Software Requirements Specification for Chipmunk2D

Alex Halliwushka

August 19, 2015

Contents

1 Reference Material **2**
- 1.1 Table of Units . . . 2
- 1.2 Table of Symbols . . . 2
- 1.3 Abbreviations and Acronyms . . . 3

2 Introduction **4**
- 2.1 Purpose of Document . . . 4
- 2.2 Scope of Requirements . . . 4
- 2.3 Organization of Document . . . 4

3 General System Description **4**
- 3.1 User Characteristics . . . 5
- 3.2 System Constraints . . . 5

4 Specific System Description **5**
- 4.1 Problem Description . . . 5
  - 4.1.1 Terminology and Definitions . . . 5
  - 4.1.2 Physical System Description . . . 6
  - 4.1.3 Goal Statements . . . 6
- 4.2 Solution Characteristics Specification . . . 6
  - 4.2.1 Assumptions . . . 6
  - 4.2.2 Theoretical Models . . . 6
  - 4.2.3 General Definitions . . . 9
  - 4.2.4 Data Definitions . . . 12
  - 4.2.5 Instance Models . . . 14
  - 4.2.6 Detailed derivation of Rotation on a 2D rigid body . . . 18
  - 4.2.7 Data Constraints . . . 19

5 Requirements **20**
- 5.1 Functional Requirements . . . 20
- 5.2 Nonfunctional Requirements . . . 21

6 Likely Changes **21**

7 Off the Shelf Solutions **21**

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. For each unit, the symbol is given followed by a description of the unit with the SI name.

symbol	unit	SI
m	length	meter
kg	mass	kilogram
s	time	second
N	force	Newton
rad	angle	radians

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The symbols are listed in alphabetical order.

symbol	unit	description
a	$\mathrm{m/s^2}$	acceleration
α	$\mathrm{rad/s^2}$	angular acceleration
C_{R}	unitless	Coefficient of restitution
d	m	Displacement
d_{p}	m	Distance between particle and the axis of rotation
$\mathbf{F}$	N	Force
g	N	Acceleration due to gravity
G	$\mathrm{m^3/kgs^2}$	Gravitational constant $(6.673 * 10^-11)$
$\mathbf{I}$	$\mathrm{kgm^2}$	Moment of inertia
$\mathbf{J}$	N s	Impulse
k	N/m	Spring constant

L	m	Length
m	kg	Mass
ω	rad/s	Angular velocity
$\mathbf{p}_{\text{i}}$	m	Initial position
$\mathbf{p}_{\text{f}}$	m	Final position
$\mathbf{p}_{\text{com}}$	m	Position of the center of mass
$\mathbf{p}_{\text{force}}$	m	Position the force vector is acting on the object
ϕ	rad	Orientation
r	m	Distance between the force and the axis of rotation
ρ	kg/m^3	Density
t	s	Time
τ	N m	Torque
$\mathbf{v}_{\text{i}}$	m/s	Initial velocity
$\mathbf{v}_{\text{f}}$	m/s	Final velocity
V	m^3	Volume
X	m	Displacement of spring from equilibrium
ζ	unitless	Damping coefficient

1.3 Abbreviations and Acronyms

symbol	**description**
A	Assumption
COM	Center of mass
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

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, an open source 2D rigid body physics library. This section explains the purpose of this document, the scope of the system, 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 goals and the theoretical models 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. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation.

2.2 Scope of Requirements

The scope of the requirements includes the physical simulation of 2D rigid bodies acted on by forces. Given 2D rigid bodies, Chipmunk2D is intended to simulate how these rigid bodies interact with one another .

2.3 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [**?**] and [**?**]. 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 4.2.5 and trace back to find any additional information they require.

The goal statements are refined to the theoretical models, and theoretical models to the instance models.

3 General System Description

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

3.1 User Characteristics

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

3.2 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 for the physics library.

