Interactive Falling Object Simulator
Report

Submitted for the MSc in
Advanced Computer Science

December 17

By
Alexander C Whitehead

Word Count: 1642

Table of Contents

Table of Figures

1 Introduction

This is a report describing how the motion physics, collision detections and responses have been implemented in the project to create an interactive falling object simulator.

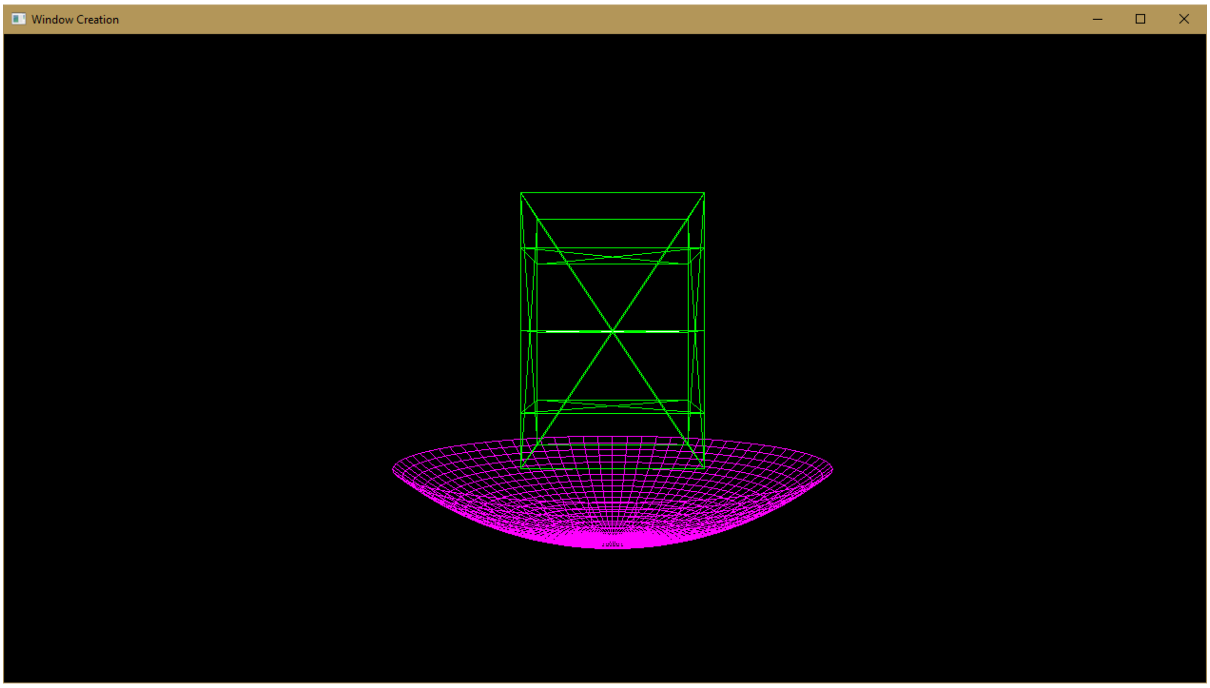

Figure 1: This image shows the solution to the project to create an interactive falling object simulator.

2 Motion Physics

2.1 Ball and Cube

There are two numerical integration methods which can be used to calculate the velocity and position of each ball and cube for each frame of the simulation. These numerical integration methods take the time since the last frame and the acceleration, in this case the acceleration of the object due to gravity, of the object in as arguments.

The first numerical integration method used is the semi-implicit Euler integration method:

$$Velocity = Velocity + Acceleration * DT$$

$$Position = Position + Velocity * DT$$

This is the integration method used by most commercial game physics engines as it balances performance and accuracy.

The second numerical integration method used is the Runge Kutta 4th order integration method:

$$Y = F(T, X)$$

$$Yn+1 = Yn + 1/6 * (K1 + 2 * K2 + 2 * K3 + K4) * DT$$

$$K1 = F(Tn, Xn)$$

$$K2 = f(Tn + ½ * DT, Xn + ½ * K1)$$

$$K3 = f(Tn + ½ * DT, Xn + ½ * K2)$$

$$K4 = f(Tn + DT, Xn + K3)$$

Runge Kutta 4th order integration as its name would suggest has an accumulated error which is on the order of the 4th derivative, meaning that it is very accurate far more accurate than even semi-implicit Euler, which is a 1st order integration method. However this does not mean that Runge Kutta 4th order integration is superior to semi-implicit Euler, because of its nature Runge Kutta is far more complicated to implement and uses up more resources to calculate.

The method of integration can be swapped in the simulation using a keypress.

2.2 Plane

The velocity, position and therefore overall energy of the plane is less important than that of the ball object and cube object, as covered above.
Thus to move the plane out of the box in a given time a fixed velocity relative to the distance to be moved and the time to be moved in is given to the plane, then a simple version of Euler is used to calculate the planes new position for each frame of the simulation.

3 Collision Detection

3.1 Ball to Ball and Cube to Cube

In order to properly simulate the reaction of two objects colliding the point when those two objects first touch needs to be found. In these calculations each object is treated as a point with dimensions.

To improve the efficiency of the calculations used to determine collisions objects which are in the box can only collide with other objects in the box and the box itself, whereas objects in the bowl or net can only collide with other objects in the bowl or net and the bowl or net itself.

To find if two balls are colliding and to find the point when this happens on each frame, each ball in each section of the simulation is checked to find if the distance between the centres of each ball is less than or equal to the combined radius of the balls concerned, if this is true a pointer to each ball is added to a list known as the ball collision manifold.

