PLTSC-SDET-Assignment MTP

Author: Daniel Moreno

Latest Review: 2023/03/24

Reviewer(s): Daniel Moreno

Introduction

This document aims to specify the needs and requirements of the PLTSC Roomba Service as to conduct a proper set of test cases to assess its faults and potential issues based on the attributes and specifications given by the PLSTC team for the *Platform Science Software Development Engineer in Test assignment*. All of the aforementioned elements will be discussed more in depth by going through the following points:

1. Objectives
2. Testing Scope
3. Preparations
4. Test Matrix –REMOVED
5. Results

Objectives

Listed below are the main targets of the testing efforts of the PLTSC Roomba Service; these range from technical to business requirements, which should be validated, enforced and(or), maintained by the test cases and suites later addressed:

- Validate that the service's output provides valid final coordinates.
- Validate that the service's output provides an accurate number of patches cleaned.
- Ensure the service input processing complies with the following specifications:
    - Validate that the service has a failsafe for incomplete service calls.
    - Confirm that instructions input that drive the Roomba towards the border of the map instead make the Roomba skid in place.
    - Verify the roomSize and coords inputs have a valid failsafe for out-of-bounds or invalid input.
        - Not an array
            - Single Int
            - Single String
            - Single Float
        - String array
        - Float array
        - Mixed array
            - Int-Str
            - Int-Float

- Verify the roomSize input has valid size limits and restrictions
- Verify the coords input has a valid failsafe for out-of-bounds values (x or y bigger than roomSize equivalent).
- Verify the patches input has a valid failsafe for out-of-bounds of invalid input.
    - Not an array
        - Single Int
        - Single String
        - Single Float
    - Not a 2-dimentional array
        - Int array
    - Mixed 2-dimentional array
        - Individual element not an array
- Verify the patches input has a valid failsafe for out-of-bounds values (x or y bigger than roomSize equivalent)
- Verify the instructions input has a valid failsafe for out-of-bounds or invalid input.

Testing Scope

This testing effort will encapsulate the components of the PLTSC Roomba Service into categories, as per the specifications defined on the project's description.

Processing

- CastService
    - Main Goal:
        - Validating that the service is processing data correctly and it enforces proper restrictions on which types of data can be used when casting it.
    - Submodules:
        - StandardCall
            - ValidData
                - Control Call
                - Empty Element
                    - Empty Room Size
                    - Empty Coords
                    - Empty Patches Array
                    - Empty Movement Instructions
            - InvalidData
                - Invalid roomSize
                    - InvalidType
                        - Pass String
                        - Pass Float
                    - Pass Mixed Data
                        - IntFloat
                        - IntString
                - Invalid coords
                    - InvalidType

                - Pass String
                - Pass Float
            - Pass Mixed Data
                - IntFloat
                - IntString
            - Out of bound coordinates
        - Invalid Patches
            - Not2DArray
                - Pass String
                - Pass 1D Int Array
            - 2D Array
                - Pass Mixed Data
                    - Valid Coordinates and Invalid Coordinates
                        - IntFloat
                        - IntString
                - Out of bound coordinates
    - AlteredCall
        - Missing Elements
            - Pass empty JSON
            - Pass without element
                - No roomSize
                - No coords
                - No Patches
                - No Movement
        - Additional Elements

Movement

- Regular Input
    - Controlled Movement
    - Skid In Place
    - Instruction Load
        - 25 Instructions
        - 50 Instructions
        - 100 Instructions
- Abnormal Input
    - Mixed Input
    - Random String
    - Random Int

Cleaning

- End on Patch
- Skid on Patch
- Clear all Patches twice
- Clear single Patch

- ❖ Start on Patch
- ❖ Multiple patches on single coordinate

Preparations

Test Cases will be automated by creating a Framework that runs on Cucumber and generates JSON files for each test execution.

The priority of each test case will be determined by the following criteria:

- Repeatability
- Criticality
- Reproducibility

Results

A report containing all information regarding the latest run can be found in the reports folder.

Bugs:

Cleaning-PatchRegistry-MemoryLeak: Patches coordinates are passed onto consecutive service calls when running service calls with differing patches inputs.

Criticality: Critical

Priority: Critical

Movement-RectangularMaps-UnexpectedBoundaries: Movement is restricted by unexpected boundaries when passing a roomsize with differing x and y values in service calls.

Criticality: Critical

Priority: High

Movement-Instructions-LowercaseValues: Service crashes when attempting to pass valid lowercase values in the instructions field.

Criticality: Medium

Priority: Medium

Notes: Product Owner approval is required to validate this bug.

Processing-ErrorHandling-InvalidPatches: Service crashes instead of returning 400 status when calling the service with invalid patches values.

Criticality: Low

Priority: Medium

Processing-ErrorHandling-InvalidInstructions: Service crashes instead of returning 400 status when calling the service with invalid instructions values.

Criticality: Low

Priority: Medium

Processing-ErrorHandling-EmptyRoomsize: Service returns 500 status instead of 400 when executing a call with an empty array as the roomSize attribute.

Criticality: Low

Priority: Low

Processing-ErrorHandling-EmptyCoords: Service crashes instead of returning 400 status when executing a call with an empty array as the coords attribute.

Criticality: Low

Priority: Medium

Processing-ErrorHandling-OOBCoords: Service crashes instead of returning 400 status when executing a call with coords values that surpass either the x or y values of the roomSize array.

Criticailty: Low

Priority: Medium

Processing-ErrorHandling-OOBPatches: Service crashes instead of returning 400 status when executing a call with patches values that surpass either the x or y values of the roomSize array.

Criticailty: Low

Priority: Medium