
NEW!! Installation with pip

To use this toolbox, please make sure to install Blender (version 4.0.0) and then install the toolbox with the following commands

```
1 conda create -n blender python=3.10
2 source activate blender
3 pip install blendertoolbox
4 pip install bpy
```

Please make sure you're using python 3.10, as Blender Python `bpy` is only compiled for that version of python.

Once it is installed successfully, one can simply type in, e.g.,

```
1 python template.py
```

to render the mesh.

If for some reasons, `pip` has difficulty installing the packages. One can still fall back to the old way: downloading the toolbox and import the module `blendertoolbox` manually. This will require some changes to the demo scripts. After the modifications, you would need to call the script with blender background mode as

```
1 blender -b -P template.py
```

(We especially want to thank Otman Benchekroun for making the pip integration possible!)

Lazy Pipeline

This is the step-by-step tutorial on how to use the `default_mesh.py` to render a mesh in few minutes.

First, in `default_mesh.py`, you can set `mesh_path` to the desired mesh path you would like to render

```
1 "mesh_path": "path/to/your/mesh.obj"
```

Now go to your terminal, and then run the script using

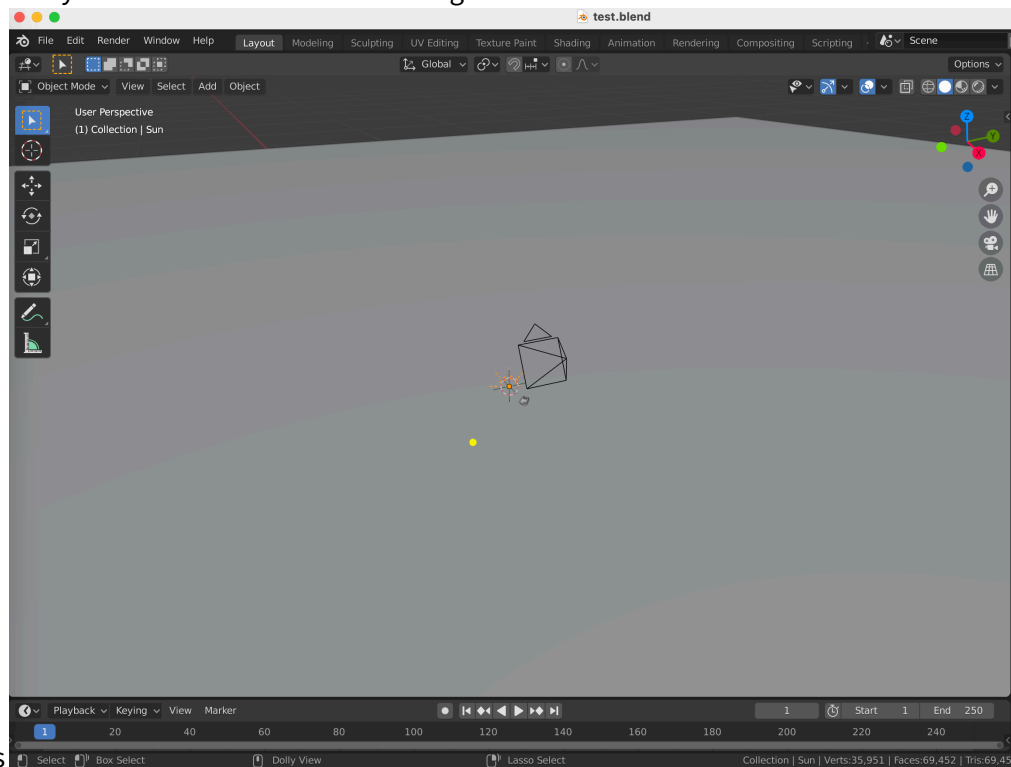
```
1 python default_mesh.py
```

This will execute the script and you will see the progress of rendering like

```
1 ...
```

