

CS148 Homework 8 - Texturing

Grading on Monday, Dec 1 – see the pinned posts on Ed for details!

0.1 Assignment Outline

The goal of this homework is to help you start picking textures for your final project. Like with HW7, it is highly recommended that you use this assignment as a milestone towards your final project, since you will be able to get some preliminary feedback from the teaching team during grading. **The actual assignment is detailed within the first few pages of this document. The rest of this PDF contains tutorials and resources for finding and creating your own textures.**

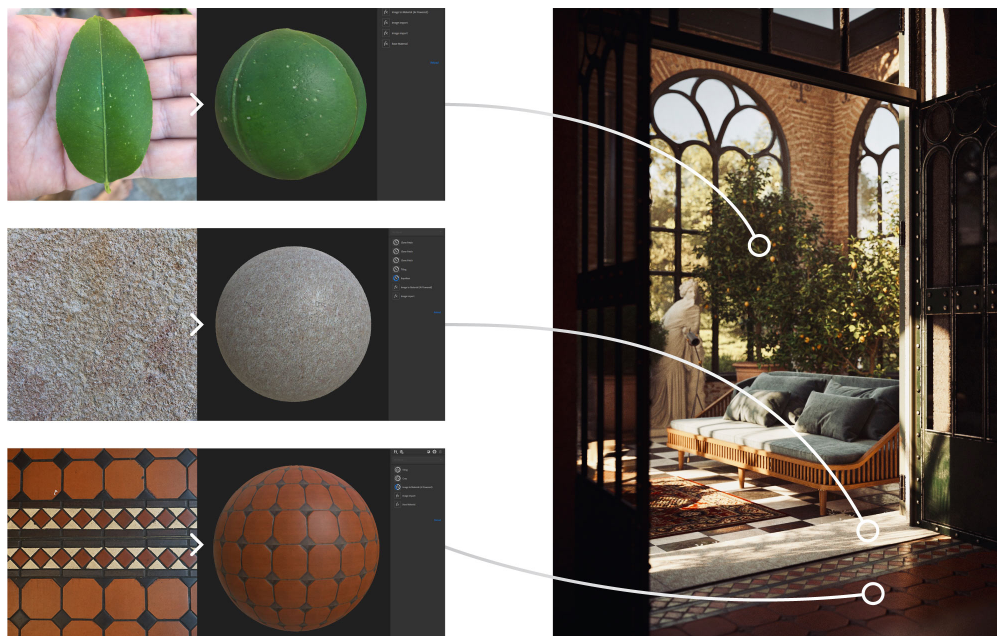


Figure 1: Image texture applied to objects. This particular scene uses AI to create albedo, normal and height maps from a single image. Source: [Substance3D](#), a commercial software.

0.2 Collaboration Policy

All policies from here on are the same as they were for HW2. See the HW2 document for details.

0.3 AI Policy

You may use AI tools to generate HDRI images and texture images to place in your scenes and apply to objects for this HW. You may use AI tools to generate geometry for this HW to supplement the geometry you create yourself. You may NOT use AI tools to generate anything else for this HW, especially your rendered images. All rendered images must be the final ray traced result from Blender Cycles.

Quiz Questions (1 pt)

You will be randomly asked one of these questions during the grading session:

- What are texture (u, v) coordinates and how are they mapped to the points on the surface of an object mesh? Given a texture coordinate (u, v) , how do we determine what color to apply from the texture image to the corresponding point on the object?
- How do we do texture mapping for points on the surface of an object mesh that are not explicitly the corners of the triangles (or polygons) that make up the mesh? Why must we be careful of the specific coordinate space in which we do texture mapping?
- What is the difference between a MIP map vs a RIP map? What is a common texture pattern that is often used for debugging purposes to check whether your texture is being mapped appropriately to an object?
- What is texture tiling? How does patch-based texture synthesis achieve this effect?
- Why is adding noise necessary for procedurally generating realistic looking textures? Explain at the high level how Perlin Noise makes natural looking noise more effectively than a pure random number generator.

1 Assignment

1.1 UV unwrap two objects.



Figure 2: “Unwrapping” a Christmas chocolate. Source: [Twitter](#).

In lecture, we discussed UV mapping (for a mesh) as the process of assigning UV coordinates to each vertex of the mesh. The UV coordinates are then used to sample the texture for the mesh.

UV coordinates do not magically appear when we need them though — we have to actually start somewhere and create the UV coordinates from scratch. To get started with learning the process, we suggest watching [this official Blender tutorial on UV unwrapping](#) first, then read through the instructional material on UV unwrapping later in this document.

Show us: (1 pt) Two objects (or two different parts of a complex, multi-part object), each with their own UV maps (0.5 pt each). You will be required to explain how these two objects will contribute toward to your final project and how texturing them will be relevant. If you cannot explain this, then you will receive no credit! You can show everything via the render preview if you want (remember to be working in Blender Cycles for ray tracing!).

1.2 Recreate two reference textures.

As discussed in lecture, texturing is a widely adopted method to add fine-grained detail to an object. There are different kinds of textures for different purposes: base color, normal, roughness, ambient occlusion, displacement, etc. **For this HW, we want you to first find reference photos or images of two objects that have textures that you find interesting (e.g. maybe the marble patterns of a fancy sink, or the bumpy rock surfaces of a mountain range). Then, try to recreate the looks in Blender using a combination of texture mapping (e.g. base color, normal, etc).** Ideally, you use this as an opportunity to texture at least one object that you'll want to include for your project – we can actually give you feedback on it during grading!

Show us: (2 pt) Your own recreation of two textured objects based on a reference photo or image (1 pt each). Alternatively, if you have a complex, multi-part object, then each textured part of it can be considered a separate object. Have your references ready, and be ready to explain what types of texture mapping you used. You can show everything via the render preview if you want.