4.1 Problem Description

Creating a gaming physics library is a difficult task. Games need physics libraries that simulate objects acting under physics conditions, while at the same time they need to be efficient and fast 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. These barriers of entry 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 7). 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 velocities of the two colliding objects after and before the collision.
- Center of Mass: The mean location of the distribution of mass of the object
- Cartesian coordinates: 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 Physical System Description

N/A

4.1.3 Goal Statements

GS1: Given the initial position and velocity, determine the new positions and velocities over a period of time of a 2D rigid body acted upon by a force

GS2: Given the initial positions and velocities, determine if any of the rigid bodies will collide with one another over a period of time

GS3: Given the initial positions and velocities, determine the new positions and velocities over a period of time of 2D rigid bodies that have undergone a collision

GS4: Given the initial orientation and angular velocity, determine the new orientation and angular velocity over a period of time of a 2D rigid body acted upon by a force

GS5: Given the initial positions and velocities, determine the positions and velocities over a period of time of 2D rigid bodies with constraints or joints between them

4.2 Solution Characteristics Specification

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 data definition, or the instance model, in which the respective assumption is used.

A1: All objects are rigid bodies

A2: All objects are 2D

A3: The damping coefficient is constant throughout the simulation

A4: The library uses a Cartesian coordinate system

A5: The axes are defined using a right hand system

4.2.2 Theoretical Models

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

Number	T1
Label	**Newton's second law of motion**
Equation	$\mathbf{F} = \text{m}\mathbf{a}$
Description	$\mathbf{F}$ = Net force applied m =Mass of the object $\mathbf{a}$ = Acceleration of the object The net force on an object is proportional to the objects acceleration
Source	
Ref. By	IM2, GD1

Number	T2
Label	**Newton's third law of motion**
Equation	$\mathbf{F_1} = -\mathbf{F_2}$
Description	$\mathbf{F}_1$ =The force being exerted on the second object by the first $\mathbf{F}_2$ =The force being exerted on the first object by the second Every action has an equal and opposite reaction
Source	
Ref. By	GD2

Number	T3
Label	**Newton's Law of universal gravitation**
Equation	$\mathbf{F} = G \frac{m_1 m_2}{\|\|\mathbf{r}\|\|^2} \frac{\mathbf{r}}{\|\|\mathbf{r}\|\|}$
Description	$\mathbf{F}$ = Force between the objects m_1 =Mass of the first object m_2 =Mass of the second object G = Gravitational constant $(6.673 * 10^{-11} m^3 kg^{-1} s^{-2})$ $\|\|\mathbf{r}\|\|$ = The distance between the centers of the objects $\mathbf{r}$ = The displacement vector between the centers of the objects Any two bodies in the universe attract each other with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them.
Source	
Ref. By	IM2

Number	T4
Label	**Hooke's law**
Equation	$\mathbf{F} = \mathrm{k}\mathbf{X}$
Description	$\mathbf{F}$ = Force between the objects $\mathbf{X}$ =The distance the spring is extended or compressed from its equilibrium position k =Spring constant
Source	
Ref. By	

Number	T5
Label	**Equation of Rotational motion**
Equation	$\boldsymbol{\tau} = \mathbf{I}\boldsymbol{\alpha}$
Description	$\boldsymbol{\tau}$ =Torque $\mathbf{I}$ =The moment of inertia $\boldsymbol{\alpha}$ =Angular acceleration
Source	
Ref. By	GD4,GD5

4.2.3 General Definitions

This section collects the laws and equations that will be used in deriving the data definitions, which in turn are used to build the instance models.

Number	GD1
Label	**Impulse**
Equation	$\mathbf{J} = \int \mathbf{F}\ dt = \Delta\ \mathbf{P} = m\ \Delta\ \mathbf{v}$
Description	$\mathbf{J}$ =Impulse $\mathbf{F}$ = Force $\Delta\mathbf{P}$ = Change in momentum m =Mass of the object $\Delta\mathbf{v}$ =Change in velocity of the object An impulse occurs when a force $\mathbf{F}$ acts over an interval of time
Source	
Ref. By	T1, GD2

Derivation of Impulse

Newton's second law of motion states:

$$\mathbf{F} = \mathrm{m}\mathbf{a} = \mathrm{m}\frac{\mathrm{d}\mathbf{v}}{\mathrm{dt}}$$

Rearranging:

$$\int_{t_1}^{t_2} \mathbf{F}\mathrm{dt} = \mathrm{m}\int_{\mathrm{v}_1}^{\mathrm{v}_2} \mathrm{d}\mathbf{v}$$

Integrating the right hand side:

$$\int_{t_1}^{t_2} \mathbf{F}\mathrm{dt} = \mathrm{m}\mathbf{v}_2 - \mathrm{m}\mathbf{v}_1 = m\Delta\mathbf{v}$$