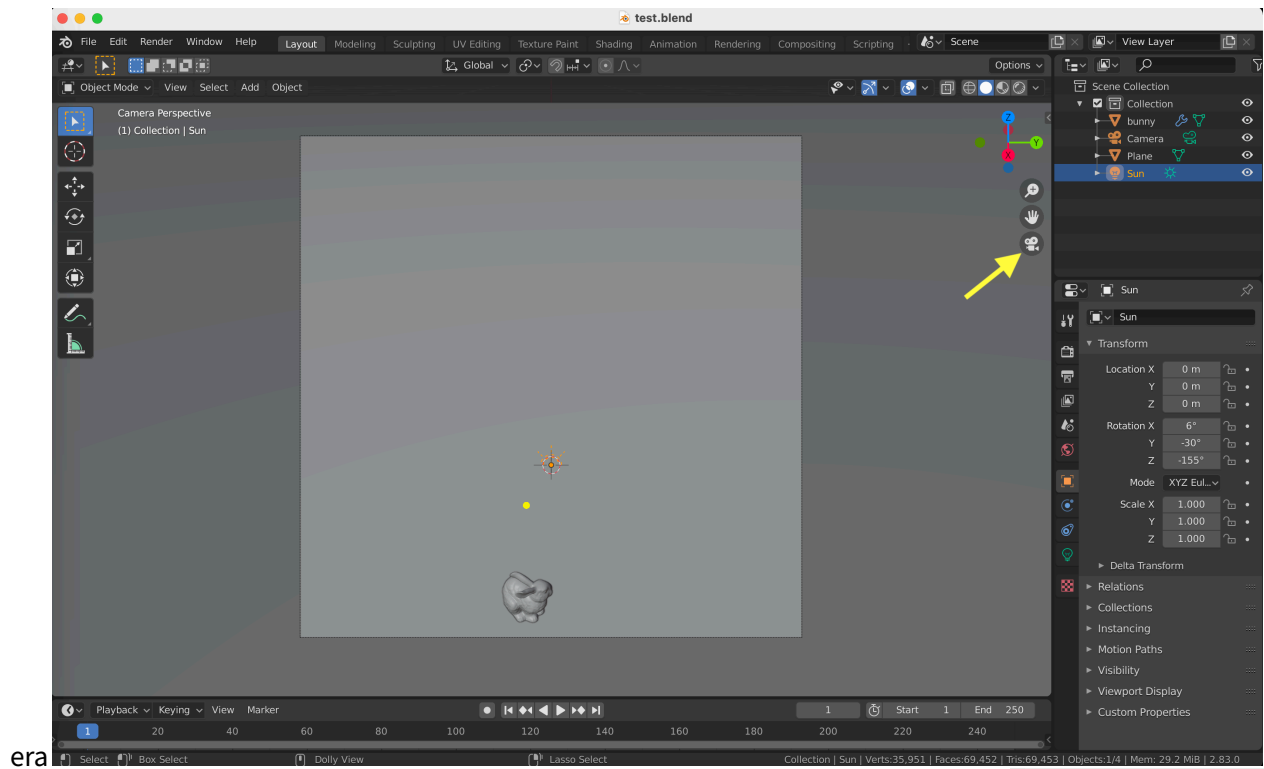
```
2 Fra:1 Mem:57.95M (0.00M, Peak 58.09M) | Time:00:02.00 | Remaining  
:00:08.06 | Mem:23.22M, Peak:23.22M | Scene, View Layer | Rendered  
1/64 Tiles, Denoised 0 tiles  
3 Fra:1 Mem:58.45M (0.00M, Peak 58.65M) | Time:00:02.69 | Remaining  
:00:09.32 | Mem:23.72M, Peak:23.72M | Scene, View Layer | Rendered  
2/64 Tiles, Denoised 0 tiles  
4 ...
```

Once you see those messages, you can manually terminate the program (e.g., `Ctrl + C` on Mac). The purpose of running this is to create a `test.blend` file so that we can adjust the mesh position in the Blender UI. So now open the newly created `test.blend` file using Blender software. You should

