SOFTWARE TEST DOCUMENTATION

**VERSION 1.1**

**06.06.2012**

ROBOT ANIMATION PROJECT

Prepeared by:

ŞERAFETTİN ÖZTÜRK

ZEYNEP MAVUŞ

**Preface**

This document contains the system test information for Robot Animation Project.

The document is prepared according to the “IEEE Standard for Software and

System Test Documentation – Software Test Descriptions – IEEE Std 829 – 2008”.

This Software Design Documentation provides a complete description of test plan of

the Robot Animation Project. The first section of this document includes the document

purpose, scope and References used though out the document. The following sections

include test processes of the system.

# 

Table of Content

[1.0. Introduction 3](#_Toc324113659)

[1.1. Document Identifier 3](#_Toc324113660)

[1.2. Scope 3](#_Toc324113661)

[1.3. References 3](#_Toc324113662)

[1.4. Glossary 3](#_Toc324113663)

[Institute of Electrical and Electronics Engineers - IEEE 4](#_Toc324113664)

[1.6. Overview of Document 4](#_Toc324113665)

[2.0. System Test Plan 4](#_Toc324113666)

[2.1. Test Items 4](#_Toc324113667)

[2.2. Features To Be Tested 5](#_Toc324113668)

[2.3. Features Not To Be Tested 5](#_Toc324113669)

[2.4. Details 6](#_Toc324113670)

[2.4.1. Use Case: Select Scenario Test Case 6](#_Toc324113671)

[2.4.2. Use Case: Create Obstacles Test Case 7](#_Toc324113672)

[2.4.3. Use Case: Control the Robot Test Case 8](#_Toc324113673)

[2.4.4. Use Case: Pause Test Case 9](#_Toc324113674)

[2.4.4. Use Case: Replay Test Case 10](#_Toc324113675)

[2.4.5. Use Case: Resume Test Case 11](#_Toc324113676)

[2.4.5. ‘P’ and ‘Q’ Buttons Test Cases 12](#_Toc324113677)

[2.5. Environmental Needs 13](#_Toc324113678)

[2.5.1. Hardware 13](#_Toc324113679)

[2.5.2. Software 13](#_Toc324113680)

# 1.0. Introduction

## 1.1. Document Identifier

This document is aimed at defining test strategies, bug-tracking procedures and providing a testing schedule of the Robot Animation Project.

## 1.2. Scope

The test scope of the Robot Animation Project includes testing of all functional requirements, non-functional requirements, application performance, time and design constraints specified in the software requirements document.

## 1.3. References

IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation.

IEEE Computer Society, 2008.

## 1.4. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Simulate | Imitation or enactment, as of something anticipated or intesting. |
| User | Person who is using the software. |
| Obstacle | Something that obstructs or hinders progress. |
| Scenario | A description of the starting positions of obstacles on the display screen |
| Software Test  Documentation | A document describing the scope, approach, resources of intended test  activities |
| Test Case | A set of test inputs, execution conditions, and correct results to prove  compliance with a particular requirement |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Term** | | **Abbreviation of** |
| JDK | Java Development Kit |
| JRE | Java Runtime Environment |
| IEEE | [Institute of Electrical and Electronics Engineers - IEEE](http://www.google.com.tr/url?sa=t&rct=j&q=ieee&source=web&cd=4&sqi=2&ved=0CIkBEBYwAw&url=http%3A%2F%2Fwww.ieee.org%2Fportal%2Findex.jsp&ei=dOWmT8-QDMfStAa22s3JBA&usg=AFQjCNFG7kSY8tJHBrOZpApbxy492HbAuw) |

## 1.6. Overview of Document

In the first chapter, document related issues are handled namely the document

identifier, scope, references and glossary sections.

In the next chapter, system test plan including items to be tested, features to be and not

be tested, test cases which identify each test case with expected test results and

environmental needs to be able to conduct test plan.

# 

# 2.0. System Test Plan

## 2.1. Test Items

This part is aimed at identifying the items to be tested by the test cases. The items are

given in following:

* Software Requirements Specifications
* Software Design Documentation

## 2.2. Features To Be Tested

The features below will be tested in test stages:

* Use cases
* Functionalities
* Interfaces
* Content
* Usability

## 2.3. Features Not To Be Tested

Security, accessibility will not be tested. All features to be tested will be tested

according to their priorities. Therefore if time does not allow, some low priority test cases

may be skipped.

