Verification and Validation Plan for Open Source Physics Game Library

Alex Halliwushka

June 14, 2015

Contents

1 General Informationl

The following section provides an overview of the Verification and Validation (V&V) Plan for an open source physics library for video games. This section explains the purpose of this document, the scope of the system, common definitions, acronyms and abbreviations that are used in the document, and an overview of the following sections

1.1 Purpose

The main purpose of this document is to describe the verification and validation process that will be used to test an open source physics game library. This document is indented to be used as a reference for all future testing and will be used to increase confidence in the software implementation.

This document will be used as a starting point for the verification and validation report. The test cases presented within this document will be executed and the output will be analyzed to determine if the software is implemented correctly.

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

symbol	description
QA	Quality assurance
SRS	Software requirements specification
V&V	Verification and validation
V&VP	Verification and validation plan
V&VR	Verification and validation report

1.4 Overview of Document

The following sections provide more detail about the V&V of an open source 2D phyiscs rigid body library. Information about the testing process is provided, and the software specifications that were discussed in the SRS document are stated. The evaluation process that will be followed during testing is outlined, and test cases for both the system testing and unit testing are provided

2 Plan

This section provides a description of the software that is being tested, the team that will perform the testing, the milestones for the testing phase, and the budget allocated to the testing.

2.1 Software Description

The software being tested is an open source 2D rigid body physics library used for games. Given the size, shape and location from the user, the software constructs 2D rigid bodies and simulates how the rigid bodies react to forces and how they interact with one another.

2.2 Test Team

The team that will execute the test cases, write and review the V&VR consist of:

- Alex Halliwushka.
- Dr. Spencer Smith
- Nolan Driessen

2.3 Milestones

2.3.1 Location

The location that the testing will be performed is Hamilton Ontario. The institution that will be performing the testing is McMaster University.

2.3.2 Dates and Deadlines

Test Case:
The creation of the test cases for both system testing and unit testing is scheduled to begin June 1st 2015. The deadline for the creation of the test cases is June 15th 2015.

Test Case Implementation:
Implementing code for the automation of the unit testing is scheduled to being June 15th 2015. The implementation period is expected to last approximately two weeks and has a deadline of June 30th 2015.

Verification and Validation Report:
The writting of the V&VR is scheduled to begin July 1st 2015 and end on July 15th 2015.

2.4 Budget

The budget for the testing of this system is being funded by McMaster University and NSERC

3 Software Specification

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

3.1 Functional Requirements

- Input the initial mass, velocities, positions, orientations, angular velocities, and constraints of objects
- Verify that the inputs satisfy the requried physical constraints
- Determine the position and velocities over a period of time of the 2D rigid bodies acted upon by gravity
- Determine the position and velocities over a period of time of 2D rigid bodies that have undergone a collision
- Determine the orientation and angular velocities over a period of time of the 2D rigid bodies
- Determine the position and velocities over a period of time of 2D rigid bodies with constraints or joints between them

3.2 Nonfunctional Requirements

Games are very resource intensive, so performance is a high priority. Other non-functional requirements that are a priority are: correctness, understandability, portability, reliability, and maintainability.

4 Evaluation

This section first presents the methods and constraints that are to be used during the evaluation process. This is followed by how the data obtained by the testing will be evaluated, which includes: how the data will be recorded, how to move from one test to the next, and how to determine if the test was successful.

4.1 Methods and Constraints

4.1.1 Methodology

The testing of the physics game library will be seperated into two sections: system testing and unit testing.

The system testing will be done manually, the tester will set up the initial conditions as described in the test cases and compare the actual results of the test to the expected results. If the results match then the test passed, otherwise the test failed.

The unit testing will be automated. The tester will implement the unit tests into the code using a unit testing framework. Once the unit tests are implemented, the software will be run and any incorrect results will be outputted by the system

4.1.2 Extent of Testing

The extent of testing that will be employed is extensive testing. The unit test cases below provide complete code coverage and will increase confidence in the verification of the software. The system test cases increase confidence in the validation of the system

4.1.3 Test Tools

A unit testing framework will be used to implement the unit test cases and run them automatically.

