

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 brief tutorial, we will examine a simple Blender scene to use for **creating a turntable animation** of any 3D model.

These animations are required by some stores where you can publish your 3D models: they consist of a series of images in which **the 3D model to be proposed rotates 360 degrees**, typically around its vertical axis.

The tutorial was created with **version 3.3 of Blender**, but the elements we will discuss are available from several previous versions and, presumably, will still be in the future.

It is a tutorial aimed at beginners, in which we will examine some basic tools such as the **Constraint Track To** for the camera and the lighting of the scene.

In this tutorial, I will use one of my 3D models: **Tableware Set 1**.

This is a model containing multiple objects, which will allow us to see how to handle models of this type to create turntables; however, it is not necessary to have this 3D model to learn from this tutorial: you can also watch the tutorial without downloading this particular 3D model!



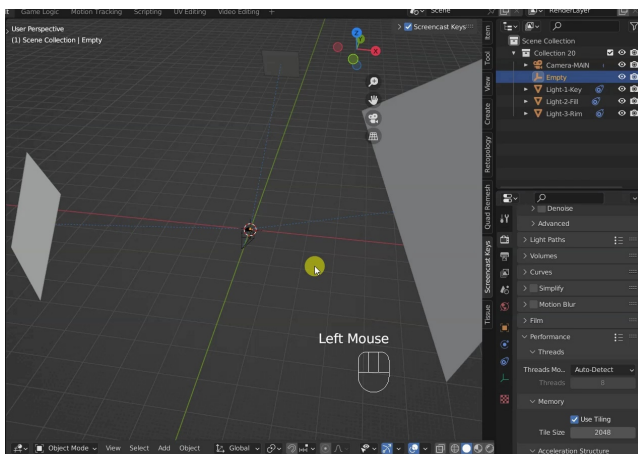
In the first part of this tutorial, I will show you **the basic scene, without a 3D model**, to list the various settings of the **template file**, to be used as a starting point for all the models to be framed and rendered.

In the second part, we will see **a practical example** with the 3D model I mentioned earlier, so as to see how to create the rotation animation.

So let's start with the first part of the tutorial!

The virtual scene contains **only five objects**:

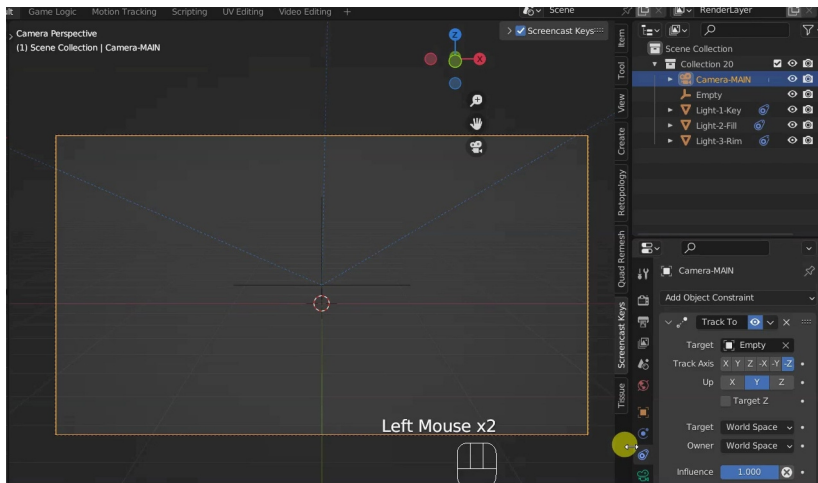
- **three light-emitting Planes**,
- a virtual **Camera**,
- and an **Empty object**, positioned at the **center** of the virtual universe.



The **Empty** object serves as a **reference point** so that the virtual camera and the three light-emitting Planes correctly point towards it.

This is important because we may need to adjust the framing based on the size of the object to be framed: by using the Empty, we won't have to operate individually on the camera and the Planes, but it will be enough **to move the Empty for the other objects to automatically follow it**.

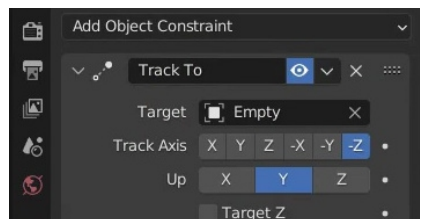
This **Constraint** is achieved through the use of the **Track To Constraint**, which, as the name suggests, makes the object that owns it automatically **follow the target object**. To correctly set the **Constraint Track To**, we need to specify two directions: the **front vector** and the **upper** one.



