

Software Requirements Specification for Chipmunk2D

Alex Halliwushka and Luthfi Mawarid

May 3, 2019

Contents

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

The unit system used throughout is SI (Système International d'Unités). In addition to the basic units, several derived units are also used. For each unit, the table lists the symbol, a description and the SI name.

Symbol	Description
J	energy (joule)
kg	mass (kilogram)
m	length (metre)
N	force (newton)
rad	angle (radian)
s	time (second)

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. Throughout the document, symbols in bold will represent vectors, and scalars otherwise. The symbols are listed in alphabetical order.

Symbol	Description	Units
a	Acceleration	$\frac{m}{s^2}$
a_i	The I-Th Body's Acceleration	$\frac{m}{s^2}$
a(t)	Linear Acceleration	$\frac{m}{s^2}$
C_R	Coefficient of restitution	—
F	Force	N
F₁	Force exerted by the first body (on another body)	N
F₂	Force exerted by the second body (on another body)	N
F_i	Force Applied to the I-Th Body at Time T	N
G	Gravitational constant	$\frac{m^3}{(kg s^2)}$
g	Gravitational acceleration	$\frac{m}{s^2}$
I	Moment of inertia	kgm ²
I_A	Moment of Inertia Of Rigid Body A	kgm ²

Symbol	Description	Units
\mathbf{I}_B	Moment of Inertia Of Rigid Body B	kgm^2
j	Impulse (scalar)	Ns
KE	Kinetic energy	J
L	Length	m
M	Total Mass of the Rigid Body	kg
m	Mass	kg
m_1	Mass of the first body	kg
m_2	Mass of the second body	kg
m_A	Mass Of Rigid Body A	kg
m_B	Mass Of Rigid Body B	kg
m_j	Mass Of the J-Th Particle	kg
\mathbf{n}	Collision Normal Vector	m
\mathbf{p}	Position	m
\mathbf{p}_{CM}	Center of Mass	m
\mathbf{p}_j	Position Vector of the J-Th Particle	m
\mathbf{r}	Displacement	m
\mathbf{r}_{OB}	Displacement vector between the origin and point B	m
$\mathbf{r}(t)$	Linear Displacement	m
\mathbf{r}	Displacement unit vector	m
t	Time	s
t_c	Denotes the time at collision	s
\mathbf{v}_f^{AB}	Final Relative Velocity Between Rigid Bodies of A and B	$\frac{\text{m}}{\text{s}}$
\mathbf{v}_i^{AB}	Initial Relative Velocity Between Rigid Bodies of A and B	$\frac{\text{m}}{\text{s}}$
\mathbf{v}^{AP}	Velocity Of the Point of Collision P in Body A	$\frac{\text{m}}{\text{s}}$
\mathbf{v}^{BP}	Velocity Of the Point of Collision P in Body B	$\frac{\text{m}}{\text{s}}$
\mathbf{v}_A	Velocity At Point A	$\frac{\text{m}}{\text{s}}$
\mathbf{v}_B	Velocity At Point B	$\frac{\text{m}}{\text{s}}$
\mathbf{v}_i	Velocity Of the I-Th Body's Velocity	$\frac{\text{m}}{\text{s}}$
\mathbf{v}_O	Velocity At Point Origin	$\frac{\text{m}}{\text{s}}$
\mathbf{v}	Velocity	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}(t)$	Linear Velocity	$\frac{\text{m}}{\text{s}}$
$\ \mathbf{n}\ $	Length of the Normal Vector	m
$\ \mathbf{r}_{AP} * \mathbf{n}\ $	Length of the Perpendicular Vector To the Contact Displacement Vector of Rigid Body A	m

Symbol	Description	Units
$ \mathbf{r}_{BP} * \mathbf{n} $	Length of the Perpendicular Vector To the Contact Displacement Vector of Rigid Body B	m
$ \mathbf{r} $	Euclidean norm of the displacement	m
$ \mathbf{r} ^2$	Squared distance	m^2
\pm	Angular Acceleration	$\frac{\text{rad}}{\text{s}^2}$
	Angular Displacement	rad
	Torque	Nm
i	Torque applied to the i-th body	Nm
	Angular Velocity	$\frac{\text{rad}}{\text{s}}$
	Orientation	rad

1.3 Abbreviations and Acronyms

	Abbreviation	Full Form
2D		two-dimensional
A		Assumption
CM		Centre of Mass
Chipmunk2D		Chipmunk2D game physics library
DD		Data Definition
GD		General Definition
GS		Goal Statement
IM		Instance Model
LC		Likely Change
ODE		Ordinary Differential Equation
R		Requirement
SRS		Software Requirements Specification
T		Theoretical Model
UC		Unlikely Change
Uncert.		Typical Uncertainty