Number	GD2
Label	**Conservation of momentum**
Equation	$\sum_{k=0}^{n} m_{\mathrm{k}}\ \mathbf{v}_{\mathrm{ik}} = \sum_{k=0}^{n} m_{\mathrm{k}}\mathbf{v}_{\mathrm{fk}}$
Description	m_{k} =Mass of the object $\mathbf{v}_{\mathrm{ik}}$ =Initial velocity of the object $\mathbf{v}_{\mathrm{fk}}$ =Final velocity of the object In an isolated system, where the sum of external impulses acting on the system is zero the total momentum of the objects is constant
Source	
Ref. By	T2, GD1, IM3

Derivation of the conservation of momentum

When objects collide they exert an equal force on each other in opposite directions. This is Newton's third law:

$$\mathbf{F}_1 = -\mathbf{F}_2$$

The objects collide with each other for the exact same amount of time (t):

$$\mathbf{F}_1\mathrm{t} = -\mathbf{F}_2\mathrm{t} \tag{1}$$

The above equation is equal to the impulse:

$$\mathbf{F}_1 = \int \mathbf{F}_1\mathrm{dt} = \mathbf{J}$$

The impulse is equal to the change in momentum:

$$\mathbf{J} = \Delta\mathbf{P} = \mathrm{m}\Delta\mathbf{v} \tag{2}$$

substitution 2 into 1

$$m_1\Delta\mathbf{v}_1 = -\mathrm{m}_2\Delta\mathbf{v}_2$$

Expanding and rearranging the above formula gives:

$$m_1\mathbf{v}_{\text{i1}} + \text{m}_2\mathbf{v}_{\text{i2}} = \text{m}_1\mathbf{v}_{\text{f1}} + \text{m}_2\mathbf{v}_{\text{f2}}$$

Generalizing for multiple colliding objects:

$$\sum_{k=0}^{n} m_{\text{k}}\mathbf{v}_{\text{ik}} = \sum_{k=0}^{n}\text{m}_{\text{k}}\mathbf{v}_{\text{fk}}$$

Number	GD3
Label	**Coefficient of restitution**
Equation	$C_{\text{R}} = \frac{\mathbf{v}_{\text{f2}}-\mathbf{v}_{\text{f1}}}{\mathbf{v}_{\text{i1}}-\mathbf{v}_{\text{i2}}}$
Description	C_{R} = Coefficient of restitution $\mathbf{v}_{\text{i1}}$ =Initial velocity of the first object $\mathbf{v}_{\text{i2}}$ = Initial velocity of the second object $\mathbf{v}_{\text{f1}}$ =Final velocity of the first object $\mathbf{v}_{\text{f2}}$ =Final velocity of the second object The coefficient of restitution determines the elasticity of the collision. A $C_{\text{R}} = 1$ results in an elastic collision, while a $C_{\text{R}} < 1$ results in an inelastic collision
Source	
Ref. By	IM3

Number	GD4
Label	**Calculating torque from a force**
Equation	$\boldsymbol{\tau} = \mathbf{r}\times\mathbf{F} = \mathbf{r}\mathbf{F}_{\perp}$
Description	$\boldsymbol{\tau}$=Rotational force on the object $\mathbf{F}$ = The force on the lever arm $\mathbf{F}_{\perp}$ = The force perpendicular to the lever arm $\mathbf{r}$ =The distance between the force and the axis of rotation (position vector)
Source	
Ref. By	T5, IM4

Number	GD5
Label	**Moment of Inertia**
Equation	$\mathbf{I} = \sum_{i=0}^{n} m_{\text{i}} d_{\text{pi}}^2$
Description	$\mathbf{I}$=Moment of Inertia n = The number of particles m_{i} =The mass of the particle i d_{pi} = Distance between particle i and the axis of rotation
Source	
Ref. By	T5

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Kinematics**
Symbol	**d, v**
SI Units	m, m/s
Equation	$\mathbf{a}(\text{t}) = \frac{\text{d}\mathbf{v}(\text{t})}{\text{dt}}$, $\mathbf{v}(\mathbf{t}) = \frac{\text{d}\mathbf{x}(\text{t})}{\text{dt}}$
Description	$\mathbf{a}(\text{t})$ = Object's acceleration over time $\mathbf{v}(\text{t})$ = Object's velocity over time $\mathbf{x}(\text{t})$ =Object's displacement over time t = Time
Sources	
Ref. By	T1, IM2