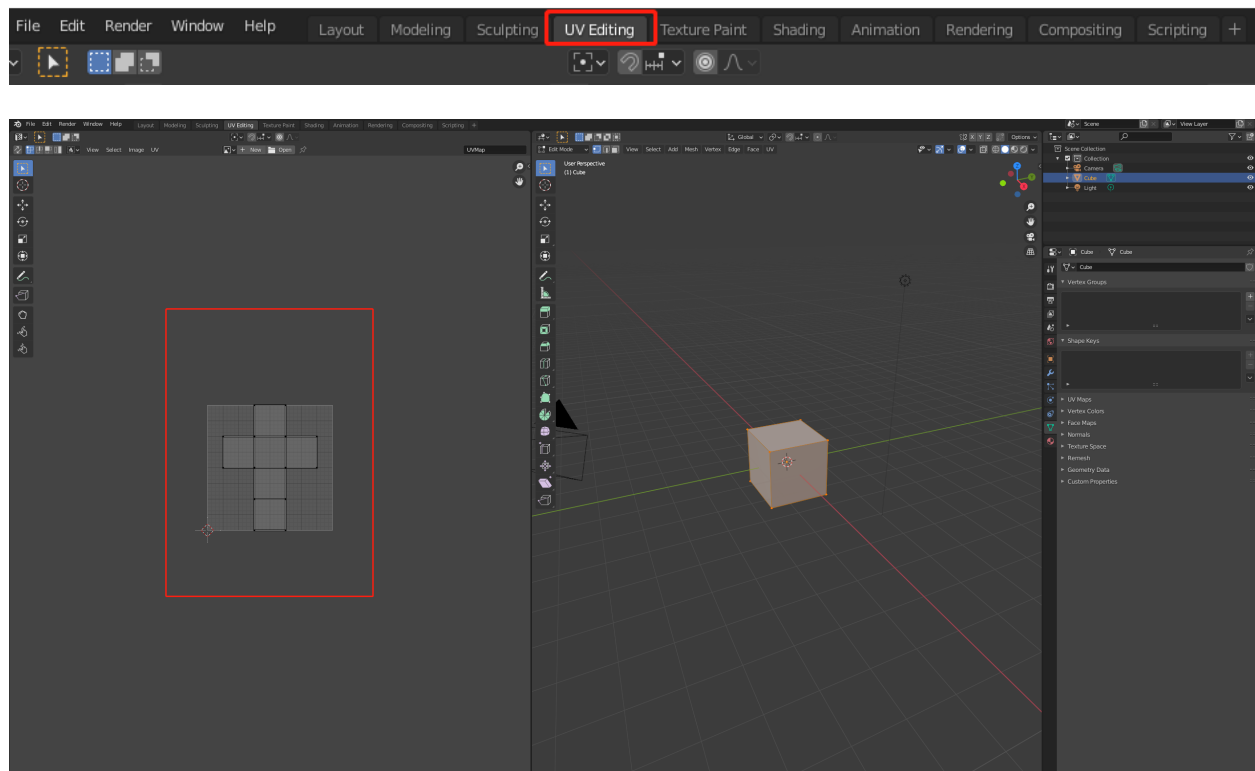
1.3 Prototype your final project.

Show us: (1 pt) A prototype of your final project scene with any geometry, texturing, and lighting you have so far, plus a quick explanation of your plan for the remaining week and half. The prototype can be shown via the render preview. It doesn't have to look good (yet) as long as you can convince us that you have a plan!

2 UV Unwrapping

2.1 Visualizing a UV Map

The first step to UV mapping is do a process called UV unwrapping (of an object mesh) to generate a UV map. Let's visualize what a UV map looks like in Blender with a simple example. Open a new Blender project and select the default cube. Then, toggle to the UV map editor view by clicking on **UV Editing**. You should be able to enter **Edit Mode** and see the UV map of the default cube object on the left side of the screen.

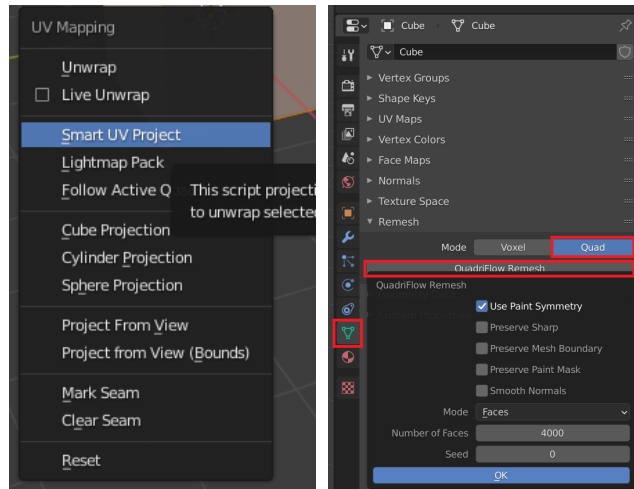


Notice how the cube “unwraps” its surface area into 2D.