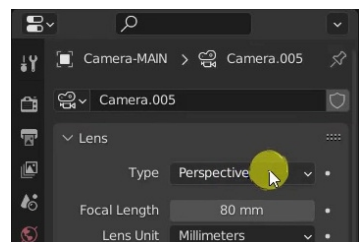
The front vector, called **Track Axis**, indicates in which direction the object will point; in the case of the **Main Camera**, it is the **local negative Z axis**, as can be seen by selecting the **Main Camera** and setting, in a 3D view, the Viewport Gizmo Move and the **Local Transform Orientation**: the positive local Z axis points towards the back of the frame, so we need to set **minus Z** in the **Track Axis** field of this object.

This display mode also allows us to easily identify **the upper vector of the Main Camera**: it is positive Y, so in the **Up** field of the **Track To Constraint** we will select **Y**.

All that remains is to specify the **Empty object as the Target** and the camera will always point towards the Empty, even when we move the Main Camera within the scene.

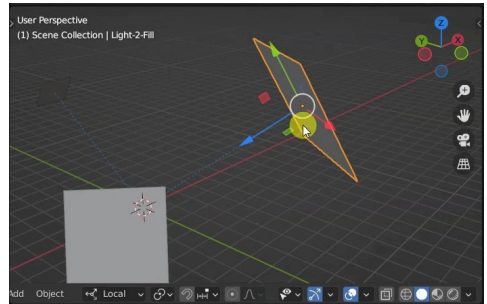


Another setting concerning the **Main Camera** is its **Focal Length**, which is found in the **Object Data Properties** tab of the object. In this case, it is best to set a **high value, for example 80 or 110 mm**, to reduce the distortion effects due to perspective; however, we will see a practical example shortly.



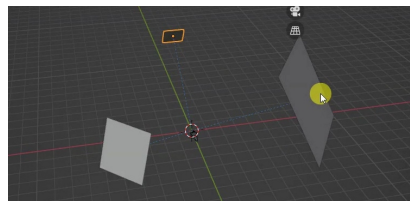
Now that we know that the camera's framing axis is local Z, we can also **move the camera away** from or closer to the Empty by moving along that axis, simply by using **G Z Z**.

The **Constraint Track To** should also be set for the **three light-emitting Planes**; here, however, we notice that the positive axis perpendicular to the Planes, which coincides with their normals, is **positive Z**, so in their Constraint Track To I will specify **Z**.

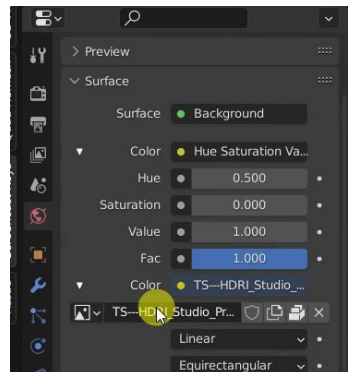


Regarding these three light-emitting Planes, there isn't much to say: they are, indeed, **three Planes provided with an Emitter material**, which provide default white lighting with equal intensity (but the three Planes are of different sizes and this affects light intensity and shading in general).

The three Planes are arranged to achieve a **standard three-point lighting**, but the settings listed so far are just the starting points: nothing prevents you from changing the intensity, color or arrangement of these three light backgrounds in the scene, depending on your needs!



The three light-emitting Planes are not, however, the only elements related to lighting: in the **World Properties** tab I have set an **EXR** type image, with **Linear Color Space**, to illuminate the object.



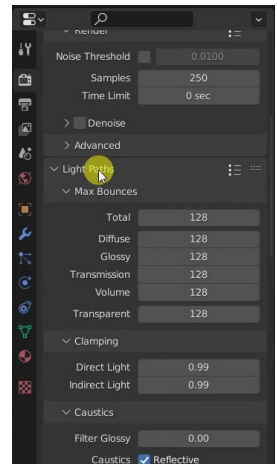
The choice of the image depends on various factors, but in general, I use **colored images**, rather than images that reproduce studio lighting, when framing objects with many metallic components, in order to achieve interesting and non-monotonous shading effects on those very reflective surfaces.

As previously stated, these are just the initial settings to have different lighting options, but of course, **you can also deactivate the three light-emitting Planes** or reduce the intensity of the World Background lighting, depending on the needs of the moment.

