HW1 Blender Basics

CS 148 Fall 2021-2022
Due Date: Monday, 27 September 2021 by 7 pm PT

Follow the instructions carefully. If you encounter any problems in the setup, please do not hesitate to reach out to CAs on Piazza or attend office hours.

Be aware of the **Checkpoints** below. Make sure you complete each one since we will do the grading based on them.

I. Blender Basics

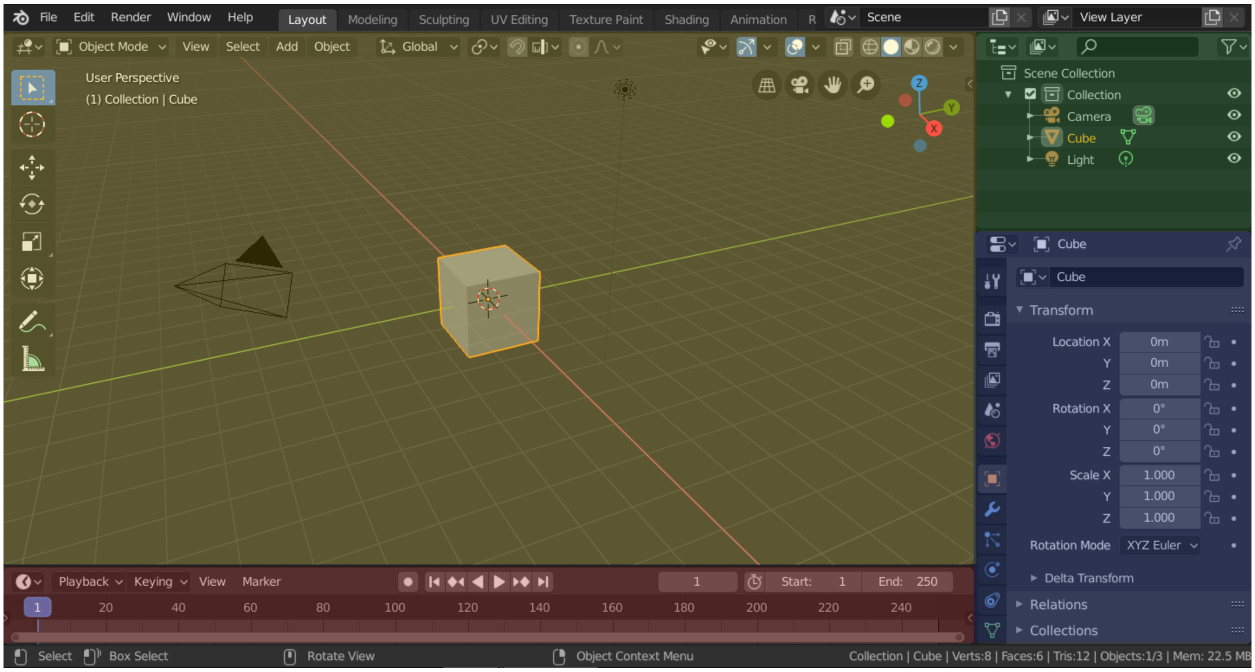
💡 What is Blender? Blender is a free and open-source 3D creation suite. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. Examples from many Blender-based projects are available in the showcase. (from Blender website)

💡 What we'll do with Blender in this course? We will learn the basics of modeling, texturing, shading, and rendering inside Blender. We'll also use Blender's API for Python scripting to write a custom raytracing render engine. Blender is well suited to this class because of its unified pipeline and responsive development process.

Download Blender 2.93.4 from https://www.blender.org/ and install it.

☐ **Checkpoint 0: Run Blender.**

There should be a default scene with a cube, a light, and a camera:



Location X 0m
Y 0m
Z 0m
Rotation X 0°
Y 0°
Z 0°
Scale X 1.000
Y 1.000
Z 1.000
Rotation Mode XYZ Euler

Start: 1 End: 250

Collection | Cube | Verts:8 | Faces:6 | Tris:12 | Objects:1/3 | Mem: 22.5 MB

3D Viewport (yellow), Outliner (green), Properties (blue) and Timeline (red). From Blender documentation.

