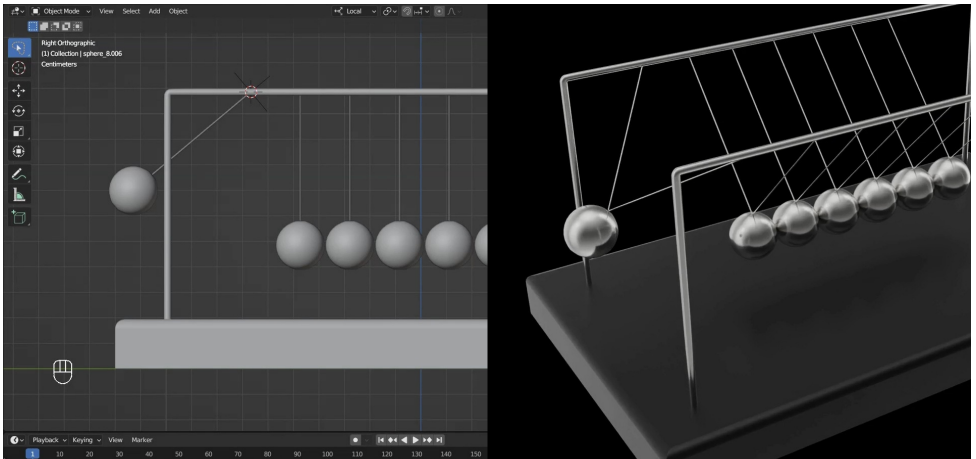


This PDF is available for free; feel free to share it!

[Subscribe to my YT channel](#) & turn on notifications to get updates on new videos!



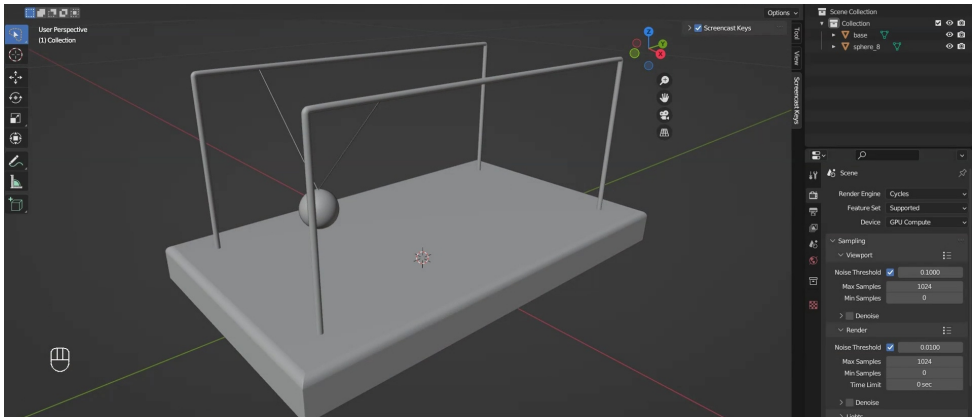
In this tutorial, we will see how to set up some **Rigid Bodies** and an **Hinge Constraint** to create the animation of Newton's pendulum. The tutorial was created using Blender version 3.6.

To begin, I am using a pre-made 3D model that features the basic structure and a considerable number of spheres with their respective supports.

In this 3D model, the spheres and supports have their origins correctly set along what will be the axis of rotation, but they are all **parented to the structure**.