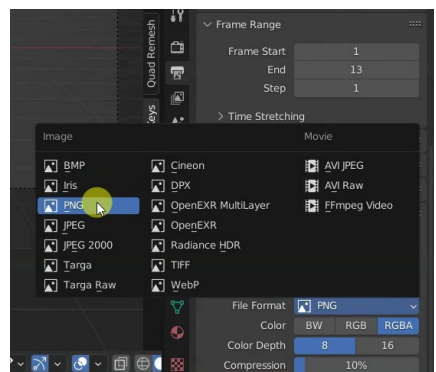
The rendering preview can be evaluated **in real-time** by switching to the **Rendered display mode**, as we will see later with a practical example.

Regarding the rendering of the scene, most of the time the stores require **sequences of images**. The specifications vary for each store and the parameters to be set in the **Render Properties** and **Output Properties** tabs must be set accordingly.

In my case, in **Render Properties** I have set just **250 samples** for rendering and most of the time it is not even necessary to activate the **Denoise**, because the noise, in a scene like this, is minimal and the fireflies can be reduced by setting the value **0.99** in the **Clamping fields**; moreover, **Denoise** could have **effects on the details** of the surfaces of the objects, so I prefer not to use it in this case.



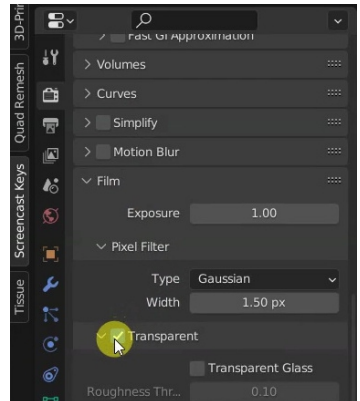
As for the **Output** specifications, in my case, I set the **resolution** to 1920 by 1080, because for many stores it's sufficient; also, I chose the **PNG RGBA format**, which is a **sequence of single images with a transparent background**, because various stores have different background requests, so here I prefer output with transparency.



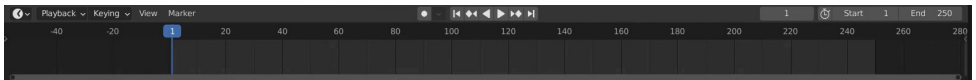
Speaking of the transparent background: **to not display the background** of the virtual universe in the renderings, you need to open the **Film** tab in the **Render Properties** window and select the **Transparent** option.

The last setting for this template file concerns the **Timeline** and, in general, the **duration of the animation** to be made: these parameters depend on the requirements of the various stores, so here I will show a couple of examples.

If the store requires a real **movie**, or if you intend to make a movie to publish on Youtube, on social media or on your website, then you need to set a **number of frames** equal to the duration of the animation in **seconds multiplied by the desired number of frames per second**.



For example, to make a 10-second movie to play at 25 frames per second, you will need to set **250 in the End field of the Timeline**.



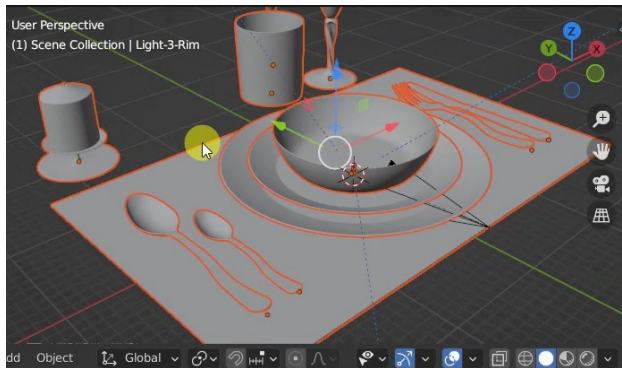
In this case, moreover, you can opt for the direct realization of a **movie file**, rather than a sequence of images, in the **Output Properties** tab.

If instead you need to create a low number of single images, for example 13, then it will be sufficient to set the exact number of images to be created in the **End field** of the **Timeline**: Blender will create an image for each frame of the animation.

These are the general settings of the file, valid for all 3D models to be rendered.

This file can then be saved as a general template in a separate folder, so as to make copies to modify for specific projects.