This is the default workspace for blender (see more at blender documentation):

- 3D Viewport on the top left (yellow)
- Outliner on the top right (green)
- Properties editor on the bottom right (blue)
- Timeline on the bottom left (red)

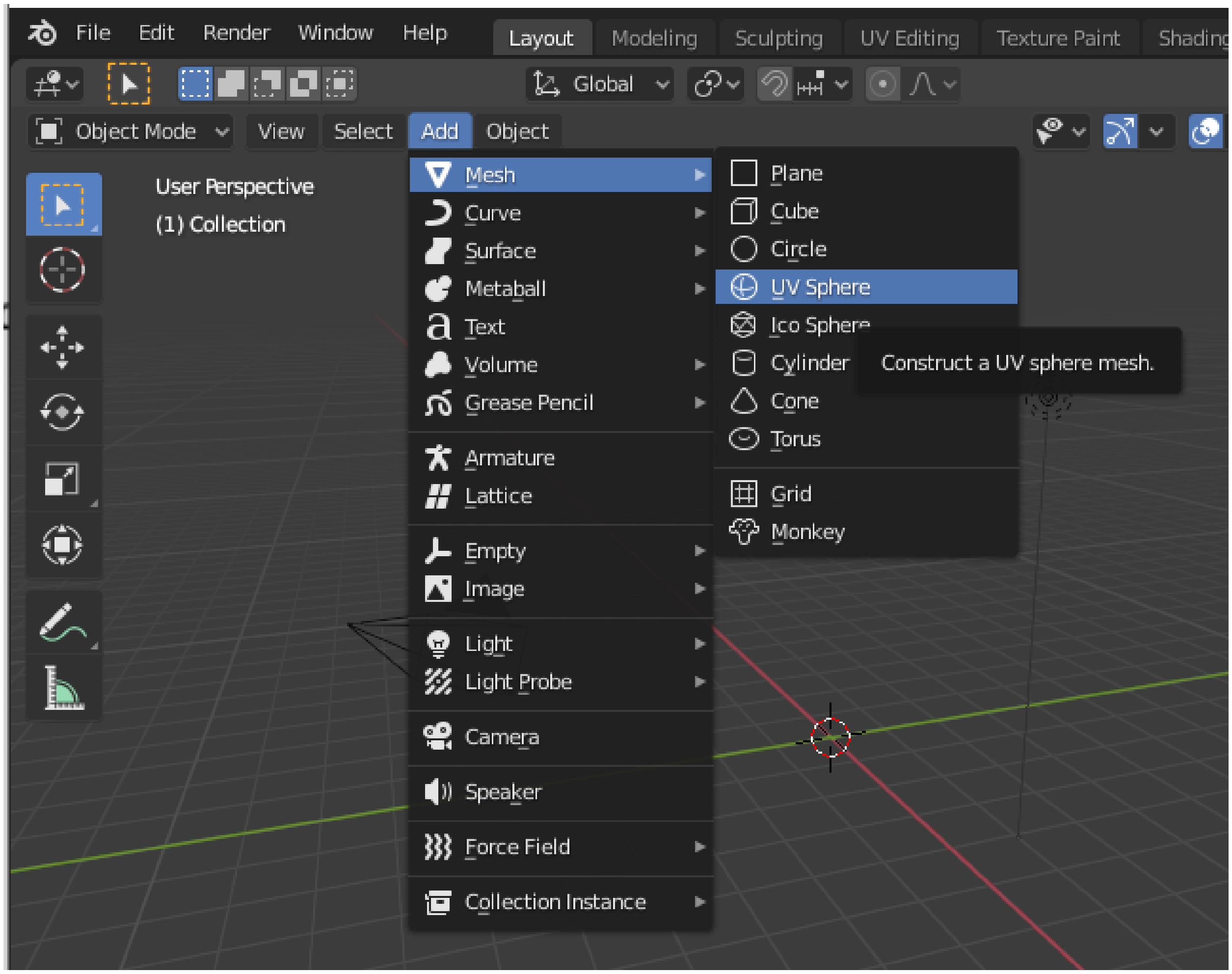
Blender has a nice official introduction playlist for 2.8 on their youtube channel. They're compatible with 2.93. If you haven't used Blender before, we suggest watching the first two videos to learn the basics of how to get around inside Blender. But don't worry, we will still provide detailed instructions below to help you get started. We will also include some links to Blender documentation along the way if you want to further investigate the tool.

1. Manipulate the Scene

> 📖 The default scene comes with a cube. We can add/delete objects in the scene. Blender stores some basic objects that we can directly add to the scene without constructing them from scratch. Here we will delete the cube and add a UV sphere.

Move your mouse over the cube, left click to select. An orange outline will appear around the cube, indicating it's being selected. Hit the Delete key to delete the cube.

Then go to the menu bar on top, go to Add ▸ Mesh ▸ UV sphere (or leave your mouse inside the 3D viewport, and use the shortcut Shift A). A sphere should appear in the center of the scene.

