

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!



Hello everyone!

In this short tutorial, we will see how to provide **a random value for the intensity of a light source** in Blender. Specifically, we will discuss using the **Noise modifier** on animation curves.

This modifier can be used with all **animation curves** present in Blender, so its use is not limited to light sources, but it can involve colors, movements, rotations, and generally anything that can be animated in Blender.

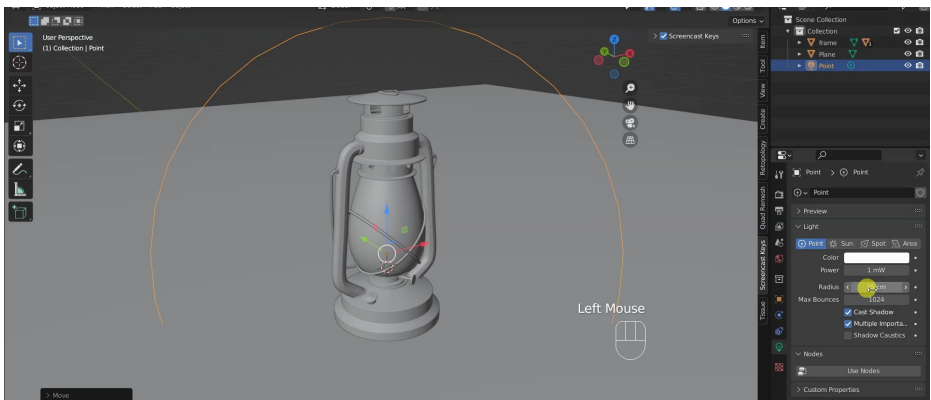
This tutorial was created using **version 3.3 of Blender**, but the concepts and tools discussed here are valid from many previous versions of the software.

I **imported the model** into an empty Blender scene, where I then inserted a **Plane** on which to place the lamp. This will make the rendering more interesting, as we will see the **light projected onto the surface** of the plane.

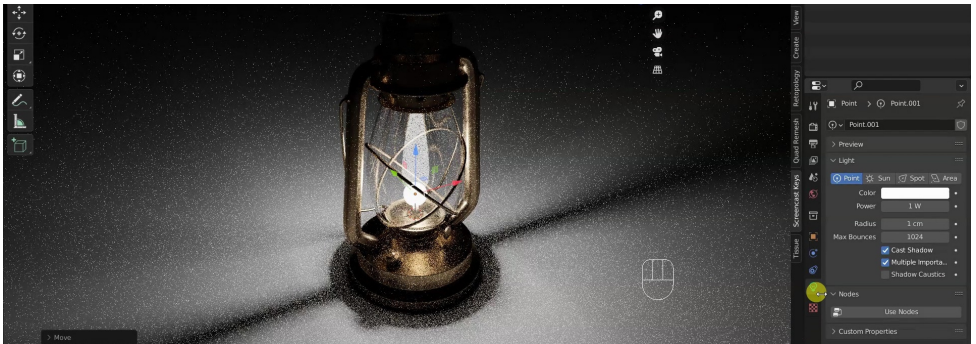


I insert a **Point Light** light source inside the glass container, setting the intensity of the light source to 1 Watt, to do a test.

As seen in the video tutorial about the Cycles modifier for animations, I need to **check the Radius value of the newly created Point Light** light source, which is actually too large for the Kerosene Lamp object, so the resulting lighting is not correct.