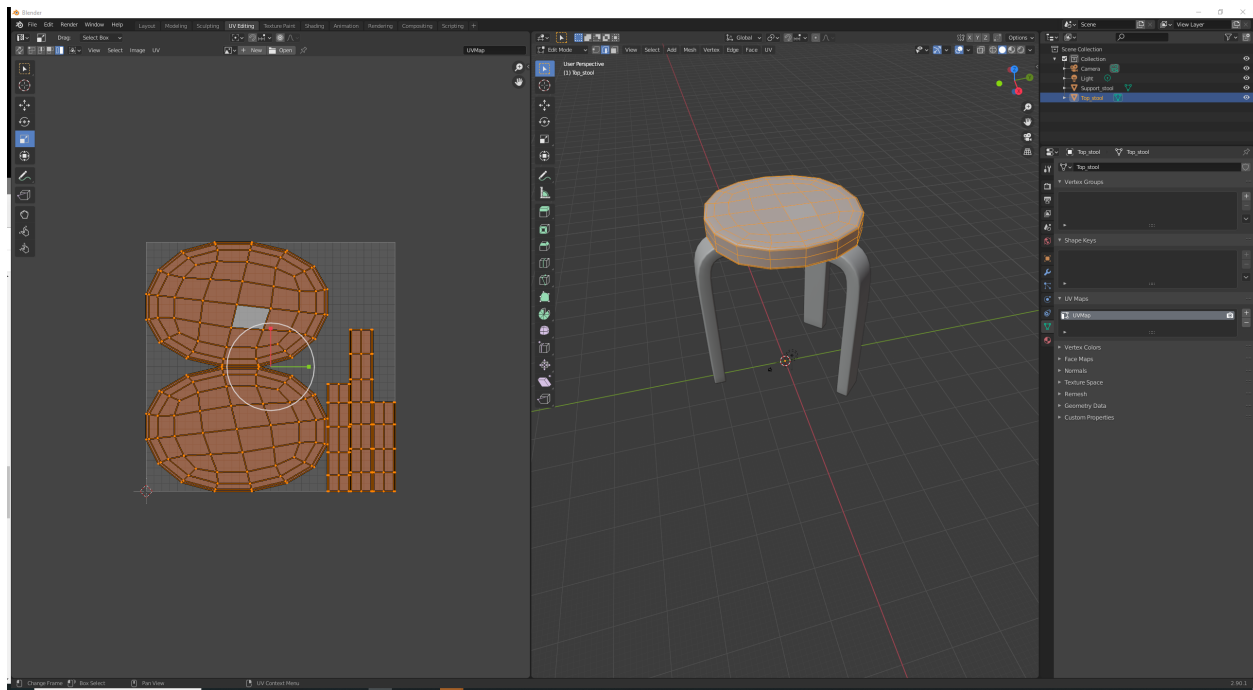
2.2 Generating your own UV Map

Now let's generate a UV map for your own object. Open your Blender project from HW7 and select the object you created yourself. For demonstration purposes, we will use a simple stool object for the example in this HW. Toggle to the UV map editor view, select all the vertices of your object, and press **U**, then click on **Smart UV Project**. You will only UV unwrap what you've selected. If you accidentally deselected an object in **Edit Mode**, then you can press **A** with the 3D Viewport in focus to select all its vertices.

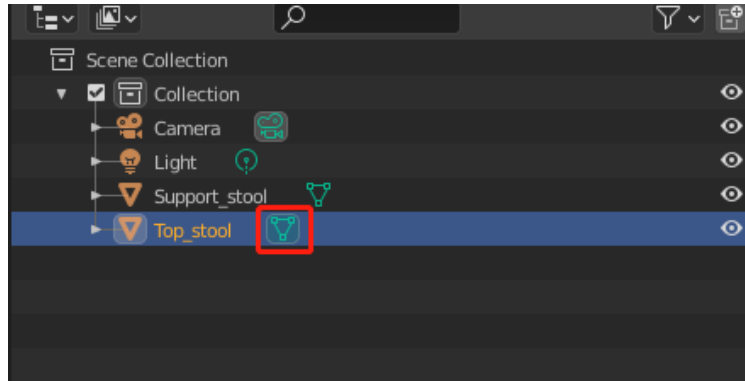
Note: If your object does not have a clean topology (especially if you sculpted it with dynamic topology), then you may need to tidy up the mesh before unwrapping using retopology tools such as [quadriflow](#) from **Object Data Properties** → **Remesh** → **Quad** → **Quadriflow Remesh** in the Properties Editor.



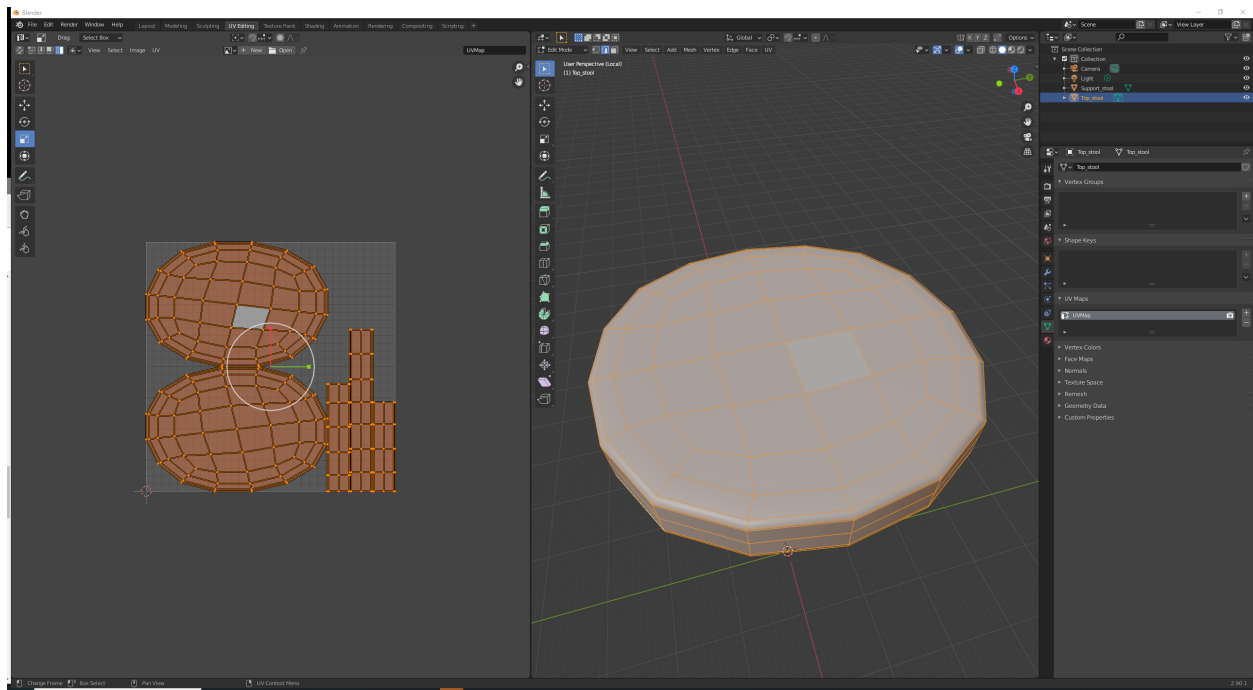
You should be able to see an auto-generated UV map for your object, like so:



