

Homework 3: Blender

By: Junwen Yu and Yuqi Meng Instructor: Prof. Jeong Joon Park
EECS 498-014 Graphics and Generative Models

2024 Fall

v1.0.1

Contents

1 Objectives	2
2 Setup and Basics	3
2.1 Installing Blender	3
2.2 Basic Manipulation	3
2.2.1 Rotation and Moving Around	3
2.2.2 Editor and Workspace	4
2.2.3 Controlling Objects	4
3 Tasks	6
3.1 Scene Setup	6
3.2 Texturing	8
3.3 Point lights	9
3.4 Skybox	11
3.5 Camera animation	13
4 (Optional) Exploration Task	17
5 Submission and Grading	17

5.1	Rendering the Animation for Submission	17
5.2	Packing The Blender Project	19
5.3	Exploration Task Submission	19
5.4	Grading	20

1 Objectives

This homework is a 10-day project on building and rendering your own scene in Blender. You are not required to build a very complex scene - one that meets the requirements described in the following sections would be sufficient. See [the grading section](#) for specific requirements.

Feel free to complete this homework on your own if you are confident with using Blender. The spec only serves as a guide.

The learning objectives of this assignment are to:

- Gain familiarity with industrial graphics techniques, like texture mapping and image-based rendering.
- Gain hands-on experience with applying ray tracing to render a scene, and identify the advantages of ray tracing, like better soft shadows and global illumination effects.



2 Setup and Basics

2.1 Installing Blender

You can download the latest version of blender at [the official website](#). This homework specification is based on version 4.2, but as we are not using many modern functionalities, an older version may be sufficient.

After installing, you can run Blender and create a default scene.



2.2 Basic Manipulation

We recommend [this tutorial video](#) for details. A short summary is provided below.

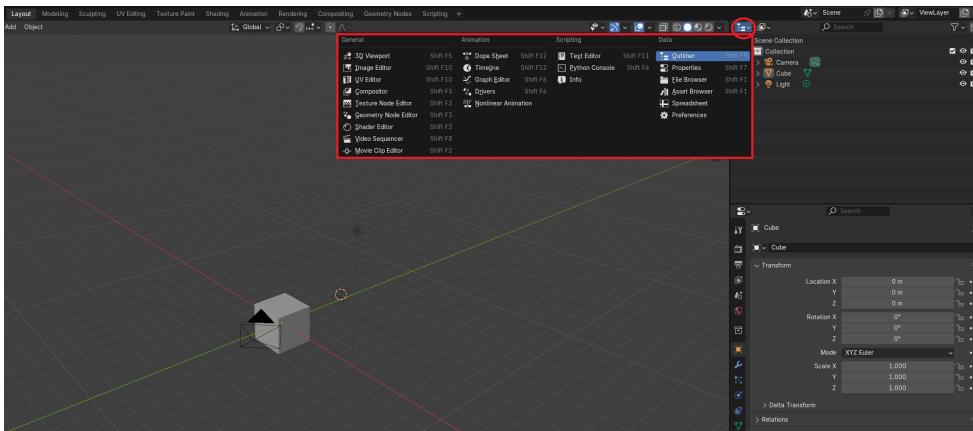
2.2.1 Rotation and Moving Around

At the bottom left, you can find what you can currently do with your cursor. For example, in normal mode, you can rotate your perspective by holding and dragging the middle mouse button (left), while when holding Shift, this operation moves your view camera around (right).

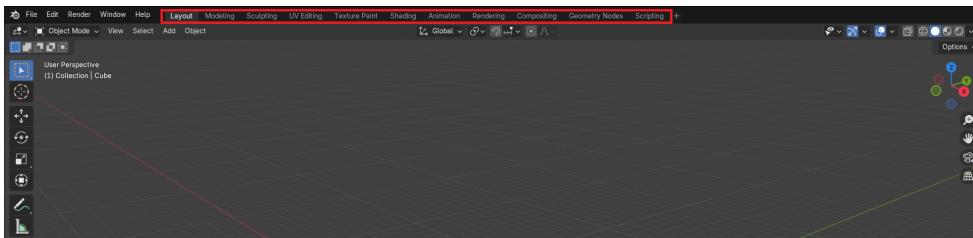


2.2.2 Editor and Workspace

Blender provides a number of editors for different purposes. For example, in the default workspace, we have a 3D viewport showing the current scene, a timeline at the bottom for animation, an outliner at the top right that helps to organize the scene data, and a properties editor at the bottom right. You can switch each area to another editor by simply changing the editor type at the top left corner of each area.



A workspace is essentially a predefined set of editors. You can switch between different workspaces by clicking the tabs at the bottom.

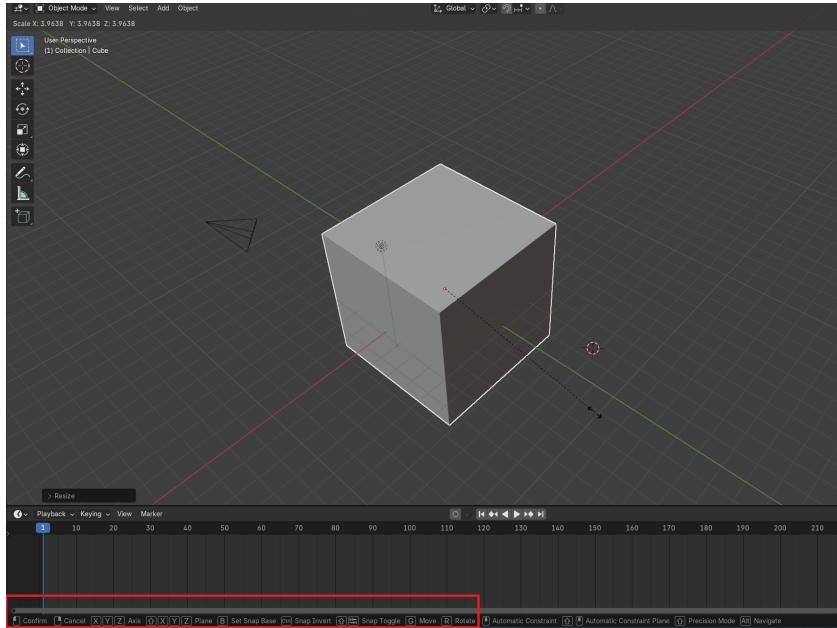


We will mainly use "layout" and "shading" in this homework.

2.2.3 Controlling Objects

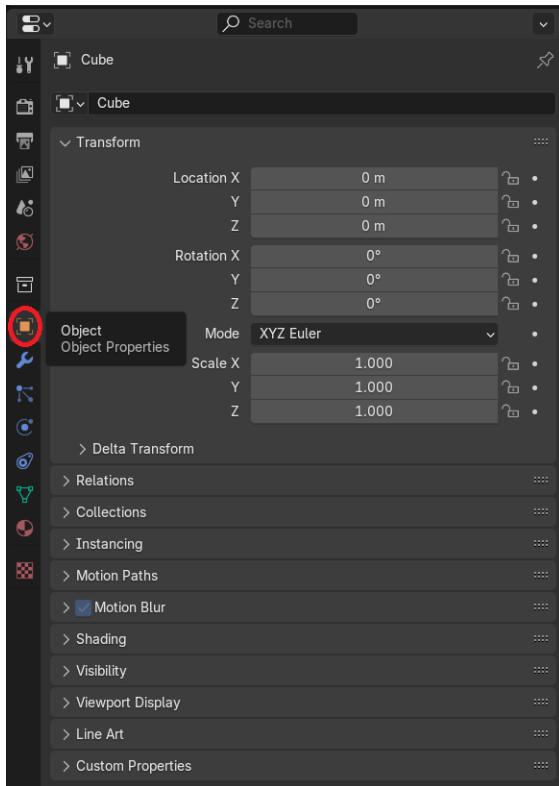
To select an object, simply click on it in the viewport. Alternatively, you can select it from the Scene Collection editor at the top right. After that, you can adjust its properties.