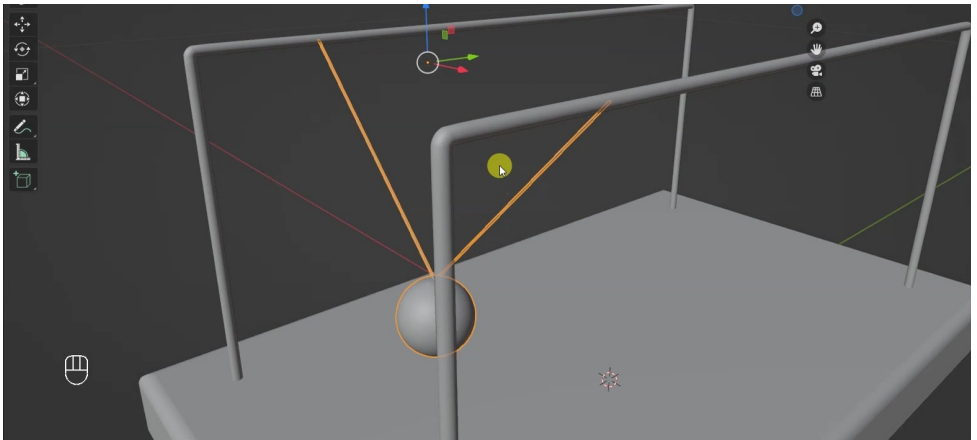
This is unnecessary and, in fact, could pose an issue for the physics simulation. Therefore, as a first step, I am selecting all of them and releasing their parenting using the shortcut ALT P. Next, I choose the option "**Clear Parent and Keep Transformation**" from the **Separate** menu that appears on-screen.

Actually, I need to perform operations that will be repeated for all spheres, so I can indeed **delete all of them except one**. This way, I can carry out the operations on the remaining sphere and obtain pre-configured copies.



We now have two objects: the base and the first sphere with its supports.
We can ignore the base, as it will not be part of the physics simulation.

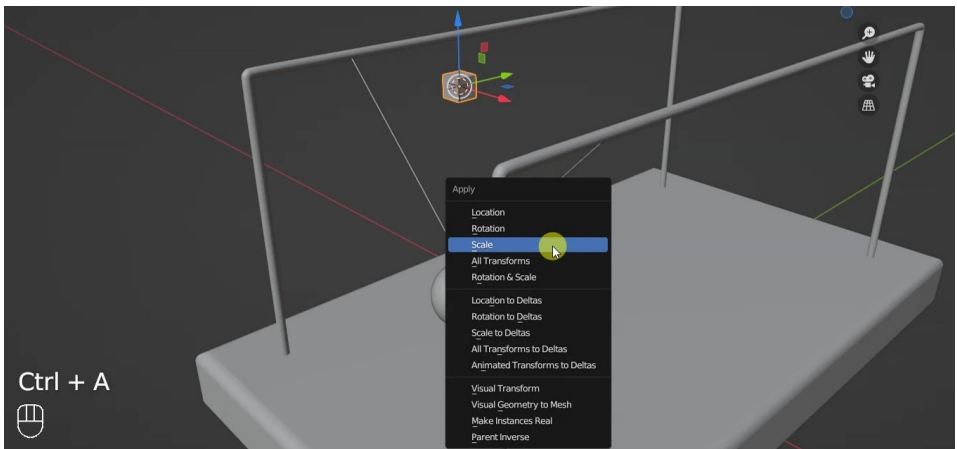
The **sphere** has its origin at the top, along the axis of rotation.
This is very useful because **right at that point**, on the vertical axis of the sphere,
we need to add an **accessory object** necessary to use the **Hinge Constraint**.



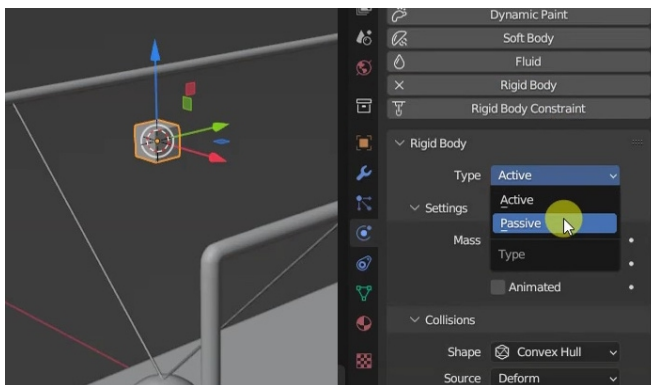
When we add a new object to the scene, it is positioned with its origin where the 3D Cursor is located. Therefore, I position the 3D Cursor at the sphere's origin using the shortcut SHIFT S, and then choose the "Cursor to Selected" option.