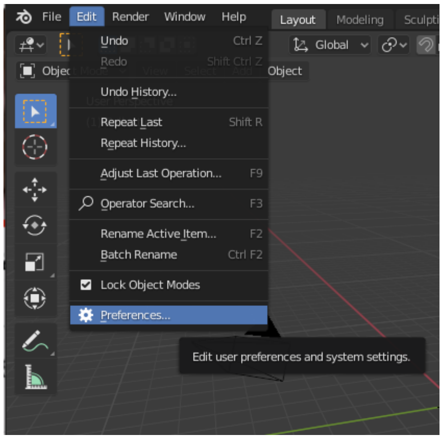


Construct a UV sphere mesh.

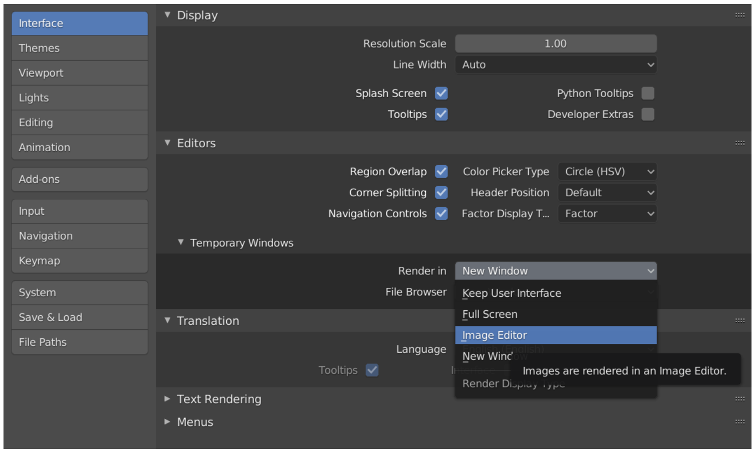
1½ (Optional) Set up editor and workspace

📖 (Optional but recommended for first-time users) First set up the editor. This will keep the render in the same window, without popping out a new window every time.

Go to the top bar, Edit ▸ Preferences (fig_1). A new window should pop up, on the sidebar select Interface, then change the option for Editors ▸ Temporary Window ▸Render in to Image Editor (fig_2).



Edit user preferences and system settings.



▾ Display

Resolution Scale 1.00

Line Width Auto

Splash Screen ☑ Python Tooltips □

Tooltips ☑ Developer Extras □

▾ Editors

Region Overlap ☑ Color Picker Type Circle (HSV)

Corner Splitting ☑ Header Position Default

Navigation Controls ☑ Factor Display T... Factor

▾ Temporary Windows

Render in New Window

File Browser

Images are rendered in an Image Editor.