To scale an object, first select it, then press "S" on your keyboard (and note the information displayed at the bottom right), and move the cursor to dynamically adjust its size. Finally, left click to set the size. You can press "Ctrl-Z"("Command-Z" for Mac) to undo.



Likewise, press "G" to move the objects around, and "R" to rotate.

Alternatively, you can directly adjust the location, rotation, and scale parameters of the object to achieve these. You can access them in the properties editor, which is by default at the bottom right in the "layout" workspace.

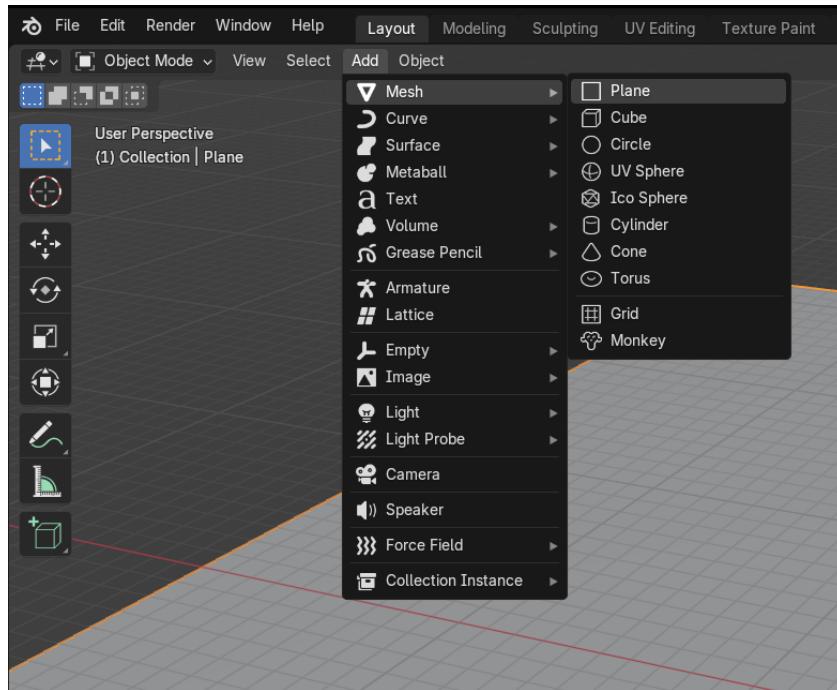


3 Tasks

3.1 Scene Setup

In this step, we will build up a simple scene.

First, you may delete the cube by pressing "x" on the keyboard, if you don't want it to appear in your final scene. Then, add a plane to the scene.

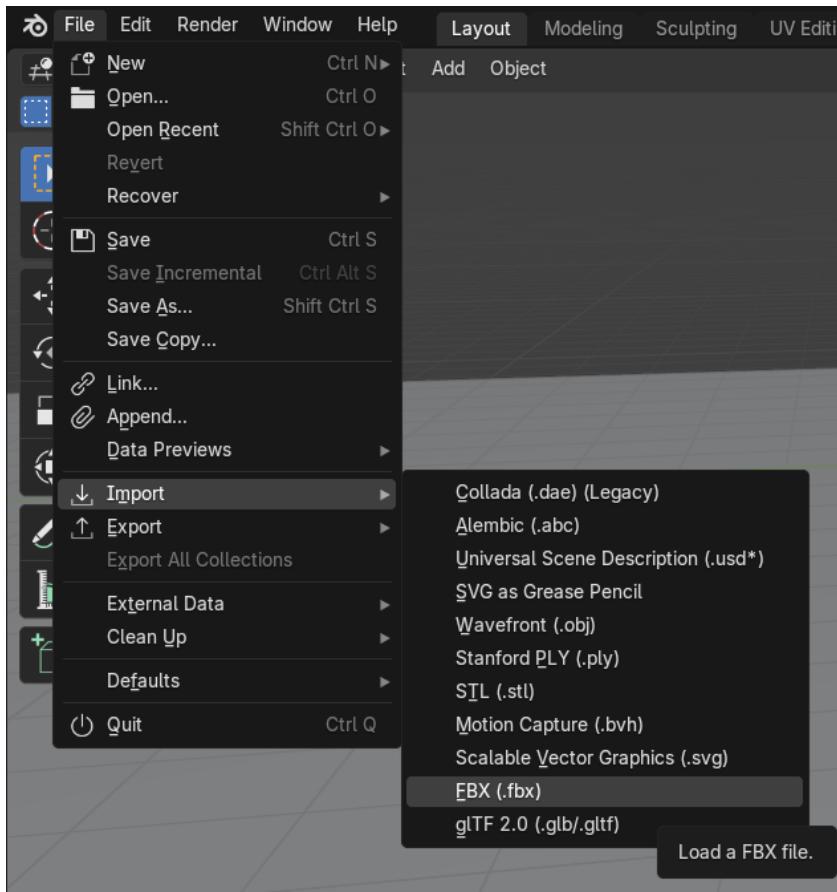


Next, we need to download some models and place them on the plane. You can find free 3D models on websites like [Turbosquid](#). We suggest you to download the .fbx files because they usually require less setup. At least one of your models should have corresponding textures, which most of the relatively complex models do.

Here is a concrete example: For [a chair model with texture](#), Click download (you may need to register first) and find the fbx model file.

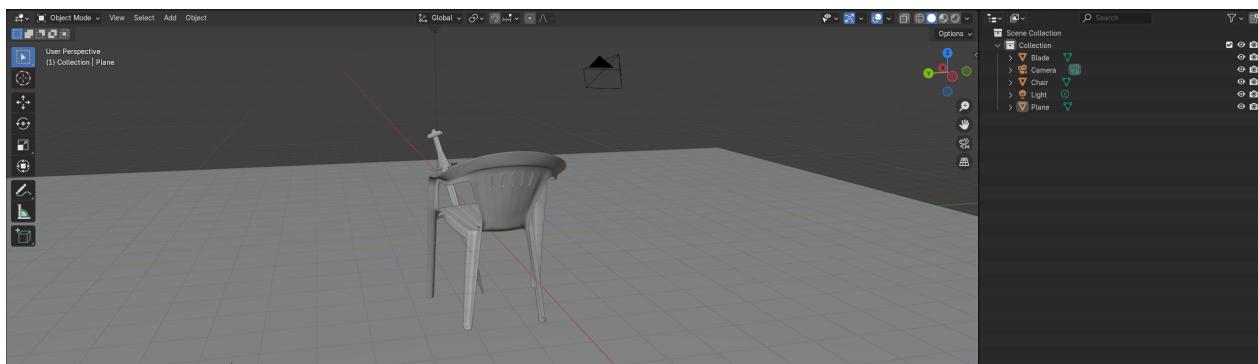


Unzip it to find a .fbx file and a folder containing the textures. Then, import the .fbx.