2 Introduction

Due to the rising cost of developing video games, developers are looking for ways to save time and money for their projects. Using an open source physics library that is reliable and

free will cut down development costs and lead to better quality products. The following section provides an overview of the Software Requirements Specification (SRS) for Chipmunk2D. This section explains the purpose of this document, the scope of the system, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

This document describes the modeling of an open source 2D rigid body physics library used for games. The theoretical models and goal statements used in Chipmunk2D are provided. This document is intended to be used as a reference to provide all necessary information to understand and verify the model. This document will be used as a starting point for subsequent development phases, including writing the design specification and the software verification and validation plan. The design document will show how the requirements are to be realized, including decisions on the numerical algorithms and programming environment. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation. Although the SRS fits in a series of documents that follow the so-called waterfall model, the actual development process is not constrained in any way. Even when the waterfall model is not followed, as Parnas and Clements point out [parnasClements1986], the most logical way to present the documentation is still to “fake” a rational design process.

2.2 Scope of Requirements

The scope of the requirements includes the physical simulation of 2D rigid bodies acted on by forces. Given the appropriate inputs, Chipmunk2D simulates how these rigid bodies interact with one another.

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have an understanding of rigid body dynamics and high school calculus. The users of Chipmunk2D can have a lower level of expertise, as explained in Section: User Characteristics.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [dParnas1972] and [parnasClements1984]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the instance models in Section: Instance Models and trace back to find any additional information they require. The goal statements (Section: Goal Statements) are refined to the theoretical models, and the theoretical models (Section: Theoretical Models) to the instance models (Section: Instance Models).



Figure 1: System Context

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics, and lists the system constraints.

3.1 System Context

Fig:sysCtxDiag shows the system context. A circle represents an external entity outside the software, the user in this case. A rectangle represents the software system itself (Chipmunk2D). Arrows are used to show the data flow between the system and its environment. The interaction between the product and the user is through an application programming interface. The responsibilities of the user and the system are as follows:

- User Responsibilities
 - Provide initial conditions of the physical state of the simulation, rigid bodies present, and forces applied to them.
 - Ensure application programming interface use complies with the user guide.
 - Ensure required software assumptions (FIXME REF) are appropriate for any particular problem the software addresses.
- Chipmunk2D Responsibilities
 - Determine if the inputs and simulation state satisfy the required physical and system constraints (FIXME REF).
 - Calculate the new state of all rigid bodies within the simulation at each simulation step.
 - Provide updated physical state of all rigid bodies at the end of a simulation step.

3.2 User Characteristics

The end user of Chipmunk2D should have an understanding of first year programming concepts and an understanding of high school physics.

3.3 System Constraints

There are no system constraints.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, and definitions that are used.

4.1 Problem Description

Creating a gaming physics library is a difficult task. Games need physics libraries that simulate objects acting under various physical conditions, while simultaneously being fast and efficient enough to work in soft real-time during the game. Developing a physics library from scratch takes a long period of time and is very costly, presenting barriers of entry which make it difficult for game developers to include physics in their products. There are a few free, open source and high quality physics libraries available to be used for consumer products (Section: Off-The-Shelf Solutions). By creating a simple, lightweight, fast and portable 2D rigid body physics library, game development will be more accessible to the masses and higher quality products will be produced.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements

- Rigid body: A solid body in which deformation is neglected.
- Elasticity: Ratio of the relative velocities of two colliding objects after and before a collision.
- Centre of mass: The mean location of the distribution of mass of the object.
- Cartesian coordinate system: A coordinate system that specifies each point uniquely in a plane by a pair of numerical coordinates.
- Right-handed coordinate system: A coordinate system where the positive z-axis comes out of the screen.

4.1.2 Goal Statements

Given the inputs, the goal statements are:

GS1: Determine their new positions and velocities over a period of time.

GS2: Determine their new orientations and angular velocities over a period of time.