▾ Translation

Language

Tooltips ☑

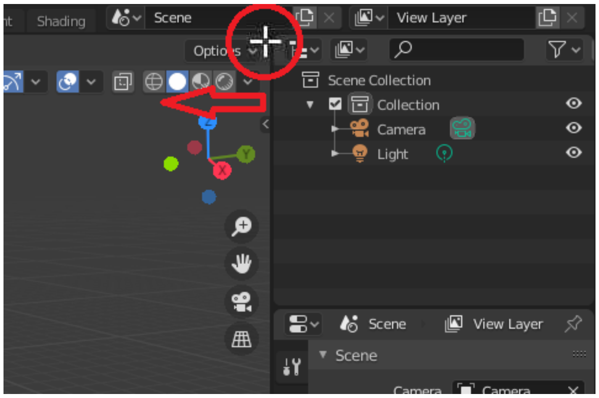
▸ Text Rendering

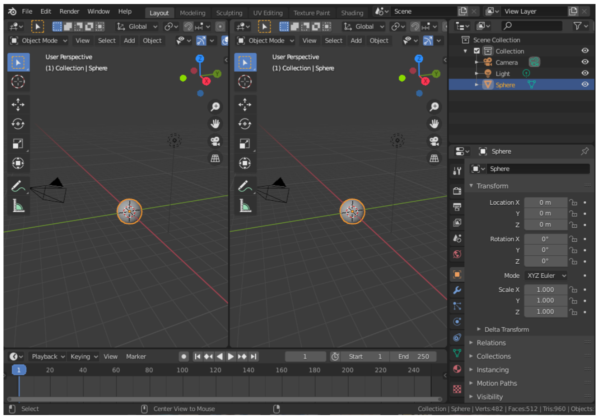
▸ Menus

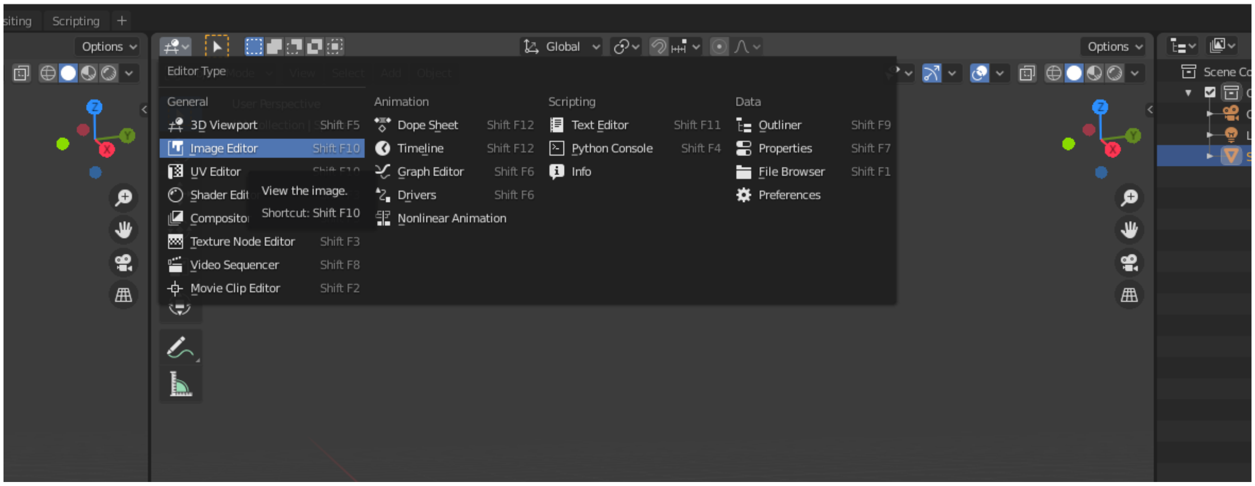
📖 (Optional but recommended for first-time users) Then set up the workspace: adding an Image Editor. This will be where we see our render results inside Blender. It will be more convenient than going to your file explorer and open the rendered image.

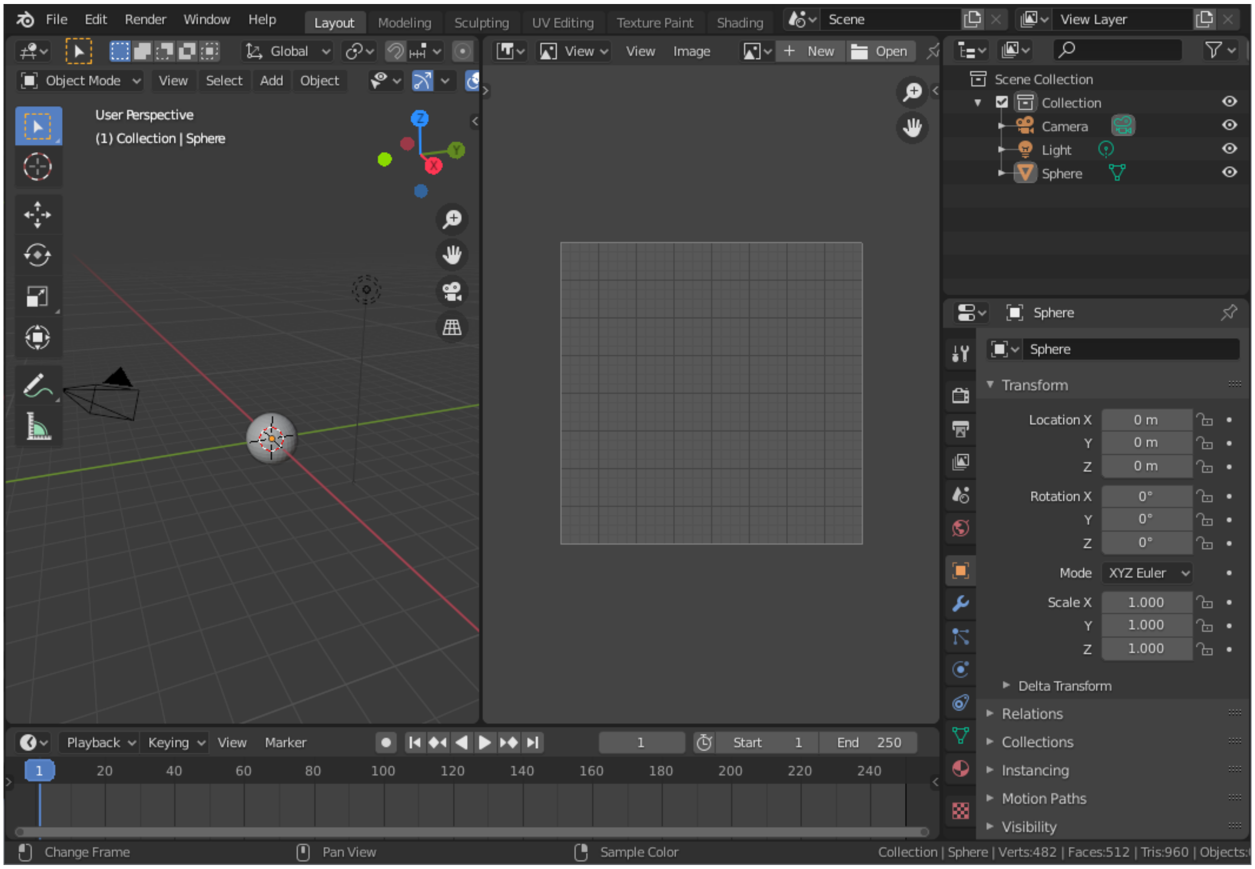
Hover the mouse over the top right corner of the 3D viewer. The cursor will change to a cross (fig 1). Drag left to create a new area (fig 2). In the top left of the new area, change the editor type to Image Editor (fig 3). Your editor should look like fig 4 now.

