct according to the rotated coordinate system.

Source The first client meeting.

Status Not Implemented.

Dependencies R0006 Rover Map.

Complexity Medium.

Priority Low.

R0038: User name shown on Rover Log

Description When the Rover Log is opened by clicking on the Status Bar, the user name

registered on the computer will be shown at the beginning of the log.

Rationale The user's identification will make the user aware of the role he/she is assuming in operating the system. It's not important at the initial stage of this project, but in the final working version, there may be various levels of privileges for different user groups, then the current user name can help them identify their roles and permissions.

Acceptance criteria This requirement can be verified by running the UI program on many computers with different users. It will be considered acceptable if the different user names showed up correctly on the Rover Log.

Source The fifth client meeting.

Status Implemented.

Dependencies R0015 Status Bar.

Complexity Low.

Priority Low.

R0039: Transmitting data to the Rover

Description The GUI shall be able to send data such as new user drawn NGZs' information, current operation mode, information of imported maps, etc. to the robot wirelessly.

Rationale The GUI can only control the Rover when it can send data successfully to the Rover.

Acceptance criteria This requirement can be verified by sending commands to the Rover on the GUI, and see if the robot responded to the command correctly, if at all. It will be considered successful if the Rover's following behaviour were expected from the given command.

Source The client project specification.

Status Implemented.

Dependencies Any requirement from R0007 Options to import and export Rover Maps to R0014 Emergency stop button.

Complexity Medium.

Priority High.

R0040: Receiving data from the Rover

Description The GUI shall be able to receive data such as the Rover's current position and heading, and newly found elements from the Rover wirelessly.

Rationale The GUI requires the Rover's data feedback to track it's location on the Rover Map, and to update the new elements on the Rover Map.

Acceptance criteria This requirement can be verified by letting the Rover start an automatic survey, and see if the information of the area it has searched through can be updated on the GUI correctly. It will be considered acceptable if the real map was updated on the GUI, and the colours were detected correctly at the right places.

Source The client project specification.

Status Implemented.

Dependencies Any requirement from R0001 Target Locating to R0005 Moving to designated location.

Complexity Medium.

Priority High.

4 Use Cases

4.1 Automated Use Cases

UC0001: Automated search for the target

Description The robot searches for the mission target following its programme.

Flow

1. The Rover will look for target radiation with its sensors at the very beginning. If no radiation is detected, it will keep surveying the area (refer to UC0002 until it finds the target radiation.
2. It will proceed directly towards the target while mapping its path.
3. If there are obstacles or NGZs in its calculated route, it will modify its path to go around them, and at the same time update the Rover Map.
4. If it cannot find a safe path to the target, it will continue to survey the unexplored area (refer to UC0002).
5. After reaching the target, the Rover will go around it to mark it completely on the map.
6. After the target is marked, the Rover will pause for 3 seconds, displaying Target marked successfully on the status bar, then go on surveying the area if there's still unexplored area, or return to the landing site by itself.
7. END 1: The target's radiation was never found after completing the survey of the whole map. The Rover will display Target could not be found. on the status bar and wait for user command.
8. END 2: The Rover couldn't calculate a safe path to reach the target. The Rover will then display Target could not be reached. on the status bar and wait for user command.
9. END 3: After part 6 is done, the Rover will wait at the landing site and display Mission Complete! on the status bar.

UC0002: Automated survey of the area

Description The robot surveys the whole area, marking elements of interest on the map (in coloured areas and lines just as shown in the area), and update its route in the meantime.

Flow

1. When the target is found, the Rover will always go after it first (refer to UC0001) whenever possible, unless it is ordered by a user to go to a specific location first (refer to UC0003).
2. The Rover scans the 180 degrees of area in front of it, updating the Rover Map, and will go ahead for a short distance, turn left, or turn right to reveal or go near unexplored areas. It will not come in contact with dangerous elements represented by coloured areas and lines or physical objects detected by its sensors.
3. (NOTE) The ENDs for **UC0002 always happen right before the ENDs of UC0001.**

4. END 1: Some areas couldn't be surveyed because of NGZs. The Rover will display Survey within safe areas complete for 3 seconds after stopping movement.
5. END 2: The map had been completely surveyed. The Rover will display Survey Complete! for 3 seconds after stopping movement.