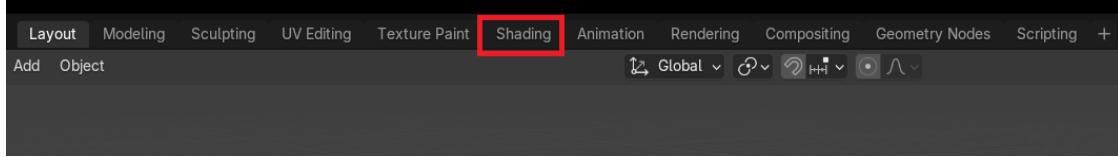
As long as the texture folder are placed together with the .fbx model, Blender should be able to automatically find the textures as well. Models that you import may look tiny in the viewport (you may not even be able to see them). In this case, you can select the model through the outliner editor, and press "S" to scale it. In addition, the model may contain extra light sources, or even cameras. You may delete them through the outliner as well.

The following is how the scene should look like after this step.

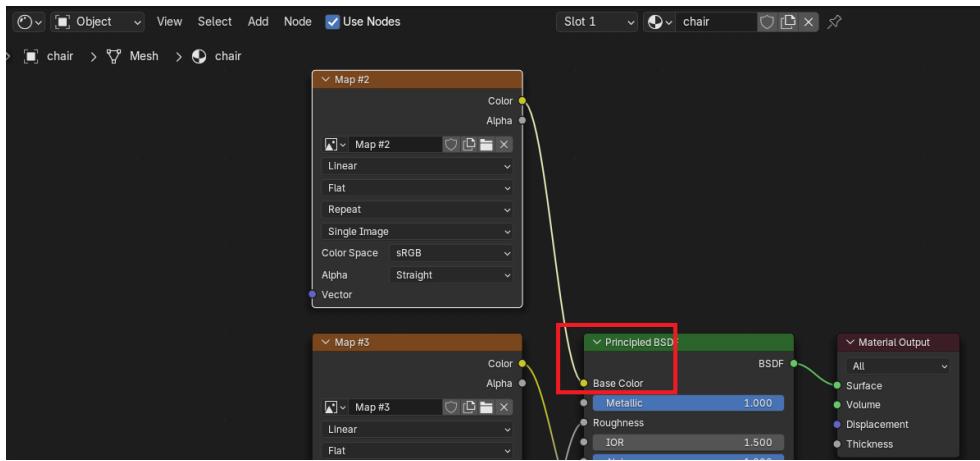


3.2 Texturing

In the previous step we imported a model with textures. To view the textures and materials, go to the "Shading" workspace.

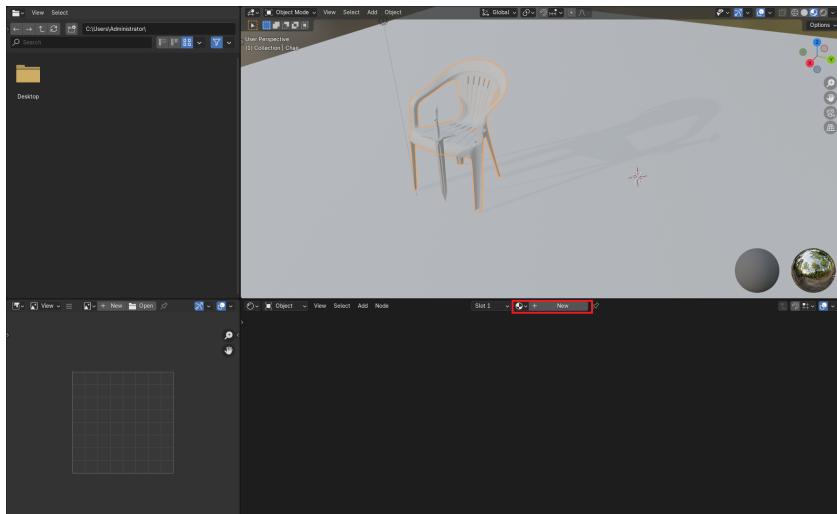


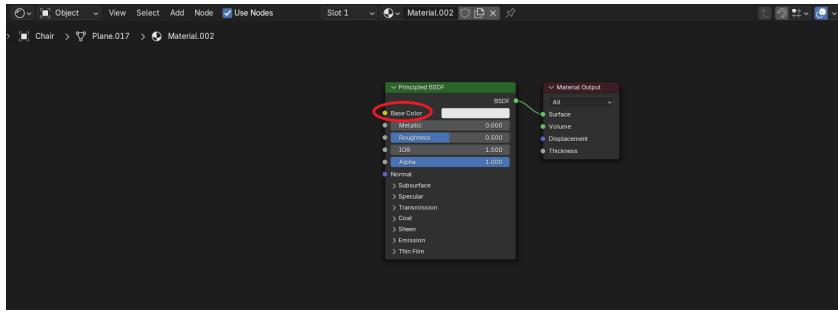
Here you should see the models, together with the materials. By selecting a model, you can find in the Shader Editor below about how the textures are applied to the model. Usually, each texture (map) specify a certain property. For example, a color (or albedo) map is connected to the Base Color property of the model, which defines per-vertex color.



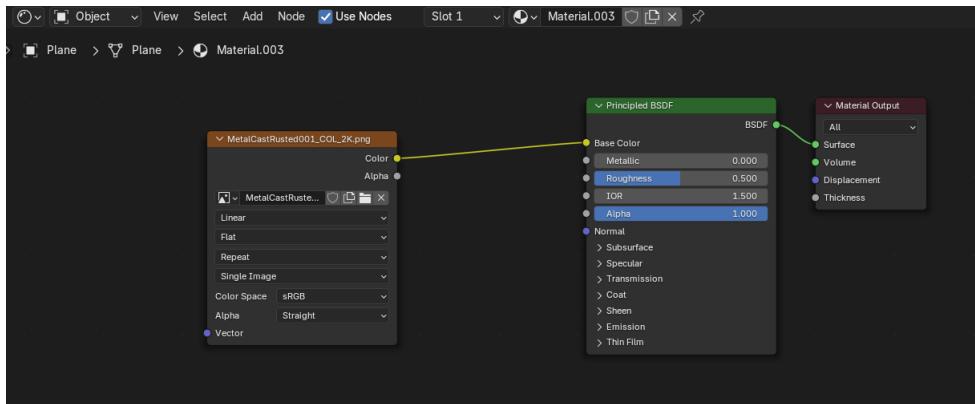
Now, your task is to download a color map and apply it to the plane we created. You can download free materials in websites like [poliigon](#), and extract the image (usually .png or .jpg) corresponding to the color property.

By default, a newly added geometry won't have any node in the Shader Editor. We therefore add a default material to it.

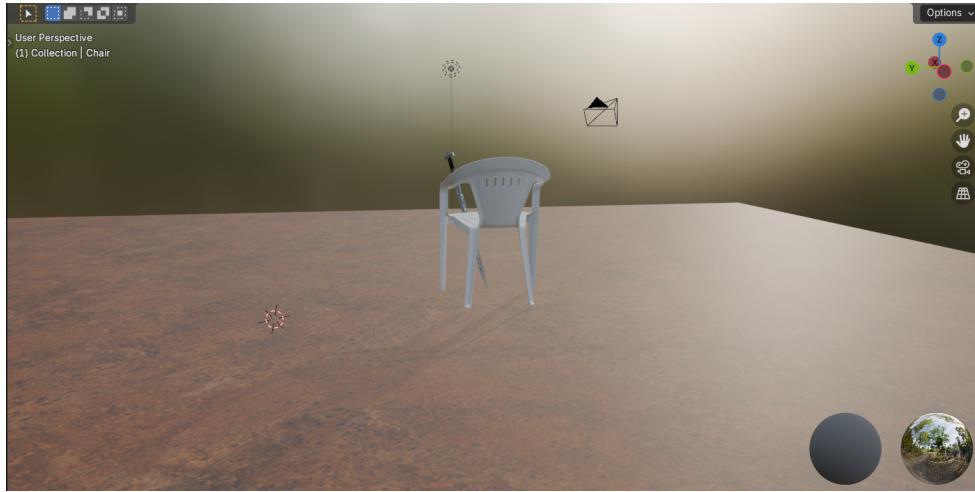




To achieve this, navigate to the folder of the textures, drag the color texture into the shader editor, and connect it to the "base color" property of the floor,



which gives us a floor with rusted color.