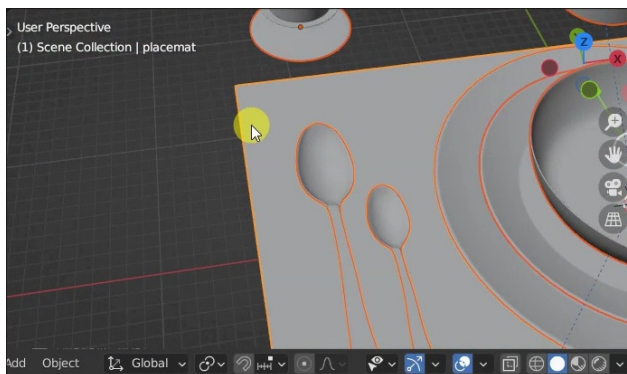
So let's see a **practical example with a 3D model**, which can be imported into the scene using **Append** or **Asset Browser**.



I chose the **Tableware Set 1** model because it consists of **many objects**, so it will be necessary to group them or merge them into a single object, in order to **set its Origin** and easily perform some transformations.

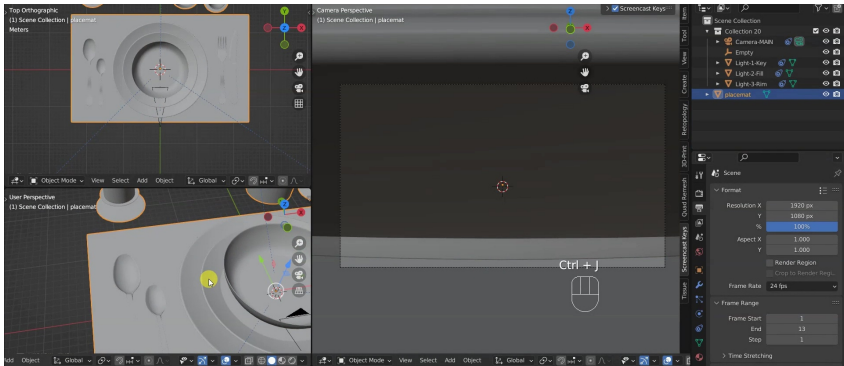
As soon as it's imported, the model is huge and doesn't fit within the frame; before resizing it, however, **I'll merge it into a single object with Join**. At this moment, though, all the individual objects that make up the set are selected, but **there isn't an active object**, so Join cannot be done correctly.

To solve this problem, I press **SHIFT** and the left mouse button on an object, such as the **Placemat**, whose outline will change color to indicate that it is now the active object.

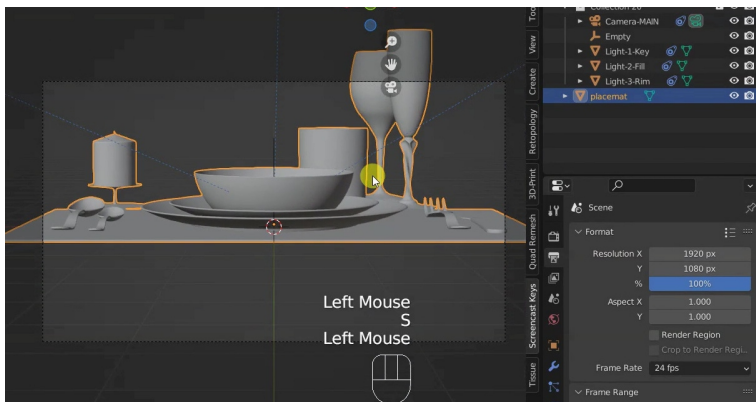


I then press **CTRL J** to merge all selected elements with the Placemat object.

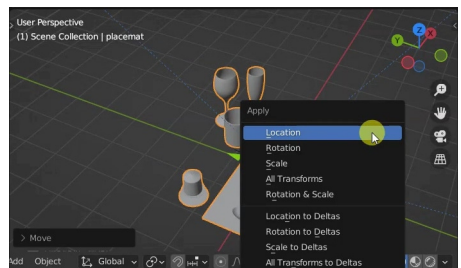
BLENDER (basics) - How to make a simple scene for a turntable animation, tutorial
www.francescomilanese.com --- 2024 --- video version: <https://youtu.be/hG1Xf3xEISw>



The Placemat has its **origin** coinciding with the center of the scene, so I can **resize** it by simply pressing the **S** key and moving the mouse, then pressing Enter when the object fits into the frame.



In case the imported object does not have its **Origin** in the center of the scene, you can manually position it as you wish, then press **CTRL A** and choose **Location**: this operation will have the effect of resetting the scene's Location coordinates; the result of this operation is that the **object's Origin** will coincide with the center of the virtual universe, as desired.

