

HW4 Lighting and Shading

CS 148 Fall 2021-2022

Due Date: Monday, October 18th 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 on Nooks.

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



A wide variety of looks can be achieved through shading and lighting. Source: [Paperclip](#).

This week, we will explore the lights and shading from Blender's raytracing render engine: Cycles. We will implement a portion of these functionalities into your SimpleRT ray tracer next week.

I. Smoothing Geometry



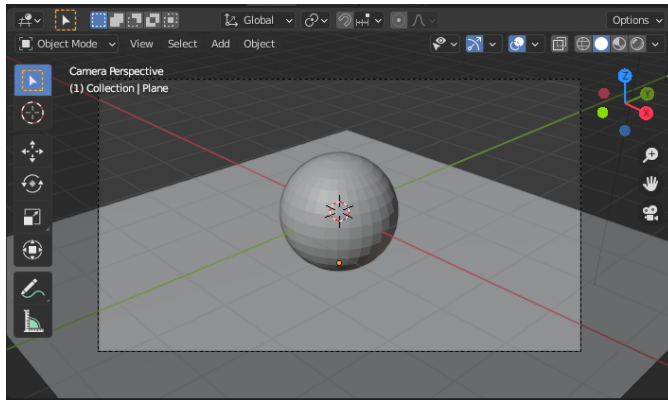
In the lecture, we talked about shading techniques to make the geometry appear smoother without adding more geometry. We can toggle Phong shading for objects in Blender's viewport (scanline renderer), and also in Cycles. We will compare how Phong shading looks, and how it compares with adding actual geometry (subdivide).

I.1. Scene Setup

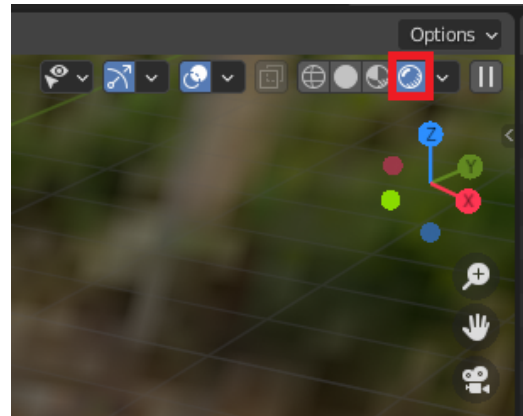
Open Blender, delete the default cube. Keep the camera and the point light.

Add a UV sphere. Add a plane, scale it 5 times (shortcut S, 5), then move it down along the z-axis by 1 (shortcut G, Z, -1).

Go to Properties Editor → Render Properties, and set Render Engine to Cycles. To preview your render, set your viewport shading to Rendered.



Scene setup.



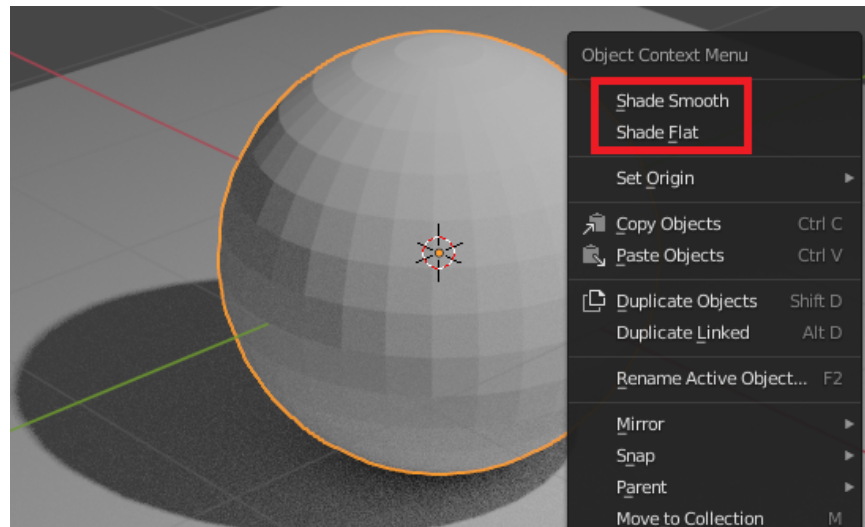
Viewport Shading: Rendered.



For this homework, keep the Resolution at 1920x1080 and Percentage at 50% in the Output Properties.

I.2. Shade Flat vs Shade Smooth

Select the UV sphere. Right-click to choose between Shade Flat and Shade Smooth.