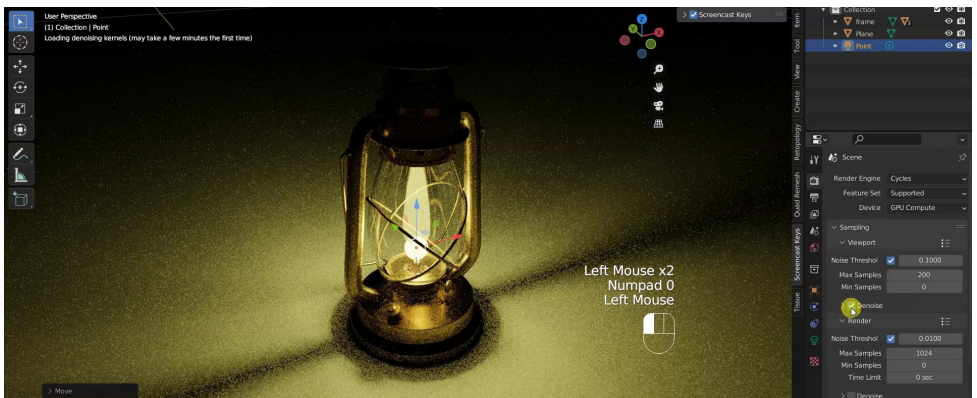


I, therefore, **reduce the value of Radius** until the circle representing the physical size of the light source fits inside the glass container.

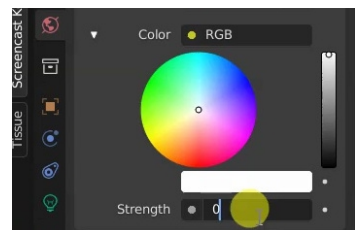


To analyze the effect achieved up to this point, I switch to **Rendered display mode** in the 3D Viewport.

I am setting **200 samples** and turning on **Denoise** for the render preview, to be able to evaluate the effect more quickly.



To make the effect more noticeable, I set the **intensity of the World Background lighting** to **0**.



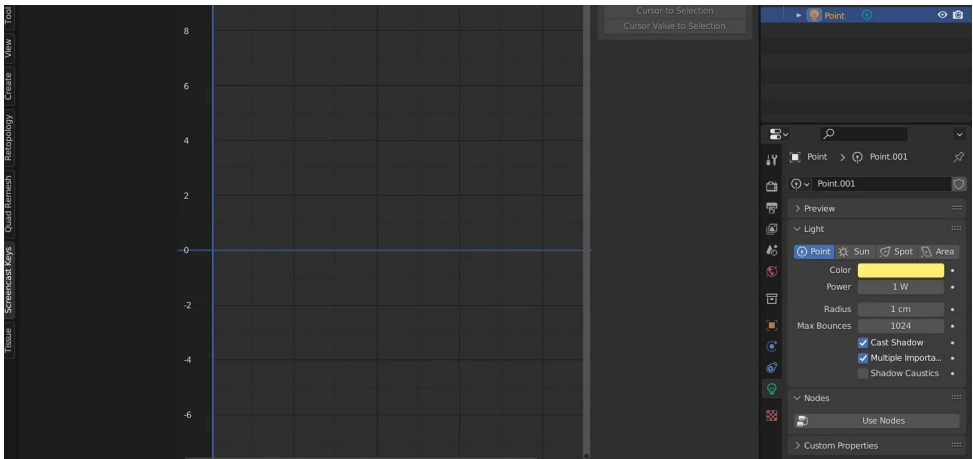
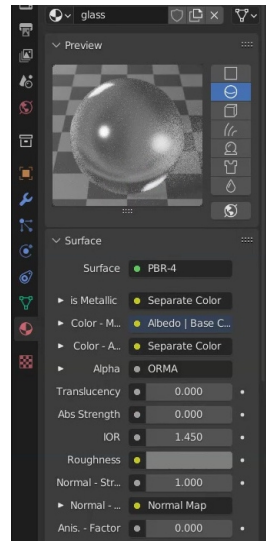
In my 3D model, **the glass is completely transparent**, as if it were brand new.

To make it more **frosted**, I disconnect the texture associated with the **Roughness** channel of the Material and manually set a light gray for this channel.

I analyze the result in real-time in the 3D Viewport in Rendered mode, until I reach an acceptable result.

