

# Panda3D

## New Features to the Collision Detection System

### Student Information

- **Name:** Christopher He, [hecris](#)
- **University:** CUNY Hunter College/Undergraduate/Class of 2022
- **Time Zone:** UTC-4

### Code Samples

🔗 [Pull Request: Add Collision Tests #583](#)

🔗 [Pull Request: Add More Collision Tests #609](#)

In these pull requests, I wrote test cases for several collision scenarios between different solids. In addition, I provide a foundation for future contributors, including myself, to easily write tests for collisions between any two solids.

### Project Info

#### Abstract

Collision detection is essential in a game engine. It is the reason you go bonkers playing Flappy Bird. However, it would be difficult and expensive to represent each object using its exact geometry, so a better idea would be to put those objects in bounding volumes or CollisionSolids. These CollisionSolids are mathematically defined, so by using *some math*, a collision system would be able to detect their intersections with each other. Additional information on the collision should be provided, including the point of intersection and the vector normal to the contact (to push back on).

Currently, Panda3D is missing 3 collision tests: parabola into box, parabola into inverse sphere, box into capsule. Here are some use cases that we've probably seen before:

1. An Angry Bird hitting a box (parabola into box)
2. Protecting the audience from a football in a spherical stadium (parabola into inverse sphere)
3. Some box object hitting a player (box into capsule)

My first goal would be to add these collision tests.

My second goal would be to add another CollisionSolid named CollisionHeightfield. The idea is that we can represent heightfields using a grayscale image, with lighter (taller) and darker (lower) pixels. We can use this concept to efficiently deal with collisions in uneven terrain, rather than representing the terrain as individual CollisionPolygons.

### Implementation Details

#### Preface

Since part of the project is figuring out the algorithms to use, this section consists of investigation on existing collision tests in Panda3D, and how they can be used as bases. At the end, I give a Minimal-Viable-Product outline of CollisionHeightfield.

## Details

- Parabola into InvSphere

Currently, there exists a Parabola into Sphere test, which splits the Parabola into tiny line segments and checks if they intersect the Sphere. Implementing Parabola into InvSphere will use the previous method as a basis, but needs further investigation on how it differs from the regular Sphere. I also have to keep in the mind edge cases, which includes the Parabola being entirely inside the bubble, or entirely outside.

- Parabola into Box

Since the Box is not quadratic, the previous method is not necessary. The Parabola into Box test might resemble the Parabola into Polygon test, which uses the Parabola into Plane test. The Parabola into Plane test uses a quadratic equation, implementation can be found [here](#). Edge cases include the Parabola being entirely inside the Box.

- Box into Capsule

This one I'm not too sure about. For inspiration, I've taken a look at [Ericson's chapter on Sphere-Swept Volumes](#) as well as [Capsule into Box](#) and [Box into Sphere \(Arvo's Algorithm\)](#), but further investigation is required.

- CollisionHeightfield

- MVP

1. Can be loaded from PNMIImages\*
2. Can be read/written from/to BAM files\*\*
3. Test intersection from sphere, box, and lines (rays, segments)

- Extra Features

1. Test intersection from capsule
2. Can be read/written from/to EGG files\*\*\* (will have to add new syntax)

\* PNMIImages are 2D arrays of pi(xels). For CollisionHeightfield, we need only a grayscale image, so the xels will have gray components

\*\* BAM files allow users to store Panda3D objects as a file, but is not human-readable

\*\*\* EGG files are the same except they are human-readable, thus the need for new syntax

## Timeline

Week	Dates (Mon - Sun)	Description
Community Bonding	05/06 - 05/27	<ul style="list-style-type: none"><li>• Review related math topics</li><li>• Algorithm study and research</li></ul>
Phase 1		
1	05/27 - 06/02	<ul style="list-style-type: none"><li>• Implement parabola into box detection</li><li>• Write unit tests for parabola into box collisions.</li></ul>
2	06/03 - 06/09	<ul style="list-style-type: none"><li>• Implement parabola into inverse sphere detection</li><li>• Write unit tests for parabola into inverse sphere collisions.</li></ul>
3	06/10 - 06/16	<ul style="list-style-type: none"><li>• Implement box into capsule detection</li><li>• Write unit tests for box into capsule collisions.</li></ul>
4	06/17 - 06/23	<ul style="list-style-type: none"><li>• Discuss with mentors for requirements specification on CollisionHeightfield</li><li>• Begin implementing CollisionHeightfield MVP</li></ul>

		<ul style="list-style-type: none"> <li>◦ Generate from PNMIImage</li> <li>◦ Test collision from Ray</li> </ul>
Phase 2		
5 & 6	06/24 - 06/30 07/01 - 07/07	<ul style="list-style-type: none"> <li>• Continue implementing CollisionHeightfield <ul style="list-style-type: none"> <li>◦ Test collision from Segment</li> <li>◦ Test collision from Line</li> <li>◦ Test collision from Sphere</li> </ul> </li> </ul>
7 & 8	07/08 - 07/14 07/15 - 07/21	<ul style="list-style-type: none"> <li>• Finish implementing CollisionHeightfield MVP <ul style="list-style-type: none"> <li>◦ Test collision from Box</li> <li>◦ Bam serialization</li> <li>◦ Possibly implement <a href="#">extra features</a></li> </ul> </li> </ul>
Phase 3		
9 & 10	07/22 - 07/28 07/29 - 08/04	<ul style="list-style-type: none"> <li>• Write more unit tests for all collisions with CollisionHeightfield</li> <li>• Write documentation for the <a href="#">3 new collision tests</a> and CollisionHeightfield</li> </ul>
11 & 12	08/05 - 08/11 08/12 - 08/19 (Monday)	<ul style="list-style-type: none"> <li>• Make a sample game that implements CollisionHeightfield and add it to panda3d/samples</li> <li>• Complete any last unit tests and documentation</li> </ul>

## More information

### Roadblocks

One the most challenging aspect of this project is that it requires an understanding of 3D space and linear algebra. I do have a introductory understanding of 3D space, but I will have to self-study some linear algebra. Thankfully, the [Collision Detection Book](#) has an overview of the math required for this project, which will be very useful. I will be utilizing the book and other online resources to prepare myself with the math required for this project.

### Research Materials

- [Erison, C. \(2005\) Real-Time Collision Detection Book](#)
- [Bullet Heightfield Shape](#)