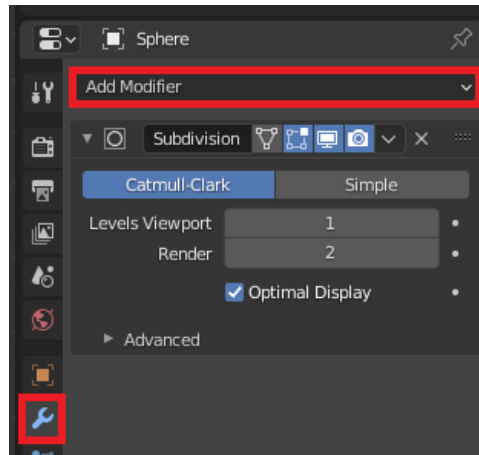
Shade smooth and shade flat.

- ☐ **Checkpoint 1.1: Render the image with shade flat.**
- ☐ **Checkpoint 1.2: Render the image with shade smooth.**
- ☐ **Checkpoint 1.3: Discuss the effect of smooth shading.**

I.3. Subdivision

To add actual geometry to our sphere, we can add subdivision.

Keep the sphere selected, go to Properties Editor → Modifier Properties and add a Subdivision Surface Modifier. (Shortcut Ctrl/Cmd+1, remember to turn off Emulate Numpad in preference setting)



Add subdivision surface modifier.

- ☐ **Checkpoint 1.4: Render the image with subdivision + shade flat.**
- ☐ **Checkpoint 1.5: Render the image with subdivision + shade smooth.**
- ☐ **Checkpoint 1.6: Discuss the effect of subdivision vs smooth shading, and how we can use them in combination to achieve a balance between render quality and scene complexity.**

II. Lights



We talked about point lights and area lights in the lecture. We discussed how they affect the scene and how their shadows look different. We will play with these lights in Cycles, and compare their rendering results.

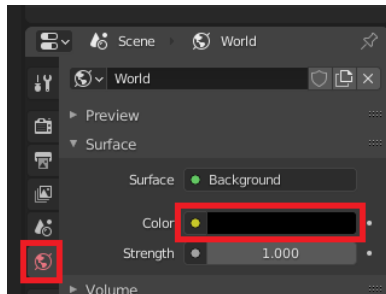
II.1. How Irradiance Change

The irradiance of light depends on its distance to the object, as well as its power. Here we will change the power and position of the light to see their relationship.

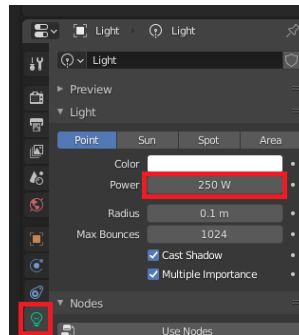
First, turn off the environment light (ambient). In **Properties Editor** → **World Properties**, set **Color** to **black**. Rerender the image for checkpoint 1.5. The background should be completely black.

Then, we will change the power of our light. Select the light in the scene, go to Properties Editor → Object Data Properties, and change the Power to 250 W.

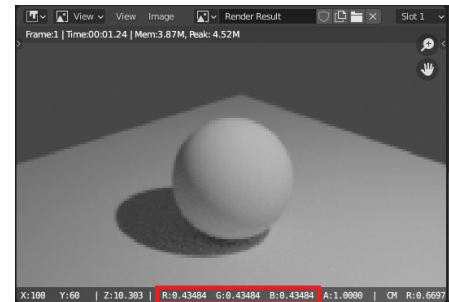
Switch to another slot to render the new image. This way you can then toggle between this render and checkpoint 1.5, and right-click on the render result to view the pixel value on the bottom. This is the raw value before view transform, so it is normal for the values to exceed 1.



Set world background to black.



Set light power.



View pixel value.

☐ Checkpoint 2.1: Render the image with lower light power.

Pick a pixel in the same location from both renders (the X and Y value should be the same), write down their RGB values, and find their relationship.

☐ Checkpoint 2.2: Compare the rerendered checkpoint 1.5 with 2.1. State the relationship between light power and irradiance.

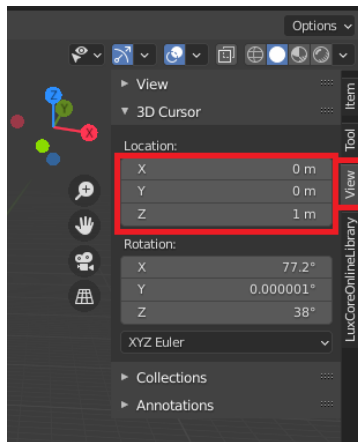
Next, we will change the distance between the object and the light. We do this by scaling the light origin closer to the sphere.

