Custom Physics Documentation

Eight-Ball

Connor Mills

2023

Contents

[1.0 - Custom Physics Simulation Class Diagram 2](#_Toc125539830)

[2.0 - Custom Physics Simulation Interactions 3](#_Toc125539831)

[3.0 - Custom Physics Simulation Potential Improvements 4](#_Toc125539832)

[3.1 - Improvement #1 4](#_Toc125539833)

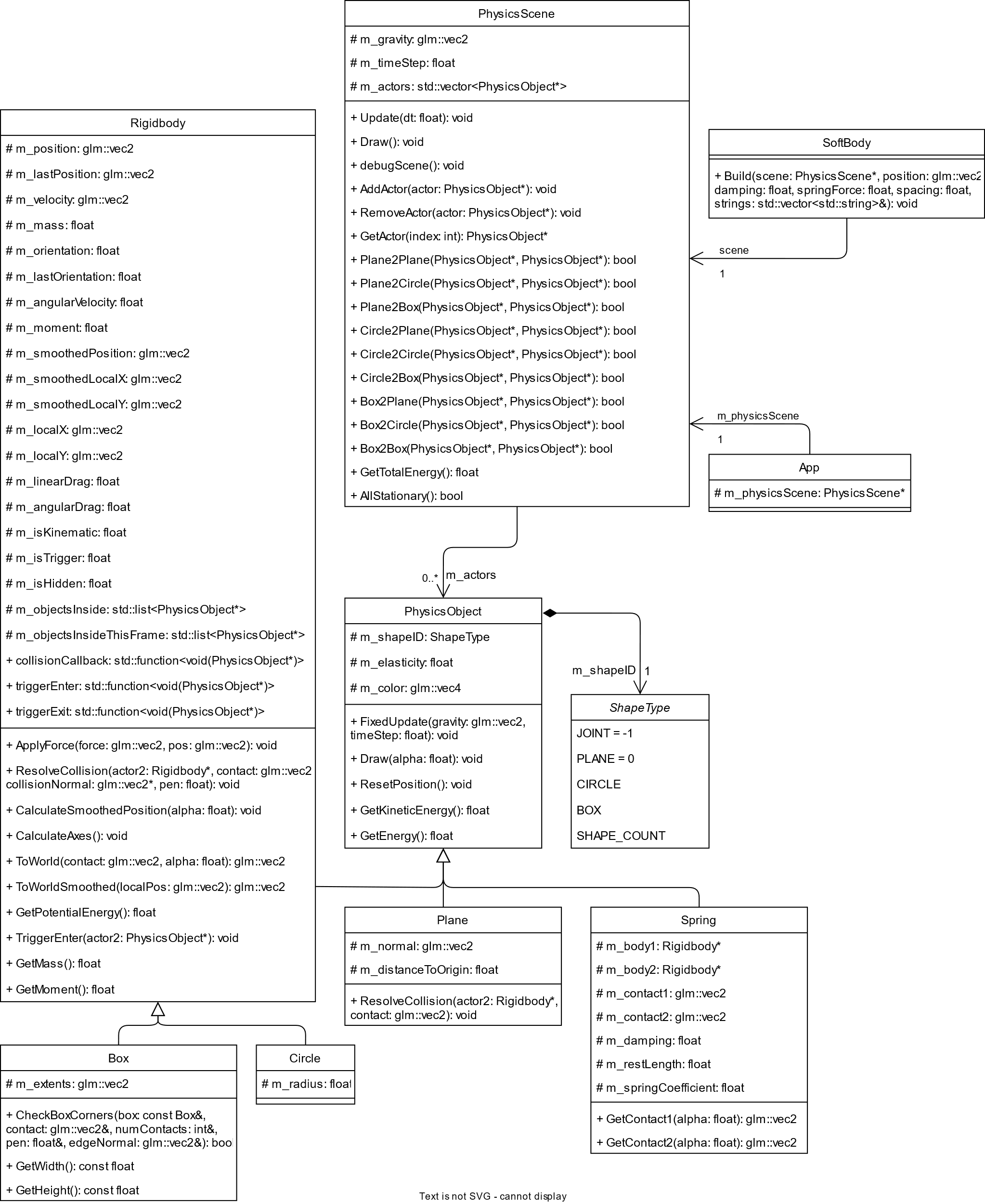
[3.2 - Improvement #2 4](#_Toc125539834)

[4.0 - Visualised Game Using Your Custom Physics Simulation 5](#_Toc125539835)

[5.0 - Third Party Libraries 6](#_Toc125539836)

[6.0 - References 6](#_Toc125539837)

# 1.0 - Custom Physics Simulation Class Diagram



# 2.0 - Custom Physics Simulation Interactions

I created my Custom Physics Simulation to demonstrate simple physics and to visualise the interactions and outcomes of different situations. The Custom Physics Simulation can be used as a controlled environment to demonstrate how objects interact and behave in certain situation so that experimentation can be done quicker.