## 2.4. Details

## 2.4.1. Use Case: Select Scenario Test Case

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| The user has executed the program, select scenario window should be opened and user should be able to select the scenario for the program in this use case. | | | | | | | | | | |
| SS-01 | | Select a valid scenario | | Functional Test | | Enter a valid scenario text file which contains necessary information about display area and robot in right order. | 1) Select a valid scenario which contains coordinates of obstacles. | | | Simulation starts with obstacles specified in the scenario text file. |
| 2) Select a valid scenario which does not contain coordinates of obstacles. | | | Simulation starts with no obstacles. |
| SS-02 | | Select an invalid scenario | | Functional Test | | Enter an invalid scenario text file. | 1) Enter an empty scenario file or a scenario file which does not contain the necessary information or which contains the necessary information in wrong order. | | | Simulation does not start an error message displayed. |

## 2.4.2. Use Case: Create Obstacles Test Case

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| The user has executed the program and after selecting the scenario, user should be able to create obstacles on the display screen manually. | | | | | | | | | | |
| CO-01 | | Create Obstacles | | Functional Test | | By left-clicking the mouse, the user is able to create as many obstacles as needed. | 1) Creating obstacles on display screen with no obstacles. | | | The obstacles are seen on the screen right after the left-clicking on the clicked coordinate. |
| 2) Creating obstacles on display screen with obstacles. | | | The obstacles are seen on the screen right after the left-clicking on the clicked coordinate. |
| 3) Creating obstacles on an existing obstacle. | | | A new obstacle is added onto the screen by ignoring the existing one. |

## 2.4.3. Use Case: Control the Robot Test Case

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| The user has executed the program and selected the scenario , the user should be able to control the robot with arrow keys. | | | | | | | | | | |
| CR-01 | | Right-Left arrow keys testing | | Functional Test | | By pressing the right-left arrow keys, the robot changes its direction accordingly. | 1)Pressing the right arrow key | | | The robot changes its direction to the right by 5 degree per second. |
| 2) Pressing the left arrow key | | | The robot changes its direction to the left by 5 degree per second. |
| CR-02 | | Up-Down arrow keys testing | | Functional Test | | By pressing the up-down arrow keys, the robot changes its speed accordingly. | 1) Pressing the up arrow key | | | The robot speeds up by 5 units per second. |
| 2) Pressing the down arrow key | | | The robot slows down by 5 units per second. |

## 2.4.4. Use Case: Pause Test Case

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| The user has been controlling the robot, it should stop its movement when user pressed ‘2’. | | | | | | | | | | |
| P-01 | | Pause testing | | Functional Test | | After pressing the ‘2’ button, the robot stops and its rotational and translational speed should be equal to zero. | 1) Pressing the ‘2’ button | | | The robot is paused and it waits and saves speed its last speed. |

## 2.4.4. Use Case: Replay Test Case

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| The user has been controlling the robot, it should replay the last five seconds of the simulation. | | | | | | | | | | |
| REP-01 | | Replay  testing | | Functional Test | | After pressing the ‘3’ button, the last five seconds of the simulation is displayed on the screen. | 1) Pressing the ‘3’ button | | | If the robot’s simulation time exceeds five seconds, the replay operation causes the last five seconds of the robot to be displayed on the display area. If the robot’s simulation time does not exceed five seconds, it replays the simulation in passed time interval. |

## 2.4.5. Use Case: Resume Test Case

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| User can resume from its last position by pressing the ‘1’ button after pausing. | | | | | | | | | | |
| RES-01 | | Resume  testing | | Functional Test | | After pausing the simulation of the robot, the robot simulation continues from where it is left after pressing the ‘1’ button. | 1) Pressing the ‘1’ button | | | The robot continues to play the simulation and it restores the last speed of the robot before being paused. |

## 2.4.5. ‘P’ and ‘Q’ Buttons Test Cases

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | | Test Type | | Test Case Description | | | Test Steps | | |
| Step | Expected | |
| While program execution is continuing up, if ‘P’ button is pressed, the user should have the ability to see the number of obstacles and all of their positions with their coordinate information.  While program execution is continuing up, if ‘Q’ button is pressed, the user should have the ability to exit the program. | | | | | | | | | | |
| PB-01 | | ‘P’ button testing | | Functional Test | | By pressing the ‘P’ button, the number of obstacles and the positions of all obstacles are printed. | 1) Pressing the ‘P’ button. | | | The number of obstacles and coordinates of all of them are printed on the output window. |
| QB-02 | | ‘Q’ button  testing | | Functional Test | | By pressing the ‘Q’ button, the program execution is quitted, the display screen should be shut down. | 1) Pressing the ‘Q’ button. | | | The program exits. |

## 2.5. Environmental Needs

In order to be able to test the system, the needed hardware and software components are

given in the following subtitles:

## 2.5.1. Hardware

Hardware having JRE (Java Runtime Environment) is required in order to be

able to run the code.

## 2.5.2. Software

In order to be able to compile the code, JDK 6 compiler or its higher versions are required.

As developer tool, version 7.0.1 of NetBeans IDE is required.