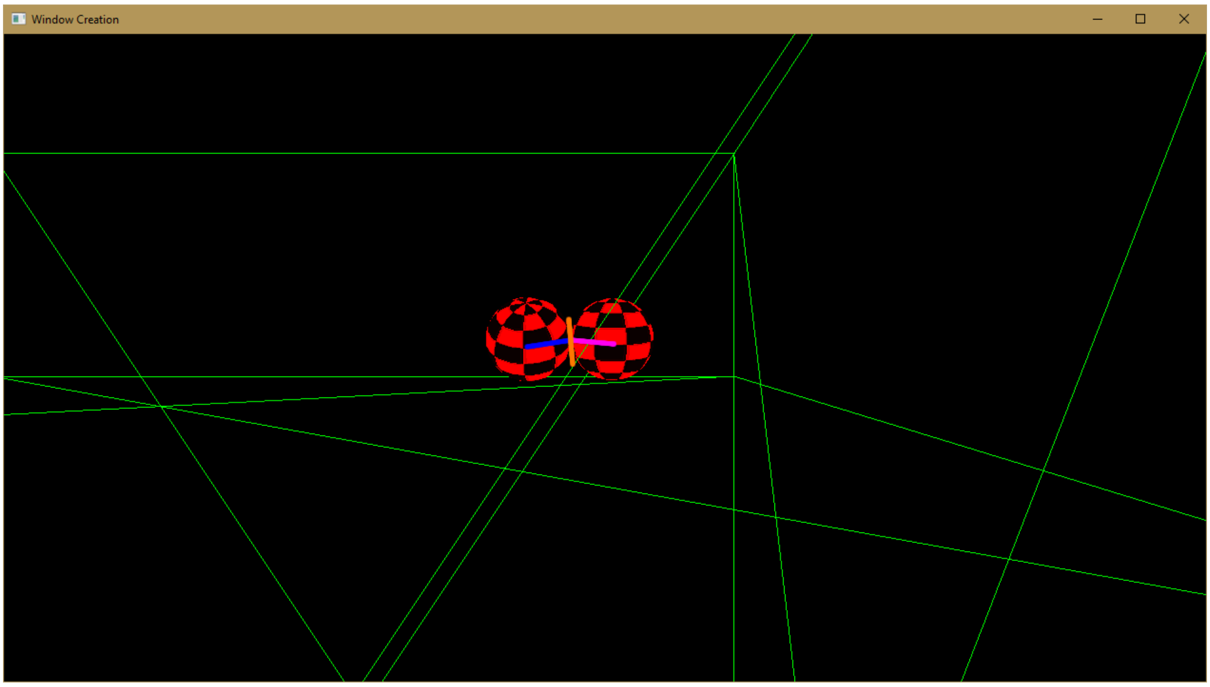

Figure 2: The above image shows two balls colliding.

The image above shows two balls which have just collided head on, the pink line shows the radius of the ball on the left and the indigo line shows the radius of the ball on the right. The point where the blue line and the pink line touch is the point where the collision takes place. The orange line shows the tangent to the collision.

In this simulation balls are used to represent the dimensions of the cube objects, ideally the object would be 'tessellated' and multiple balls would be inserted inside the cube object to represent the cube to a higher resolution. However, in this solution very few balls are used to represent the cube object.

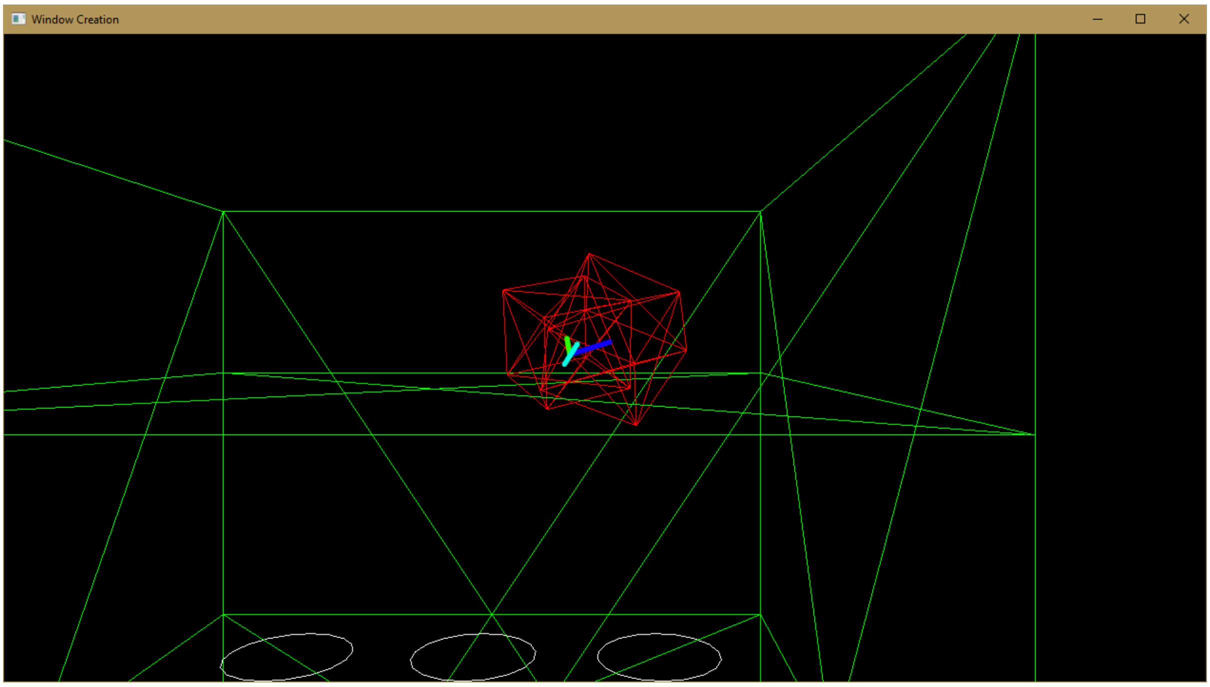

Figure 3: The above image shows two cubes colliding.