The Custom Physics Simulation has four Physics Objects, box, circle, plane, and spring. The box and circle are both rigidbodies that can be either dynamic or static. They are the only Physics Objects that react to game physics. The plane is a static object that is just a line that span across the screen. The spring is non-interactable object that only applies forces to other objects. It acts just like a real-world spring in that it tries to pull two objects together.

In the simulation boxes, circle and planes can all interact with each other.

If both objects are dynamic and they collide, depending on their mass the objects will be moved with different forces applied to them. For example, if a circle with a mass of 10 moving a 2m/s collides with another circle with a mass of 10 that is stationary, then a force of 2m/s will be applied to the second circle and a opposite force of -2m/s will be applied to the first circle coursing it to stop. If you increase the mass of the second circle, then it will be harder to accelerate meaning the other circle will be accelerated more making it move backwards instead of being stationary.

If one object is static and the other is dynamic then none of the force can be applied to the static object so all of it is applied to the dynamic object. This means that the dynamic object will bounce of the static object with the exact speed as it collided with.

# 3.0 - Custom Physics Simulation Potential Improvements

## 3.1 - Improvement #1

An improvement that could be implemented in the physics simulation is better friction and angular velocity. The friction and angular velocity that is currently in the simulation is very basic and not accurate. Angular velocity just spins the object and doesn’t change anything with the physics of the object.

The improvement I am suggesting is to include angular velocity in the collision detection. So for example when a ball rotating clockwise collides with a wall (static box) it applies a force on the wall moving clockwise and conversely the wall applies a force to the ball moving in counter-clockwise direction. The counter-clockwise force that the wall applies to the ball will push it in that direction and course the ball to spin counter-clockwise.

## 3.2 - Improvement #2

An improvement that could be used for the physics simulation is using Quadtrees. Quadtrees is a method of dividing up the screen and when collision detection is run it first checks if the to objects could collided by looking at the quadtree and checking if they are in nearby areas of the screen. This would improve the quality of the physics simulation by cutting down on unnecessary uses of collision detection on objects that could not have possibly been touching, for example two objects that are on opposite sides of the screen.

A Quadtree is basically a binary tree but instead of each node only have two children it has four. The quadtree starts of as a single node but as objects are added it splits into four subnodes. Each object is put into these subnodes according to their position on the screen and any object that doesn’t fit into a single nodes area it is put in the parent node. Each of these subnodes can then also be split into four subnodes further dividing the screen up.

Chart, scatter chart

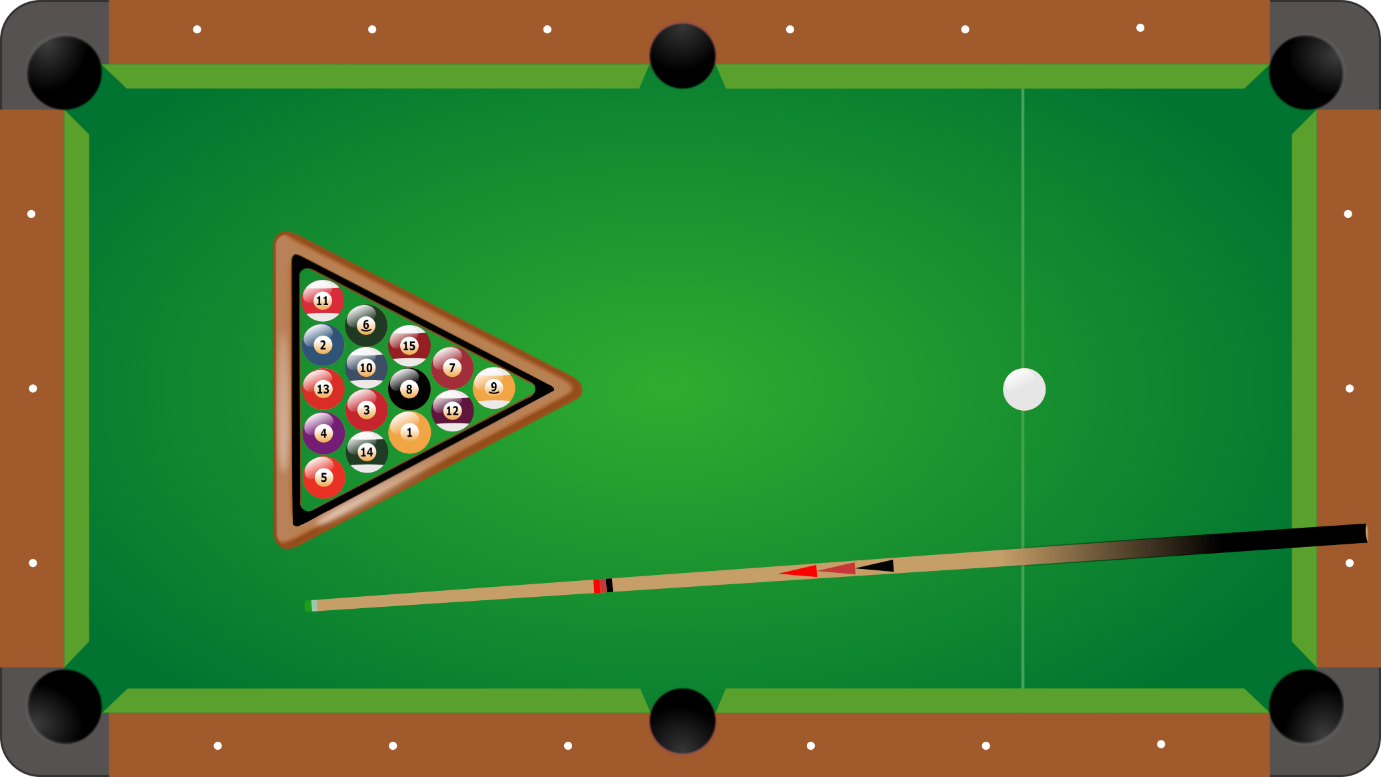
Description automatically generated Chart, qr code