Now we need an **accessory** object to apply the Hinge Constraint. Any object will do, so I am adding a **simple cube** and resizing it for better visibility.

After resizing, I **apply the scale** transformations using the shortcut **CTRL A**. This operation might be unnecessary in this context, but it's a good habit to reinforce.

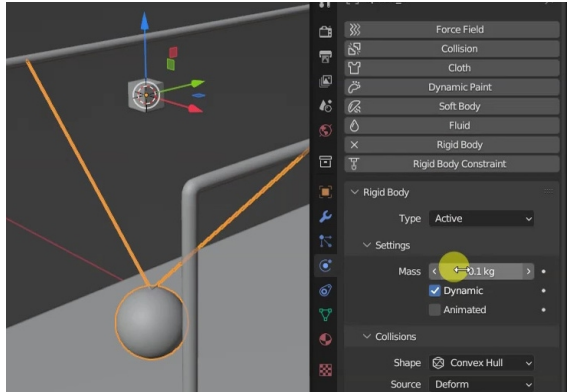


With the cube selected, open the **Physics** tab in the Properties Window and add a **Rigid Body** component, specifically setting it as **Passive**.



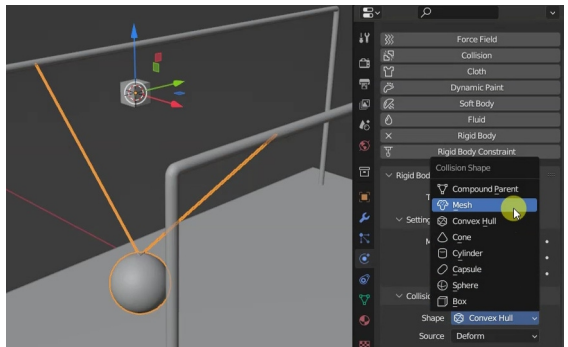
This object will then participate in the Rigid Body simulation but will not react to forces. As mentioned, it will serve as the constraint for the rotation of the sphere.

Next, let's add a **Rigid Body** to the sphere, but change its type to **Active**, as the sphere needs to move according to gravity and other forces. The default mass value for the object is too low, so I set it to 100 grams.

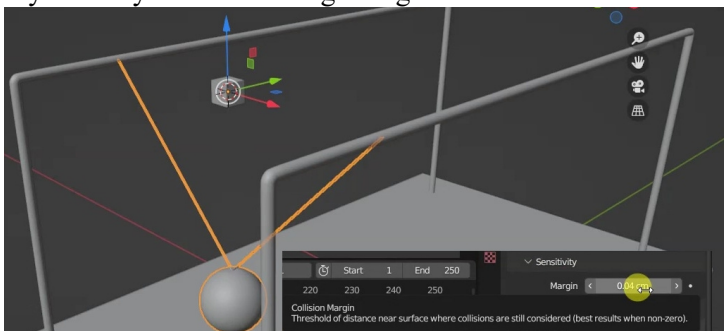


Another crucial modification to make concerns the **Collision Shape** of the sphere. This parameter is initially set to Convex Hull by default, but the hull created might cause some issues during interaction with others.

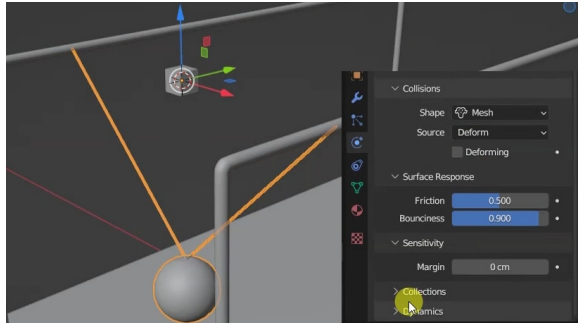
Therefore, I change the mode to **Mesh**. This option is more computationally intensive but will provide a more accurate result.