4.2 Solution Characteristics Specification

The instance models that govern Chipmunk2D are presented in Section: Instance Models. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the Theoretical Models Section: Theoretical Models, General Definitions Section: General Definitions, Data Definitions Section: Data Definitions, Instance Models Section: Instance Models, Likely Changes Section: Likely Changes, or Unlikely Changes Section: Unlikely Changes, in which the respective assumption is used.

objectTy: All objects are rigid bodies. DD: chalses DD: chalses DD: reVeInColl IM: transMot IM: rotMot DD: ctrOfMass DD: linVel DD: linDisp DD: linAcc DD: kEnergy DD: impulse IM: col2D TM: ChaslesThm DD: angVel DD: angDisp DD: angAccel.

objectDimension: All objects are 2D. DD: chalses IM: transMot IM: rotMot DD: ctrOfMass TM: NewtonSecLawRotMot DD: linVel DD: linDisp DD: linAcc DD: kEnergy DD: impulse IM: col2D DD: angVel DD: angDisp DD: angAccel.

ordinateSystemTy: The library uses a Cartesian coordinate system.

axesDefined: The axes are defined using right-handed coordinate system. IM: rotMot DD: impulse IM: col2D.

collisionType: All rigid bodies collisions are vertex-to-edge collisions. LC: Expanded-Collisions DD: impulse IM: col2D.

mpingInvolvement: There is no damping involved throughout the simulation and this implies that there are no friction forces. DD: chalses IM: transMot IM: rotMot DD: linVel DD: linDisp DD: linAcc LC: Include-Dampening DD: kEnergy IM: col2D DD: angVel DD: angDisp DD: angAccel.

JointsInvolvement: There are no constraints and joints involved throughout the simulation. IM: transMot IM: rotMot LC: Include-Joints-Constraints IM: col2D.

4.2.2 Theoretical Models

This section focuses on the general equations and laws that Chipmunk2D is based on.

Refname	TM:NewtonSecLawMot
Label	Newton's second law of motion
Equation	$\mathbf{F} = m\mathbf{a}$
Description	\mathbf{F} is the force (N) m is the mass (kg) \mathbf{a} is the acceleration ($\frac{\text{m}}{\text{s}^2}$)
Notes	The net force \mathbf{F} (N) on a rigid body is proportional to the acceleration \mathbf{a} ($\frac{\text{m}}{\text{s}^2}$) of the rigid body, where m (kg) denotes the mass of the rigid body as the constant of proportionality.
Source	
RefBy	

Refname	TM:NewtonThirdLawMot
Label	Newton's third law of motion
Equation	$\mathbf{F}_1 = -\mathbf{F}_2$
Description	<p>\mathbf{F}_1 is the force exerted by the first body (on another body) (N)</p> <p>\mathbf{F}_2 is the force exerted by the second body (on another body) (N)</p>
Notes	<p>Every action has an equal and opposite reaction. In other words, the force \mathbf{F}_1 (N) exerted on the second rigid body by the first is equal in magnitude and in the opposite direction to the force \mathbf{F}_2 (N) exerted on the first rigid body by the second.</p>
Source	
RefBy	

Refname	TM:UniversalGravLaw
Label	Newton's law of universal gravitation
Equation	$\mathbf{F} = G \frac{m_1 m_2}{\ \mathbf{r}\ ^2} \mathbf{r} = G \frac{m_1 m_2}{\ \mathbf{r}\ ^2} \frac{\mathbf{r}}{\ \mathbf{r}\ }$
Description	<p>\mathbf{F} is the force (N)</p> <p>G is the gravitational constant ($\frac{\text{m}^3}{(\text{kg s}^2)}$)</p> <p>$m_1$ is the mass of the first body (kg)</p> <p>m_2 is the mass of the second body (kg)</p> <p>$\ \mathbf{r}\$ is the Euclidean norm of the displacement (m)</p> <p>\mathbf{r} is the displacement unit vector (m)</p> <p>\mathbf{r} is the displacement (m)</p>
Notes	<p>Two rigid bodies in the universe attract each other with a force \mathbf{F} (N) that is directly proportional to the product of their masses, m_1 and m_2 (kg), and inversely proportional to the squared distance $\ \mathbf{r}\ ^2$ (m^2) between them. The vector \mathbf{r} (m) is the displacement between the centres of the rigid bodies and $\ \mathbf{r}\$ (m) represents the Euclidean norm of the displacement, or absolute distance between the two. \mathbf{r} denotes the displacement unit vector, equivalent to the displacement divided by the Euclidean norm of the displacement, as shown above. Finally, G is the gravitational constant ($6.673 \cdot 10^{-11}$) ($\frac{\text{m}^3}{(\text{kg s}^2)}$).</p>
Source	
RefBy	