Description automatically generated

This allows for easy and faster collision detection as the only objects that an object can collide with are objects in the same node, in the parent nodes or the children nodes.

# 4.0 - Visualised Game Using Your Custom Physics Simulation

The game that I chose to make was Eight-Ball Pool. Eight-Ball Pool is a discipline of Pool played with sixteen billiard balls including the cue ball and fifteen object balls, cue stick and played on a billiard table with six pockets. The object balls are split into two types striped and solid with an additional black ball. The side a player is on is decided by which ball is sunk first and by whom. To win the game a player must first sink all of their balls and then the black ball.



To created Eight-Ball Pool I first created a box out of kinematic boxes for the boundaries with gaps, for the pockets, in the corners and the middle of the top and bottom sides. Then in those gaps I added circles, as the pockets, that where set as triggers. This meant that any object within the box would bounce of the walls and then when it passed through one of the gaps and into the circles I will know and be able execute code based on what passed through.

The object balls have their own class that is a child of the circle class from the physics library. The object balls would act just like the standard circle in the physics library with one additional bit of information its state as an integer. 0 for the black ball, 1 for solids and 2 for stripes. This allows me to tell which ball is sunk.

The cue ball also has a class of its own which is also a child of the circle class. The cue ball class stores information about if the mouse button is being held, current position of the mouse, position of the mouse when it was pressed and when it is released. This allows the player to control how much power to apply to the cue ball and when to apply it by pressing and dragging the mouse to increase the power and then releasing the apply that power to the cue ball.

This all comes together, with some code to check the game state and the state of the balls, to create the game Eight-Ball Pool.

# 5.0 - Third Party Libraries

The only third-party non-physics library that was used was Bootstrap. Bootstrap was the library provided by AIE, I used for rendering and visualising all the physics objects in the eight-ball game. I also used it for getting the input information from the mouse and keyboard. I did not use any other third-party libraries as they were not needed.

# 6.0 - References

*Newton's laws of Motion Tutorial* (no date) *The Physics Classroom*. The Physics Classroom. Available at: https://www.physicsclassroom.com/Physics-Tutorial/Newton-s-Laws (Accessed: February 20, 2023).

*World 8 ball rules* (no date) *West Australian Eight Ball Federation Inc*. Available at: https://poolwest.com.au/world-8-ball-rules/ (Accessed: February 13, 2023).

Game\_assets\_66 (2016) *8-ball pool assets*, *OpenGameArt*. OpenGameArt.org. Available at: https://opengameart.org/content/8-ball-pool-assets (Accessed: February 13, 2023).

Lambert, S. (2012) *Quick tip: Use quadtrees to detect likely collisions in 2d Space*, *Game Development Envato Tuts+*. Envato Tuts. Available at: https://gamedevelopment.tutsplus.com/tutorials/quick-tip-use-quadtrees-to-detect-likely-collisions-in-2d-space--gamedev-374 (Accessed: February 20, 2023).

Kang, J. (2014) *An interactive explanation of quadtrees*, *An interactive explanation of Quadtrees.* Available at: https://jimkang.com/quadtreevis/ (Accessed: February 20, 2023).

user249315, Jacob Bach and Claudio Saspinski (2019) *Acting forces on a curve ball (billiards)*, *Physics Stack Exchange*. Available at: https://physics.stackexchange.com/questions/518823/acting-forces-on-a-curve-ball-billiards (Accessed: February 20, 2023).