The tooltip for the **Margin** parameter suggests leaving a value greater than 0. However, through experimentation, I've found that in this context, it's better to set it to 0 to prevent the physics simulation from detecting collisions between objects even when they are stationary but very close at the beginning of the simulation.



Additionally, we need to set a very high value, such as 0.9, for the **Bounciness** parameter. This ensures that **the spheres bounce a lot**, almost perfectly elastic collisions.



In the **Dynamics** section, you'll find other parameters related to **Damping**, which manages the loss of velocity or angular momentum over time. We'll revisit these parameters later when we conduct tests with all the spheres.

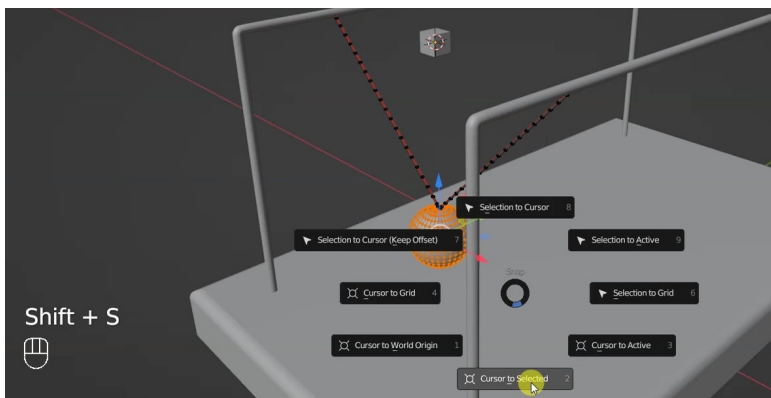
Before proceeding, **let's save the scene to a new file.**

Throughout the tutorial, I'll advise you to save the project before pressing Play or initiating a Bake, as Blender has crashed several times with this particular simulation when I pressed Play without saving changes first.

Starting the animation at this point, **the sphere will fall into the void.**

This is, in fact, the correct behavior because we haven't created an Hinge Constraint yet to constrain the sphere to the cube we added earlier.

Before that, however, we need to make another modification: we have to **move the center of mass of the sphere right inside its geometry**. This can be done by placing the 3D Cursor at the center of the sphere with SHIFT S in Edit Mode, then choosing "Origin to 3D Cursor" with the right mouse button in Object Mode.



Now the sphere and the cube are perfectly aligned vertically.

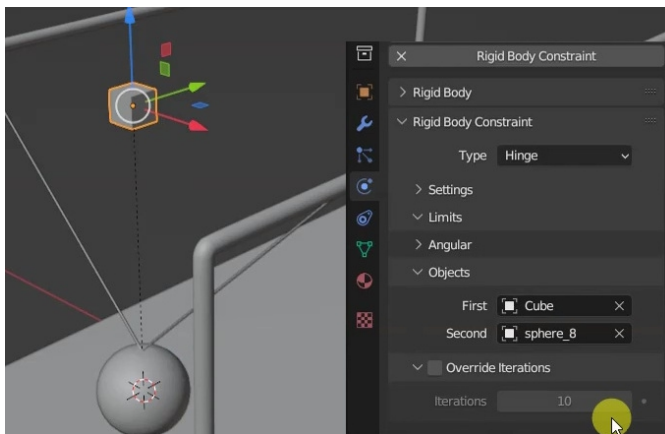
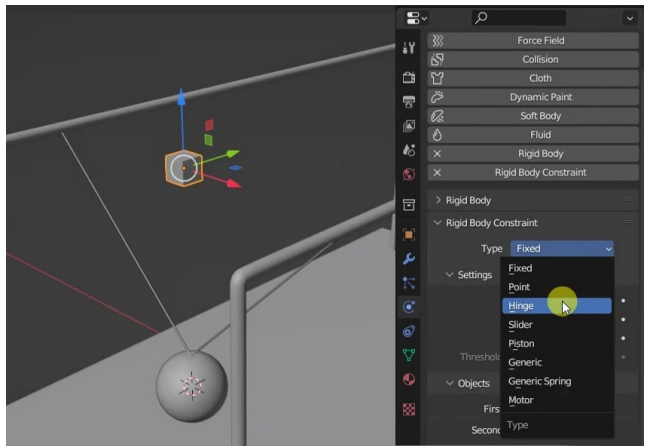
Before rendering, we can hide **the cube**, but for now, it's useful to keep it visible. Select it, and let's proceed to **create the Hinge Constraint**.