4.1.4 Testing Constraints

There are currently no anticipated limitations on the testing

4.2 Data Evaluation

4.2.1 Data Recording

After each test is run the results of the test should be recorded in the following format:
Test ID:
Input:
Expected Output:
Actual Output:
Result:

4.2.2 Test Progression

For the system test cases: Follow the preperation instructions given for the test case to get the system initialized correctly. Follow the procedures given for the test case and use the inputs provided. Run the test and record the results, record any dsicrepancies between the actual output and the expected output. Move on to the next test case and repeat the process again.

4.2.3 Testing Criteria

The actual results of each test will be evaluated against the expected results to see if the software is working as intended.

4.2.4 Testing Data Reduction

The results of the test data will be evaluated on a PASS/FAIL basis. If the actual results match the expected results the test will be considered a PASS, otherwise the test is considered a FAIL.

5 System Test Description

5.1 Gravity with no initial velocity

5.1.1 Means of Control

Manual

5.1.2 Input

Use the input stated in 7.1.1

5.1.3 Expected Output

$v_{\mathrm{x}}(t) = 0$, $v_{\mathrm{y}}(t) = -9.8t$
$p_{\mathrm{x}}(t) = 0$, $p_{\mathrm{y}}(t) = 50 - 4.9t^2$

5.1.4 Procedure

Record the position and velocity of the object as it falls. Determine if the position and velocity follow the equations in the expected output.

5.1.5 Preperation

Create an object and give it the initial values indicated in 7.1.1. Output the object's position and velocity at 0.5 s intervals.

5.2 Gravity with initial velocity in y

5.2.1 Means of Control

Manual

5.2.2 Input

Use the input stated in 7.1.2

5.2.3 Expected Output

$v_{\mathrm{x}}(t) = 0$, $v_{\mathrm{y}}(t) = 2 - 9.8t$
$p_{\mathrm{x}}(t) = 0$, $p_{\mathrm{y}}(t) = 50 + 2t - 4.9t^2$

5.2.4 Procedure

Record the position and velocity of the object as it falls. Determine if the position and velocity follow the equations in the expected output.

5.2.5 Preperation

Create an object and give it the initial values indicated in 7.1.2. Output the object's position and velocity at 0.5 s intervals.

5.3 Gravity with initial velocity in x

5.3.1 Means of Control

Manual

5.3.2 Input

Use the input stated in 7.1.3

5.3.3 Expected Output

$v_{\mathrm{x}}(t) = 5.75$, $v_{\mathrm{y}}(t) = -9.8t$
$p_{\mathrm{x}}(t) = 5.75t$, $p_{\mathrm{y}}(t) = 50 - 4.9t^2$

5.3.4 Procedure

Record the position and velocity of the object as it falls. Determine if the position and velocity follow the equations in the expected output.

5.3.5 Preperation

Create an object and give it the initial values indicated in 7.1.3. Output the object's position and velocity at 0.5 s intervals.

5.4 Gravity with initial velocity in x and y

5.4.1 Means of Control

Manual

5.4.2 Input

Use the input stated in 7.1.4

5.4.3 Expected Output

$v_{\mathrm{x}}(t) = 15.9$, $v_{\mathrm{y}}(t) = 4.8 - 9.8t$
$p_{\mathrm{x}}(t) = 15.9t$, $p_{\mathrm{y}}(t) = 50 + 4.8t - 4.9t^2$

5.4.4 Procedure

Record the position and velocity of the object as it falls. Determine if the position and velocity follow the equations in the expected output.

5.4.5 Preperation

Create an object and give it the initial values indicated in 7.1.4. Output the object's position and velocity at 0.5 s intervals.