Number	DD2
Label	**Kinematics with damping**
Symbol	d, v
SI Units	m, m/s
Equation	$\mathbf{p}(\mathrm{t}) = \mathbf{p}_{\mathrm{i}} + \mathbf{v}_{\mathrm{i}} t \zeta^{\mathrm{t}} + \frac{1}{2}\mathbf{a}t^2$, $\mathbf{v}(\mathrm{t}) = \mathbf{v}_{\mathrm{i}}\zeta^{\mathrm{t}} + \mathbf{a}\mathrm{t}$
Description	$\Delta\mathbf{d}$ =Change in displacement $\mathbf{p}(\mathrm{t})$ = Object's position over time $\mathbf{p}_{\mathrm{i}}$ = Object's initial position $\mathbf{v}(\mathrm{t})$ = Object's velocity over time $\Delta\mathbf{v}$ =Change in velocity $\mathbf{v}_{\mathrm{i}}$ = Object's initial velocity $\mathbf{a}$ = Object's acceleration t = Time ζ = The damping coefficient
Sources	
Ref. By	T1

Number	DD3
Label	**Rotational Kinematics**
Symbol	$\boldsymbol{\phi}, \boldsymbol{\omega}$
SI Units	rad, rad/s
Equation	$\boldsymbol{\alpha}(t) = \frac{d\boldsymbol{\omega}(t)}{dt}$, $\boldsymbol{\omega}(t) = \frac{d\boldsymbol{\phi}(t)}{dt}$
Description	$\boldsymbol{\alpha}(t)$ = Object's angular acceleration over time $\boldsymbol{\omega}(t)$ = Object's angular velocity over time $\boldsymbol{\phi}(t)$ =Object's orientation over time t = Time
Sources	
Ref. By	T5

Number	DD4
Label	**Rotational Kinematics with damping**
Symbol	$\boldsymbol{\phi}, \boldsymbol{\omega}$
SI Units	$\mathrm{rad}, \mathrm{rad/s}$
Equation	$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_{\mathrm{i}} + \boldsymbol{\omega}_{\mathrm{i}} t \zeta^{\mathrm{t}} + \frac{1}{2}\boldsymbol{\alpha} t^2$, $\boldsymbol{\omega}(t) = \boldsymbol{\omega}_{\mathrm{i}} \zeta^{\mathrm{t}} + \boldsymbol{\alpha} t$
Description	$\boldsymbol{\phi}(t)$ = Object's orientation over time $\boldsymbol{\phi}_{\mathrm{i}}$ = Object's initial orientation $\boldsymbol{\omega}(t)$ = Object's angular velocity over time $\boldsymbol{\omega}_{\mathrm{i}}$ = Object's initial angular velocity $\boldsymbol{\alpha}$ = Object's angular acceleration t = Time ζ = The damping coefficient
Sources	
Ref. By	T5

4.2.5 Instance Models

This section transforms the problem defined in the Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in the Sections 4.2.2 and 4.2.3.

Number	IM1
Label	**Gravity on a 2D rigid body without damping**
Input	g, $\mathbf{p}_{\text{i}}$, $\mathbf{v}_{\text{i}}$
Output	$\mathbf{p}(\text{t})$, $\mathbf{v}(\text{t})$ such that $\mathbf{a}= \frac{d\mathbf{v}(\text{t})}{dt} = \begin{bmatrix} 0 \\ -g \end{bmatrix}$
Description	$\mathbf{a}$ =Acceleration $\mathbf{p}_{\text{i}}$ = Initial position of the object $\mathbf{v}_{\text{i}}$ =Initial velocity of the object $\mathbf{p}(\text{t})$ =Position of the object as a function of time $\mathbf{v}(\text{t})$ =Velocity of the object as a function of time t = Time g = Gravity
Sources	
Ref. By	T1, T3, DD1

Derivation of gravity

Netwon's first law of motion states:

$$\mathbf{F} = \text{m}\mathbf{a} \tag{3}$$

Newton's law of Universal Gravitation states:

$$\mathbf{F} = G\frac{m_1 m_2}{||\mathbf{r}||^2}\frac{\mathbf{r}}{||\mathbf{r}||} \tag{4}$$