The image above shows two cubes which have just collided at quite an acute angle, the indigo line shows the radius of the ball used to represent the cube on the right and the green line shows the radius of the ball used to represent the cube on the left. The point where the blue line and the pink line touch is the point where the collision takes place. The cyan line shows the tangent to the collision.

3.2 Ball to Cube and Cube to Ball

The calculation to find if a ball and a cube are colliding is identical to the method mentioned above. If these objects are found to be colliding a pointer to the ball and a pointer to the cube is added to a list known as the ball cube collision manifold.

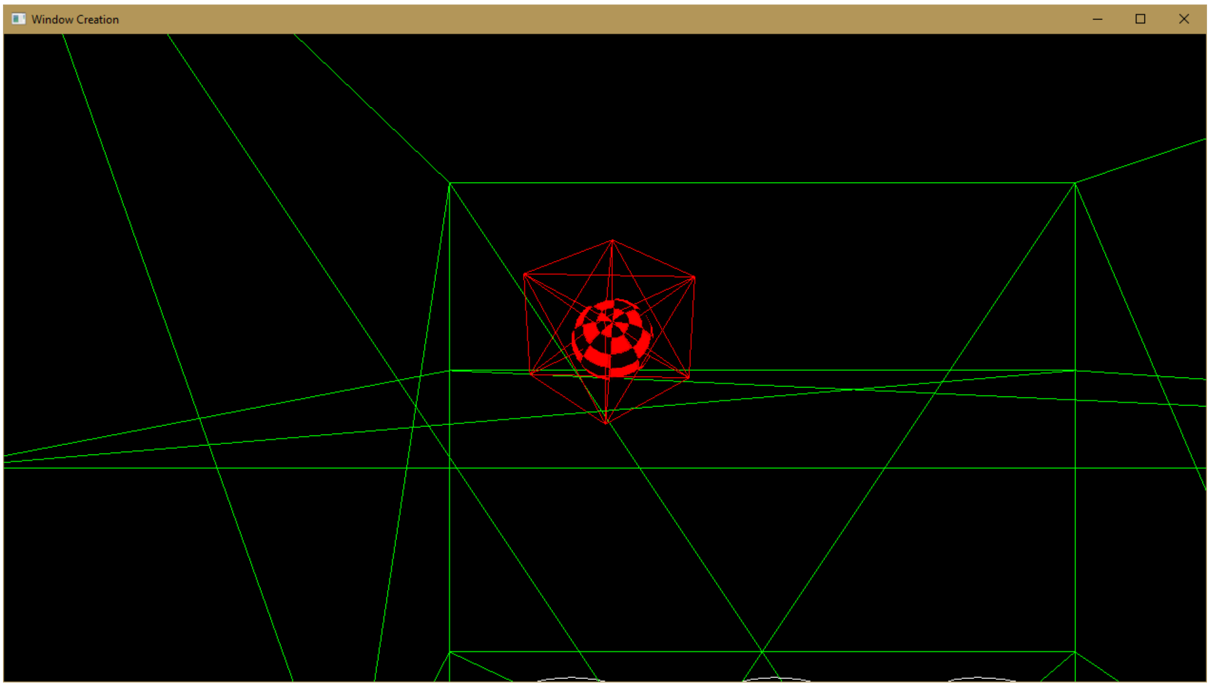

Figure 4: The above image shows a ball and a cube colliding.

3.3 Ball to Plane and Cube to Plane

To find if an object is colliding with a plane on each frame, each object in the box is checked to find if the distance between the centre of each object and a point on the plane where the normal of the plane intersects the centre of the object is less than or equal to the radius of the object concerned. An additional check is also performed to determine if an object has moved from one side of the plane concerned to the other side of the plane concerned during the simulation period. If either of these situations has occurred then the object has collided with the plane.

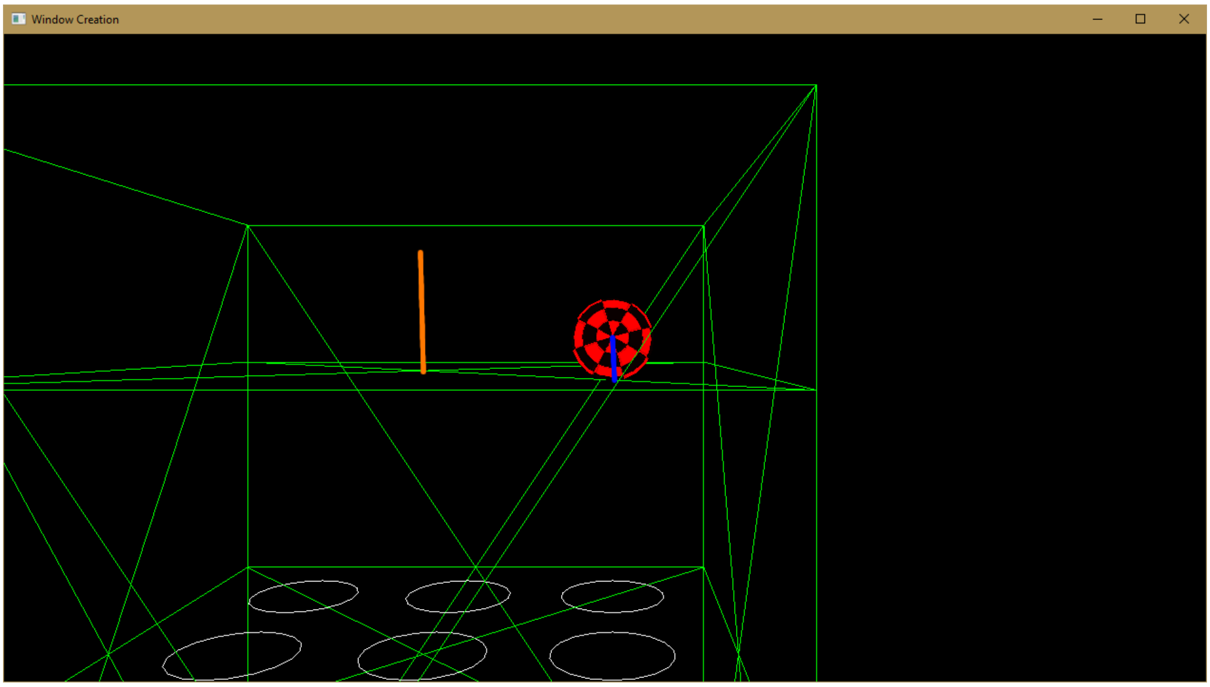

Figure 5: The above image shows a ball colliding with a plane

The image above shows a ball which has just collided with a plane, the indigo line shows the radius of the ball and the orange line shows the plane normal at the centre of the plane.