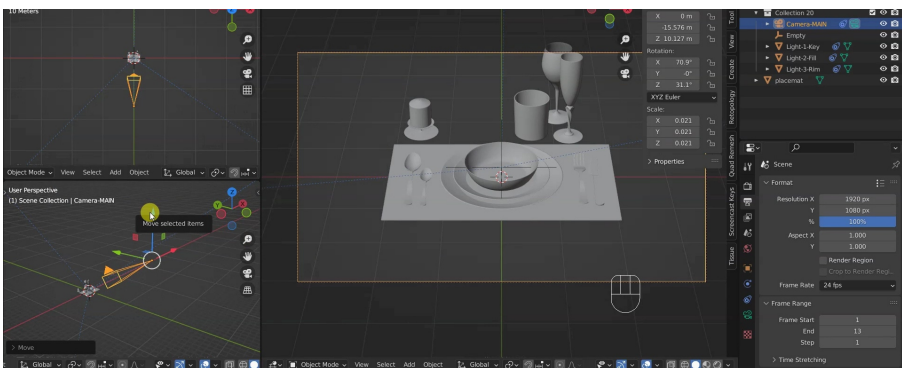


The set of objects now fits within the frame, but there's a problem: this frame is not ideal for displaying this particular model.

Here we have two possible solutions: moving the virtual camera up or rotating the object around its X axis. The choice depends on the model in question: if the object also has details on the bottom, then it's probably better to rotate it around the X axis, tilting it, before performing the animation around the vertical axis.

If you decide to follow this method, remember to **apply the rotation transformation with CTRL A Rotate** and, if necessary, move the object and apply the **Location** transformation again, in order to correctly set the **Origin** of the rotated object in the center of the virtual scene.

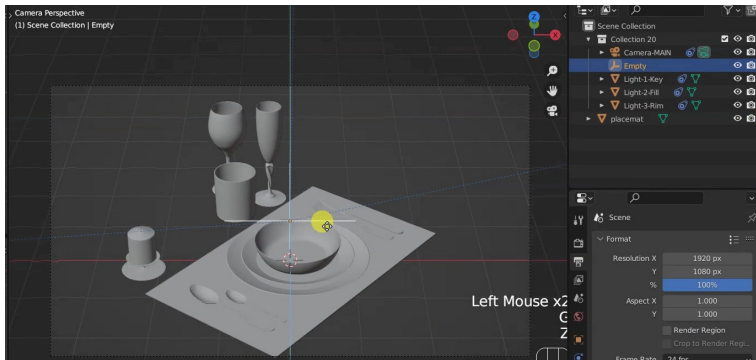
In the case of **Tableware Set 1**, however, there are no details on the lower part of the model, so we can proceed with the second option: **moving up the virtual camera**. This operation can be easily performed by selecting the virtual camera and moving it with a transformation gizmo in the 3D view or with the key sequence **G Z**.



Nothing prevents you from starting with a different framing, moving the virtual camera to other points in the scene.

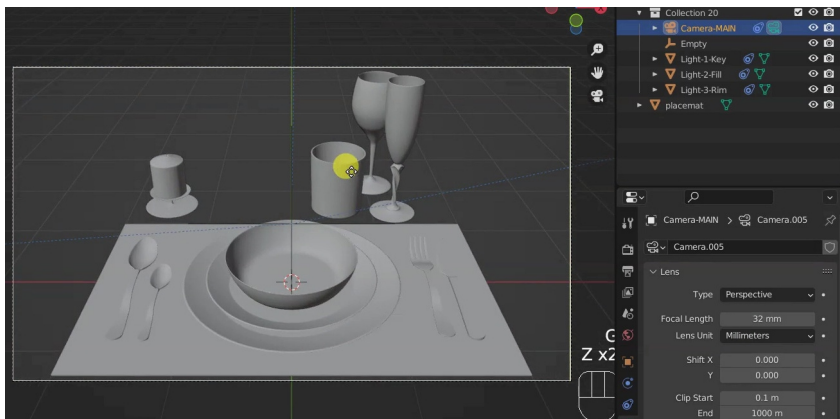
Note that, since **the virtual camera always points towards the Empty** located at the center of the scene, there will be no need to modify the rotation of the camera.