Ok, the initial setup of the sample scene is complete!
Let's move on to the **flickering effect**!

I open a **Graph Editor** window, with the Point Light light source selected.

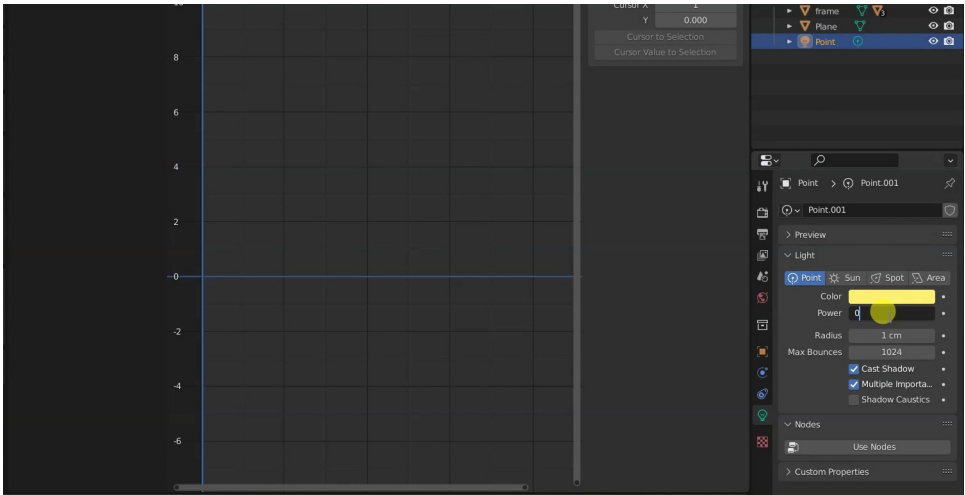


To be able to provide a **modifier to the curve of the Power parameter** of the Point Light object, I must first create **an animation curve** for this parameter.

This can be done simply by **inserting an animation keyframe** for this parameter, for example at the **first frame** of the Timeline.

Before inserting the keyframe, I set the **value of the Power field to 0**.

This will make the rendering completely black, but trust me: the reason will become very clear as soon as we add the Noise modifier to this parameter.

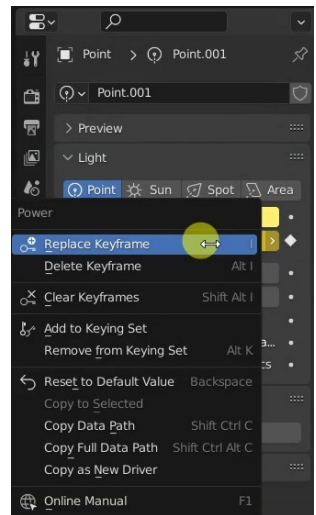


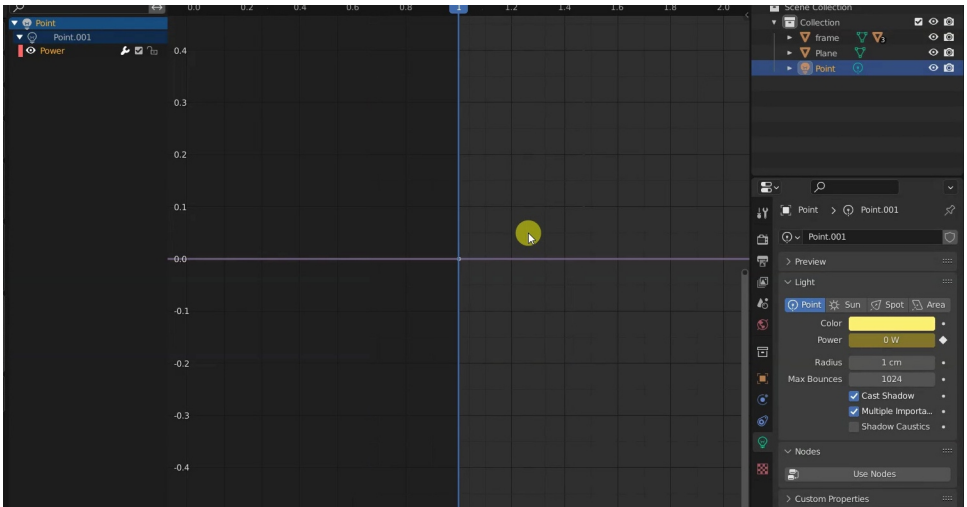
To insert the keyframe, I place the **mouse cursor over the Power parameter** and press the **I** key. Alternatively, I can **right-click** on the **Power** value and select **Insert Keyframe** from the menu that will appear on the screen.