Change the light power back to 1000W.

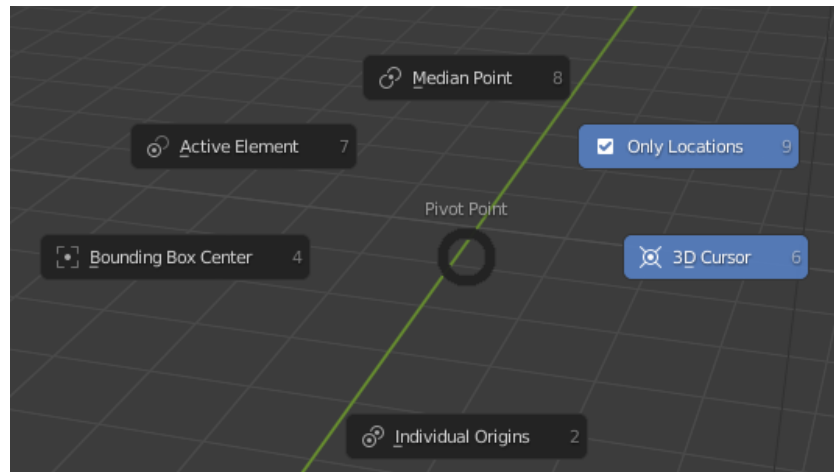
We first set the location of our scaling with the 3D cursor. In our 3D viewport, open the sidebar with N, find the View tab, and set the 3D Cursor location to (0,0,1). You should see the 3D cursor on top of the sphere. This will be our pivot for scaling.

We then set the scale options. Press the period key to reveal the pie menu, and select Only Locations and 3D Cursor. This way we will only transform the location of objects

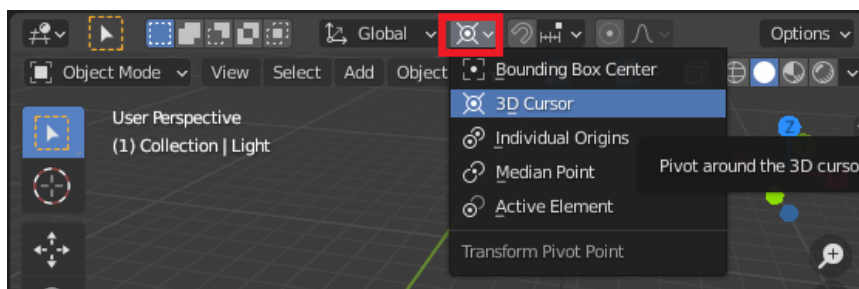
(without transforming the actual object), and use the 3D cursor as the pivot. You can also change these two settings separately from the 3D viewport top bar and Active Tool Settings.



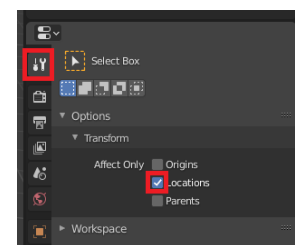
Change 3D cursor location.



Pivot point option.



Top bar — pivot around 3D cursor.



Affect only Locations.

Now, with everything set up, select the light, and use the scale tool to scale by 0.5 (S, 0.5). The light should be half the distance to the top of the sphere now.

Switch to another slot and render the image.

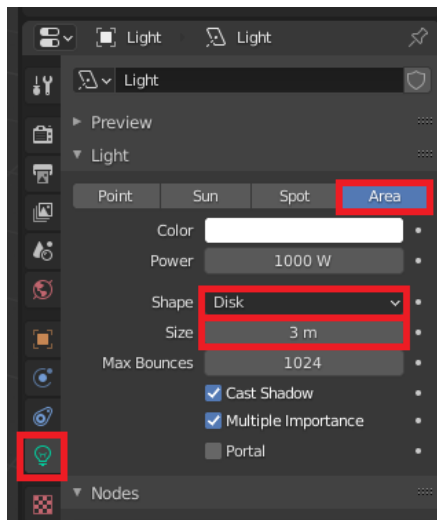
- ☐ **Checkpoint 2.3: Render the image with a closer light.**
- ☐ **Checkpoint 2.4: Compare checkpoint 1.5 with 2.3. State the relationship between light distance and irradiance.** Since the light is only roughly half the distance to the sphere surface, your proof does not need to be exact.