Refname	TM:ChaslesThm
Label	Chasles' theorem
Equation	$\mathbf{v}_B = \mathbf{v}_O + \boldsymbol{\omega} \times \mathbf{r}_{OB}$
Description	<p> \mathbf{v}_B is the velocity at point B ($\frac{\text{m}}{\text{s}}$) \mathbf{v}_O is the velocity at point origin ($\frac{\text{m}}{\text{s}}$) $\boldsymbol{\omega}$ is the angular velocity ($\frac{\text{rad}}{\text{s}}$) \mathbf{r}_{OB} is the displacement vector between the origin and point B (m) </p>
Notes	<p> The linear velocity \mathbf{v}_B ($\frac{\text{m}}{\text{s}}$) of any point B in a rigid body A: objectTy is the sum of the linear velocity \mathbf{v}_O ($\frac{\text{m}}{\text{s}}$) of the rigid body at the origin (axis of rotation) and the resultant vector from the cross product of the rigid body's angular velocity ($\frac{\text{rad}}{\text{s}}$) and the displacement vector between the origin and point B, \mathbf{r}_{OB} (m). </p>
Source	
RefBy	

Refname	TM:NewtonSecLawRotMot
Label	Newton's second law for rotational motion
Equation	$\tau = \mathbf{I}\pm$
Description	<p>τ is the torque (Nm)</p> <p>\mathbf{I} is the moment of inertia (kgm^2)</p> <p>\pm is the angular acceleration ($\frac{\text{rad}}{\text{s}^2}$)</p>
Notes	The net torque (Nm) on a rigid body is proportional to its angular acceleration \pm ($\frac{\text{rad}}{\text{s}^2}$). Here, \mathbf{I} (kgm^2) denotes the moment of inertia of the rigid body. We also assume that all rigid bodies involved are two-dimensional A: objectDimension.
Source	
RefBy	

4.2.3 General Definitions

There are no general definitions.

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models.

Refname	DD:ctrOfMass
Label	Center of Mass
Symbol	\mathbf{p}_{CM}
Units	m
Equation	$\mathbf{p}_{CM} = \frac{\sum m_j \mathbf{p}_j}{M}$
Description	<p>\mathbf{p}_{CM} is the Center of Mass (m) m_j is the mass of the j-th particle (kg) \mathbf{p}_j is the position vector of the j-th particle (m) M is the total mass of the rigid body (kg)</p>
Notes	A: objectTy A: objectDimension
Source	
RefBy	IM: transMot IM: col2D.

Refname	DD:linDisp
Label	Linear Displacement
Symbol	$\mathbf{r}(t)$
Units	m
Equation	$\mathbf{r}(t) = \frac{d\mathbf{p}(t)}{dt}$
Description	<p>$\mathbf{r}(t)$ is the linear displacement (m) t is the time (s) \mathbf{p} is the position (m)</p>
Notes	A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	IM: transMot.

Refname	DD:linVel
Label	Linear Velocity
Symbol	$\mathbf{v}(t)$
Units	$\frac{\text{m}}{\text{s}}$
Equation	$\mathbf{v}(t) = \frac{d\mathbf{r}(t)}{dt}$
Description	$\mathbf{v}(t)$ is the linear velocity ($\frac{\text{m}}{\text{s}}$) t is the time (s) \mathbf{r} is the displacement (m)
Notes	A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	IM: transMot.

Refname	DD:linAcc
Label	Linear Acceleration
Symbol	$\mathbf{a}(t)$
Units	$\frac{\text{m}}{\text{s}^2}$
Equation	$\mathbf{a}(t) = \frac{d \mathbf{v}(t)}{dt}$
Description	$\mathbf{a}(t)$ is the linear acceleration ($\frac{\text{m}}{\text{s}^2}$) t is the time (s) \mathbf{v} is the velocity ($\frac{\text{m}}{\text{s}}$)
Notes	A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	IM: transMot.

Refname	DD:angDisp
Label	Angular Displacement
Symbol	
Units	rad
Equation	$= \frac{d(t)}{dt}$
Description	<p>is the angular displacement (rad)</p> <p>t is the time (s)</p> <p>is the orientation (rad)</p>
Notes	A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	IM: rotMot.

Refname	DD:angVel
Label	Angular Velocity
Symbol	
Units	$\frac{\text{rad}}{\text{s}}$
Equation	$= \frac{d(t)}{dt}$
Description	<p>is the angular velocity ($\frac{\text{rad}}{\text{s}}$)</p> <p>$t$ is the time (s)</p> <p>is the angular displacement (rad)</p>
Notes	A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	IM: rotMot.

Refname	DD:angAccel
Label	Angular Acceleration
Symbol	\pm
Units	$\frac{\text{rad}}{\text{s}^2}$
Equation	$\pm = \frac{d(t)}{dt}$
Description	\pm is the angular acceleration ($\frac{\text{rad}}{\text{s}^2}$) t is the time (s) is the angular velocity ($\frac{\text{rad}}{\text{s}}$)
Notes	A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	IM: rotMot.