Now I have the **Power curve** available in the **Graph Editor window**.

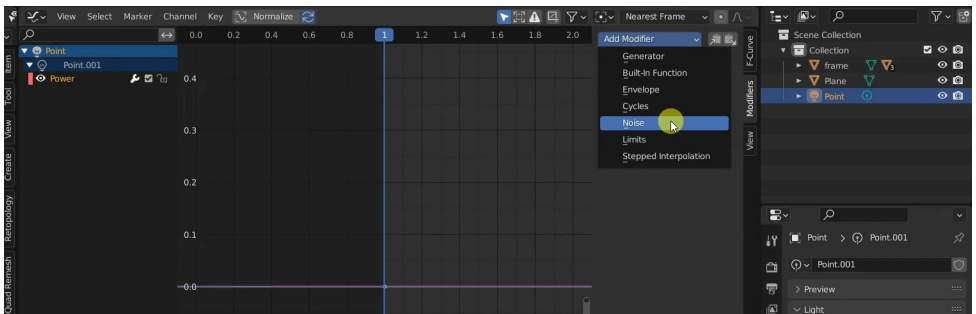
In case the keyframe is not easily identifiable, place the mouse cursor in the window, press the **A** key and then the **period key** on the numeric keypad.

This will have the effect of **framing the keyframe**, selected with A as it is the only element present in the window.

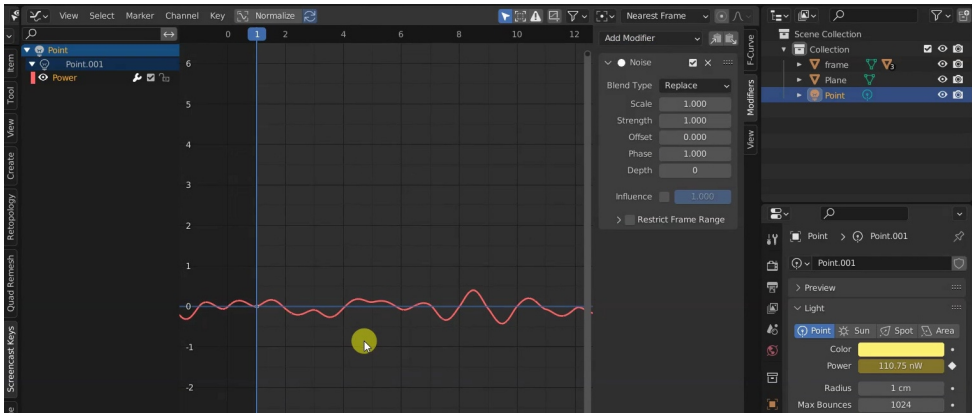




I insert a **Noise modifier** from the side panel of the **Graph Editor** window. This panel can be opened or closed with the **N** key.



The first **parameter** of the newly created **Noise** modifier is **Blend Type**, and it is very important to know the options it provides, because it will be possible to modify the value of the parameter in a very different way depending on the mode selected.