The Rigid Body Constraint needs to be added, like the Rigid Body component, in the **Physics** tab of the Properties editor.

The first thing to do is to **change the type** of Constraint from Fixed to **Hinge**.

In the **Objects** section, you'll find **two fields** where you specify two objects in the scene. The Constraint works by connecting the object named **Second**, which **will move**, to the object named **First**, which will remain **stationary**.

In our case, we can set the Cube as First and the sphere as Second.



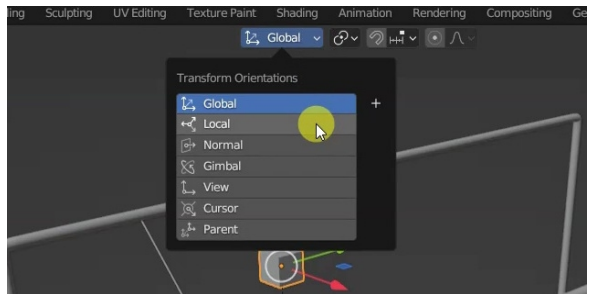
We're not done yet because, with the **Hinge Constraint**, rotation occurs around **the local Z-axis** of the passive Rigid Body object.

In essence, Hinge is similar to the Point Constraint but constrains the swing to this axis. Since, in this project, the rotation needs to occur around the global X-axis of the scene, which currently corresponds to the local X-axis of the cube, **we need to rotate the cube by 90 degrees** around the Y-axis. To display the local axes, I changed the **Transform Orientation** of the 3D view to **Local**.

Let's perform this rotation by pressing **R Y 90**.

Saving the project and starting the simulation now...
nothing will happen!

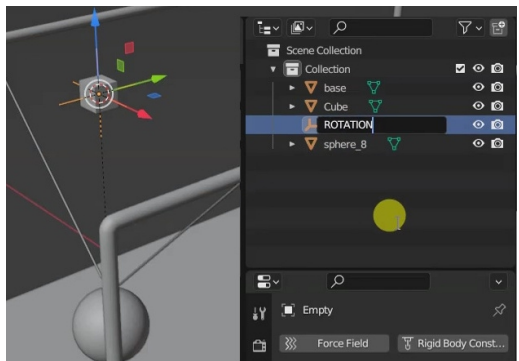
The center of mass of the sphere is correctly set inside, but the sphere and the constraint are **perfectly aligned vertically, in balance!**



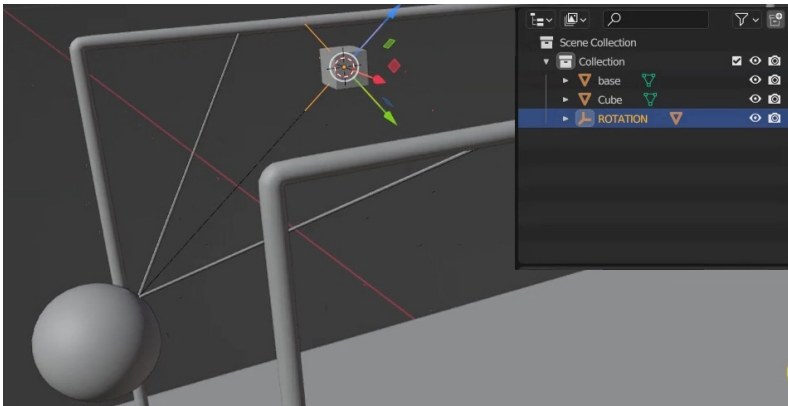
Therefore, we need to provide **an initial rotation** to the sphere, and then physics will take care of the rest.