Note: You will likely UV unwrap multiple objects in the same .blend file as you work on any project. If you want to quickly switch between them when editing, then you can click on the mesh icon next to your target object in the outliner:



On the flipside, you can also use “local view” to hide non-selected objects so that they don’t get in the way of your editing. Select an object and press `/` (the forward slash on the keyboard) or go to **View → Local View → Toggle Local View** to toggle to local view. In our case, we did this for the top part of the stool, thus hiding the leg parts from view.



2.3 Editing a UV Map

Sometimes the auto-generated UV map of a complex object will be far from perfect. Each connected part of the unwrapped mesh is called a “UV Island”. A good UV map should maximize its area covered by the islands with small distortion (i.e. little stretching) and a reasonable amount of empty space between the islands. There are ways to improve a UV map though if it is not what we want. We’ll explore two such techniques here. For a full list of UV editing tools in Blender, you can check [the official manual here](#).

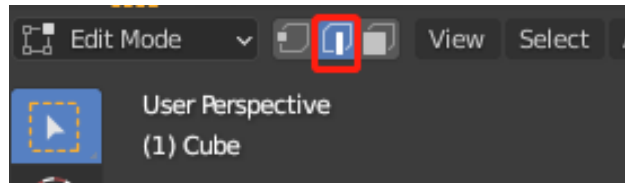
2.3.1 Adding UV Seams

Just like in sewing, a seam is where the ends of the image (cloth for sewing) are sewn together. In UV unwrapping, the mesh is unwrapped at the seams. Think of this method as peeling an orange.

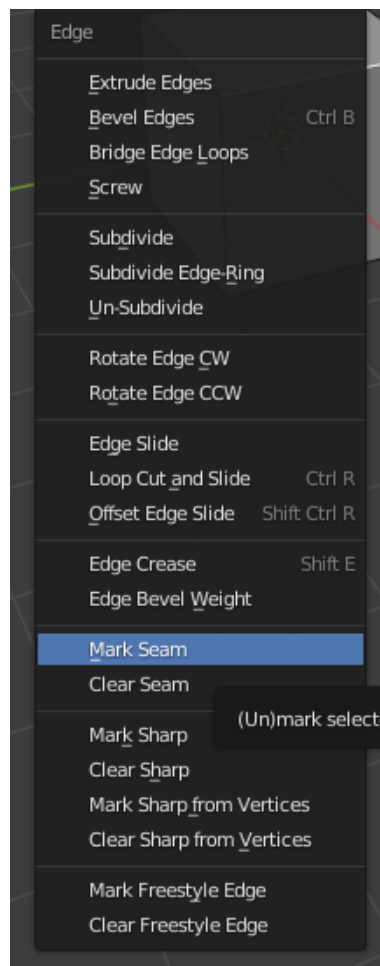
You make a series of cuts in the skin, then peel it off. You could then flatten it out, applying some amount of stretching. The cuts are the same as seams.

Smart UV Project creates seams automatically, but we can specify the seams to have finer control over the unwrapping process. Generally, artists place seams at sharp edges, or places where non-overlapping textures are less noticeable (e.g. the back or the bottom of an object). Faces with edges not marked as seams will strictly adhere to each other. Sometimes we might want to add seams to separate a flat surface so the mesh could be better unwrapped. You can read more about seams in Blender in the [official manual here](#).

Let's try adding a seam to your object. In the 3D Viewport, select edge selection mode via the following icon:



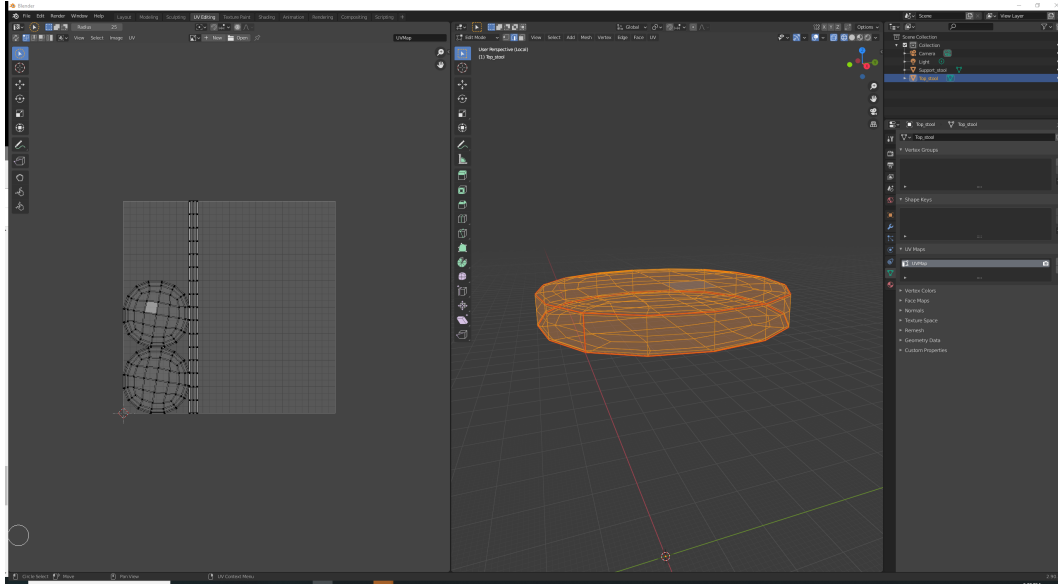
Hold shift, then start clicking to select all the edges that you want to mark as seams. When you're done selecting, press **Ctrl/Cmd + E** for the **Edge** menu and click **Mark Seam**. Seams will be marked in red in the 3D Viewport.