The second fundamental parameter of this modifier is **Strength**, which allows defining to what extent to modify the starting value. **The value of Strength can also be negative.**

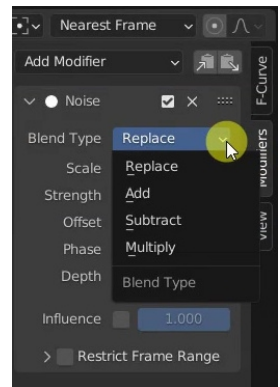
The names of the modes easily suggest how the value of the parameter to be modified will be altered.

Add will add a random value **in the range between 0 and Strength**, so we will be sure that the final value will never fall below the value we set in the first keyframe, unless we set a negative value for Strength.

Subtract will, of course, perform a **subtraction between the values**. In our case, it would be useless, because the initial value is 0, so any value resulting from the subtraction would be **negative**, and the Point Light object would not emit light.

Multiply will multiply the original value by a value between 0 and Strength. This could lead to much more extreme variations than those given by the Add mode, **if the original value of the parameter to be modified is different from 0**. In our case, **Power** has an initial value of **0**, so Multiply will have no effect, because any value **will be multiplied by 0**.

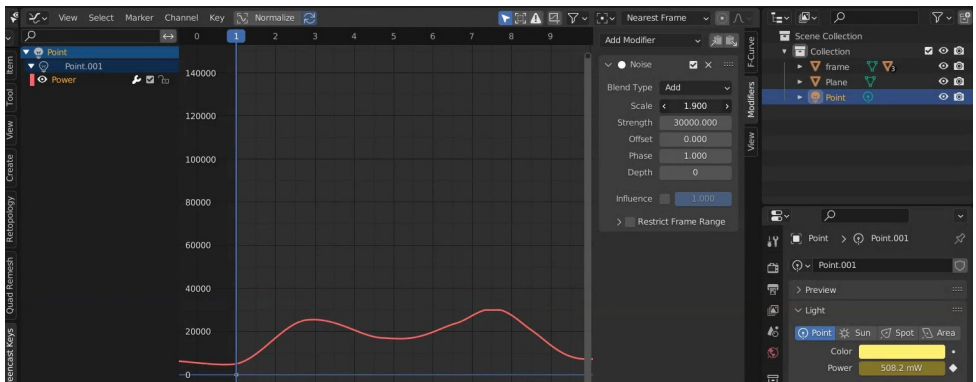
Replace will completely replace the starting value with the one given by Strength, but be careful when using this parameter, because **Strength represents the**



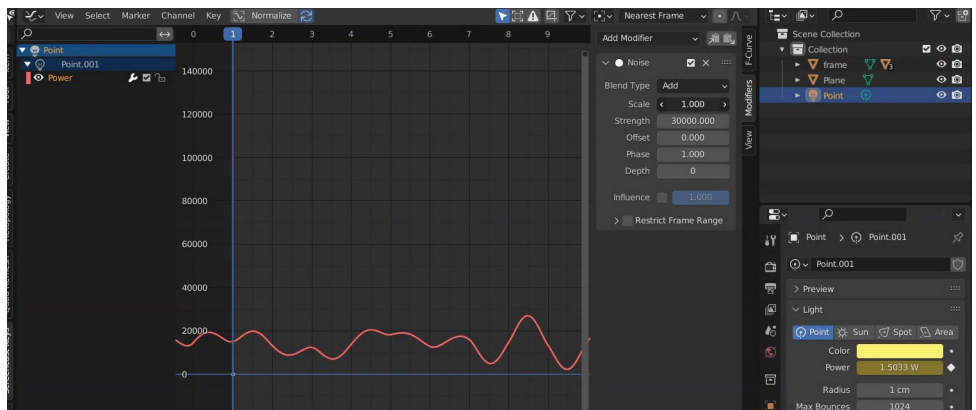
amplitude of the disturbance, so it might also make the parameter to be modified take on **negative values**.

For this case, we will use the **Add** mode.