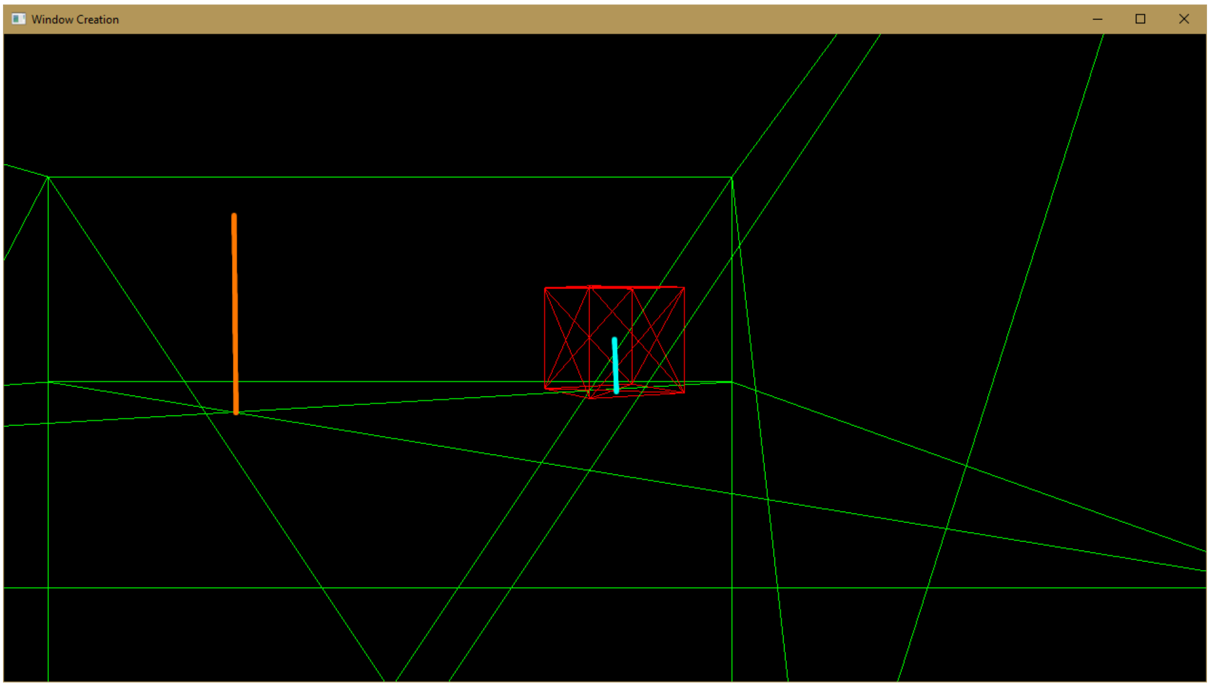

Figure 6: The above image shows a cube colliding with a plane.

3.4 Ball to Bowl and Cube to Bowl

To find if an object is colliding with the bowl on each frame, each object in the bowl is checked to find if the distance between the centre of each object and the centre of the bowl is greater than or equal to the radius of the bowl minus the radius of the object.
The collision detection is performed this way on the bowl because it is highly efficient as only one calculation is needed to determine if an object has collided with the bowl.