Substituting 4 into 3:

$$m\mathbf{a} = G\frac{m_1 m_2}{||\mathbf{r}||^2}$$

Simplifying:

$$\mathbf{a} = G\frac{m_1}{||\mathbf{r}||^2}$$

Let $\mathbf{g} = G\frac{m_1}{||\mathbf{r}||^2}$, The acceleration due to gravity is:

$$\mathbf{a} = \mathbf{g}$$

Number	IM2
Label	**Force on a 2D rigid body without damping**
Input	g, $\mathbf{p}_\mathrm{i}$, $\mathbf{v}_\mathrm{i}$, $\mathbf{F}$
Output	$\mathbf{p}(\mathrm{t})$, $\mathbf{v}(\mathrm{t})$ such that $\mathbf{a}= \frac{d\mathbf{v}(\mathrm{t})}{dt} = \begin{bmatrix} F_\mathbf{x} \\ F_\mathbf{y} \end{bmatrix}$
Description	$\mathbf{a}$ =Acceleration $\mathbf{p}_\mathrm{i}$ = Initial position of the object $\mathbf{v}_\mathrm{i}$ =Initial velocity of the object $\mathbf{p}(\mathrm{t})$ =Position of the object as a function of time $\mathbf{v}(\mathrm{t})$ =Velocity of the object as a function of time t = Time $\mathbf{F}$ = Force acting on the object
Sources	
Ref. By	T1, T3, DD1

Number	IM3
Label	**Collisions on 2D rigid bodies**
Input	m_1, m_2, $\mathbf{v}_{\text{i1}}$, $\mathbf{v}_{\text{i2}}$, $\mathbf{p}_{\text{i1}}$, $\mathbf{p}_{\text{i2}}$. C_{R}
Output	$\mathbf{v}_1(\text{t})$, $\mathbf{v}_2(\text{t})$, $\mathbf{p}_1(\text{t})$, $\mathbf{p}_2(\text{t})$ such that: $\sum_{k=0}^{n} m_{\text{k}}\mathbf{v}_{\text{ik}} = \sum_{k=0}^{n} m_{\text{k}}\mathbf{v}_{\text{fk}}$, $C_{\text{R}} = \frac{\mathbf{v}_{\text{f2}}-\mathbf{v}_{\text{f1}}}{\mathbf{v}_{\text{i1}}-\mathbf{v}_{\text{i2}}}$, $\mathbf{a}_1 = \frac{\text{d}\mathbf{v}_1(\text{t})}{\text{dt}}$, $\mathbf{a}_2 = \frac{\text{d}\mathbf{v}_2(\text{t})}{\text{dt}}$
Description	$\mathbf{p}_{\text{i1}}$ = Initial position of the first object $\mathbf{p}_{\text{i2}}$ = Initial position of the second object $\mathbf{v}_{\text{i1}}$ =Initial velocity of the first object $\mathbf{v}_{\text{i2}}$ =Initial velocity of the second object $\mathbf{p}_1(\text{t})$ =Final position of the first object as a function of time $\mathbf{p}_2(\text{t})$ =Final position of the second object as a function of time $\mathbf{v}_1(\text{t})$ =Final velocity of the first object as a function of time $\mathbf{v}_2(\text{t})$ =Final velocity of the second object as a function of time $\mathbf{a}_1$ = Acceleration of the first object $\mathbf{a}_2$ =Acceleration of the second object C_{R} = Coefficient of restitution
Sources	
Ref. By	GD2, GD3, DD1

Number	IM4
Label	**Rotation on a 2D rigid body**
Input	$\boldsymbol{\phi}_{\text{i}}$, $\boldsymbol{\omega}_{\text{i}}$, $\mathbf{F}$, $\mathbf{p}_{\text{com}}$, $\mathbf{p}_{\text{force}}$, $\mathbf{I}$
Output	$\boldsymbol{\phi}(t)$, $\boldsymbol{\omega}(t)$ such that: $\boldsymbol{\alpha}(t) = \frac{d\boldsymbol{\omega}(t)}{dt}$
Description	$\boldsymbol{\phi}_{\text{i}}$ = Initial orientation of the object $\boldsymbol{\omega}_{\text{i}}$ =Initial angular velocity of the object $\mathbf{F}$ =Force acting on the object $\mathbf{I}$ =Moment of Intertia of the object $\mathbf{p}_{\text{com}}$ =Center of mass of the object $\mathbf{p}_{\text{force}}$ = Point the force is applied on the object $\boldsymbol{\phi}(\text{t})$ =Orientation of the object as a function of time $\boldsymbol{\omega}(\text{t})$ =Angular velocity of the object as a function of time $\boldsymbol{\alpha}(\text{t})$ =Angular acceleration of the object as a function of time
Sources	
Ref. By	T5, GD4,DD3