In particular, to create the animation you are seeing on the screen right now, I set **30000** as the value for the **Strength** parameter, so the luminous power of the light source actually varies **between 0 and 3 Watts**.



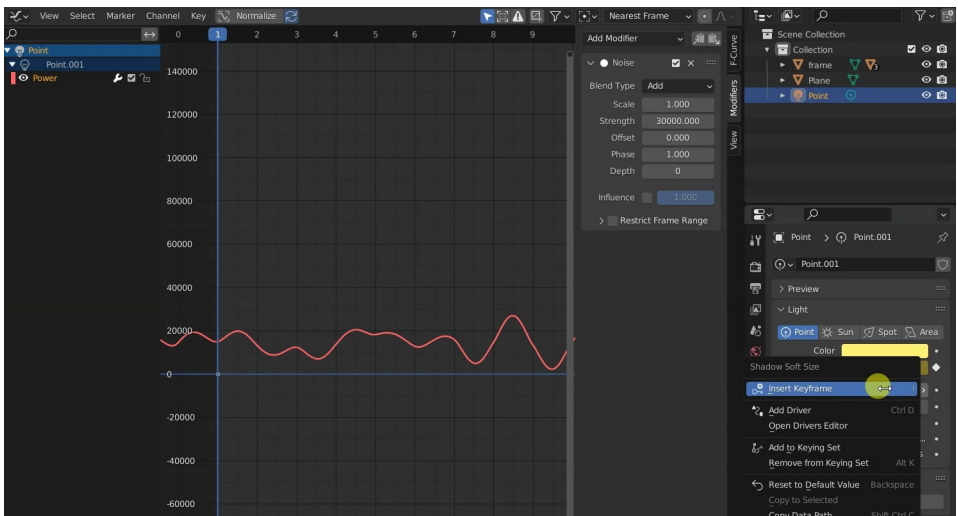
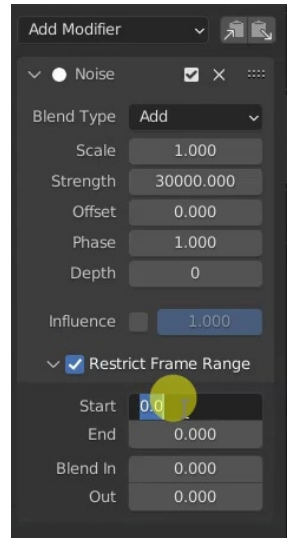
Another very interesting parameter of the **Noise** modifier is **Scale**, which allows you to **make the flicker more or less frequent**: with a value between 0 and 1, we will have sudden changes between values, while **for values much greater than 1**, we will have very **smooth interpolations extended over time**. For the rendering I showed you earlier, I used the value 1 for this parameter.



The **Restrict Frame Range** section is also very interesting, allowing you to specify the range of frames in which to apply the effect. This can be useful for **inserting a flickering effect at certain moments** during an animation, for example for a light source that suggests that the lighting might fail at any moment, in a scene full of suspense.

As I told you at the beginning of the video tutorial, the Noise modifier can be applied **to any parameter that can be animated in Blender**. So, I'm doing a test with the **Radius** field of the **Point Light** light source, but this time with a different mode for the Noise modifier.