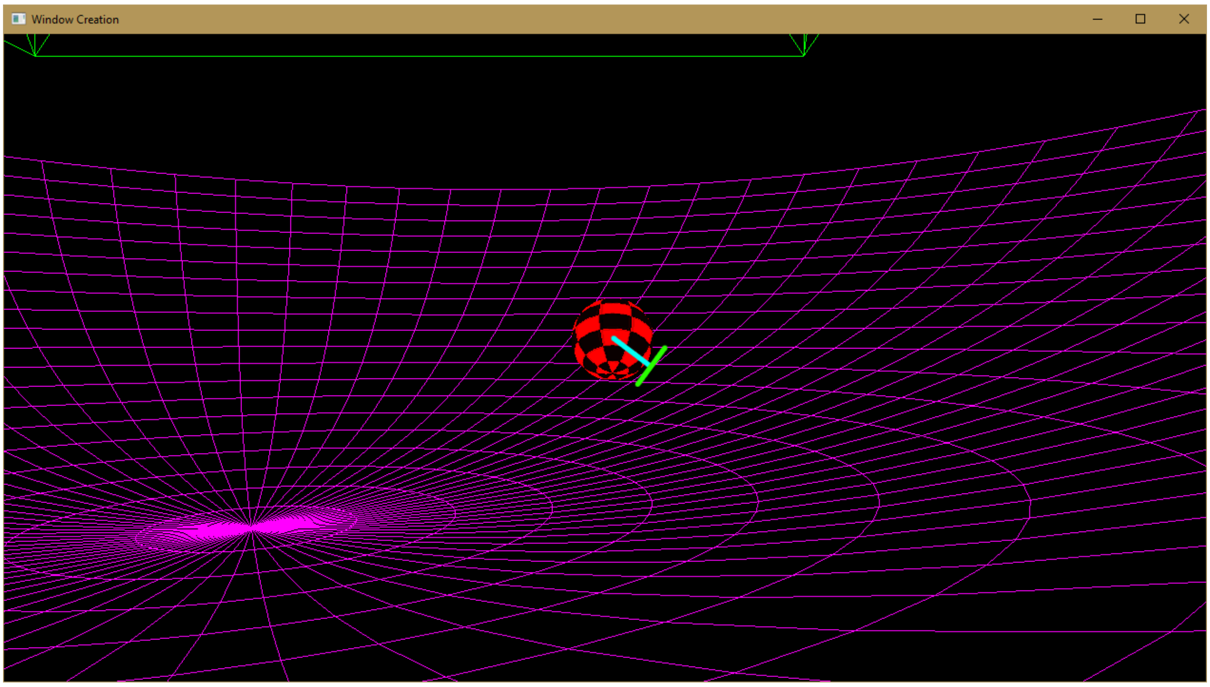

Figure 7: The above image shows a ball colliding with the bowl.

The image above shows a ball which has just collided with the bowl, the cyan line shows the radius of the ball and the green line shows the tangent to the collision.

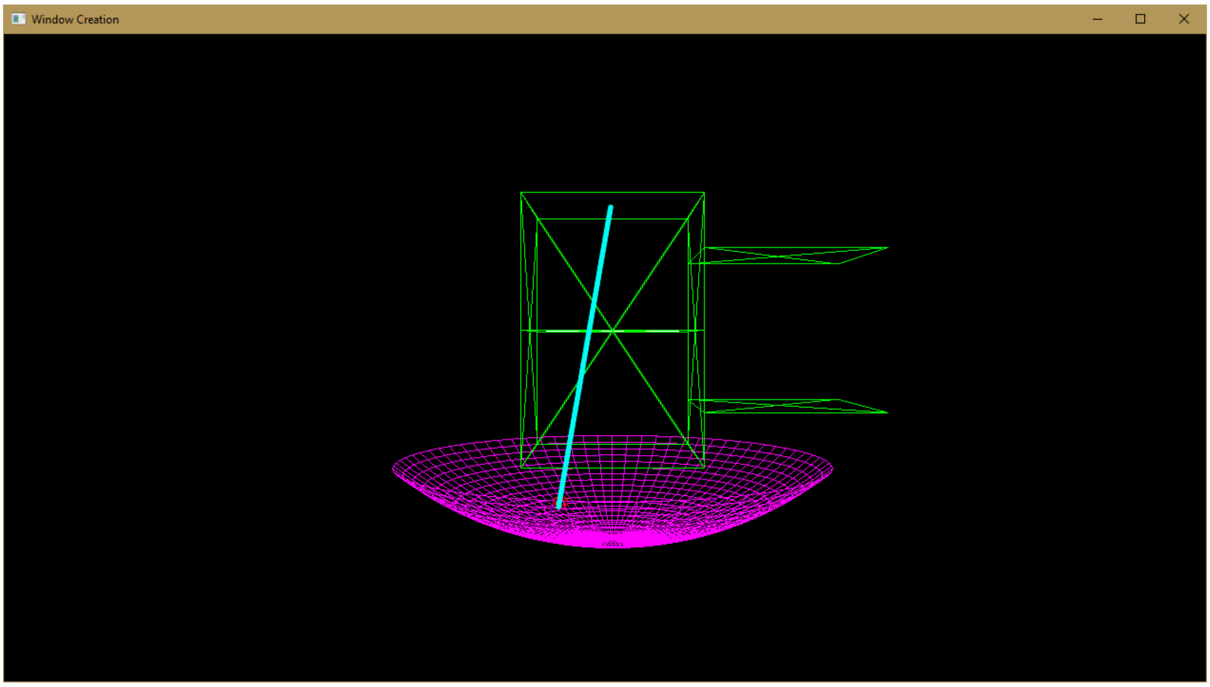

Figure 8: The above image shows the radius of the bowl.

The image above shows a ball which has just collided with the bowl, the cyan line shows the radius of the bowl.

3.5 Ball to Net and Cube to Net

To find if an object is colliding with the net on each frame, each object in the net is checked to find if the distance between the centre of each object and each point on the net less than or equal to the radius of the object.
The collision detection is performed this way on the net because it would allow calculations to be performed to deform the net after a collision.