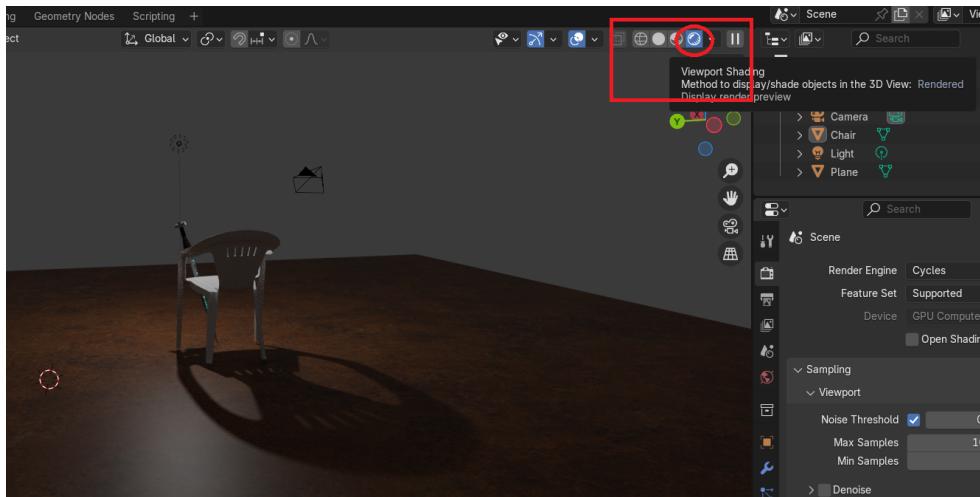


Please note that in the example image, the chair has no texture, for a comparison with the textured sword and plane.

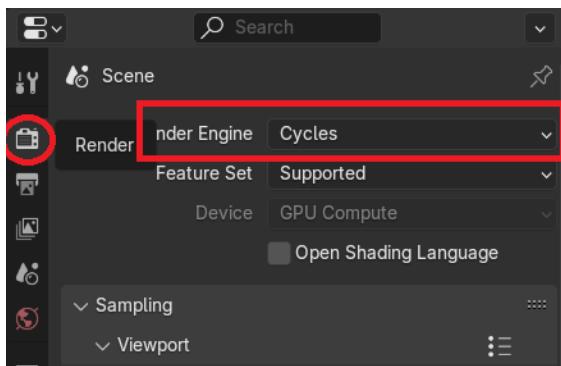
3.3 Point lights

The scene looks good under the default sky light in the material preview mode. However, we don't actually have such nice lighting for now. To view the actual scene, switch to the rendered mode, where you can preview how the final image will look like. Note that previewing in this mode consumes quite a lot of both CPU and GPU resources, so you

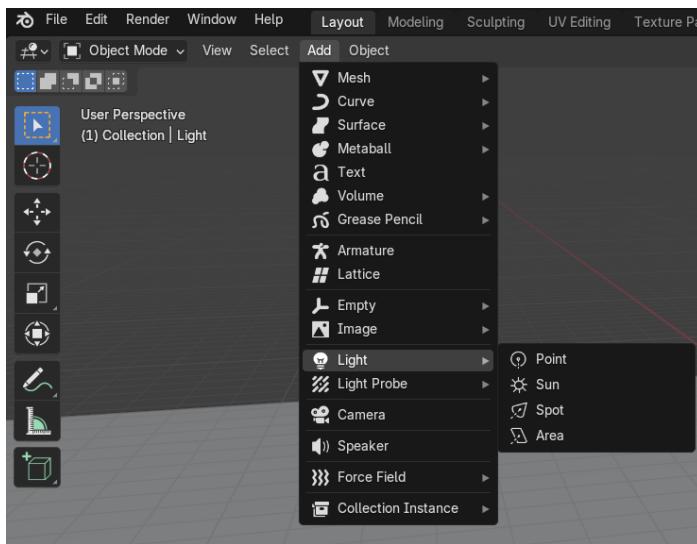
may switch back to material or solid preview when you don't need to preview the rendered results.



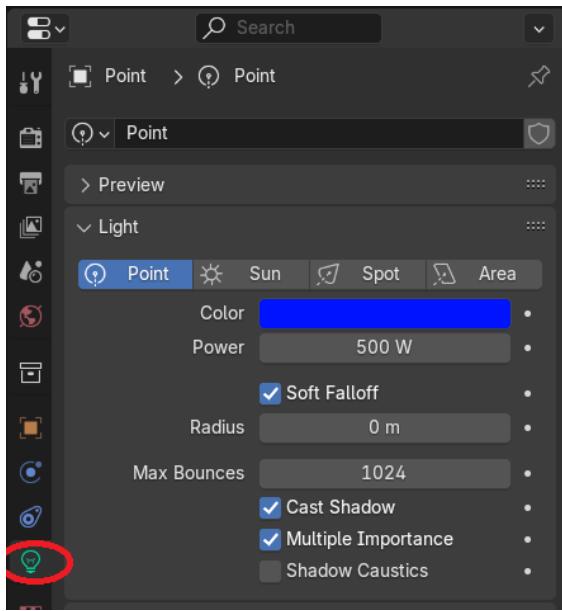
Also, we need to switch Blender's render engine. By default it uses EEVEE, which uses rasterization. To gain better results, we switch to the ray tracing engine - Cycles, in the Properties Editor at the bottom right.



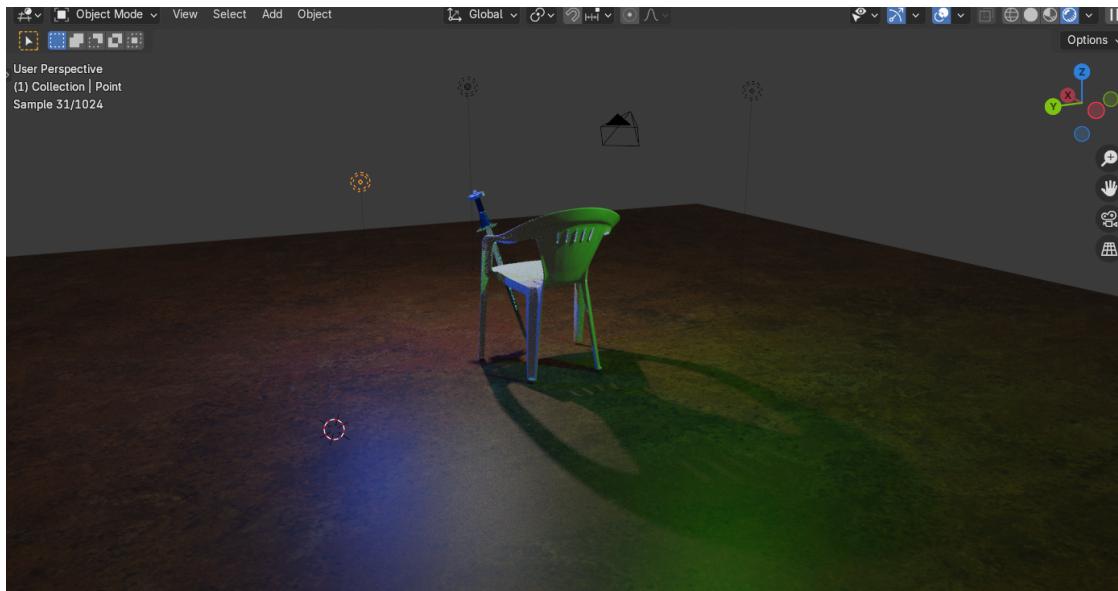
Now you may find your scene looks quite dark, as the default scene only provides a single point light. In this part, your task is to add at least 2 more lights, with different colors, to light up the models.