For more detail about areas and how to manipulate them, go to the documentation or the tutorial video.


Shading
Scene
View Layer
Options
Scene Collection
Collection
Camera
Light
Scene
View Layer
Scene
Camera
Camera


File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading
Scene
View Layer
Object Mode View Select Add Object
User Perspective
(1) Collection | Sphere
Scene Collection
Collection
Camera
Light
Sphere
Transform
Location X 0 m
Y 0 m
Z 0 m
Rotation X 0°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Visibility
Playback Keying View Marker
Start 1 End 250
Select
Center View to Mouse


siting
Scripting
Options
Global
Options
Scene C
Editor Type
General
3D Viewport Shift F5
Image Editor Shift F10
UV Editor
Shader Edit
Composito
Texture Node Editor Shift F3
Video Sequencer Shift F8
Movie Clip Editor Shift F2
View the image.
Shortcut: Shift F10
Animation
Dope Sheet Shift F12
Timeline Shift F12
Graph Editor Shift F6
Drivers Shift F6
Nonlinear Animation
Scripting
Text Editor Shift F11
Python Console Shift F4
Info
Data
Outliner Shift F9
Properties Shift F7
File Browser Shift F1
Preferences



Sphere

Transform

Location X 0 m
Y 0 m
Z 0 m

Rotation X 0°
Y 0°
Z 0°

Mode XYZ Euler

Scale X 1.000
Y 1.000
Z 1.000

Delta Transform

Relations

Collections

Instancing

Motion Paths

Visibility

Start 1 End 250

Collection | Sphere | Verts:482 | Faces:512 | Tris:960 | Objects:

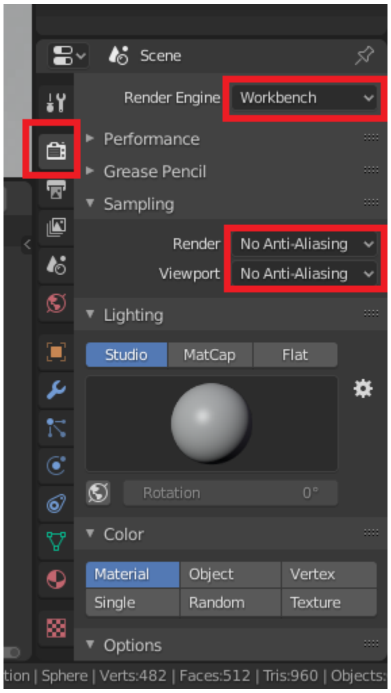
2. Render and save image

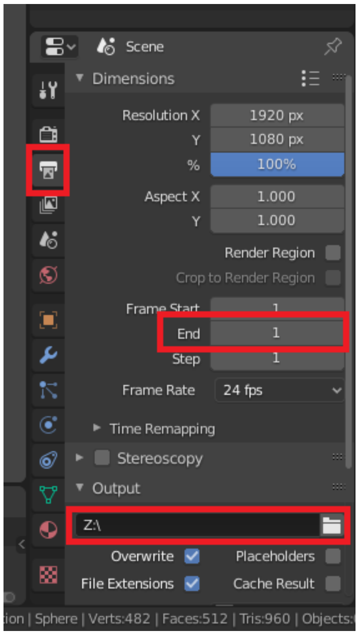
📖 After creating the scene, we can now change the render settings and render the image. By default Blender only saves animations, so here we change the length of the animation to 1. Also, we are playing with the default viewport scanline renderer for this homework, so we will use the Workbench render engine.