Refname	DD:impulse
Label	Impulse (scalar)
Symbol	j
Units	Ns
Equation	$j = \frac{-(1 + C_R) \mathbf{v}_i^{AB} \cdot \mathbf{n}}{\left(\frac{1}{m_A} + \frac{1}{m_B}\right) \ \mathbf{n}\ ^2 + \frac{\ \mathbf{r}_{AP} * \mathbf{n}\ ^2}{\mathbf{I}_A} + \frac{\ \mathbf{r}_{BP} * \mathbf{n}\ ^2}{\mathbf{I}_B}}$
Description	<p>j is the impulse (scalar) (Ns)</p> <p>C_R is the coefficient of restitution (Unitless)</p> <p>\mathbf{v}_i^{AB} is the initial relative velocity between rigid bodies of A and B ($\frac{\text{m}}{\text{s}}$)</p> <p>$\mathbf{n}$ is the collision normal vector (m)</p> <p>m_A is the mass of rigid body A (kg)</p> <p>m_B is the mass of rigid body B (kg)</p> <p>$\ \mathbf{n}\$ is the length of the normal vector (m)</p> <p>$\ \mathbf{r}_{AP} * \mathbf{n}\$ is the length of the perpendicular vector to the contact displacement vector of rigid body A (m)</p> <p>\mathbf{I}_A is the moment of inertia of rigid body A (kgm^2)</p> <p>$\ \mathbf{r}_{BP} * \mathbf{n}\$ is the length of the perpendicular vector to the contact displacement vector of rigid body B (m)</p> <p>\mathbf{I}_B is the moment of inertia of rigid body B (kgm^2)</p>
Notes	A: objectTy A: objectDimension A: axesDefined A: collisionType
Source	
RefBy	IM: col2D.

Refname	DD:chases
Label	Velocity At Point B
Symbol	\mathbf{v}_B
Units	$\frac{\text{m}}{\text{s}}$
Equation	$\mathbf{v}_B = \mathbf{v}_O + \boldsymbol{\omega} \times \mathbf{r}_{OB}$
Description	<p> \mathbf{v}_B is the velocity at point B ($\frac{\text{m}}{\text{s}}$) \mathbf{v}_O is the velocity at point origin ($\frac{\text{m}}{\text{s}}$) $\boldsymbol{\omega}$ is the angular velocity ($\frac{\text{rad}}{\text{s}}$) \mathbf{r}_{OB} is the displacement vector between the origin and point B (m) </p>
Notes	<p> The linear velocity \mathbf{v}_B ($\frac{\text{m}}{\text{s}}$) of any point B in a rigid body A: objectTy is the sum of the linear velocity \mathbf{v}_O ($\frac{\text{m}}{\text{s}}$) of the rigid body at the origin (axis of rotation) and the resultant vector from the cross product of the rigid body's angular velocity ($\frac{\text{rad}}{\text{s}}$) and the displacement vector between the origin and point B, \mathbf{r}_{OB} (m). A: objectTy A: objectDimension A: dampingInvolvement </p>
Source	
RefBy	

Refname	DD:torque
Label	Torque
Symbol	
Units	Nm
Equation	$= \mathbf{r} \times \mathbf{F}$
Description	<p>is the torque (Nm)</p> <p>\mathbf{r} is the displacement (m)</p> <p>\mathbf{F} is the force (N)</p>
Notes	The torque on a body measures the the tendency of a force to rotate the body around an axis or pivot.
Source	
RefBy	

Refname	DD:kEnergy
Label	Kinetic energy
Symbol	KE
Units	J
Equation	$KE = \frac{m\mathbf{v}^2}{2}$
Description	KE is the kinetic energy (J) m is the mass (kg) \mathbf{v} is the velocity ($\frac{\text{m}}{\text{s}}$)
Notes	The kinetic energy of an object is the energy it possess due to its motion. A: objectTy A: objectDimension A: dampingInvolvement
Source	
RefBy	

Refname	DD:coeffRestitution
Label	Coefficient of restitution
Symbol	C_R
Units	Unitless
Equation	$C_R = - \left(\frac{\mathbf{v}_f^{AB} \cdot \mathbf{n}}{\mathbf{v}_i^{AB} \cdot \mathbf{n}} \right)$
Description	<p> C_R is the coefficient of restitution (Unitless) \mathbf{v}_f^{AB} is the final relative velocity between rigid bodies of A and B $\left(\frac{\text{m}}{\text{s}}\right)$ \mathbf{n} is the collision normal vector (m) \mathbf{v}_i^{AB} is the initial relative velocity between rigid bodies of A and B $\left(\frac{\text{m}}{\text{s}}\right)$ </p>
Notes	<p> The coefficient of restitution C_R is a unitless, dimensionless quantity that determines the elasticity of a collision between two rigid bodies. $C_R = 1$ results in an elastic collision, while $C_R < 1$ results in an inelastic collision, and $C_R = 0$ results in a totally inelastic collision. </p>
Source	
RefBy	

