

Test Report: Tamias2DA 2D Rigid Body Game Physics Library

Oluwaseun Owojaiye

December 26, 2018

1 Revision History

Date	Version	Notes
Dec. 12, 2018	1.0	Initial draft
Dec. 24, 2018	1.1	Updates throughout document

2 Symbols, Abbreviations and Acronyms

The symbols, abbreviations, and acronyms used in this document include those defined in the table below, as well as any defined in the tables found in Section 2 of the Software Requirements Specification (SRS) at https://github.com/smiths/caseStudies/blob/gamephy_finaldoc/CaseStudies/gamephys/docs/SRS/GamePhysicsSRS.pdf

symbol	description
MIS	Module Interface Specification
MG	Module Guide
TC	Test Case
VnV	Verification and Validation

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Functional Requirements Evaluation	1
4	Nonfunctional Requirements Evaluation	2
4.1	Performance	2
4.2	Correctness	2
4.3	Usability	4
5	Comparison to Existing Implementation	4
6	Unit Testing	5
7	Changes Due to Testing	5
8	Automated Testing	5
9	Trace to Requirements	5
10	Trace to Modules	6
11	Code Coverage Metrics	6

List of Tables

1	Traceability Between Test Cases and Requirements	5
2	Traceability Between Test Cases and Modules	6

List of Figures

1	Output Performance of Tamias2D	2
2	Output Correctness of Tamias2D	3
3	Change in position for projectile motion test Tamias2D	3
4	Change in velocity for projectile motion test Tamias2D	4

This document outlines the results of testing for Tamias2D. Evaluation of Functional and Non Functional Requirements testing are reported in Sections 2 and 3 respectively. This document is related to the System VnV plan and should be referenced when reviewing this document for the details of System VnV test cases. System VnV Plan document can be found at https://github.com/smiths/caseStudies/blob/gamephy_finaldoc/CaseStudies/gamephys/docs/VnVPlan/SystVnVPlan/SystVnVPlan.pdf. I did some comparison with an existing library Pymunk, although just one or two scenarios were compared and my findings are reported in section 5. All changes made as a result of changes to requirement(if any), bug fixes and error detection were discussed in section 7. Section 8 describes the test framework used for test automation. Section 9 and 10 details the traceability between test cases, requirements and modules. were implemented with an automated testing framework.

3 Functional Requirements Evaluation

All functional requirement test cases executed as a part of the System VnV can be found in section 5.1. TC1-TC7 tests for the movement of a body when force or no force is applied and the history of the position and velocity of rigid bodies over time is recorded. This set of test cases cover the functional requirements to create a space for all rigid body simulation(R1), the bodies were initialized with initial values and no force or force was applied on the rigid bodies, the inputs set for the parameters were also verified (R4)and the position and velocities over a period of time was computed upon a force acting on a body (R5).

TC8 - TC9 verified the rotational motion of a body when a rotational force known as 'torque' acts on the body. The angular orientation and angular velocities are computed over a period of time and this satisfies R6.

T10 - T11 verifies the collision of rigid bodies, in collision test, the software checks for collision, i.e if 2 objects are intersecting and occupying the same position in space(R7) and the collision response is also computed by the software,Coefficient of restitution is applied in collision of 2 objects(R3) and the position and velocities over a period of time is also computed(R8). All the above test cases passed.

4 Nonfunctional Requirements Evaluation

4.1 Performance

The performance of Tamias2D was tested by stress test, using TC9 of the System VnV Plan, 50 dynamic rigid bodies were added to space and we compute the velocity and position of each body over 4 seconds. The system was able to return all the position and velocity output of all 50 bodies without crashing or slowing down. It took 38.35 secs to return all values. This result will be compared to Pymunk in a future phase. The graph below illustrates the output of performance test. [I had a hard time capturing the legend in one screen —OO]

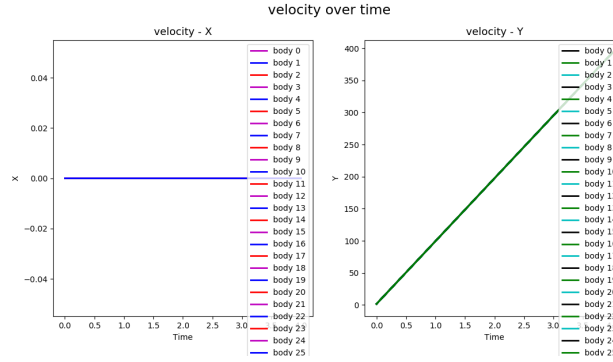


Figure 1: Output Performance of Tamias2D

4.2 Correctness

The NFR of correctness was measured by comparing actual simulation results of the velocity and position history of Tamias2D with the expected result and the relative error is shown in the image below. TC2 results are displayed. The results give more confidence to the correctness of Tamias2D

Figure 3 shows the output result of a simulation of projectile motion. The object was launched at an angle of 60° . Figures 3 and 4 respectively show the position-time graph and the velocity-time graph. With a trend of the velocity decreasing on the y-axis and constant in the x-axis and change

	Expected		Actual		Relative error	Relative error
Time(secs)	Position(x, y)	Velocity(x,y)	Position(x, y)	Velocity(x,y)	Position	Velocity
1	(22.84, 20.00)	(10.50, 0.0)	(22.8403, 20.0003)	(10.501, 0.0)	0.0001, 0	0.0001,0.0003
2	(31.02, 20.0)	(20.83, 0.0)	(31.0203, 20.0003)	(20.831, 0.0)	0.0001, 0	0.0001,0.0003
3	(44.54, 20.00)	(31.16, 0.0)	(44.5401, 20.0002)	(31.163, 0.0)	0.0003, 0	0.0003,0.0002
4	(60.66, 20.00)	(40.16, 0.0)	(60.6601, 20.0002)	(40.163, 0.0)	0.0003, 0	0.0003,0.0002

Figure 2: Output Correctness of Tamias2D

in position forming a projectile pattern, I deduce that the computation is correct as the expected results matches the actual results.