5.5 Parabolic motion with damping

5.5.1 Means of Control

Manual

5.5.2 Input

Use the input stated in 7.1.5

5.5.3 Expected Output

$v_{\mathrm{x}}(t) = 6.32 * (0.95)^t$, $v_{\mathrm{y}}(t) = -1.77 * (0.95)^t - 9.8t$
$p_{\mathrm{x}}(t) = 6.32t * (0.95)^t$, $p_{\mathrm{y}}(t) = 50 - 1.77t * (0.95)^t - 4.9t^2$

5.5.4 Procedure

Record the position and velocity of the object as it falls. Determine if the position and velocity follow the equations in the expected output.

5.5.5 Preperation

Create an object and give it the initial values indicated in 7.1.5. Output the object's position and velocity at 0.5 s intervals.

5.6 Conservation of Momentum

5.6.1 Means of Control

Manual

5.6.2 Input

Use the input stated in 7.2.1

5.6.3 Expected Output

$m_1\mathbf{v}_{\text{i1}} + \mathrm{m}_2\mathbf{v}_{\text{i2}} = \mathrm{m}_1\mathbf{v}_{\text{f1}} + \mathrm{m}_2\mathbf{v}_{\text{f2}}$
$C_{\text{R}} = \frac{\mathbf{v}_{\text{f2}}-\mathbf{v}_{\text{f1}}}{\mathbf{v}_{\text{i1}}-\mathbf{v}_{\text{i2}}}$

rearranging and substituting the above two equations gives the velocity right after the collision:
$v_{\text{f1}} = \frac{m_1 v_{\text{i1}}+m_2 v_{\text{i2}}+m_2(C_{\text{R}}(v_{\text{i2}}-v_{\text{i1}}))}{m_1+m_2}$ $v_{\text{f2}} = \frac{m_1 v_{\text{i1}}+m_2 v_{\text{i2}}+m_1(C_{\text{R}}(v_{\text{i1}}-v_{\text{i2}}))}{m_1+m_2}$

5.6.4 Procedure

Record the velocity of the objects right after they collide. Determine if the velocities are equal to the values in the expected output.

5.6.5 Preperation

Create two objects and give them the initial values indicated in 7.2.1. Output the objects' velocities right after the collision

5.7 Perfectly Elastic Collision

5.7.1 Means of Control

Manual

5.7.2 Input

Use the input stated in 7.2.2

5.7.3 Expected Output

$m_1\mathbf{v}_{\text{i1}} + \mathrm{m}_2\mathbf{v}_{\text{i2}} = \mathrm{m}_1\mathbf{v}_{\text{f1}} + \mathrm{m}_2\mathbf{v}_{\text{f2}}$
$\frac{1}{2}m_1 v_{\text{i1}}^2 + \frac{1}{2}m_2 v_{\text{i2}}^2 = \frac{1}{2}m_1 v_{\text{f1}}^2 + \frac{1}{2}m_2 v_{\text{f2}}^2$

5.7.4 Procedure

Record the velocity of the objects right after they collide. Determine if the velocities are equal to the values in the expected output.

5.7.5 Preperation

Create two objects and give them the initial values indicated in 7.2.2. Output the objects' velocities right after the collision

5.8 Rotation with no initial angular velocity

5.8.1 Means of Control

Manual

5.8.2 Input

Use the input stated in 7.3.1

5.8.3 Expected Output

$\omega(t) = \frac{\pi}{10}t$, $\phi(t) = \frac{\pi}{4} + \frac{\pi}{20}t^2$

5.8.4 Procedure

Record the orientation and the angular velcoity of the object as it rotates. Determine if the orientation and the angular velocitiy follow the equations in the expected output.

5.8.5 Preperation

Create an objects and give them the initial values indicated in 7.3.1. Output the objects' velocities right after the collision

5.9 Rotation with angular velocity

5.9.1 Means of Control

Manual

5.9.2 Input

Use the input stated in 7.3.2

5.9.3 Expected Output

$\omega(t) = \frac{\pi}{3} - \frac{\pi}{12}t$, $\phi(t) = \frac{\pi}{4} + \frac{\pi}{3}t + \frac{\pi}{24}t^2$

5.9.4 Procedure

Record the orientation and the angular velcoity of the object as it rotates. Determine if the orientation and the angular velocitiy follow the equations in the expected output.