Refname	DD:reVeInColl
Label	Initial Relative Velocity Between Rigid Bodies of A and B
Symbol	\mathbf{v}_i^{AB}
Units	$\frac{\text{m}}{\text{s}}$
Equation	$\mathbf{v}_i^{AB} = \mathbf{v}^{AP} - \mathbf{v}^{BP}$
Description	<p>\mathbf{v}_i^{AB} is the initial relative velocity between rigid bodies of A and B ($\frac{\text{m}}{\text{s}}$)</p> <p>$\mathbf{v}^{AP}$ is the velocity of the point of collision P in body A ($\frac{\text{m}}{\text{s}}$)</p> <p>$\mathbf{v}^{BP}$ is the velocity of the point of collision P in body B ($\frac{\text{m}}{\text{s}}$)</p>
Notes	In a collision, the velocity of a rigid body A: objectTy A colliding with another rigid body B relative to that body \mathbf{v}_i^{AB} is the difference between the velocities of A and B at point P.
Source	
RefBy	

4.2.5 Instance Models

This section transforms the problem defined in Section: Problem Description into one which is expressed in mathematical terms. It uses concrete symbols defined in Section: Data Definitions to replace the abstract symbols in the models identified in Section: Theoretical Models and Section: General Definitions.

Refname		IM:transMot	
Label		Force on the translational motion of a set of 2d rigid bodies	
Input		$\mathbf{v}_i, t, g, \mathbf{F}_i, m_j$	
Output		\mathbf{a}_i	
Input Constraints		$\mathbf{v}_i > 0$ $t > 0$ $g > 0$ $\mathbf{F}_i > 0$ $m_j > 0$	
Output constraints	Con-		
Equation	$\mathbf{a}_i = \frac{d \mathbf{v}_i(t)}{dt} = g + \frac{\mathbf{F}_i(t)}{m_j}$		
Description	\mathbf{a}_i is the the i-th body's acceleration ($\frac{\text{m}}{\text{s}^2}$) t is the time (s) \mathbf{v}_i is the velocity of the i-th body's velocity ($\frac{\text{m}}{\text{s}}$) g is the gravitational acceleration ($\frac{\text{m}}{\text{s}^2}$) \mathbf{F}_i is the force applied to the i-th body at time t (N) m_j is the mass of the j-th particle (kg)		
Notes	The above equation expresses the total acceleration of the rigid body A: objectTy A: objectDimension i as the sum of gravitational acceleration (GD3) and acceleration due to applied force Fi(t) (T1). The resultant outputs are then obtained from this equation using DD: linDisp DD: linVel DD: linAcc. It is currently assumed that there is no damping A: dampingInvolvement or constraints A: constraintsAndJointsInvolvement involved. DD: ctrOfMass.		
Source			
RefBy			

Refname	IM:rotMot
Label	Force on the rotational motion of a set of 2D rigid body
Input	$, t, {}_i, \mathbf{I}$
Output	\pm
Input Constraints	> 0 $t > 0$ ${}_i > 0$ $\mathbf{I} > 0$
Output constraints	$\pm > 0$
Equation	$\pm = \frac{d({}_i(t))}{dt} = \frac{{}_i(t)}{\mathbf{I}}$
Description	\pm is the angular acceleration ($\frac{\text{rad}}{\text{s}^2}$) t is the time (s) ${}_i$ is the angular velocity ($\frac{\text{rad}}{\text{s}}$) ${}_i$ is the torque applied to the i-th body (Nm) \mathbf{I} is the moment of inertia (kgm^2)
Notes	The above equation for the total angular acceleration of the rigid body A: objectTy A: objectDimension i is derived from T5, and the resultant outputs are then obtained from this equation using DD: angDisp DD: angVel DD: angAccel. It is currently assumed that there is no damping A: dampingInvolvement or constraints A: constraintsAndJointsInvolvement involved. A: axesDefined
Source	29
RefBy	