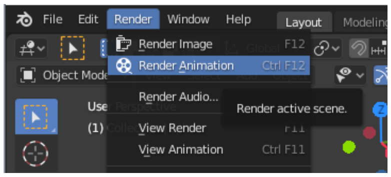
Go to the Properties Editor (bottom right), select the camera icon (Render Properties) on the sidebar. Change the Render Engine to Workbench, and the Sampling for Render and Viewport to No Anti-Aliasing (fig 1).

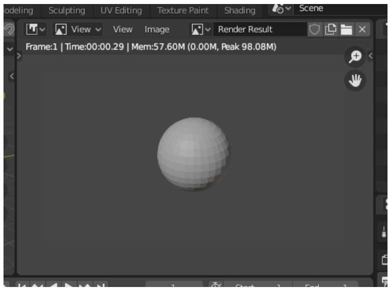
Then click on the printer icon (Output Properties) on the sidebar. Change the Frame End to 1 and Output Path to a place on your local machine where you want to save your rendered image (fig 2).

Now render the image by going to the top bar, go to Render ‣ Render Animation (or shortcut Ctrl/Cmd F12) to render the image, and save it to the specified path automatically (fig 3). The rendered image will also show up as "Render Result" in the Image Editor we created in the previous step (fig 4).


Scene
Render Engine: Workbench
▸ Performance
▸ Grease Pencil
▾ Sampling
Render: No Anti-Aliasing
Viewport: No Anti-Aliasing
▾ Lighting
Studio | MatCap | Flat
Rotation 0°
▾ Color
Material | Object | Vertex
Single | Random | Texture
▾ Options
tion | Sphere | Verts:482 | Faces:512 | Tris:960 | Objects:


Scene
▾ Dimensions
Resolution X: 1920 px
Y: 1080 px
%: 100%
Aspect X: 1.000
Y: 1.000
Render Region □
Crop to Render Region □
Frame Start: 1
End: 1
Step: 1
Frame Rate: 24 fps
▸ Time Remapping
▸ □ Stereoscopy
▾ Output
Z:\
Overwrite ☑ Placeholders □
File Extensions ☑ Cache Result □
ion | Sphere | Verts:482 | Faces:512 | Tris:960 | Objects:


File Edit Render Window Help Layout Modeling
Render Image F12
Render Animation Ctrl F12
Render Audio...
View Render F11
View Animation Ctrl F11
Render active scene.
Object Mode



Modeling Sculpting UV Editing Texture Paint Shading Scene
View View Image Render Result

Frame:1 | Time:00:00.29 | Mem:57.60M (0.00M, Peak 98.08M)

You can also use Render ‣ Render Image (or F12) to render a single image, but it will not save the image by default. This is usually for a quick test render, but you can still save the rendered image with Image ‣ Save Image (or Alt/Option R) from the image editor.

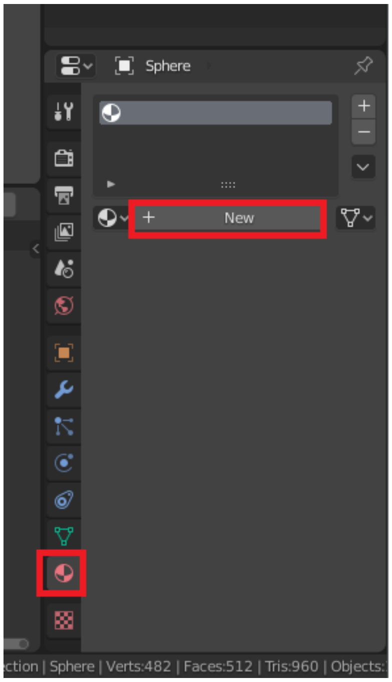
Now you have rendered your first image in Blender! Note that this image is rendered from the camera's perspective.

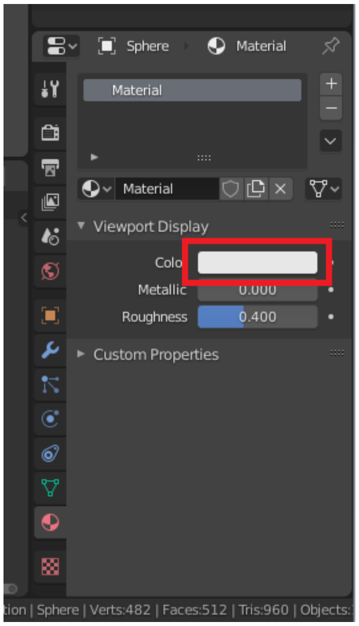
☐ **Checkpoint 1: Save the rendered image of the default sphere.**