5.9.5 Preperation

Create an objects and give them the initial values indicated in 7.3.2. Output the objects' velocities right after the collision

5.10 Rotation with damping

5.10.1 Means of Control

Manual

5.10.2 Input

Use the input stated in 7.3.3

5.10.3 Expected Output

$\omega(t) = -\frac{\pi}{6}(0.77)^t - \frac{\pi}{8}t$, $\phi(t) = \frac{\pi}{4} - \frac{\pi}{6}t(0.77)^t + \frac{\pi}{16}t^2$

5.10.4 Procedure

Record the orientation and the angular velcoity of the object as it rotates. Determine if the orientation and the angular velocitiy follow the equations in the expected output.

5.10.5 Preperation

Create an objects and give them the initial values indicated in 7.3.3. Output the objects' velocities right after the collision

6 Unit Test Description

6.1 Module Information

6.1.1 Module Inputs

6.1.2 Module Outputs

6.1.3 Related Modules

6.2 Test Data

6.2.1 Inputs

6.2.2 Expected Outputs

7 Appendix

7.1 Test for Gravity

The following test cases use:

$$\mathbf{p}_{\mathrm{i}} = \begin{bmatrix} 0 \\ 50 \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} 0 \\ -9.8 \end{bmatrix}$$

t = 0, 0.5, 1, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0

7.1.1 No Initial Velocity

Use the values in 7.1 and an initial velocity of:

$$\mathbf{v}_{\mathrm{i}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

7.1.2 Initial Velocity in y

Use the values in 7.1 and an initial velocity of:

$$\mathbf{v}_{\mathrm{i}} = \begin{bmatrix} 0 \\ 2.3 \end{bmatrix}$$

7.1.3 Initial Velocity in x

Use the values in 7.1 and an initial velocity of:

$$\mathbf{v}_\mathrm{i} = \begin{bmatrix} 5.75 \\ 0 \end{bmatrix}$$

7.1.4 Initial Velocity in both x and y

Use the values in 7.1 and an initial velocity of:

$$\mathbf{v}_\mathrm{i} = \begin{bmatrix} 15.9 \\ 4.8 \end{bmatrix}$$

7.1.5 Parabolic Motion with Damping

Use the values in 7.1, an initial velocity of:

$$\mathbf{v}_\mathrm{i} = \begin{bmatrix} 6.32 \\ -1.77 \end{bmatrix}$$

and a damping coeffiecient: $\zeta = 0.95$

7.2 Test for Collision

The following test cases use:

$$\mathbf{p}_\mathrm{i1} = \begin{bmatrix} 100.5 \\ 51.75 \end{bmatrix}$$

$$\mathbf{p}_\mathrm{i2} = \begin{bmatrix} 105 \\ 50 \end{bmatrix}$$

$$\mathbf{v}_\mathrm{i1} = \begin{bmatrix} 3.1 \\ 2.5 \end{bmatrix}$$

$$\mathbf{v}_\mathrm{i2} = \begin{bmatrix} -5.4 \\ 1.2 \end{bmatrix}$$

7.2.1 Conservation of momentum

Use the values in 7.2, and $m_1 = 5, m_2 = 15.4, CR = 0.5$

7.2.2 Perfectly elastic collision

Use the values in 7.2, and $m_1 = 7.8, m_2 = 20.1, CR = 1$

7.3 Test for rotation

The folllowing test cases use:

$$\mathbf{p}_{\mathrm{i1}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\phi = \frac{\pi}{4}$$

7.3.1 Rotation no initial angular velocity

Use the values in 7.3, and $\omega = 0$ and $\alpha = \frac{\pi}{10}$

7.3.2 Rotation with angular velocity

Use the values in 7.3, and $\omega = \frac{\pi}{3}$ and $\alpha = -\frac{\pi}{12}$

7.3.3 Rotation with damping

Use the values in 7.3, and $\omega = -\frac{\pi}{6}$, $\alpha = \frac{\pi}{8}$ and $\zeta = 0.77$