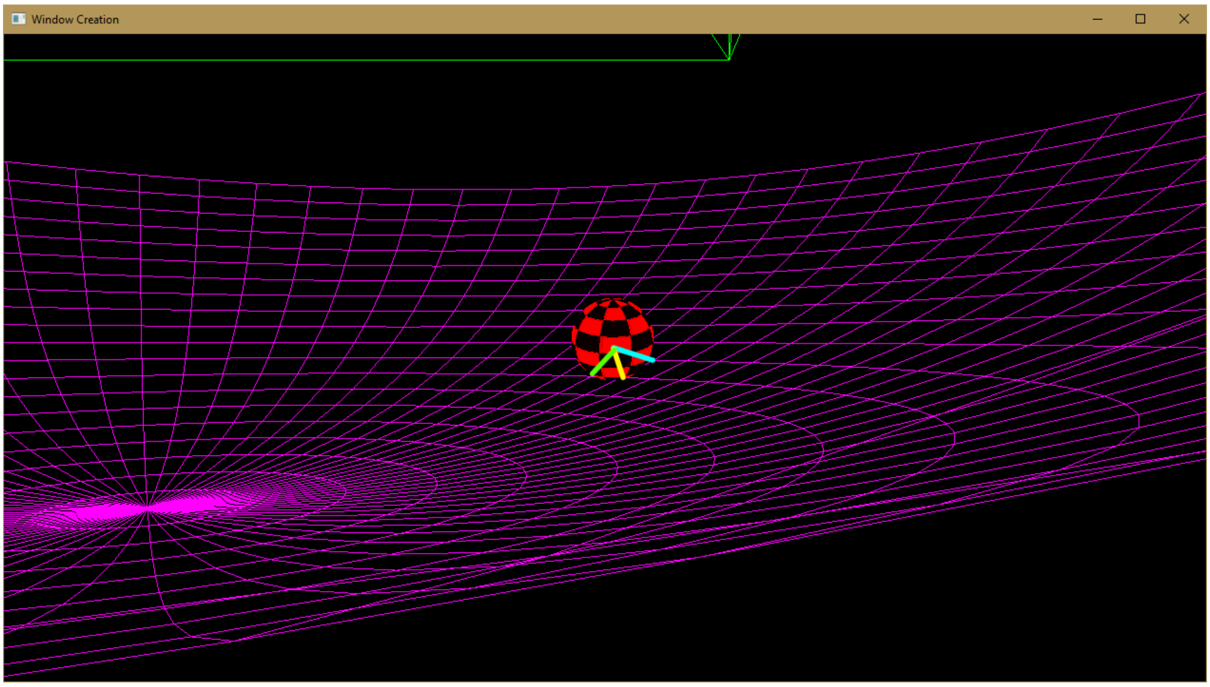

Figure 9: The above image shows a ball colliding with the net.

The image above shows a ball which has just collided with the net, the lines shows the radius of the ball to 3 different points on the net.

3.6 Plane

The motion of the planes is always purely in one direction thus the collision detection can be much simpler that that discussed above.
To detect if a plane has moved outside the box each frame the position of the first vertex of the plane, which is always the top left vertex, is checked to see if it is beyond the box.

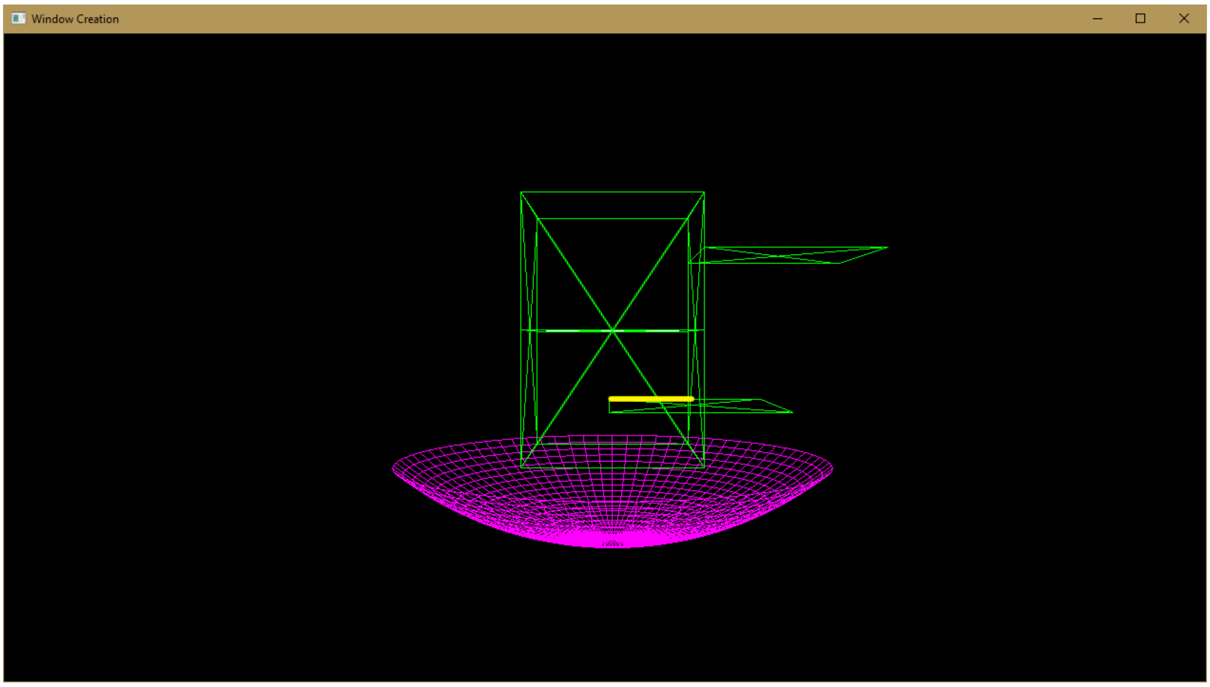

Figure 10: The above image shows a plane which has collided with the box and a plane which is moving to collide with the box.

The image above shows a plane which has collided with the box and another which is moving to collide with the box, the yellow line shows the distance from the first vertex to the box.

4 Collision Response

4.1 Ball to Ball, Cube to Cube, Ball to Cube and Cube to Ball

Once collisions have been detected and added to their relevant lists these lists are iterated over and each collision in turn is resolved.
The objects are taken and the magnitude of the force impacted in the collision based on the objects mass is amended to their velocity with regards to the collision normal.
To ensure that the simulation is as realistic as possible collision detections and collision responses are repeated multiple times during each frame for each object with a small margin for error.

When two objects collide there collisions are not usually entirely elastic and things such as friction rob energy from the objects, these factors are also applied in the collision response.