3. Add material and Change Color

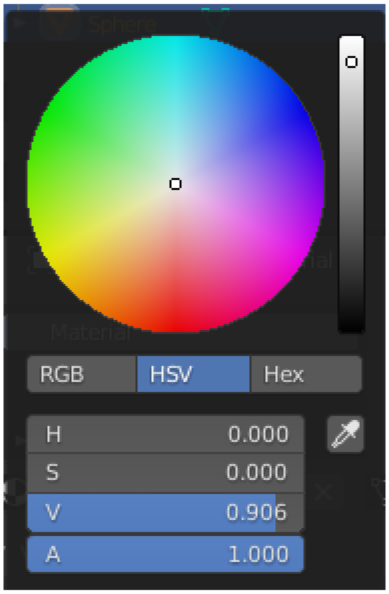
> 📖 We can add color, texture, and various shaders (which we'll talk about in two weeks) to objects to make them more appealing. In 3D software, it is usually done by first adding material to an object, then changing the settings of that material. We are only getting started now in a very basic scanline renderer, so our options are limited for now. Let's add color to the sphere first. We will play with lighting and shaders in other renderers in future homework.

Left-click on the sphere to select it. Go to Properties Editor ▸ Material Properties, click the New button to add a material to the sphere (fig 1). Under Viewport Display you can change the Color (fig 2). In the pop-up color palette (fig 3), you can move the dot in the color field and the slider on the right to change the color.


Sphere
New
ction | Sphere | Verts:482 | Faces:512 | Tris:960 | Objects:



Color
Metallic 0.000
Roughness 0.400



H 0.000
S 0.000
V 0.906
A 1.000

More on how to use color picker here: Blender Documentation.

You should be able to see the color change in the 3D viewport in real-time. You can also render the image to see the rendered result.

Before rendering to an output image (Render ▸ Render Animation or shortcut Ctrl/Cmd F12), change the Output Path or increase both Frame Start and Frame End by 1 to prevent your previous render from getting overwritten. Alternatively, if you have a rendered image in the Image Editor (Render ‣ Render Image or F12), you can also go to the top bar Image ‣ Save to save it.

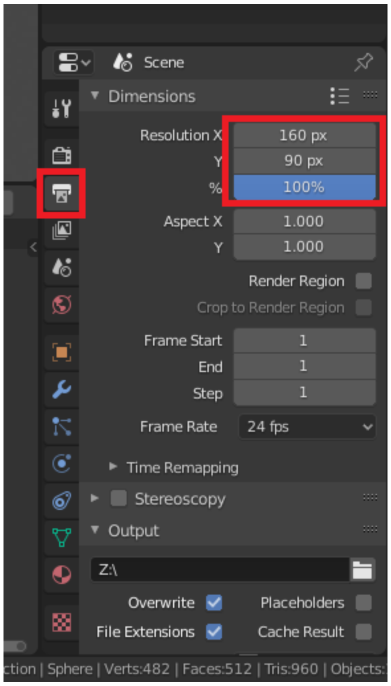
Now, pick any color other than the default light grey for the sphere and:

□ **Checkpoint 2: Save the rendered image of a colored sphere.**

4. Change Render Resolution

> 📖 We can change the resolution of our image. If you want a sharperimage, render at a higher resolution (larger X and Y). Vice versa, render at low resolution (lower X and Y). Here we want you to see and analyze the effect of tuning the resolution way down.

Go to the Properties Editor (bottom right), select the printer icon (Output Properties) on the sidebar. Change the Resolution X to 160, Resolution Y to 90. Make sure the Resolution % stays at 100 (fig 1).



Scene

Dimensions

Resolution X 160 px
Y 90 px
% 100%

Aspect X 1.000
Y 1.000

Render Region □
Crop to Render Region □

Frame Start 1
End 1
Step 1

Frame Rate 24 fps

Time Remapping

□ Stereoscopy

Output

Z:\

Overwrite ☑ Placeholders □
File Extensions ☑ Cache Result □

ction | Sphere | Verts:482 | Faces:512 | Tris:960 | Objects:

□ **Checkpoint 3: Save the rendered image of the colored sphere with a resolution of 160x90.**

□ **Checkpoint 4: Compare the images from Checkpoint 2 and Checkpoint 3. Discuss the effect of changing the resolution.** (No write-up needed, just get ready to answer this question during the grading session)