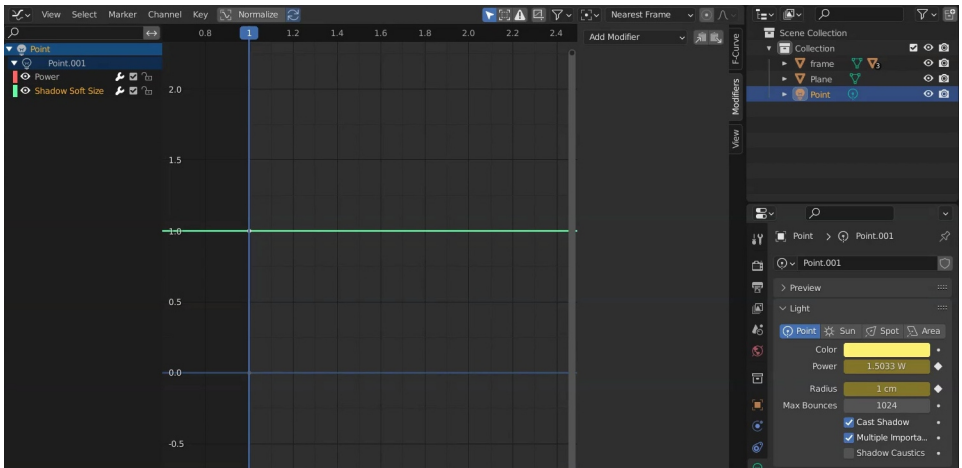
First, I go back to the first keyframe of the Timeline and **insert a keyframe** for the **Radius** field of the Point Light light source, with an initial value of 1 cm.



In the Graph Editor window, I select the **new animation curve**, called **Shadow Soft Size**.

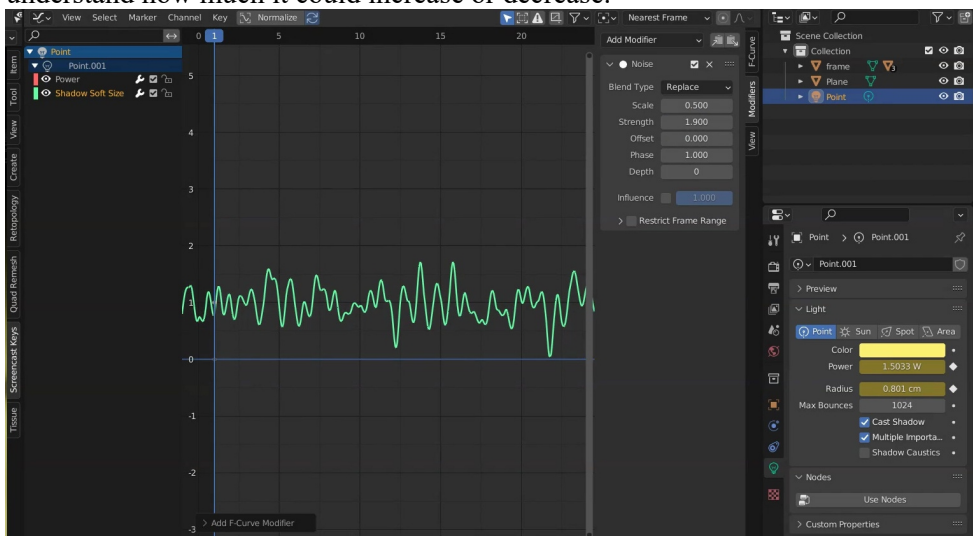
To easily locate the **Shadow Soft Size keyframe**, I can disable Power, select all with A, and press the **period key** on the numeric keypad. As you can see, this

time **the value on the vertical axis** looks more like the one in the Radius field, unlike what we saw with Power.



I then insert a **Noise modifier** for this curve, this time changing the mode to **Replace**. I now set **Scale** to **0.5**, to have more frequent and sudden variations.

Finally, I set **Strength** to **1.9**. This is, in a sense, the maximum value for this particular animation, because the starting value of Radius was 1, and Strength represents the maximum amplitude of the noise, which needs to be divided by 2 to understand how much it could increase or decrease.



Since Radius has a value of 1, if I set the value of 2 for Strength, I could get a final value between 2 and 0, which we want to avoid.

By setting 1.9, I will get very low random values for this parameter, but without the risk of reaching 0 or even negative values.

Before starting the new rendering, **I also set the value of the Scale parameter of Power to 0.5**, so as to have more sudden variations for the intensity of the light source, as well as for its **Radius**.



Well, that's it for this tutorial! I hope you found it helpful! See you soon!