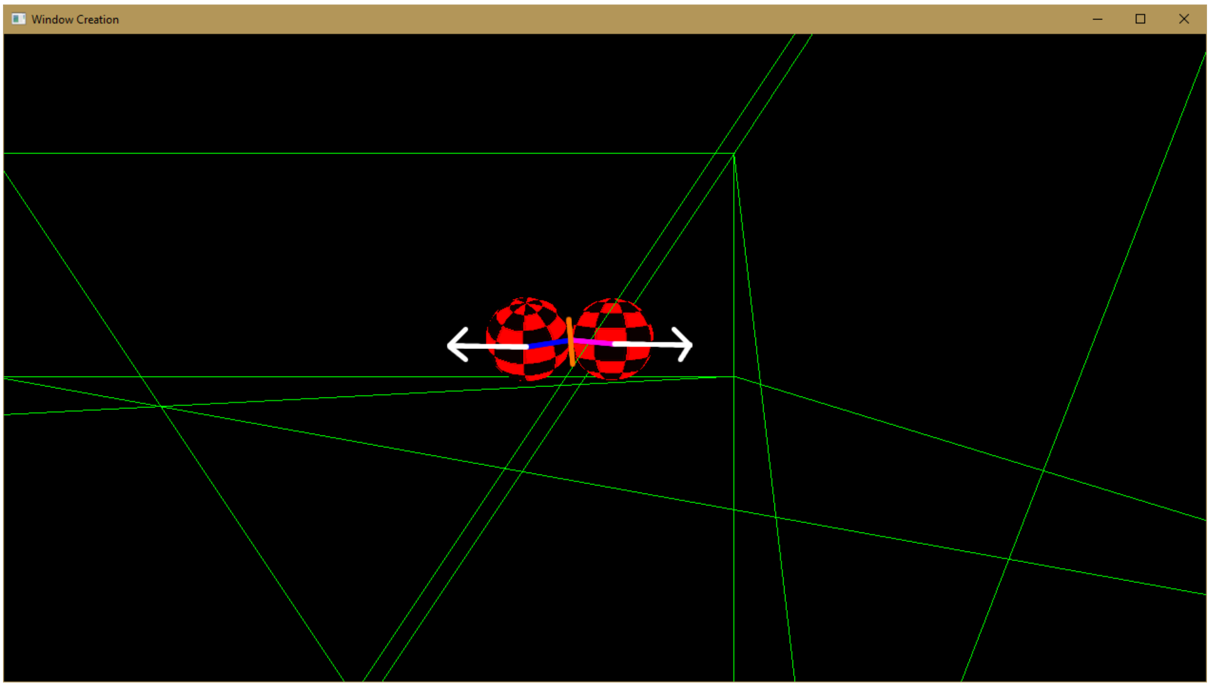

Figure 11: The above image shows the resultant velocities after two balls have collided.

The image above shows the resultant velocities for two balls which have just collided, as can be seen these velocities are in roughly opposite directions.

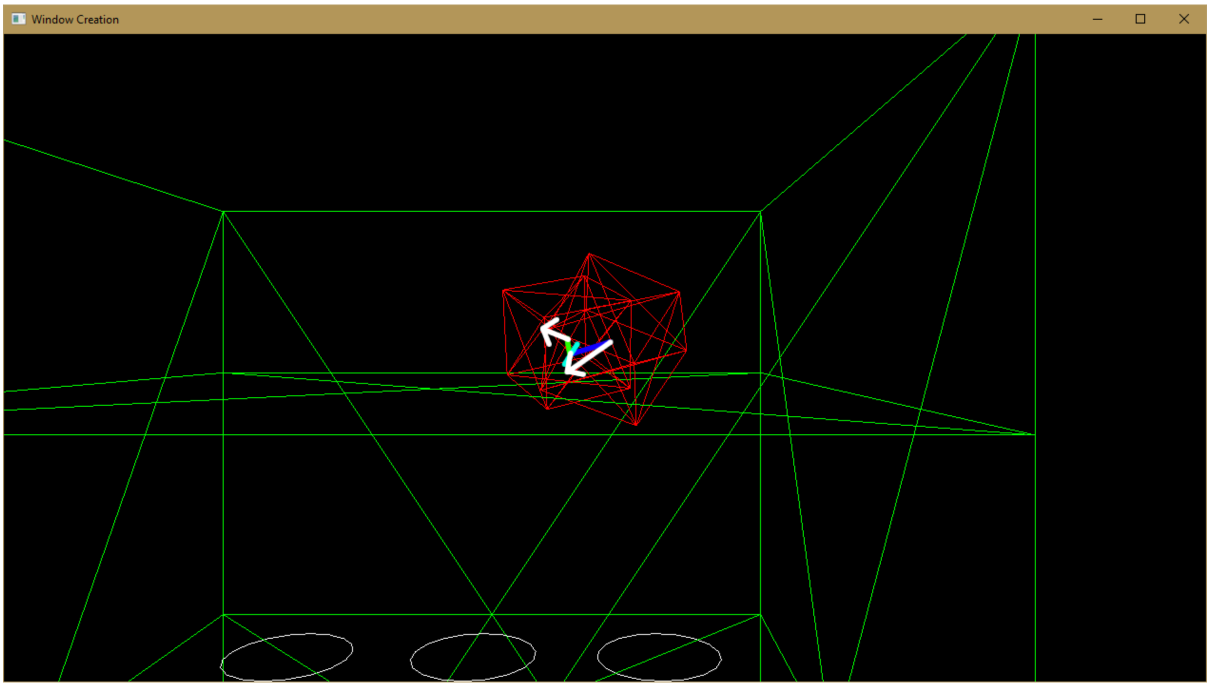

Figure 12: The above image shows the resultant velocities after two cubes have collided.

4.2 Ball to Plane and Cube to Plane

The collision response between an object and a plane is calculated in much the same way as discussed above, obviously a resultant velocity is not applied to the plane however as it is considered to be immovable.