II.2. Light Types

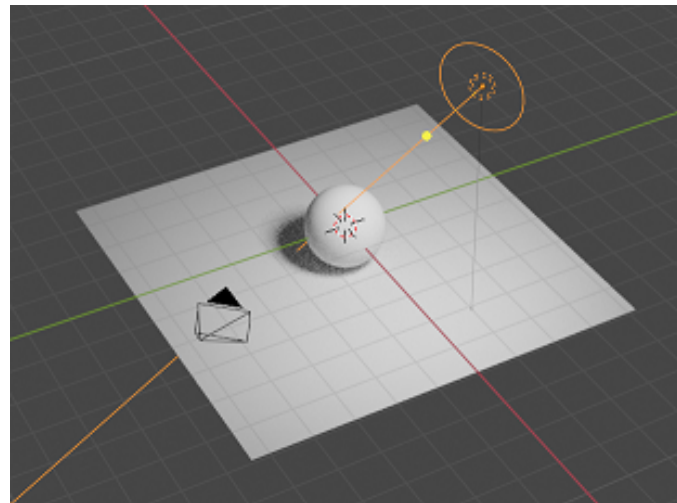
The previous images were rendered with the default point light. Now we will change it to an area light and compare them.

Keep smooth shading and subdivision on for the sphere.

Select the light in the scene. Go to Properties Editor → Object Data Properties. Change its type to Area, Shape to Disk, and Size to 2m. You will see the wireframe of the light from the viewport.



Area light setting.



Viewport preview with area light.

- ☐ **Checkpoint 2.5: Render the image with the area light.**
- ☐ **Checkpoint 2.6: Compare checkpoint 1.5 with 2.5. Discuss how the shadow looks different between the point light and area light.**

III. Environment



A good environment texture is essential to photorealism. In the lecture, we talked about how to measure incoming light and use them to render synthetic objects in the scene. Here, we will show you how to do this in Blender.

There are many types of environment textures inside Blender. You only need to do one of the following for this homework: environment texture or Nishita sky. HDRIs are captures of real-world scenes, and Nishita Sky is a sky model that approximates the sun and the sky.

To see your result, render the image or change your viewport shading to Rendered.

III.1. Environment Texture — HDRI



High Dynamic Range Image, or HDRI, is a type of image frequently used for realistic lighting in 3D environments, usually in EXR format. It is created by combining pictures taken in the same scene with different exposure values, which means it contains much more color and illumination information than normal images. It used to need professional equipment to capture HDRIs, but nowadays you can use 360 cameras to do it yourself.

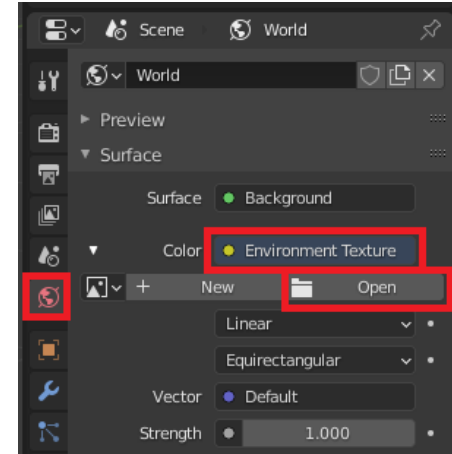


There are free CC0 HDRI libraries online, such as the popular [HDRI Haven](#). You will be able to pick a wide variety of environments from sunny outdoors to indoor studios.

To use HDRI textures in Blender, download an HDRI online (1k resolution is enough for our use) or pick one that comes with Blender (in the installation folder `Blender 2.93/2.93/datafiles/studiolights/world`). They also come from HDRI Haven). Then in the Properties Editor, go to World Properties, change the Background Color to Environment Texture. Then click Open to browse to your HDRI. The settings below should be left as default.



HDRI Haven texture.



HDRI Settings in Blender.

III.2. Nishita Sky



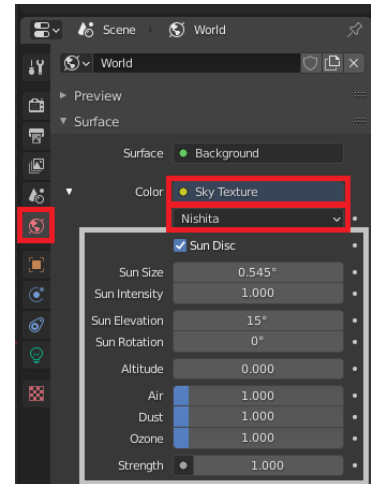
Nishita Sky is a new addition to Blender 2.9. It is an improved version of the sky model proposed by Nishita et al. in 1993. This model lets you control the sky texture with only a few intuitive parameters. Read more about the Nishita sky model here.