You can modify the properties of each light in the Properties editor.



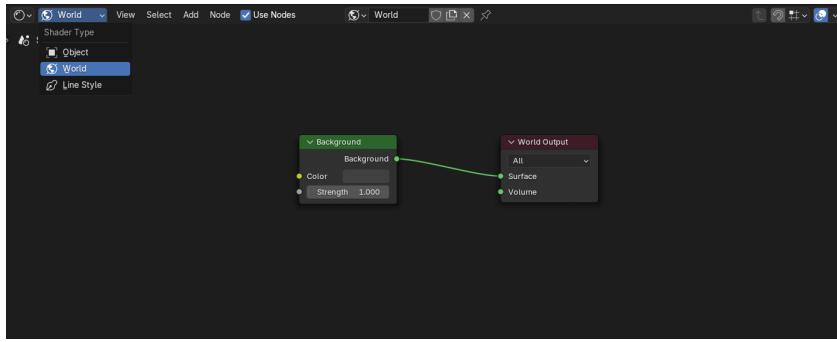
The scene after this step looks like the following.



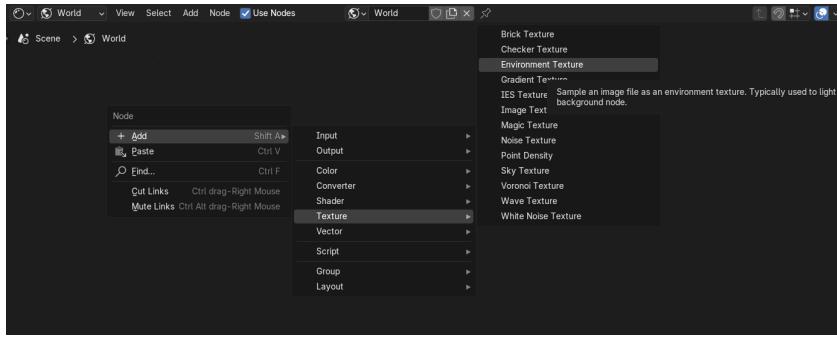
3.4 Skybox

There is still one part missing from the scene - the sky. To add a skybox, first download a HDR (high dynamic range) sky texture from websites like [polyhaven](#). It is typically in .hdr or .exr format.

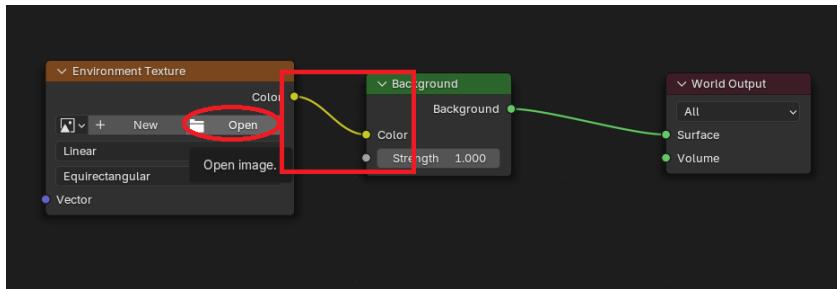
To apply the texture to our scene, first switch to the shader for the world in the Shader Editor.



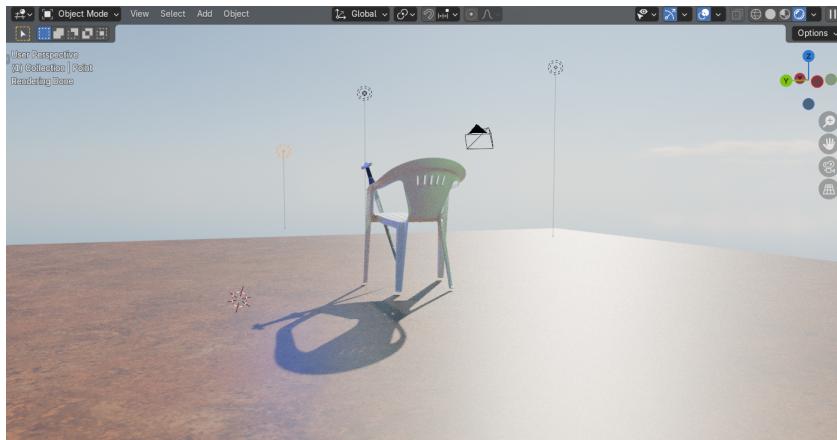
Next, right clicking in the Shader Editor (or click on "Add" add the top of the editor), and create an environment texture.



Then, apply the downloaded texture as the image of it, and link the nodes.



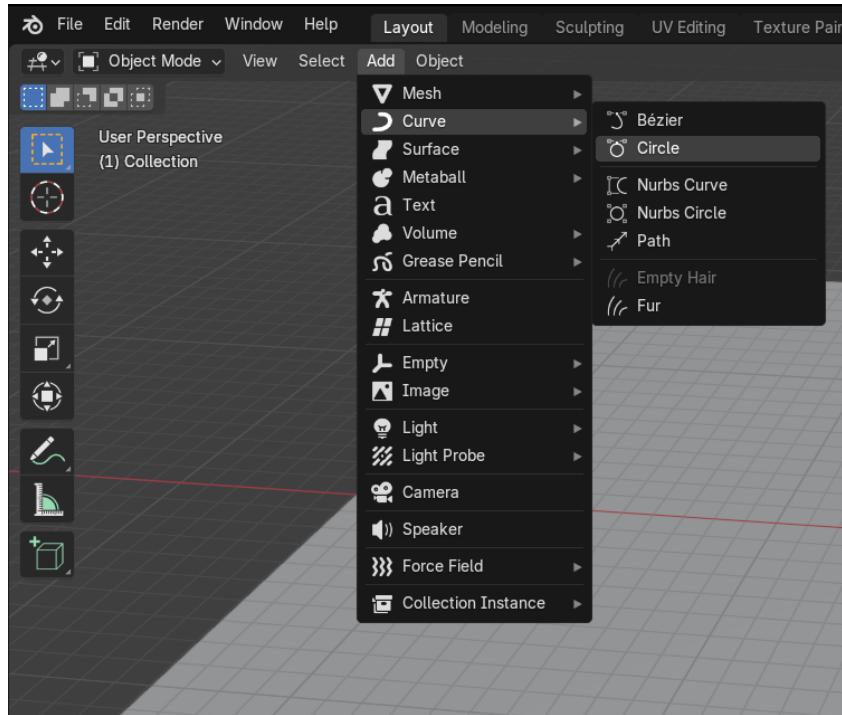
Now we have a sky texture (and by default, sky light) for our scene.