UC0003: Going to user designated location

Description Instead of the mission target, the robot will temporarily try to reach the location set by the user on Rover map as the main goal. The behaviour is very similar to **UC0001**, Although there are some differences. The differences are:

1. As the goal is set by the user, it is guaranteed to exist, so the Rover won't have to search for it. The task is then simplified to surveying and path finding.
2. If the Rover finds that the target is actually within a NGZ, it will immediately display "Target location is within NGZ, cancelled attempt to reach it." and switch to user mode automatically.
3. If the Rover finds that the target cannot be reached after the whole map is surveyed, it will display "Target cannot be reached" and switch to user mode automatically.

After reaching the target successfully, it will display "Target reached" and switch to manual mode to await further commands.

UC0004: Return to landing site

Description Either when the mission is complete, or the Rover receives a user command to abort the current mission and return, it will calculate the shortest path to return to the starting point.

4.2 Manual Use Cases

UC0005: Self protection

Description As the Rover's safety will always be its highest priority, it will disobey user commands that will make it go through or come into contact with obstacles or NGZs.

UC0006: Emergency state

Description The user can toggle emergency state by pressing the emergency button. The Rover will then cease all actions, and it will not respond to any further movement commands until the emergency state is disengaged by pressing the emergency button twice within a second.

UC0007: Arrow key movement

Description The Rover will rotate around when left or right keys are pressed. It will move forward (if it is safe to do so) when the up arrow key is pressed, at a speed of approximately 5 centimetres per second. When the down arrow key is pressed, it will go backward, following its own trail.

5 External Interface Requirements

5.1 User Interface

The Rover Controller (implemented by Java Swing) will be a command panel showing the Rover Map and options to control the Rover. Most options are buttons that can be pressed with mouse clicks, while the user can also use the keyboard to move the Rover in manual mode, zoom in and out on the map, toggle control modes, engage and disengage the emergency mode, etc.

5.2 Software Interface

The software is completely written in Java, and it will run on a Java Virtual Machine (required SE Runtime Environment 1.7).

5.3 Communication Interface

The Rover Controller reaches the Rover with Bluetooth and/or Wifi. As of this version, the communication software is set up for Bluetooth communication.

6 Other Non-functional Requirements

6.1 User Interface Friendliness Requirements

R0016: Legends

Description There shall be a legend on the Rover Map's right.

Rationale There must be legends for the user to refer to so they can understand the elements shown on the Rover Map.

Source The second client meeting.

Status Implemented.

Dependencies None.

Complexity Medium.

Priority High.

R0019: Help option

Description There should be a help button on the User Interface that, when clicked, will open the user manual with explanations on all the elements of the GUI.

Rationale The buttons' exact effects may not be obvious and could be ambiguous, so an option to show brief descriptions of them is desirable.

Source The first client meeting.

Status Not Implemented.

Dependencies None.

Complexity Medium.

Priority Medium.

R0020: User manual option

Description There should be a user manual button that will open the user manual (which is delivered along with this SRS).

Rationale The client should want to view the complete user manual for the details of the Rover Controller some time, so an option to open it would be convenient so that the user won't have to search for it.

Source The second client meeting.

Status Not Implemented.

Dependencies None.

Complexity Low.

Priority Medium.

6.2 Safety Requirements

R0017: Avoiding collision with any objects

Description The Rover will not come into contact with any objects horizontally, and will always proceed at a stable and safe speed.

Rationale While the Rover must ensure its own safety as the highest priority, it has been explicitly required that the integrity of the site be fully preserved.

Source The client project specification and the first client meeting.

Status Implemented.

Dependencies R0003 Self Protection.

Complexity High.

Priority High.

6.3 Performance Requirements

R0018: Exploration time limit

Description The Rover must complete the survey within 20 minutes, and preferably within 15 minutes.

Rationale Due to the limited resources available and the nature of the Google Lunar X-Prize competition, the Rover should finish the survey as quickly as possible with the given constraints to outmatch other contestants.

Source The second client meeting.

Dependencies R0005 Moving to designated location.

Complexity Medium.

Priority High.

A Glossary

A.1 Initialisms

GUI Graphical user interface
NGZ No-go zone
UI User interface (interchangeable with GUI)
SRS Software Requirements Specification