Refname	IM:col2D		
Label	Collisions on 2D rigid bodies		
Input	t, j, m_A, \mathbf{n}		
Output	t_c		
Input Constraints	$t > 0$ $j > 0$ $m_A > 0$ $\mathbf{n} > 0$		
Output constraints	Con-	$\mathbf{v}_A > 0$ $t_c > 0$	
Equation	$\mathbf{v}_A(t_c) = \mathbf{v}_A(t) + \frac{j}{m_A} \mathbf{n}$		
Description	\mathbf{v}_A is the velocity at point A ($\frac{\text{m}}{\text{s}}$) t_c is the denotes the time at collision (s) t is the time (s) j is the impulse (scalar) (Ns) m_A is the mass of rigid body A (kg) \mathbf{n} is the collision normal vector (m)		
Notes	This instance model is based on our assumptions regarding rigid body A: objectTy A: objectDimension collisions A: collisionType Again, this does not take damping A: dampingInvolvement or constraints A: constraintsAndJointsInvolvement into account. A: axesDefined. DD: ctrOfMass DD: impulse		
Source			
RefBy			

4.2.6 Data Constraints

Table:InDataConstraints and Table:OutDataConstraints show the data constraints on the input and output variables, respectively. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. FIXME

Var	Physical Constraints	Typical Value	Uncert.
C_R	$0 \leq C_R \leq 1$	$800.0 \cdot 10^{-3}$	10.0%
\mathbf{F}	–	98.1 N	10.0%
G	–	$9.8 \frac{\text{m}^3}{(\text{kg s}^2)}$	10.0%
\mathbf{I}	$\mathbf{I} \geq 0$	74.5 kgm ²	10.0%
L	$L \geq 0$	44.2 m	10.0%
m	$m \geq 0$	56.2 kg	10.0%
\mathbf{p}	–	$412.0 \cdot 10^{-3}$ m	10.0%
\mathbf{v}	–	$2.51 \frac{\text{m}}{\text{s}}$	10.0%
	–	200.0 Nm	10.0%
	–	$2.1 \frac{\text{rad}}{\text{s}}$	10.0%
	–	$\frac{1}{2}$ rad	10.0%

Table 2: Input Data Constraints

Var
\mathbf{p}
\mathbf{v}

Table 3: Output Data Constraints

5 Requirements

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

5.1 Functional Requirements

- Simulation-Space: Create a space for all of the rigid bodies in the physical simulation to interact in.
- Initial-Conditions: Input the initial masses, velocities, orientations, angular velocities of, and forces applied on rigid bodies.
- Surface-Properties: Input the surface properties of the bodies such as friction or elasticity.
- Physical_Constraints: Verify that the inputs satisfy the required physical constraints from Section: Solution Characteristics Specification.
- Simulation-Over-Time: Determine the positions and velocities over a period of time of the 2D rigid bodies acted upon by a force.
- Rotation-Over-Time: Determine the orientations and angular velocities over a period of time of the 2D rigid bodies.
- Determine-Collisions: Determine if any of the rigid bodies in the space have collided.
- Response-Over-Time: Determine the positions and velocities over a period of time of the 2D rigid bodies that have undergone a collision.

5.2 Non-Functional Requirements

Games are resource intensive, so performance is not a priority. Rather than performance, the non-functional requirement priorities are correctness, understandability, portability, reliability, and maintainability.

6 Likely Changes

This section lists the likely changes to be made to the game physics library.

- Variable-ODE-Solver: The internal ODE-solving algorithm used by the library may be changed in the future.
- Expanded-Collisions: The library may be expanded to deal with edge-to-edge and vertex-to-vertex collisions.
A: collisionType
- Include-Dampening: The library may be expanded to include motion with damping. A: dampingInvolvement
- Joints-Constraints: The library may be expanded to include joints and constraints. A: constraintsAndJointsInvolvement

7 Unlikely Changes

This section lists the unlikely changes to be made to the game physics library.

Simulate-Rigid-Bodies: The goal of the system is to simulate the interactions of rigid bodies.

External-Input: There will always be a source of input data external to the software.

Coordinate-System: A Cartesian Coordinate system is used.

Objects-Rigid-Bodies: All objects are rigid bodies.

8 Off-The-Shelf Solutions

As mentioned in Section: Problem Description, there already exist free open source game physics libraries. Similar 2D physics libraries are:

- Box2D: <http://box2d.org/>
- Nape Physics Engine: <http://napephys.com/>

Free open source 3D game physics libraries include:

- Bullet: <http://bulletphysics.org/>
- Open Dynamics Engine: <http://www.ode.org/>
- Newton Game Dynamics: <http://newtondynamics.com/>

9 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the row of that component that are marked with an “X” should be modified as well. Table:Tracey shows the dependencies of goal statements, requirements, instance models, and data constraints with each other. Table:TraceyReqGoalsOther shows the dependencies of theoretical models, general definitions, data definitions, instance models, and on the assumptions. Table:TraceyAssumpsOther shows the dependencies of theoretical models, general definitions, data definitions, and instance models on each other.