You may also want to toggle to the wireframe view to better inspect the object's edges:



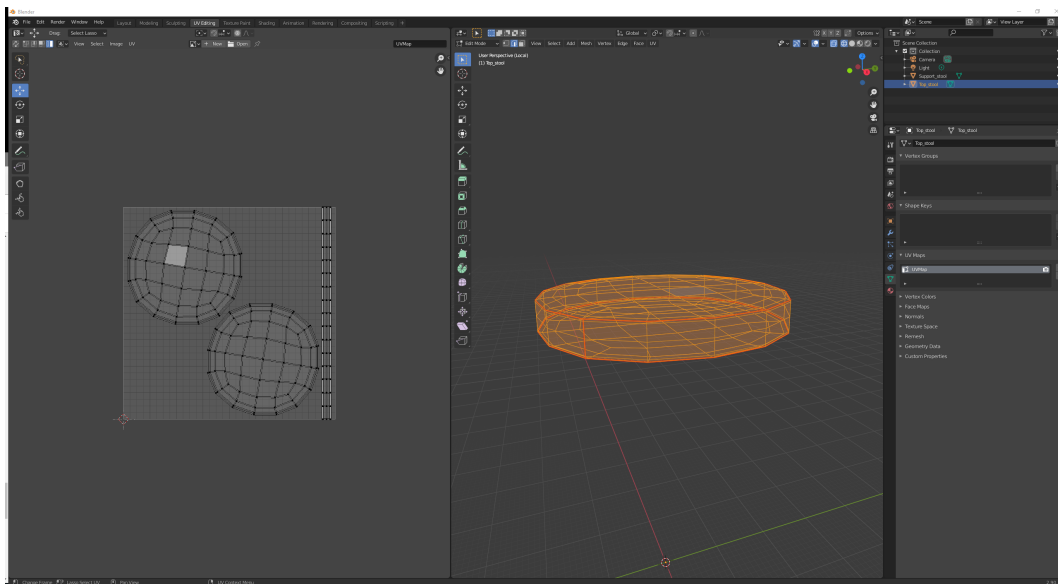
After adding seams to an object, press **U** and select **Unwrap**. You should be able to see a new UV map with the seams you specified.



2.3.2 Directly Editing the UV Map

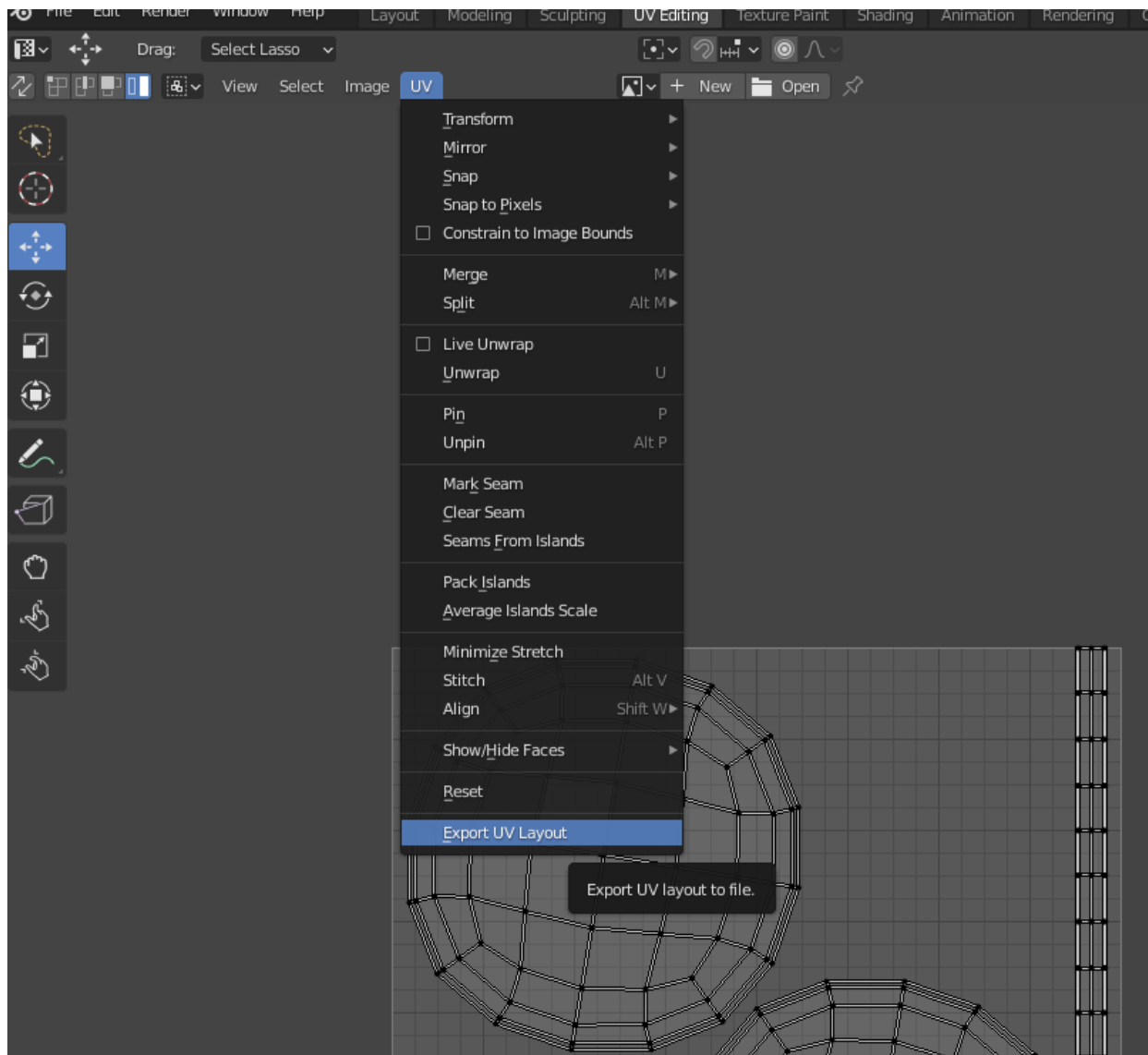
We can also directly edit the UV map. There are four different selection modes on the top left of the UV editor: **Vertex**, **Edge**, **Face**, **Island** modes. You can try whatever is suitable in your case and move, rotate, and scale the selected target. For instance, if you are in vertex mode, then you can move the selected vertices in the UV editor.