4.2.6 Detailed derivation of Rotation on a 2D rigid body

Using GD4 to calculate the toruqe

$$\boldsymbol{\tau} = \mathbf{r}\times\mathbf{F} = \mathbf{r}\mathbf{F}_{\perp} = (\mathbf{p}_{\text{com}} - \mathbf{p}_{\text{ force}})\mathbf{F}_{\perp}$$

The angular acceleration can be calculated (T5):

$$\boldsymbol{\alpha} = \frac{\boldsymbol{\tau}}{\mathbf{I}} = \frac{(\mathbf{p}_{\text{com}} - \mathbf{p}_{\text{ force}})\mathbf{F}_{\perp}}{\mathbf{I}}$$

Using DD3 The object's angular velocity and orientation can be calculated:

$$\boldsymbol{\alpha}(t) = \frac{d\boldsymbol{\omega}(t)}{dt}$$

Integrating both sides gives:

$$\boldsymbol{\omega}(t) = \boldsymbol{\alpha}t + \boldsymbol{\omega}_{\text{i}}$$

Using the above angular velocity to calculate the orientation:

$$\boldsymbol{\omega}(t) = \frac{d\boldsymbol{\phi}(t)}{dt}$$

Integrating both sides gives:

$$\boldsymbol{\phi}(t) = \frac{1}{2}\boldsymbol{\alpha}t^2 + \boldsymbol{\omega}_{\text{i}}t + \boldsymbol{\phi}_{\text{i}}$$

Number	IM5
Label	**Connections and joints on 2D rigid bodies**
Ref. By	The instance models for constraints are located in the SRS Constraints document

4.2.7 Data Constraints

Table 2 and 4 show the data constraints on the input and output variables, respectively. The column physical constraints gives the physical limitations on the range of values that can be taken by the variable. 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.

Table 2: Input Variables

Var	Physical Constraints	Typical Value
$\mathbf{a}$	None	9.8 m/s^2
$\mathbf{d}$	None	0.412 m
$\mathbf{v}$	None	0.05 m/s
m	$m \geq 0$	1.2 kg
C_{R}	$0 \leq C_{\text{R}} \leq 1$	0.8
$\boldsymbol{\alpha}$	None	1.6 rad/s^2
$\boldsymbol{\phi}$	$0 \leq \phi < 2\pi$	$\frac{\pi}{2}$ rad
$\boldsymbol{\omega}$	None	2.1 rad/s
ζ	$\zeta \geq 0$	0.91
L	$L \geq 0$	44.2 m
k	$k \geq 0$	1.4 N/m
V	$V > 0$ (*)	22 m^3
ρ	$\rho \geq 0$	1 kg/m^3

(*) These quantities cannot be equal to zero, or there will be a divide by zero in the model.

Table 4: Output Variables

Var	Physical Constraints
$\mathbf{v}$	None
$\mathbf{p}$	None
$\mathbf{F}$	None
$\boldsymbol{\phi}$	$0 \leq \phi < 2\pi$
$\boldsymbol{\omega}$	None

5 Requirements

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.

5.1 Functional Requirements

R1: Create a space for all of the rigid bodies in the physical simulation to interact in.

R2: Input the initial mass, velocities, positions, orientations, angular velocities, and constraints of objects

R3: Input the surface properties of the objects such as friction or elasticity

R4: Input the constraints of the objects

R5: Verify that the inputs satisfy the required physical constraints

R6: Determine the position and velocities over a period of time of the 2D rigid bodies acted upon by a force

R7: Determine if any of the rigid bodies in the space have collided

R8: Determine the position and velocities over a period of time of 2D rigid bodies that have undergone a collision

R9: Determine the orientation and angular velocities over a period of time of the 2D rigid bodies

R10: Determine the position and velocities over a period of time of 2D rigid bodies with constraints or joints between them

R11: Allow the user to query the space and return information about the rigid bodies.

5.2 Nonfunctional Requirements

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

6 Likely Changes

7 Off the Shelf Solutions

As mentioned in section 4.1, there already exists 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/`