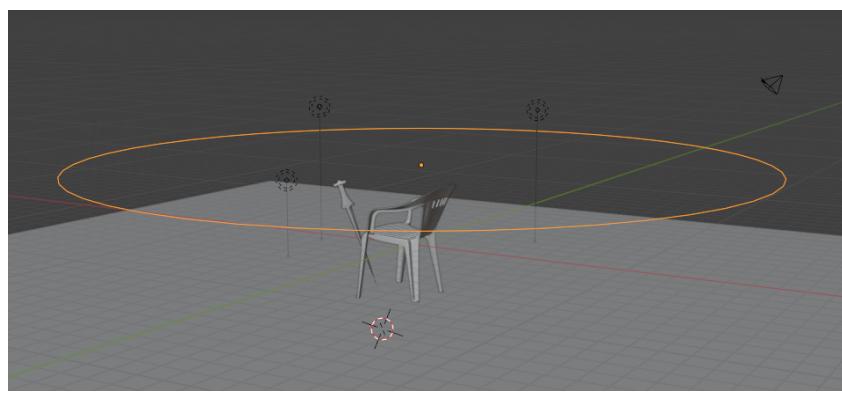
3.5 Camera animation

Finally, we want to render a short animation to demonstrate our scene. This can be done through a camera animation. In this part, your task is to produce an animation by rotating and orbiting the camera around the center of your scene for a whole cycle.

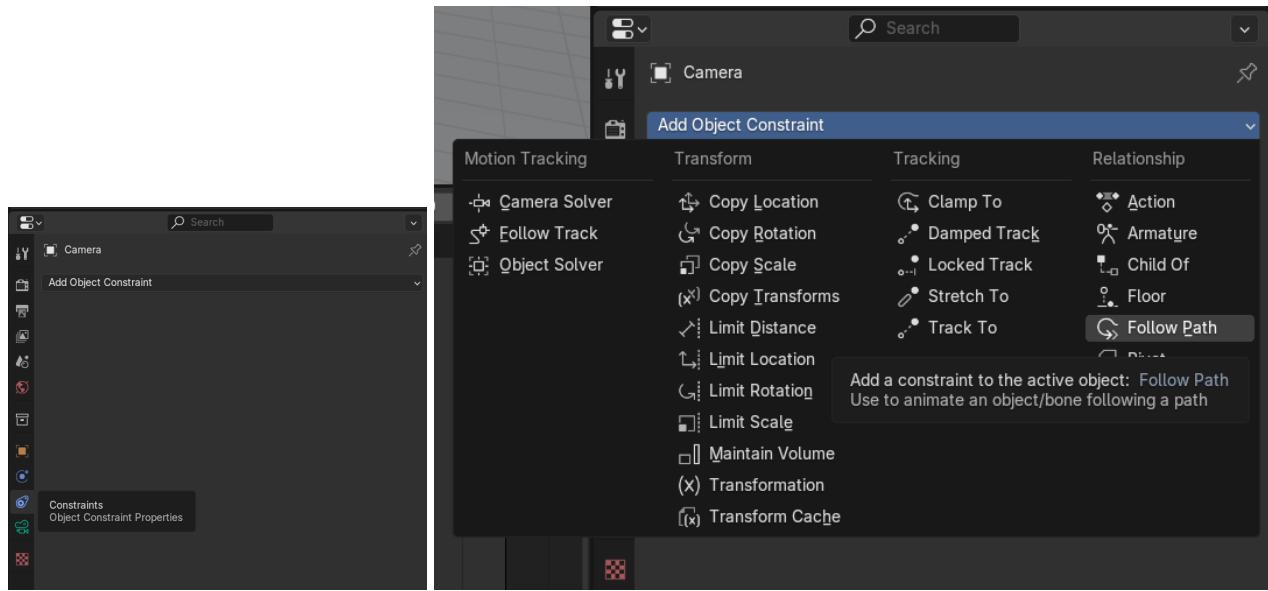
First, we need to set up a path for the camera to follow. Here it should be a circle around the scene. We go back to the "Layout" workspace, and add a circle.



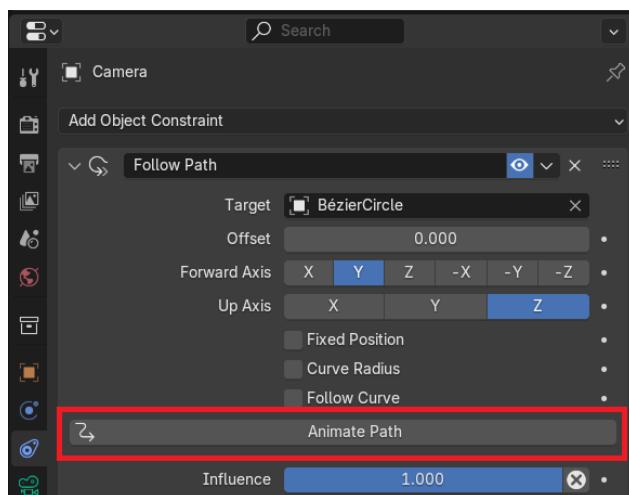
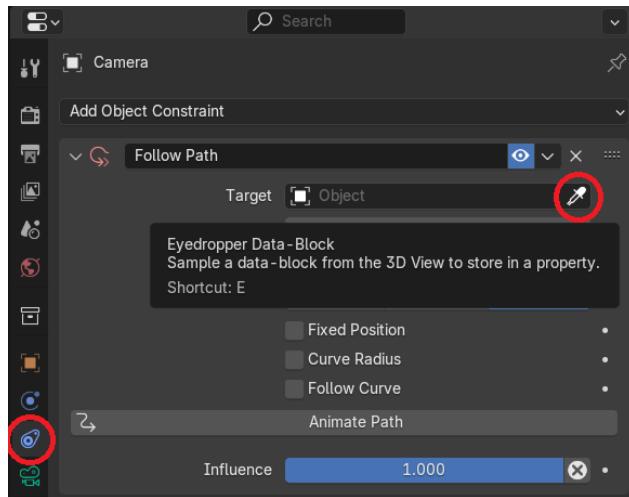
Then, adjust the size and position of the cycle, such that it becomes a path around your scene.



Next, we want to attach our camera to the circle. This can be done through adding a follow path for the camera. Select the camera object, and in the Properties Editor, add an object constraint, and select follow path.



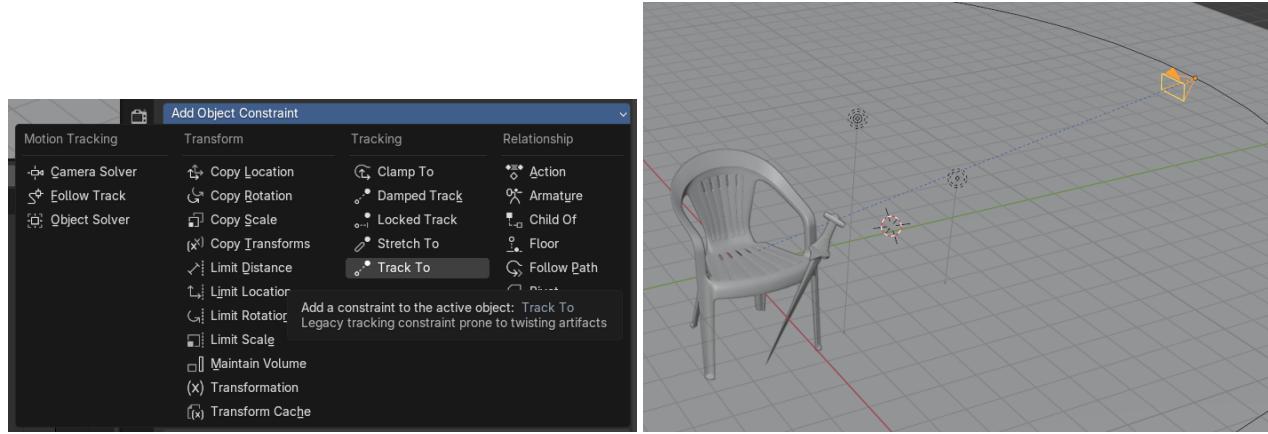
Then, use the eyedropper to select the circle as our target, and click "Animate Path" - now the camera will have a default animation, following the path.