We show a better UV map of the stool top below after editing. A larger area of the square is now covered by the unwrapped meshes. This will allow more pixels on the texture to be mapped onto the object, giving us better detail.



2.3.3 Saving the UV Map

You can save your edited UV map through the **UV → Export UV Layout** menu, so that you can create textures with it later.



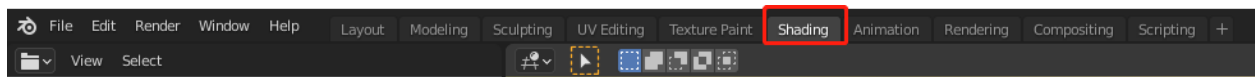
3 Texturing

For demonstration purposes, we will apply a rock ground texture to a sphere. You can download these example textures (e.g. 2k resolution diffuse maps, normal maps, and displacement maps) from [this popular website](#). If you feel comfortable making your own textures, then you can also do that and apply them to your models.

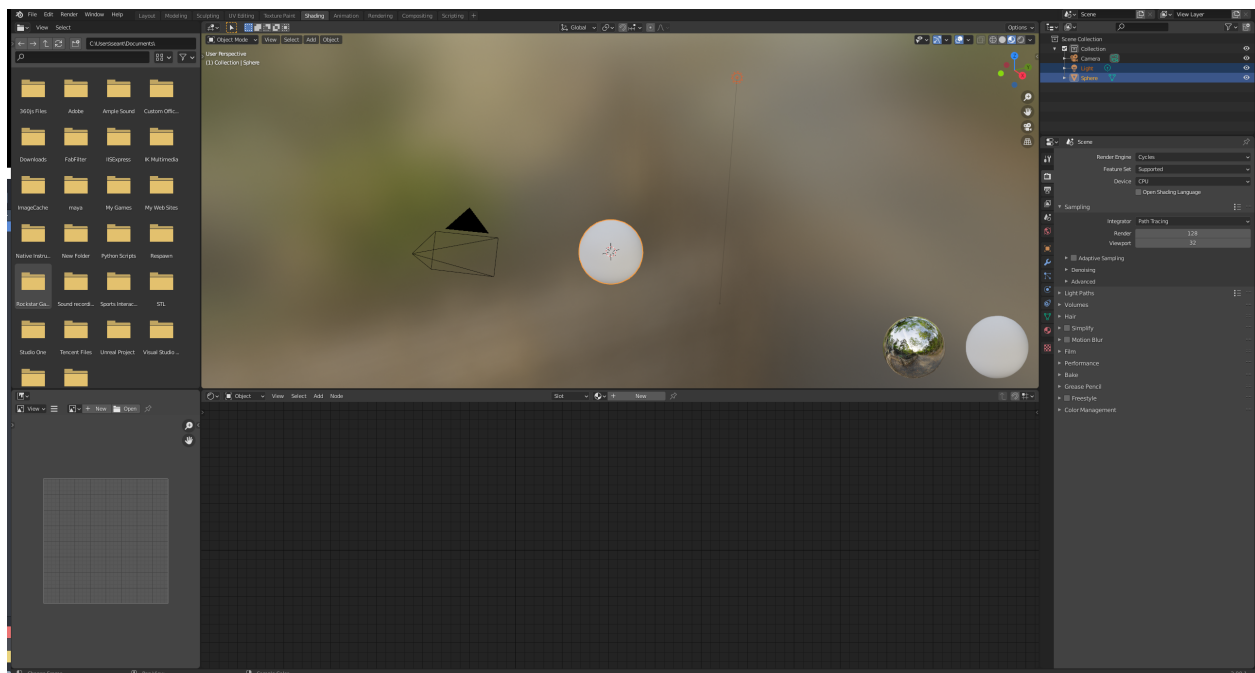
Note: Texturing an object requires that the object has an associated UV map. So make sure to complete the steps in the last section first on UV Unwrapping.

3.1 Base Color

Open a new Blender project, replace the default cube with a UV sphere, and enable smooth shading as well as Blender Cycles for ray tracing. Now open the Shading Editor via the **Shading** tab:

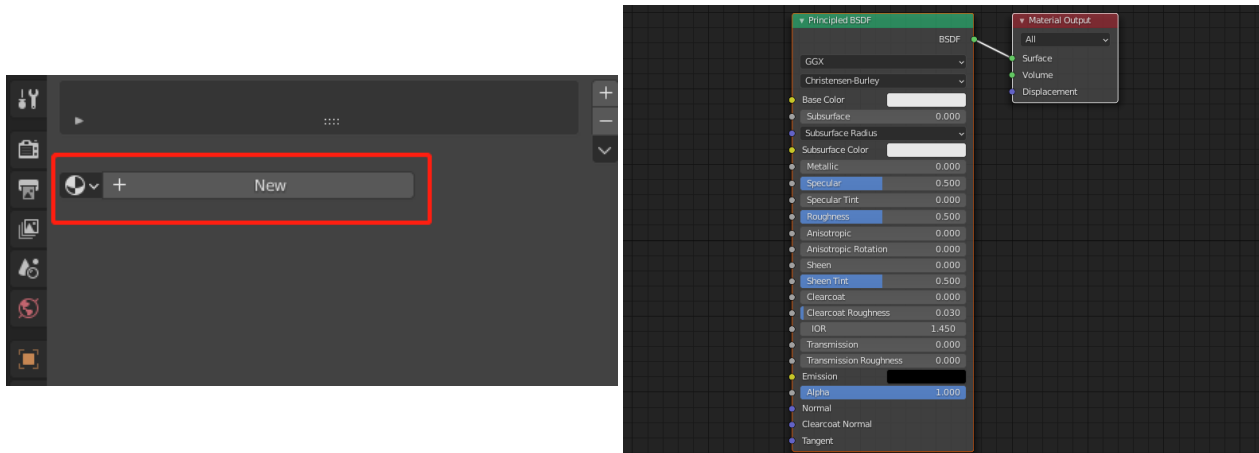


It may take a few seconds to load, but you should see an interface like the following:

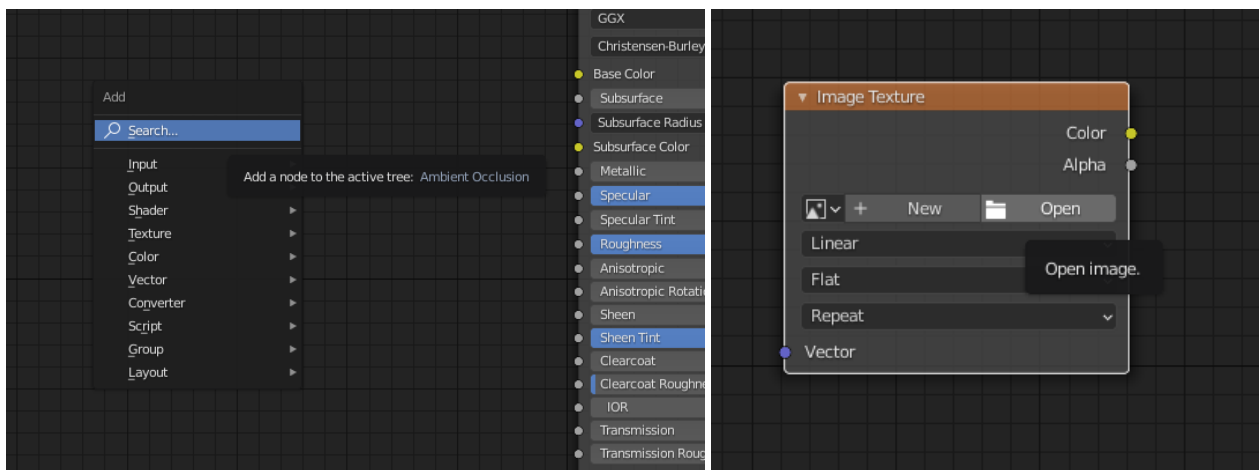


The Shader Editor is used to edit materials, which includes textures. We played a bit with it in HW2 and also in HW4 when we worked with BSDF materials in the Properties Editor as a list of parameters. Under the hood, the BSDF materials are actually handled via a tree structure in the **Shading** tab. In the Shader Editor, you can create and edit the node tree more easily. You can read more about it in the [official Blender manual](#).

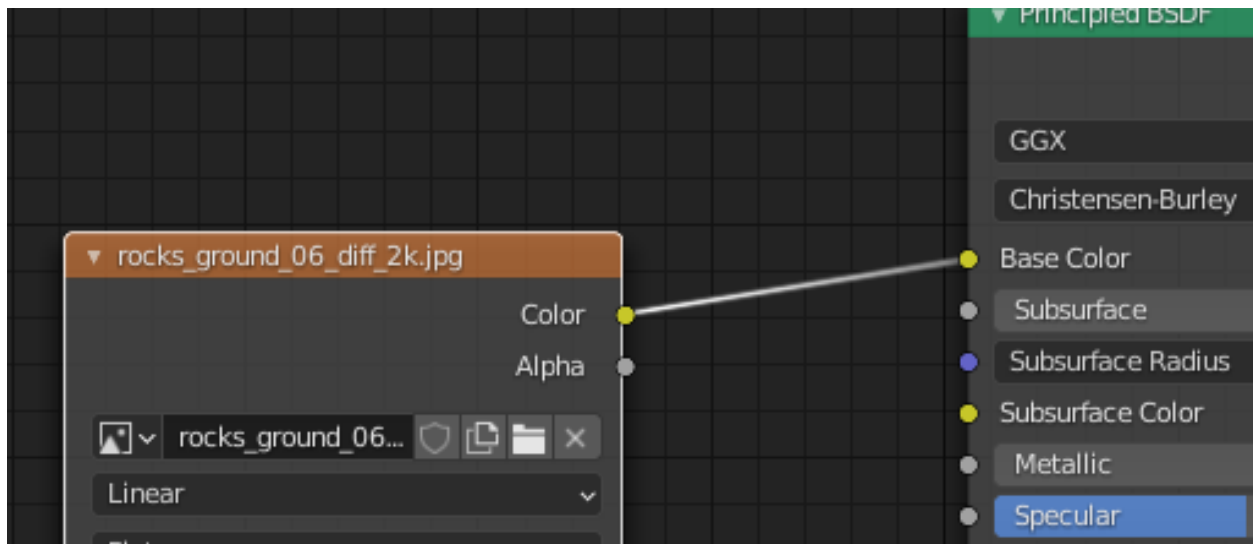
Add a new material for the sphere, and a new material node will appear in the Node Editor:



Go to the menu bar of the Shader Editor or press **Shift + A** to open the **Add** menu. Search for “image texture node” or go to **Texture → Image Texture** to get an image texture node. Click anywhere to add it to the graph.