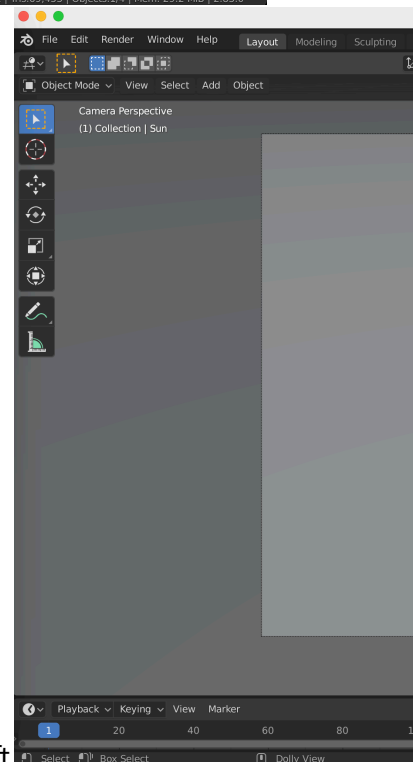


land on the default scene like this

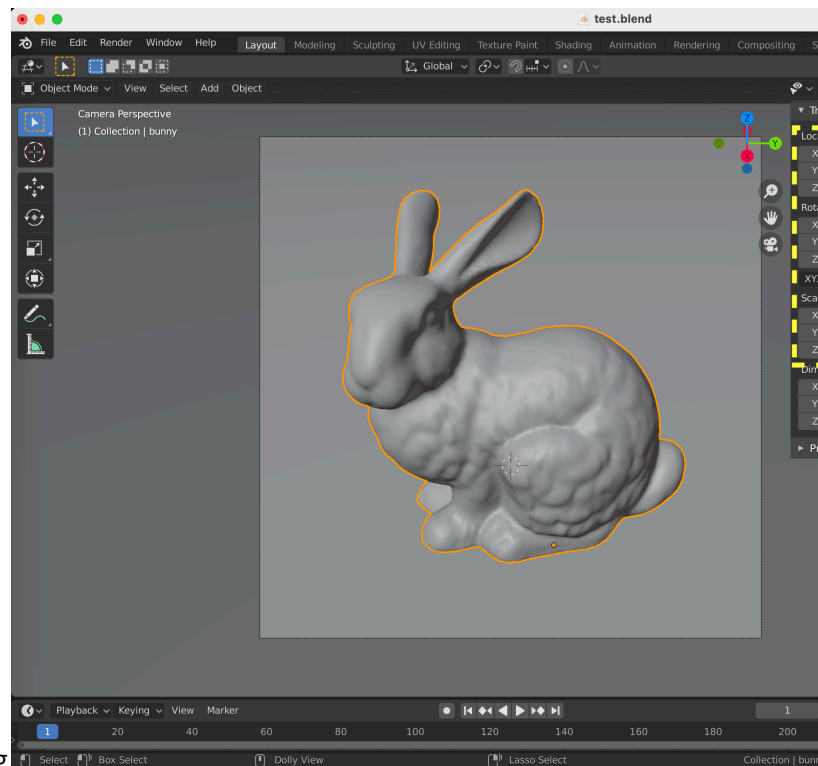
The next step is to click this “camera bottom” (yellow arrow) to view the scene from the eye of the cam-



era



The next step is to click the mesh in the scene and drag the tiny arrow towards the left. You should see a new window that displays the mesh information (position, rotation, scale). The final



step is to adjust all these parameters to your liking. Then you can copy those adjusted parameters from the Blender UI back to `mesh_position`, `mesh_rotation`, `mesh_scale` in the script respectively. The above steps are all you need in order to render a mesh, you can now re-run the script in the terminal

```
1 python default_mesh.py
```

This will output a `.png` file as your final rendering result.

We also expose other commonly adjusted parameters, such as lighting angle, mesh color, image resolution, etc. For more details, please read the comments in the `default_mesh.py`.

The sample rendering pipeline works for all the demo files in the repository. Other “non-lazy” demos expose more parameters for users to tune. However, in order to know how to adjust those parameters require some background knowledge in Blender. I refer users to a quick introduction to Blender UI from Silvia’s website. There are also a ton of online tutorials which offer more advanced stuff way beyond what I have covered here.

Contact

These scripts are tested on Blender 4.0.0. As the API may change, using a different version of the Blender may cause some functions not working properly. If you notice some bugs due to Blender updates or any questions/recommendations, please contact hsuehtil@gmail.com.