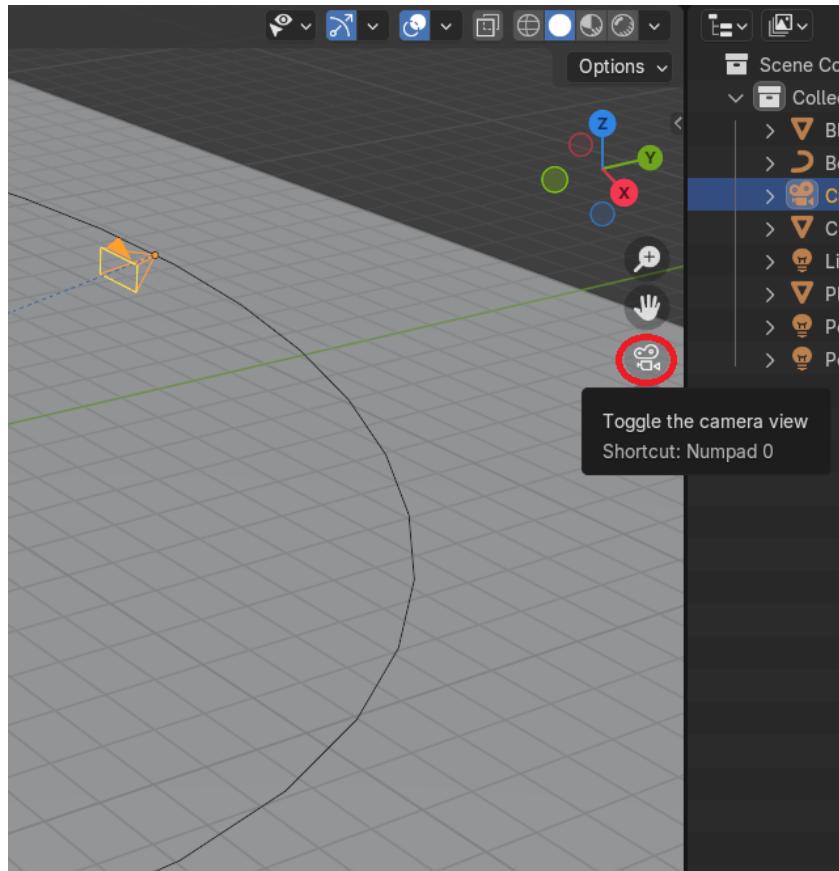
After this step, the transform property of the camera becomes relative to the circle. Since we want the camera to exactly

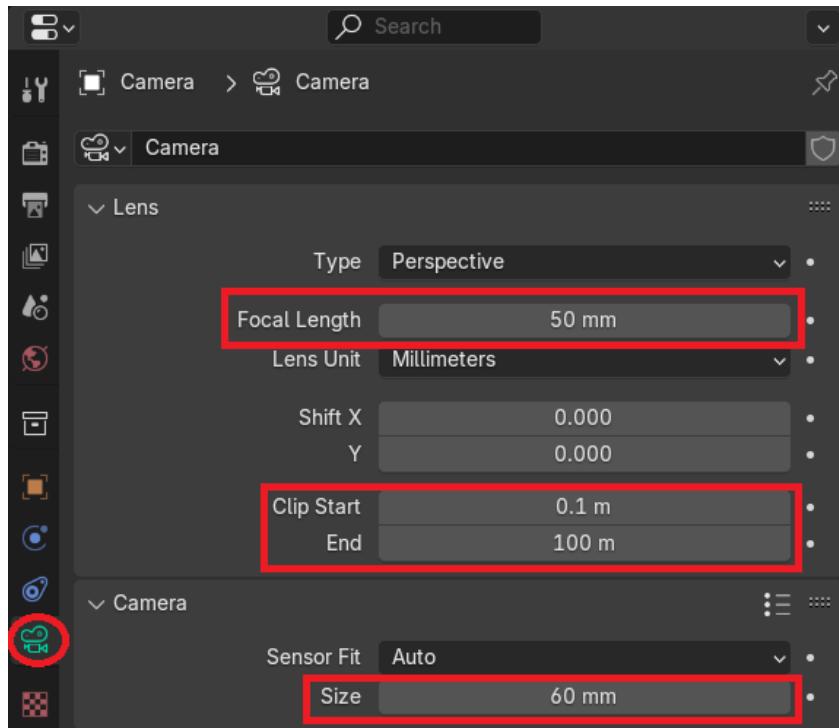
follow the path, we set its location to $(0, 0, 0)$ in the Properties Editor. Now, the camera will appear on the path.

After setting the position of the camera, we want to orient the camera towards the scene as well. To do this, add another constraint, "track to", and use the eyedropper to select the target that you want the camera to focus on.

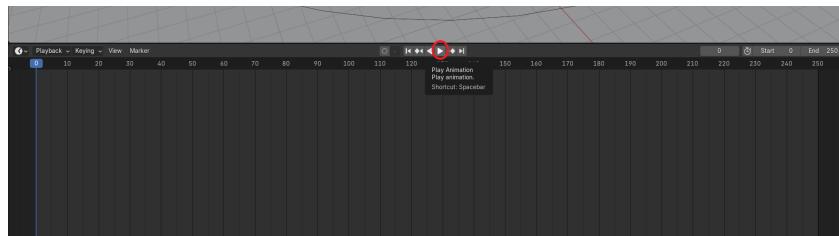


You can view from the camera by toggling the camera view, and adjust the focal length and size of your camera to fit the scene.

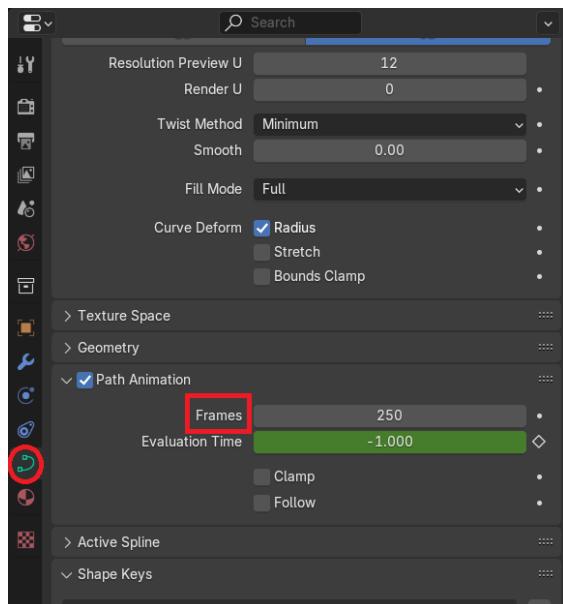




Click "Play Animation" in the Timeline Editor, and you can see the camera orbiting.



You can optionally adjust the speed of rotation through the frames property under "Path Animation" of the curve, in the Properties Editor. It specifies in how many frames the camera complete one cycle.



4 (Optional) Exploration Task

In this optional exploration task, we expect you to build a nice-looking and relatively complex scene, and render an image of it (you don't need camera animation). Also, the image should demonstrate the advantages of ray tracing. Specifically, identify the effects that can hardly be achieved with pure rasterization in real-time and without advanced rendering tricks.

Please note that, though we didn't implement texture mapping in homework1, it is not something hard to achieve for rasterization. On the other hand, the sky box texture we added in this project cannot be easily rendered in a rasterization pipeline without additional tricks, like image-based lighting and prefiltering. (As sky box is already part of the basic tasks, it cannot be counted as extra credits.)