Now that the Origin of the sphere is right inside, it would be helpful if we could rotate the sphere and its supports around the point where the cube is located.

For this reason, I am **adding an Empty object** to the scene, immediately renaming it to **ROTATION**. I then select the sphere and also select the ROTATION Empty, with a multi-selection in this order. Finally, I press CTRL P to create the parent-child relationship, with the sphere becoming a child of the ROTATION Empty.

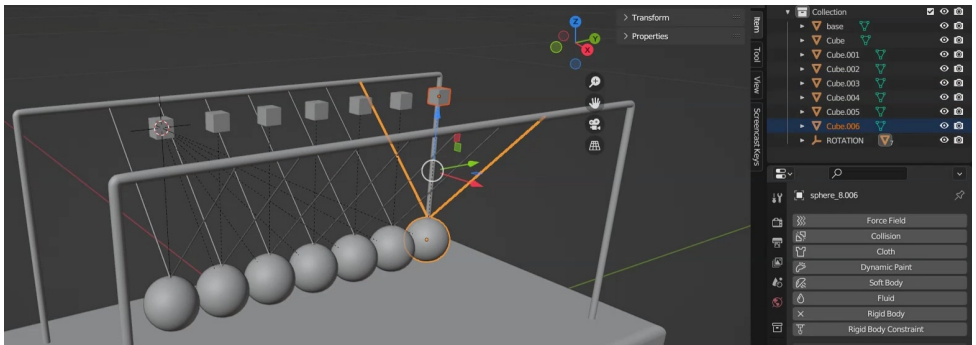


This way, we can provide **the initial rotation** to the first sphere by rotating the controlling **Empty**. Save the project now and start the simulation. This time, the sphere will move just as desired!



Creating the other spheres is actually straightforward because we can proceed with **copies** of the first sphere and the cube that has the Rigid Body Constraint.

Let's bring the **ROTATION Empty back to the vertical position**, select both the cube and the sphere with a multi-selection, and make an initial duplicate with displacement using **SHIFT D Y** and mouse movement, placing the second sphere right next to the first one.



I specified **SHIFT D Y**, with the constraint of moving the duplicate along the Y-axis, to perform the displacement of the copy along with the duplication. In other words, Blender considers these two operations as a single operation.

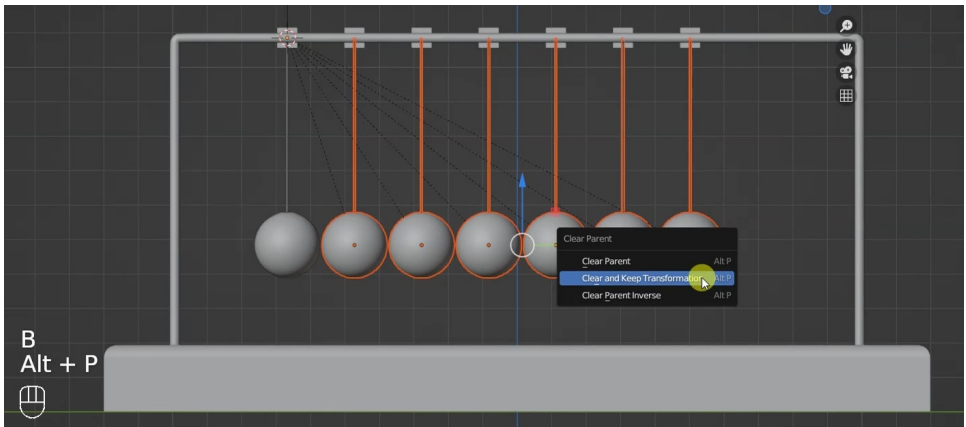
This allows us to create the other copies, already perfectly positioned, by pressing **SHIFT R**, which is the **shortcut to repeat the last operation**.