FR: Verify-Physical_Constraints	IM: rotMot	DD: impulse	IM: col2D	IM: transMot	LC: Includ
Section: Solution Characteristics Specification	X				
DD: angAccel				X	
DD: angDisp				X	
DD: angVel				X	

FR: Verify-Physical_Constraints	IM: rotMot	DD: impulse	IM: col2D	IM: transMot	LC: Includ
A: axesDefined				X	X
A: constraintsAndJointsInvolvement				X	
A: collisionType					X
A: dampingInvolvement				X	
A: objectDimension				X	X
A: objectTy				X	X
DD: impulse					
DD: linAcc					
DD: linDisp					
DD: linVel					
DD: ctrOfMass					

IM1 (IM: transMot)	IM2 (IM: rotMot)	IM3 (IM: col2D)	R1 (FR: Simulation-Space)	R4 (FR: In
GS1 (Section: Problem Description)		X		
GS2 (Section: Problem Description)			X	
GS3 (FR: Simulation-Space)				X
GS4 (FR: Input-Initial-Conditions)				X
R1 (FR: Input-Surface-Properties)				
R2 (FR: Verify-Physical_Constraints)		X	X	
R3 (FR: Calculate-Translation-Over-Time)				X
R4 (FR: Calculate-Rotation-Over-Time)				
R5 (FR: Determine-Collisions)		X		
R6 (FR: Determine-Collision-Response-Over-Time)			X	
				X

Table 5:
Requirem

A1 (A: objectTy)	A2 (A: objectDimension)	A3 (A: coordinateSystemTy)	A4 (A: axesDefined)	A5 (A: constraintsAndJointsInvolvement)
T1 (TM: NewtonSecLawMot)				
T2 (TM: NewtonThirdLawMot)				
T3 (TM: UniversalGravLaw)				
T4 (TM: ChaslesThm)		X		
T5 (TM: NewtonSecLawRotMot)				
GD1 (Section: Solution Characteristics Specification)				
GD2 (Section: Solution Characteristics Specification)				

A1 (A: objectTy)	A2 (A: objectDimension)	A3 (A: coordinateSystemTy)	A4 (A: axesDefined)	A5 (A: objectVelocity)
GD3 (Section: Solution Characteristics Specification)			X	X
GD4 (Section: Solution Characteristics Specification)				
GD5 (Section: Solution Characteristics Specification)				
GD6 (Section: Solution Characteristics Specification)				
GD7 (Section: Solution Characteristics Specification)				
DD1 (DD: ctrOfMass)		X	X	
DD2 (DD: linDisp)		X	X	
DD3 (DD: linVel)		X	X	
DD4 (DD: linAcc)		X	X	
DD5 (DD: angDisp)		X	X	
DD6 (DD: angVel)		X	X	
DD7 (DD: angAccel)		X	X	
DD8 (DD: impulse)		X	X	
IM1 (DD: chases)		X	X	
IM2 (DD: torque)		X	X	
IM3 (DD: kEnergy)		X	X	
LC1 (DD: coeffRestitution)				
LC2 (DD: reVeInColl)				
LC3 (IM: transMot)				
LC4 (IM: rotMot)				

Table 6: Traceability Matrix (Legend: M: Module Description) and O: Object Description)

T1 (TM: NewtonSecLawMot)	T2 (TM: NewtonThirdLawMot)	T3 (TM: UniversalGravLaw)	T4 (TM: ChaslesThm)
T1 (TM: NewtonSecLawMot)			
T2 (TM: NewtonThirdLawMot)			
T3 (TM: UniversalGravLaw)			
T4 (TM: ChaslesThm)			
T5 (TM: NewtonSecLawRotMot)			
GD1 (Section: Solution Characteristics Specification)	X		
GD2 (Section: Solution Characteristics Specification)		X	
GD3 (Section: Solution Characteristics Specification)	X		
GD4 (Section: Solution Characteristics Specification)			
GD5 (Section: Solution Characteristics Specification)			
GD6 (Section: Solution Characteristics Specification)			
GD7 (Section: Solution Characteristics Specification)			
DD1 (DD: ctrOfMass)			
DD2 (DD: linDisp)			
DD3 (DD: linVel)			

T1 (TM: NewtonSecLawMot)	T2 (TM: NewtonThirdLawMot)	T3 (TM: UniversalGravLaw)	T4 (TM: SpecialRelativity)
DD4 (DD: linAcc)			
DD5 (DD: angDisp)			
DD6 (DD: angVel)			
DD7 (DD: angAccel)			
DD8 (DD: impulse)			
IM1 (DD: chalses)		X	
IM2 (DD: torque)			
IM3 (DD: kEnergy)			

10 Values of Auxiliary Constants

There are no auxiliary constants.

11 References