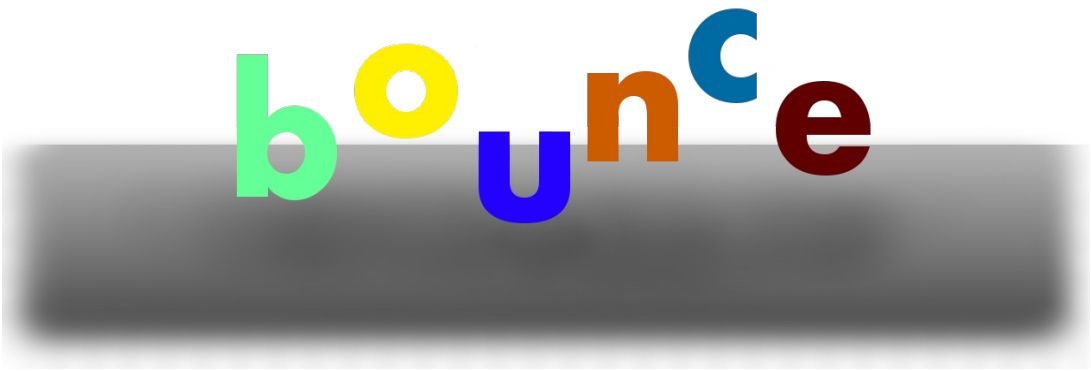
Bounce2D 1.0 Manual



1 Introduction

Description

Bounce2D is a 2D rigid body physics simulator. It is open source and free for commericial and non comerical use; under the BSD licence.

Features

- 2D rigid body physics simulator
- Collision detection and reporting
- Developed in Java
- Open Source

Download

- The source code is available from <>
- Demos are available from <>

Contact

- Donovan Benoit <donovan.benoit@gmail.com>

2 Using Bounce

Integration

Bounce has been designed to be easily integrated into an existing project or incorporated into a new project.

- Add the Bounce package to your project
- Instantiate a new Bounce World
- Add PhysicsObjects and Modifers to that world
- Add a CollisionListener to the world to report collisions
- Advance the simulation by the desired time step

3 Bounce Library Overview

Introduction

Bounce is tasked with simulating forces applied to bodies, collision detection, collision resolution, and simulation of other constraints. This section will give an overview of the components in the Bounce Package and how they interact.

Software Design

Bounce has been designed from the start to be easily customized and modular. The five main components used by the library are World, PhysicsObject, Modifer, PhysicsListener, and Vector2.

VPP fig

World

- World
  The World class is responsible for simulating the physical world. This world consists of PhysicsObjects Modifiers and other properities. The units used by the world are 100 units per meter and time is in mcroseconds. The positive x-axis points right and the positive y-axis points down. The world has gravity which acts on some PhysicsObjects(see PhysicsObject). The force of gravity can be changed through the set gravity method.

- Advance Simulation
  The `advanceSimulation` method advances the physicsl simulation by a time step. If the timestep is larger then the `maxTimeStep` then it advances by `maxTimeStep`. maxTimeStep can be set using the `setMaxTimeStep` method. `AdvanceSimulation` integrates the PhysicsObjects over time then recalculates verticies of the objects and checks for collisions. If a penetration is detected then the simulation is steped back, the time step is cut in half and resimulated until there is no penetration or the timestep is suficiently small.

- Integration
  Integration is responsible for calculating forces applied to bodies and integrating them over time to produce velocities and positions of objects.

- Collisions
  The engine checks all non static PhysicsObjects against all other PhysicsObjects this results in the order of 2^n checks. The colision detection algorithm uses a bounding circle test to see if two objects are close enough to collide. If the circles overlap the function goes into a more accurate collision detection algorithm. The algorthm used for this is a Barycentric point in triangle algorithm for collision detection.
  <http://www.blackpawn.com/texts/pointinpoly/default.html >
  If the point is outside the triangle no collision is detected, if it is inside then a collision is taking place.

- Resolving Collisions
  When a collision is detected the Resolve Collision function takes the two bodies that collided and applies an impulse and moment to the bodies dependant on the collision normal, collision point.

Physics Object

- PhysicsObject is an abstract class from whics all objects simulated by the World are extended. To create an PhysicsObject you need a name position and mass. If the mass is greater then 0 an dynamic object is created if the mass is 0 then a static object is created. Static objects cant move.

- RigidBody consists of a convex mesh of verticies.

- Particle

Modifer

A modifier is defined to be anything that modifies the properities of a set of PhysicsObjects and has an update method. Modifier is an abstract class.

- Spring
  Spring is an example of a modifier it simulates connecting two bodies with a spring. It applies a force to the objects about their center of mass proportional to the spring constant the distance between the objects and the objects masses.

Collision Listener

The Collision Listener is a way to report collisions outside the Bounce Package. It has a method that gets called with the objects names and collision normal. Extend the collision listener and add it to the Bounce World to get this information.

Vector2

Vector2 is a class that extends Point2D.Float, it has aditional methods for finding normal vectors and vectot opperations. Vector2 is not part of the Bounce package instead it is in a Util package. This is because it is used by all aspects of the game not just Bounce.