In Blender, we can use Nishita Sky by setting the **Color** to **Sky Texture** in World Properties. Play with the settings to get the sky you like.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/635675b1-b0d1-49a7-82fd-dba54da80bff/Reference_Release_Notes_2.90_Cycles_-_Blender_Developer_Wiki_1.mp4

Nishita Sky Texture for Pavilion Barcelona. Source: [Blender Wiki](#).

- **Checkpoint 3: Render the scene from checkpoint 2.5 with world texture (HDRI or Nishita).** Expect to see blurry background if you're using HDRI.



IV. BSDF materials



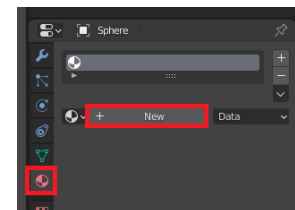
BSDF (bidirectional scattering distribution function) is a generalized version of BRDF and BSDF. It describes all the possible ways light can be scattered by a surface. Most of the materials in Blender have BSDF suffixes, which means they all scatter light, but in different ways.

Select the sphere, go to Material Properties, and add a material. Do the same for the plane. You can expand the Preview panel to see how the material looks on template models.

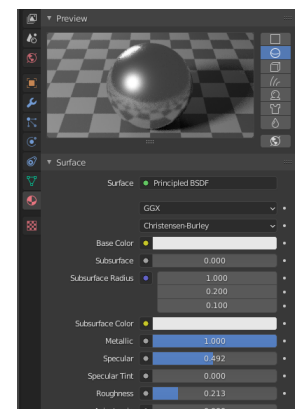
The default material is Principled BSDF, a shader based on the Disney principled model. It can emulate a wide variety of materials in our daily life. You can also switch to another BSDF from Surface.

Learn about the parameters and view some examples for the Principled BSDF here, and other shaders here.

- **Checkpoint 4: Render three images with different BSDF settings for the sphere and the plane (6 different**



Add material.



Principled BSDF.

materials in total). Discuss what shader you used, what parameters you used, and how they affect the light scattering function. You can keep the scene from checkpoint 3.



We will discuss how to use the [Shader Editor](#) to make complex materials in HW8. If you'd like to try it out now, you can watch the official tutorials [here](#) and [here](#), or Blender Guru's explanation on Principled BSDF [here](#).



You can also write your own shader code in [Open Shading Language \(OSL\)](#) and plug them into your material node tree. In fact, Cycles BSDF materials are also written in OSL. [Shadertoy](#) is a good platform to see the capability of shaders. Its founder Inigo Quilez's [website](#) has some nice articles explaining shaders and related concepts.

V. Grading (5 pts total)

This assignment will be graded on the following requirements

Complete all the checkpoints (4 pts)

1. (0.5 pt) **Render images with shade flat (1.1) and shade smooth (1.2). Discuss the effect of smooth shading. (1.3)**
2. (0.5 pt) **Render images with subdivision + shade flat (1.4), subdivision + shade smooth (1.5). Discuss the effect of subdivision vs smooth shading, and how we can use them in combination to achieve a balance between render quality and scene complexity. (1.6)**
3. (0.5 pt) **Render the image with lower light power (2.1). Compare with rerendered 1.5 and state the relationship between light power and irradiance. (2.2)**
4. (0.5 pt) **Render the image with a closer light (2.3). Compare with 1.5 and state the relationship between light distance and irradiance. (2.4)**

5. (0.5 pt) **Render the image with area light (2.5). Compare with 1.5. Discuss how the shadow looks different for point light and area light. (2.6)**
6. (0.5 pt) **Render the scene from checkpoint 2.5 with world texture (HDRI or Nishita). (3.1)**
7. (1 pt) **Render three images with different BSDF settings for the sphere and the plane (6 different materials in total). Discuss what shader you used, what parameters you used, and how they affect the light scattering function.**

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 do we interpolate a quantity at a point inside of a triangle given the quantities at each of the three vertices? Hint: for instance, suppose we have different color values at each triangle vertex; how then should we weigh the colors at each of these vertices to interpolate the color at e.g. the center of the triangle?
2. What's the difference between radiant intensity, irradiance, and radiance?
3. What is the difference between BRDF, BTDF, and BSSRDF models? Give a visual description of each.
4. Discuss the difference between Gouraud shading and Phong shading, and explain why Phong shading can render more realistic results than Gouraud shading.
5. What is smooth shading? Why does it look “smoother” than flat shading?
6. How do shading techniques change when you shade corners?
7. What is ambient lighting and how is ambient color computed?
8. Describe the difference between point light and area light. Give real-world examples of each.