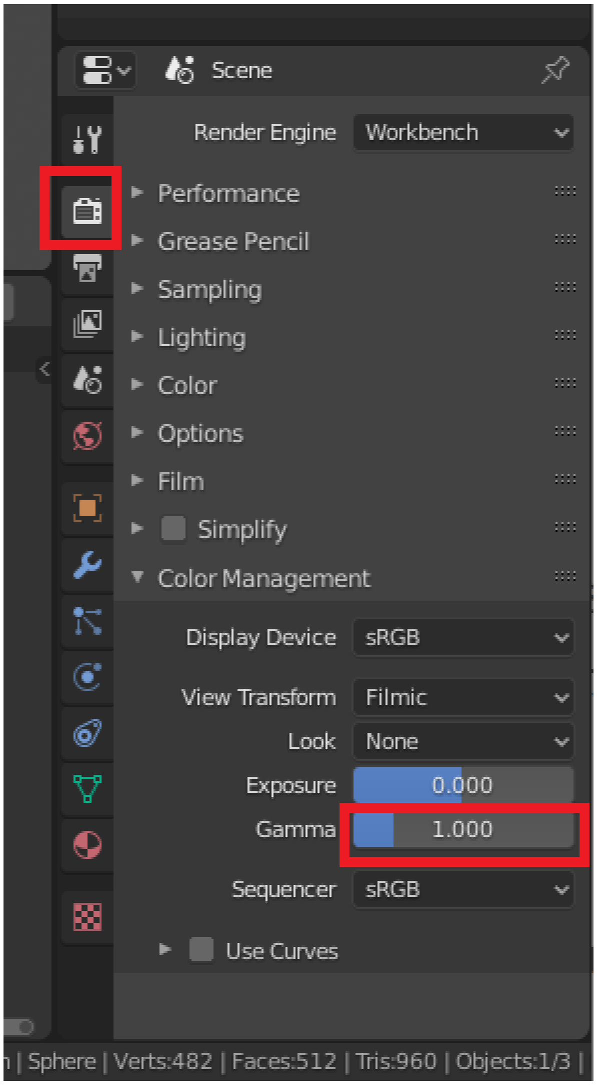
5. Change Gamma

> 📖 In the lecture, we talked about Gamma Encoding and Correction. Correct display of renders requires conversion to the display device color space. We are going to change the gamma value and compare the result.

> 📖 Note that we are not changing the actual color space conversion, but rather adding extra gamma correction applied after the default display transforms. This mainly acts as an additional effect for artistic tweaks. For more information, see the Blender documentation.

First, change the resolution in Properties Editor ▸ Output Properties back to 1920x1080. You can also right-click on the value, and select Reset to Default Value.

Then go to the Properties Editor ▸ Render Properties. Change the Color Management ▸ Gamma (fig 1) to some value other than 1.



Render Engine: Workbench

Display Device: sRGB

View Transform: Filmic

Look: None

Exposure: 0.000

Gamma: 1.000

Sequencer: sRGB

| Sphere | Verts:482 | Faces:512 | Tris:960 | Objects:1/3 |

- ☐ **Checkpoint 5: Save the rendered image of the colored sphere with a different gamma value.**
- ☐ **Checkpoint 6: Compare the images from Checkpoint 2 and Checkpoint 5. Discuss the effect of changing the gamma value.** (No write-up needed, just get ready to answer this question during the grading session)

II. Grading (5 pts total)

This assignment will be graded on the following requirements

Complete all the checkpoints (4 pts)

1. (1 pt) **Run Blender.**
2. (0.5 pt) **Save the rendered image of the default sphere.**
3. (0.5 pt) **Save the rendered image of a colored sphere.**
4. (0.5 pt) **Save the rendered image of the colored sphere with a resolution of 160x90.**
5. (0.5 pt) **Compare the images from Checkpoint 2 and Checkpoint 3. Discuss the effect of changing the resolution.**
6. (0.5 pt) **Save the rendered image of the colored sphere with a different gamma value.**
7. (0.5 pt) **Compare the images from Checkpoint 2 and Checkpoint 5. Discuss the effect of changing the gamma value.**

Quiz Question (1 pt)

The TAs will choose one of the following questions during the grading session. Please be prepared to give a one-minute answer with your partner.

1. How does light interact differently with different objects in real life? Give 3 examples.
2. Why do objects appear to have different colors to our eyes?
3. What's the advantage of using YUV color space?
4. How are colors added differently for lights compared to paint? What does R+G+B equal to in each case?
5. Why do we choose green instead of other colors for chroma keying? Hint: There are a number of practical reasons but one of the reasons comes from how camera sensors interact with green light differently. Think about the arrangement of color filters in front of the camera sensor.
6. Why is tone mapping needed for HDR images?