Rigid Body Dynamics: A Graphical Simulation

Eugene Paik
TJHSST
Alexandria, Virginia

April 2, 2009

Abstract

Dynamics is defined as the study of the interaction between masses based on the laws of physics. The possible applications of a computer simulation in which dynamics are simulated are endless. A working dynamics simulation would be useful to engineers who are interested in designing a building, students trying to grasp the law of conservation of momentum, or even video game programmers who want their games to be as realistic as possible. The goal of this project is to design and program a fully functional, efficient, rigid-body dynamics simulation capable of supporting objects of variable shape and mass. Because of the consistencies between two-dimensional and three-dimensional dynamics, this project will focus on only the two-dimensional plane for the sake of simplicity.

Keywords: rigid body dynamics, physics, simulation, graphics

1 Introduction

This project will be divided into two areas, the code behind making the dynamics work, and the graphics and interface of the simulation itself. While I have a basic understanding of physics in general, the greatest difficulty in programming this simulation will probably be handling collision response. When two bodies collide, the resulting impulse acts on both the linear and angular aspects of the bodies. The graphics and interface should be very simple and intuitive, allowing the user to create, delete, and move bodies throughout the world. The end result of this project will be a fully functional two-dimensional simulation of many convex two-dimensional rigid bodies in a world that the user can easily and intuitively interact with. This simulation will help the user understand how such bodies might act in real life. Ultimately, the project will examine different methods in which masses can be efficiently simulated. Through this project, I hope to garner a better understanding of dynamics in general. I chose this area of research because I am especially interested in the role of dynamics and physics in the gaming industry. The central focus of this project is to discover a new method of simulating dynamics that will aid future programmers in their applications of dynamics. This project will be purely applied research for now. I will be programming a working rigid-body simulation before trying to explore any new areas of research.

2 Background

In today's society, dynamics is still an area being researched and developed by scientists. While researching simular projects to mine, I have come across applications of dynamics in areas such as animation and game design. One particular set of articles of interest by Pixar Studios covers all of the elementary aspects of physically based modelling required to animate realistic characters in film. There are also various papers online that cover small areas of collision detection, collision response, or physics. Most researchers focus on only a small area of dynamics, such as friction or fluid dynamics. In contrast, my project will be approaching a much broader area of research, drawing on these other works for information.

3 Development

The first phase of this project is the design. A modular framework would be very useful because of the many different bodies in dynamics. This framework would have to be designed and programmed before anything else could be implemented. After this framework is finished, the two bodies that I would focus on implementing would be the box and circle. These bodies are the most commonly simulated bodies, and as a result should be completed first. I will be using Java and the Graphics2D library to program the simulation. The visuals of this project would be entirely based on the image produced while the simulation is running, along with some side-information that may be passed to the console for debugging purposes. The testing phase will be heavily interlaced with the development/programming phase, as I debug and fix code as I go along. Various tests will be observed such as bodies with random positions with random forces acting on them interacting with each other. To validate how well the program simulates dynamics, similar tests may be conducted in real life and compared to the tests in the simulation.

4 Overview

The main driver of the simulation is what handles the entire program. It does everything from calculating whether collisions occur, to determining what the appropiate response is to these collisions. Because of this, it makes sense that the driver is efficient, as well as robust. The most important object that the driver interacts with is the polygonal mass class. This class keeps track of the object's mass, position, orientation, speed, acceleration, as well as state. The class uses public variables for ease of use, and includes several methods to update its quantities.

5 Collision Detection

Collision detection in this simulation is done through application of the seperating axis theorem. This theorem postulates that, if there exists a plane between two objects, then the two objects must not be colliding. While at first this seems like a very simple observation, it becomes infinitely useful once one realizes that there exist a limited number of axis to test for seperation between two objects. These axis of seperation are generated by finding the perpendicular normal vectors of every edge of a polygon. Then, using basic vector maths, its possible to determine whether the two polygons intersect over the axis.

6 Collision Response

Using an impulse-based system, the dynamics simulation generates impulse for every collision that occurs. In a simulation of a large number of bodies, this means that there are a great number of impulses generated every frame of the simulation. One problem with this is that impulses can be conflicting in some cases, resulting in bodies in the simulation that are misrepresented. To fix this problem, I implemented a correcting method that essentiall forcibly seperates each body before applying impulses, based on the seperating axis theorem once again.

7 Rotating Bodies

Adding the ability for bodies to rotate in a 2D simulation opens up a number of problems. First and foremost, every body in the simulation has more variables to keep track of, such as angular velocity, angular acceleration, orientation, and moment of inertia. Keeping track of all of these variables results in a higher cost of running the simulation. In addition, the collisin detection code must be updated to take into account bodies that are rotating very quickly. Calculating collision response for rotating bodies follows similar rules to basic linear collision response.

8 Discussion

Although the simulation is far from being flawless at this point, its apparent that the collision detection method is robust and effective, determined through a series of visually confirmed tests. There appears to be no slow-down of the simulation, even when simulating over twenty complex, rotating and moving polygonal bodies at once. I do expect this to change once friction is completely integrated, considering how complicated the subject may be.