You can also **move the Empty**, to fix the framing; this is particularly useful with the model I'm using in this example, because in fact the center of its bounding box does not coincide with the center of the virtual universe, so it is necessary to **move the Empty upwards** and, consequently, the framing, to observe it better.



Also in this case, the use of the **Constraint Track To** for the camera shows its advantages. In the first part of the video I told you that it is advisable to use a high value for the **Focal Length** of the virtual camera to limit distortions due to perspective.

You can get an idea of these **distortions** by now changing the value of **Focal Length** of the Main Camera: besides the fact that you will need to move the Main Camera forward or backward to frame the object well, you will immediately notice the **distortions** with low Focal Length values, especially **at the edges of the frame**.

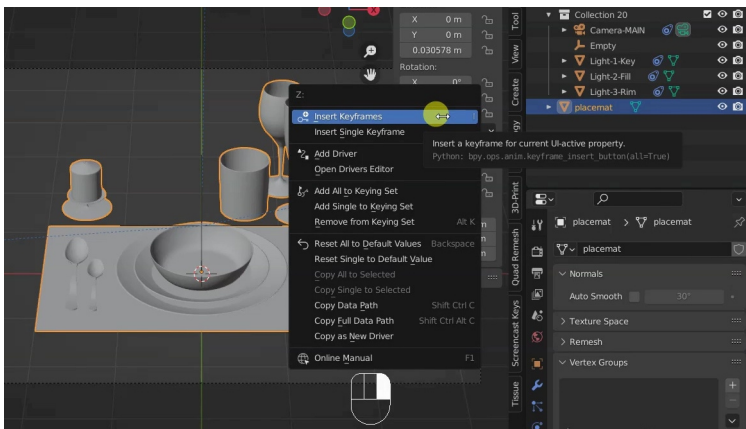


Therefore, I recommend a value between 80 and 110. I don't think there is a need for higher values but, as always, the final choice should be made by examining the specific model each time.

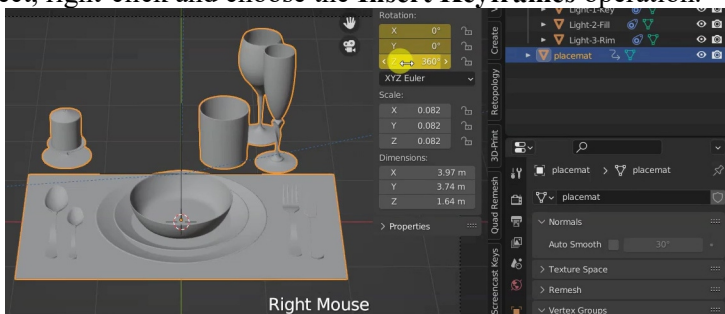
Ok, let's move on to **the animation of the model!**

Actually, the operation is very simple: since we want a **360-degree rotation around the vertical axis**, we only need to set **two animation keyframes**, at the first and last frame.

With the 3D model selected at frame 1 of the Timeline, make sure the Rotation values in the Transform tab are initially **reset to 0**, then **right-click on Rotation Z** and select the **Insert Keyframes** operation.



Then move to the **last frame** of the animation, enter **360** in the **Rotation Z** field of the object, right-click and choose the **Insert Keyframes** operation.

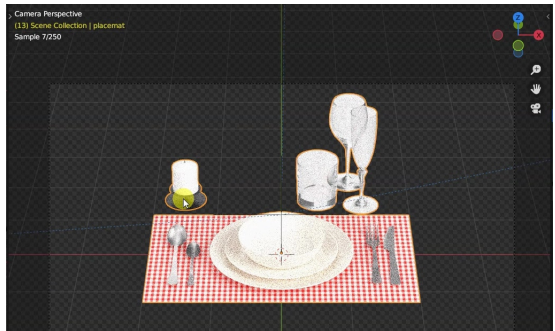


The animation of the rotation has been created, as we can see by clicking on the **Play button** of the **Timeline**; if the number of frames is very low, it might be preferable to move forward and backward between the various frames by pressing the arrow keys.

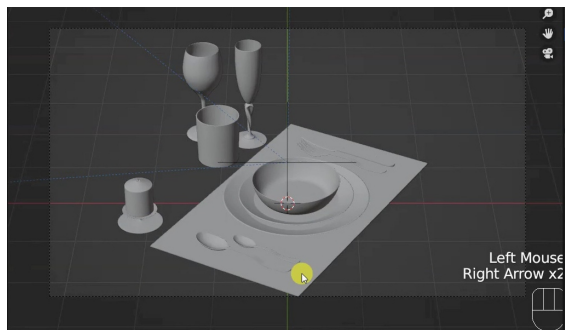
However, **before starting the rendering** of the sequence we still need to perform three operations:

1. **check the lighting** of the model;
2. **check the framing** throughout the animation;
3. modify, if necessary, **the interpolation** between the various keyframes, i.e., the way Blender will perform the rotation in the intermediate frames.

