TamiasMini2D: System Verification and Validation Plan

Oluwaseun Owojaiye

October 23, 2018

1 Revision History

Date	Version	Notes
2018-10-15	1.0	Initial draft

2 Symbols, Abbreviations and Acronyms

symbol	description
IM	Instance Model
R	Requirement
T	Test
2D	Two-dimensional
SRS	System Requirement Specification

Contents

List of Tables

List of Figures

This document provides a high-level verification and validation plan for TamiasMini2D - a 2D rigid body physics library. This document is based on the System Requirement Specification Document(SRS) located in the following project repository link: https://github.com/smiths/caseStudies/tree/master/CaseStudies/gamephys. It discusses the verification and validation requirements for TM2D, and describes the test strategy and methods that will be used to evaluate the software. The verification and validation of the software utilizes review, analysis, and testing method to determine whether a software product complies with the specifed requirements. These requirements include both functional and non-functional.

3 General Information

3.1 Summary

The software being tested is TamiasMini2D. It is a 2D rigid body physics library designed to simulate the interaction between rigid bodies. Since physics libraries are an important part of video game development, game developers will be able to make use of this library in their products.

3.2 Objectives

The purpose of verification and validation activities are to find bugs and defects in the TM2D physics library software and also to determine if it has met all the required functionality. It is also to verify that software meets the required standard and that the end product conforms with the software requirements based on the SRS. The objectives of VnV activities for TM2D are to:

- Build confidence in software correctness and performance.
- Verify the degree maintainability of the software. based of efficiency by which this product can be enhanced, modified and reused.
- Verify and demonstrate the ease of use and learning of the software.

3.3 References

1. https://github.com/smiths/caseStudies/blob/GamePhy_Olu/CaseStudies/gamephys/docs/SRS/GamePhysicsSRS.pdf

4 Plan

4.1 Verification and Validation Team

The verification and validation team consists of a one member team: Olu Owojaiye

4.2 SRS Verification Plan

The SRS for the project will be reviewed by Dr. Smith and coursemates and feedback will be provided.Some SRS feedback for this project have been provided and addressed using github issue tracker. Also once the software has been implemented, the SRS will be reviewed to ensure that software has met all the specified requirements in SRS and more feedback will be provided via github.

4.3 Design Verification Plan

To ensure that the Design Specification has been properly specified and meets software requirements, Dr. Smith and my coursemates will be verifying the software design.The Module Guide and Module Interface Specification will contain information about the software design. Feedback is expected to be rovided by reviewers via github issue tracker.

4.4 Implementation Verification Plan

The implementation of TM2D will involve inspection of the software to ensure that all the required features have been implemented successfully and are functional. Once the development activities are completed, Dr Smith and my some of CAS761 coursemates will perform the implementation verification activities. The software will be installed by the testers and system test cases specified in Section 5 will be run. Reviewers are expected to verify both functional and non-functional requirements specified below. Exploratory testing should also be performed by testers. Any implementation verification issues will be reported and tracked via github issue tracker and these issues will be resolved in order of severity by myself. After the issues raised have been fixed, they will be sent back to reviewrs for re-verification.

4.5 Software Validation Plan

There is currently no software validation plan for TM2D.

5 System Test Description

5.1 Tests for Functional Requirements

5.1.1 Area of Testing1

Title for Test

1. test-id1

   Control: Manual versus Automatic

   Initial State:

   Input:

   Output:

   How test will be performed:

2. test-id2

   Control: Manual versus Automatic

   Initial State:

   Input:

   Output:

   How test will be performed:

5.1.2 Area of Testing2

...

5.2 Tests for Nonfunctional Requirements

5.2.1 Area of Testing1

Title for Test

1. test-id1

   Type:

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

5.2.2 Area of Testing2

...

5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

6 Static Verification Techniques

[In this section give the details of any plans for static verification of the implementation. Potential techniques include code walkthroughs, code inspection, static analyzers, etc. —SS]

References

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

[This is a section that would be appropriate for some projects. —SS]