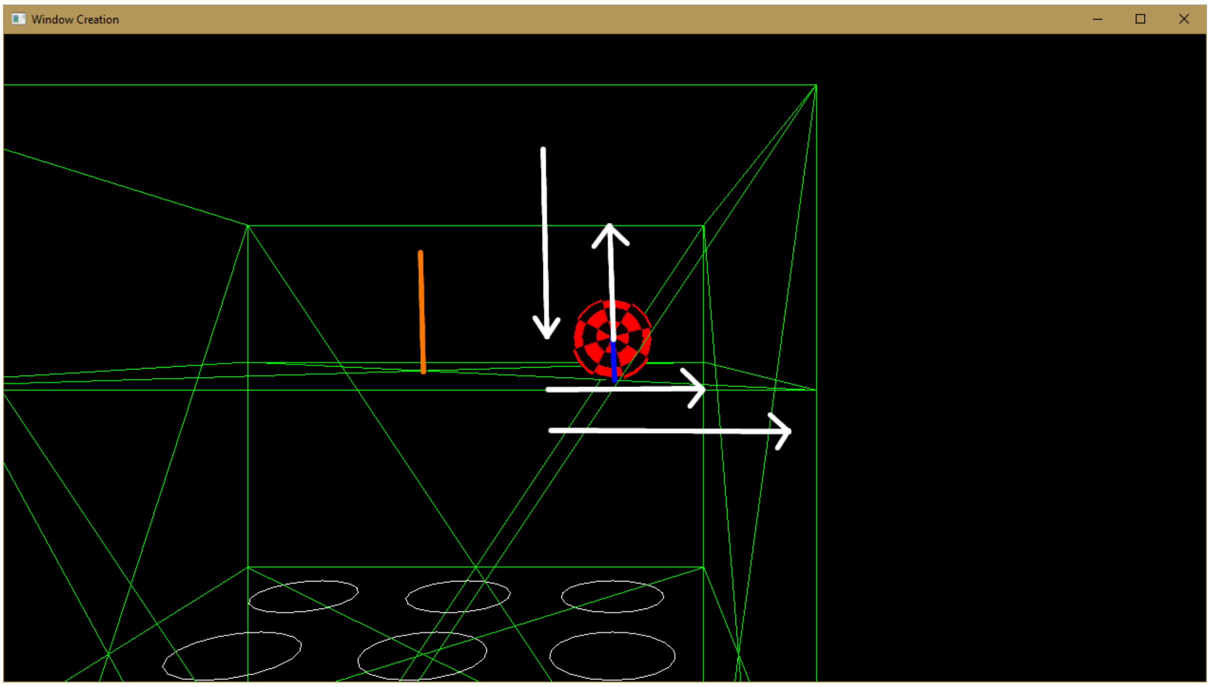

Figure 13: The above image shows the resultant velocity after a ball has collided with a plane.

The image above shows the resultant velocities in both the X and Y plane after a ball has collided with a plane. The vertical arrow on the far left shows the balls initial velocity in the Y plane, as can be seen from the balls resultant velocity next to it the elasticity of the collision robs the ball of a significant amount of velocity. The horizontal arrow at the bottom of this image shows the balls initial velocity in the X plane, as can be seen from the balls resultant velocity above this the friction of the plane robs the ball of a significant amount of velocity.

4.3 Ball to Bowl and Cube to Bowl

The collision response between an object and the bowl is calculated in much the same way as discussed above, obviously a resultant velocity is not applied to the bowl however as it is considered to be immovable.

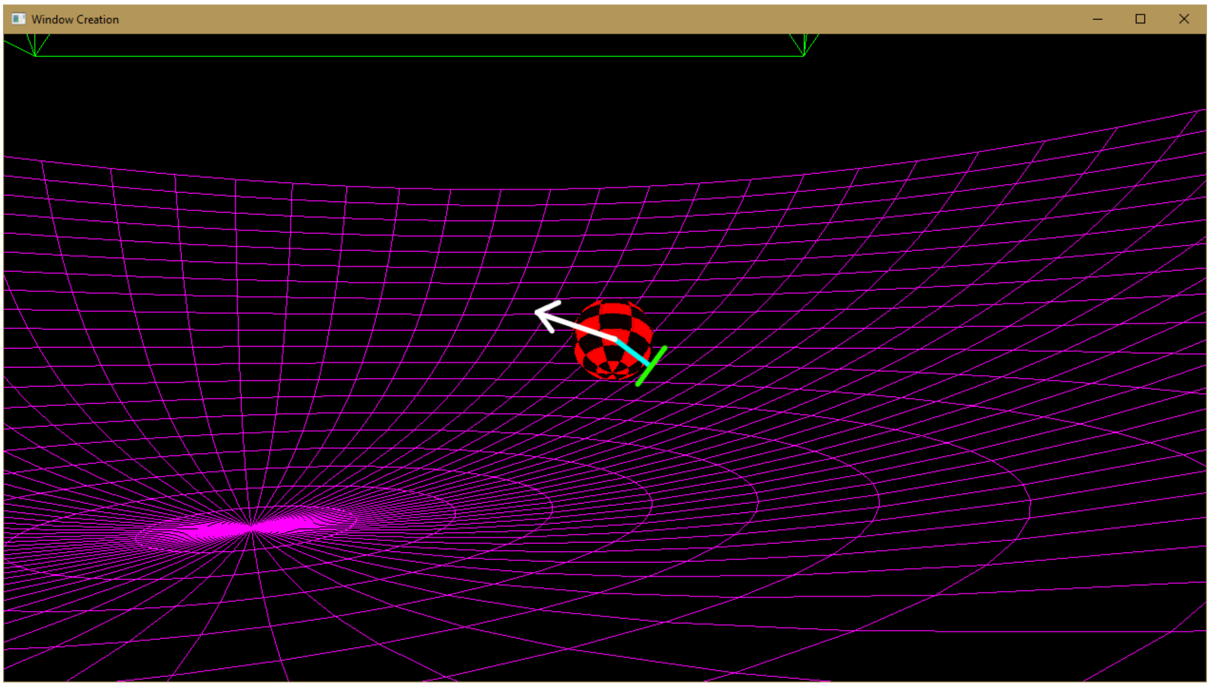

Figure 14: The above image shows the resultant velocity after a ball has collided with the bowl.

4.4 Ball to Net and Cube to Net

The collision response between an object and the net is calculated in much the same way as discussed above, obviously a resultant velocity is not applied to the net however as it is considered to be immovable.

4.5 Plane

When a plane collides with the box its position is set so that its first vertex is colliding with the plane and its velocity is set to zero.