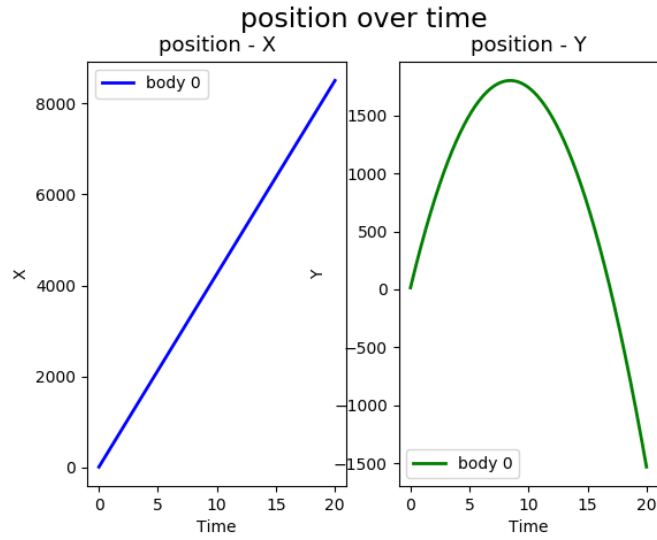


Figure 3: Change in position for projectile motion test Tamias2D

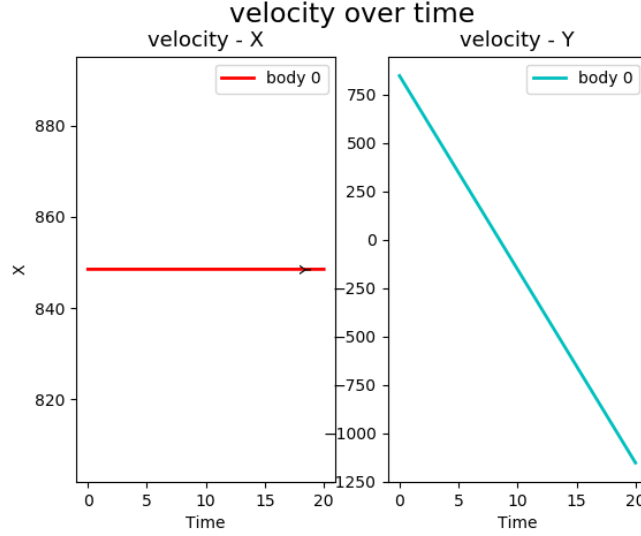


Figure 4: Change in velocity for projectile motion test Tamias2D

4.3 Usability

Usability was evaluated by how long it took a user to create a small program as per NFR TC12. After the software was downloaded, the user which was me, I was able to create a program to simulate the movement of 2 rigid bodies in space, computing their velocity and position over time in about 9.22 minutes, so an intermediate to experienced programmer especially one who is familiar with using game physics library will be able to do this in about 20 to 35 minutes and a programmer new to game development might take no less than 1 hour to create the small program described above. This proves the usability of Tamias2D

5 Comparison to Existing Implementation

Tamias2D was compared with Pymunk for TC1 and TC2, please refer to tables 1 and 2 for results, the result shows that Tamias2D output result of TC1 and TC2 matches the output result of Pymunk. More testing will be done in a later phase of this project. Pymunk will continually be used as a benchmark for Tamias2D. This section will not be appropriate for every

project.

6 Unit Testing

Please refer to the Unit VnV document located at https://github.com/smiths/caseStudies/blob/gamephy_finaldoc/CaseStudies/gamephys/docs/VnVReport/UnitVnVReport/UnitVnVReport.pdf for a detailed report on Unit VnV testing.

7 Changes Due to Testing

No changes were made due to testing. All tests were successfully executed.

8 Automated Testing

Pytest framework was used for automated testing. Please refer to test file located at

9 Trace to Requirements

The table below shows the traceability between the test cases in the System V&V Plan and the requirements

T	Requirements
TC1-TC7	R1, R2, R4, R5
TC8-TC9	R1, R2, R4, R5
TC10-TC11	R1, R3, R4, R7, R8
TC12	R1, R3, R4, R7, R8
TC13	R1, R3, R4, R7, R8
TC14	R1, R3, R4, R7, R8

Table 1: Traceability Between Test Cases and Requirements

10 Trace to Modules

T	Modules
TC1-TC7	Body, Space, Vector
TC8-TC9	Body, Space, Vector
TC10-TC11	Body, Space, Vector, Shape, Collision Solver

Table 2: Traceability Between Test Cases and Modules

11 Code Coverage Metrics

Not Applicable

References