If we have positioned the copies correctly, saving the project now and starting the simulation, the spheres will remain perfectly still in their places.

This indicates that the Rigid Body simulator is **not encountering false collisions** between objects.

The only drawback is that all the copies will be **parented to the ROTATION Empty**, which provides the initial orientation for the first sphere.

However, we can quickly solve this issue by selecting all the copies, detaching them from the Empty with the shortcut **ALT P**, and then choosing **Clear Parent and Keep Transformation**.



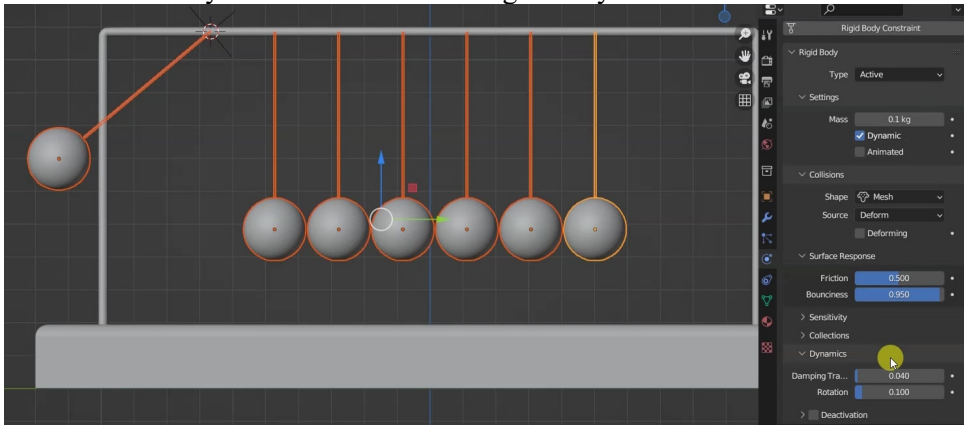
Rotate the Empty that controls **the first sphere**, **save** the project, and **start** the simulation to examine the achieved result.

If the result is satisfactory, the next step is to **hide the cubes** with Rigid Body Constraints from both the scene and the final rendering to proceed to rendering.

In case the central spheres move too much, **try repositioning them** so that they are closer to each other in the initial configuration. Remember to move the spheres **along with the cubes that have constraints**; otherwise, the spheres will start with their own initial rotation!

To **speed up the simulation** or make the spheres swing more, we can adjust a couple of parameters. To modify a parameter for all selected spheres in one go, press and **hold the ALT key while clicking** with the mouse on the parameter you want to change.

Regarding the **spheres**, I suggest **increasing the Bounciness** value and significantly **reducing the two Damping** values for both linear movements and rotations in the Dynamics section of the Rigid Body tab.



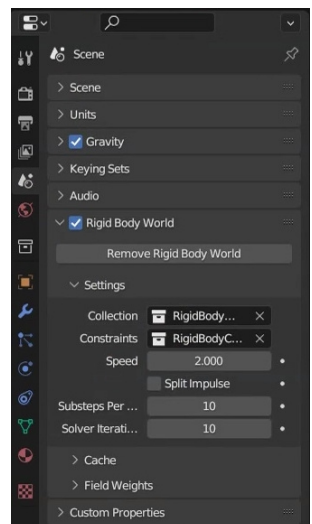
However, I **advise against lowering the Friction** value, as this could generate undesired movements in spheres that are meant to remain still.

In the **Scene tab of the Properties** editor, you will find the **Speed** parameter, which allows you to accelerate the entire physics simulation.

In the same tab, you'll also find the **Substeps** and **Solver Iterations** parameters.

I recommend increasing them only if you notice errors in the physics simulation.

These parameters instruct Blender to perform more calculations, resulting in a more accurate simulation, albeit at the expense of an increased simulation time.



Well, that's all! I hope you liked this tutorial!
See you soon!