The lighting of the model can be checked by pressing the **Z** key and **switching to Rendered view mode**; in this way, by examining the model in various frames of the animation, we can decide whether to keep the default lighting scheme or whether to change the settings of the three emitting Planes or change the EXR or HDR image used in World Background.



The **framing check** is done simply by **examining the various frames** and concerns models whose bounding box is not cubic and is not centered in the scene, just like in our case: **Tableware Set 1** is, in fact, longer than it is wide, so in some frames it might go outside the frame. In this case, it will be necessary to **adjust the distance of the camera**



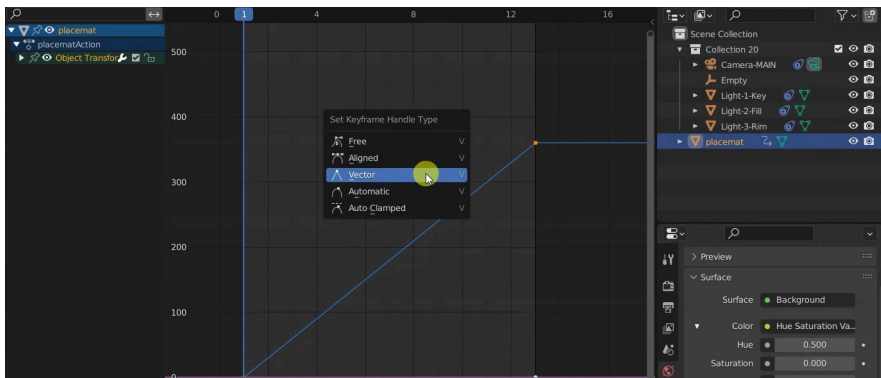
from the framed object or the height of the Empty, so that the object is within the frame in all frames.

This operation is also useful to prevent, on the contrary, the object from unnecessarily occupying too small a portion of the frame.

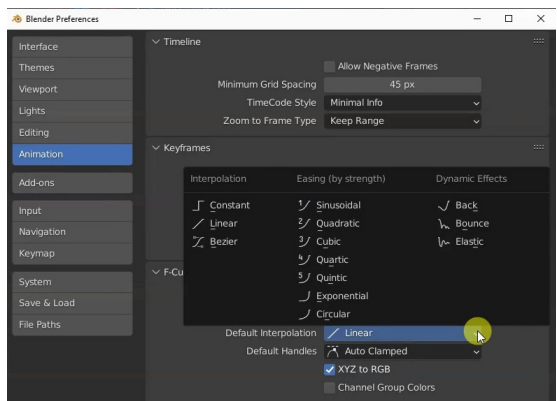
The last operation to be done concerns the analysis of the **interpolation between the various keyframes**.

Blender can indeed perform different types of transitions from one keyframe to another; for example, an interpolation with a **Bezier** curve will start and end the animation between two keyframes gradually rather than linearly.
In our case, however, a **linear animation might be preferable**.

The interpolation between the keyframes can be modified in the **Graph Editor**: with the 3D model selected, we open the Graph Editor, select the two **animation keyframes** available there, then press the **V key** and choose the type of **interpolation** we want to achieve for our animation.



In my case, the interpolation is already of the linear type, because I have set it as such by default for all my Blender projects in the Animation tab of the Preferences window, under Default Interpolation, but I wanted to show you how to modify this aspect anyway.



RECAP

To sum up: in this tutorial, we have seen how to set up a simple scene to obtain a turntable sequence of images of a 3D model in Blender, choosing an appropriate **Focal Length for the virtual camera** and setting up good **lighting** for the object to be framed.

We have seen how to set the **Track To Constraint** to point the light sources and the camera towards the center of the virtual scene.

We have also seen how to properly **set the Origin and the rotation** of the objects to be framed and how to create **animations** in a certain number of **frames**.

Finally, we have also seen how to **modify the interpolation of animation curves** to obtain linear rotations of the 3D model.

I hope this simple tutorial has been useful to you! See you soon!