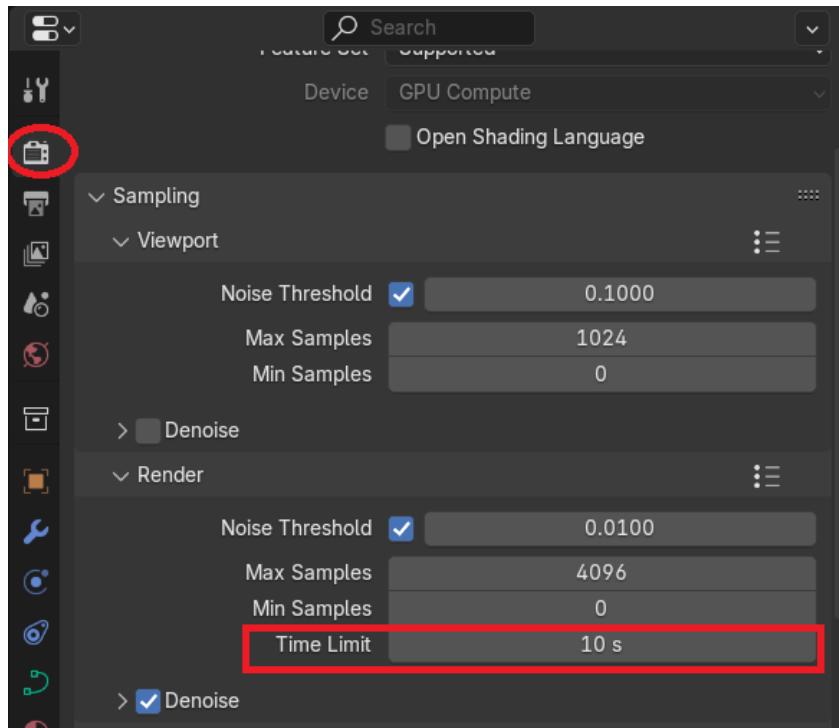
You also need to briefly explain how the features are achieved in ray tracing, and/or why they cannot be easily achieved in pure rasterization.

5 Submission and Grading

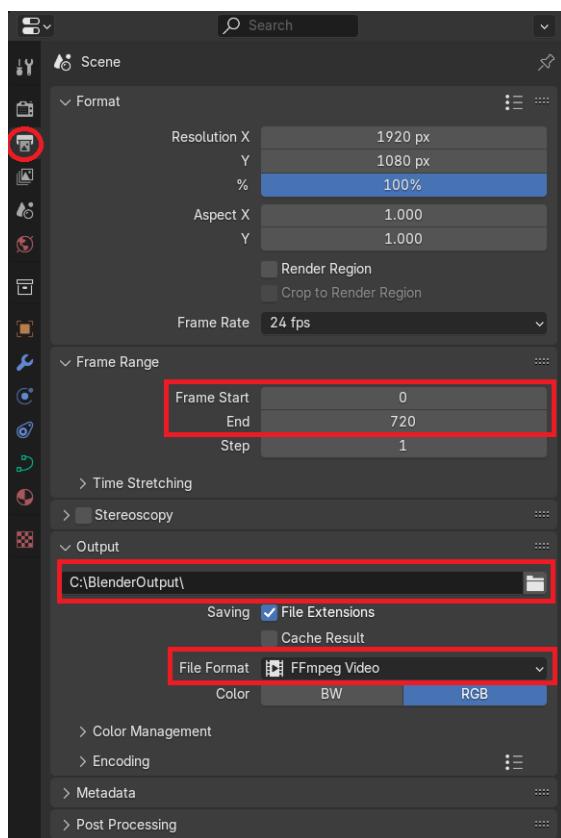
There's no coding or PDF submission for this homework. Instead, submit the animation video and the blender project file.

5.1 Rendering the Animation for Submission

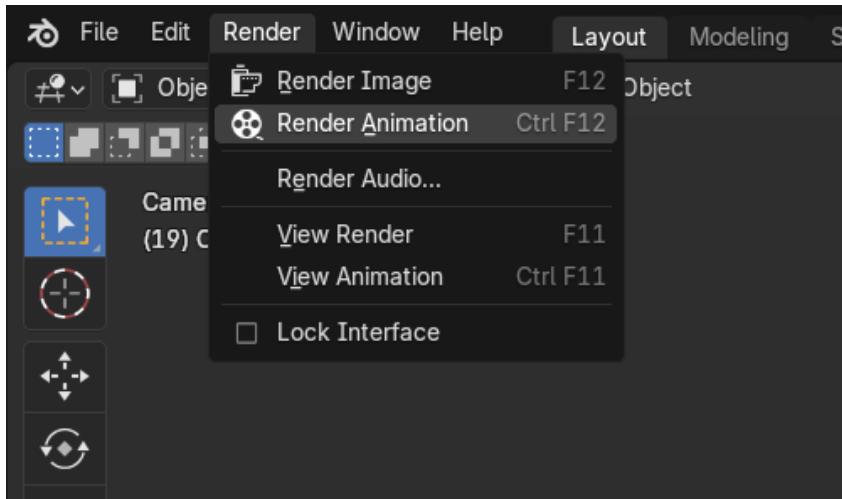
To render the final animation, we need to first adjust the rendering settings in the properties panel so that it won't take too long to render a single frame. As there is no explicit requirement for the video quality, 2 seconds would usually be sufficient. You may need to set it longer if you have a complex scene, or effects like fogs that take much longer to render.



Next, adjust the output settings. This includes the resolution, output directory, and length of the video. The resolution should be at least 800×600 px, the frame rate should be at least 24 fps, and the length should be between 10 to 60 seconds, which means 240 to 1440 frames under 24 fps. Remember to change the file format to Video, otherwise the frames will be output as images. Here is example setting:



Finally, render the animation through Render - Render Animation, and you can find the video at the output directory when it finishes rendering.



The video should clearly demonstrate the elements required in each part.

5.2 Packing The Blender Project

Your blender project file should contain all the external resources that are used. To do this, [click pack resources before saving](#) and submitting the .blend file.

This step is important, as your scene can be incomplete without packing the necessary resources.

5.3 Exploration Task Submission

Submit an image, a short text file (.txt, .pdf, etc.) that briefly explains what the features are, and the .blend project file.

If you use the same project as in the basic task, just submit a single final project file for both. Otherwise, you'll need to submit two .blend files.

Here is an example of explaining the features:



In this image, the soft shadow cast by the boxes is hard to achieve in pure rasterization. Traditional shadow map technique in rasterization only gives hard shadows, as we assume all light sources to be point lights. Therefore, every surface point in the scene is either lit or shadowed. In ray tracing, we sample multiple rays on different points within the area light source, thus it would correctly depict the effect of being partially occluded.

5.4 Grading

1. Basic Scene (20%): At least an object imported from external assets, and a plane created in blender.
2. Textures (30%): At least one of the imported objects should be accompanied with a texture. The plane should be manually textured via the material nodes.
3. Lights (30%): At least two point lights (aside from the default one) in the scene. There should also be a sky box.
4. Animation (20%): There should be a camera animation that orbits around the scene.
5. Exploration (extra credits, 10%): Based on completeness, efforts, and correctness.

Submission checklist:

- A video for the animation
- A blender project file
- (Optional) Another blend project file, if you use a separate project for exploration
- (Optional) An image, and short text explaining the extra features you implemented. We will grade the extra features part if and only if the text file presents.

Compress all items into a single .zip file, and submit to Canvas. The extra features part should be submitted together with the required part.

Remember to pack the external resources into your blender project before submission!