Click **Open** and select the downloaded base color texture **rocks_ground_06_diff.2k.jpg**. Then, connect the image texture’s **Color** channel to the material’s **Base Color**. You can also directly drag an image from your file explorer or finder window into the node editor to create an image texture node.



Now if you toggle the 3D viewport to the render preview, then you should see a textured sphere as follows. Move around in the viewport (e.g. with your mouse) to see what the sphere looks like. You may notice a seam on the sphere, which is from the UV map. If you're interested, then you can also modify the UV map and see how the texture changes accordingly.



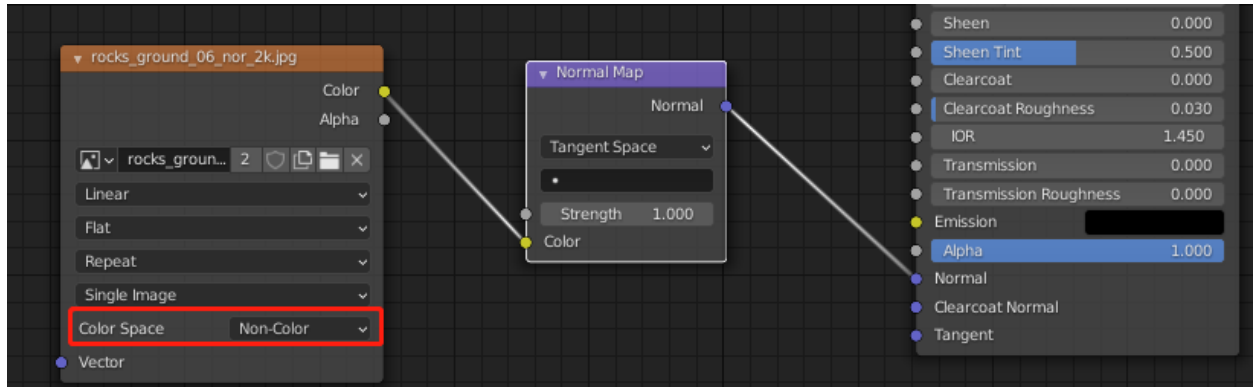
3.2 Normal Map

Normal mapping is a texture mapping technique used for faking the lighting of bumps and dents. Keep in mind that normal mapping will only change the shading of the object (i.e. how light interacts with the normals of the object), not the actual geometry of the object.

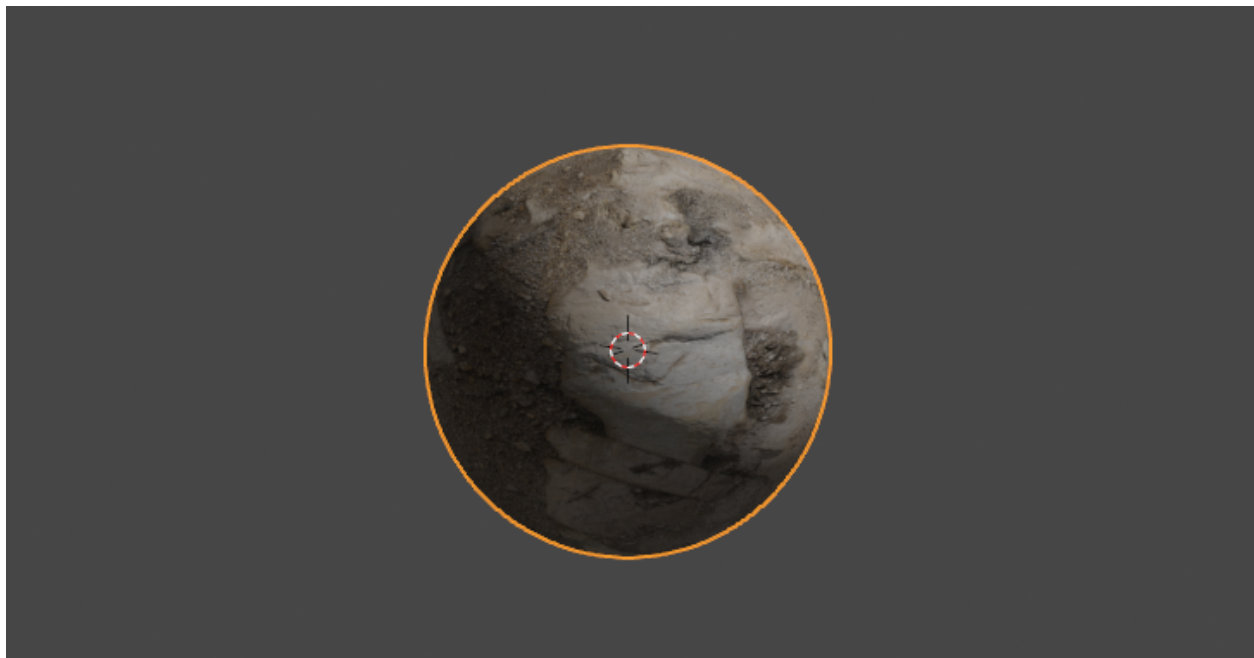
Working off the textured sphere from the last subsection, add a new image texture node, then a normal map node (**Add** → **Vector** → **Normal Map**) to the node tree. Select **rocks_ground_06_nor_2k.jpg** for the image texture node. Since the pixel value of a normal map does not represent actual colors,

but rather it stores the XYZ values of normal vectors (in the RGB channels of the image), we should set the **Color Space** of the image texture node to **Non-Color**.

Connect the **Color** channel from the image texture node to the normal map node, and the **Normal** channel from the normal map node to the Principled BSDF node like so:



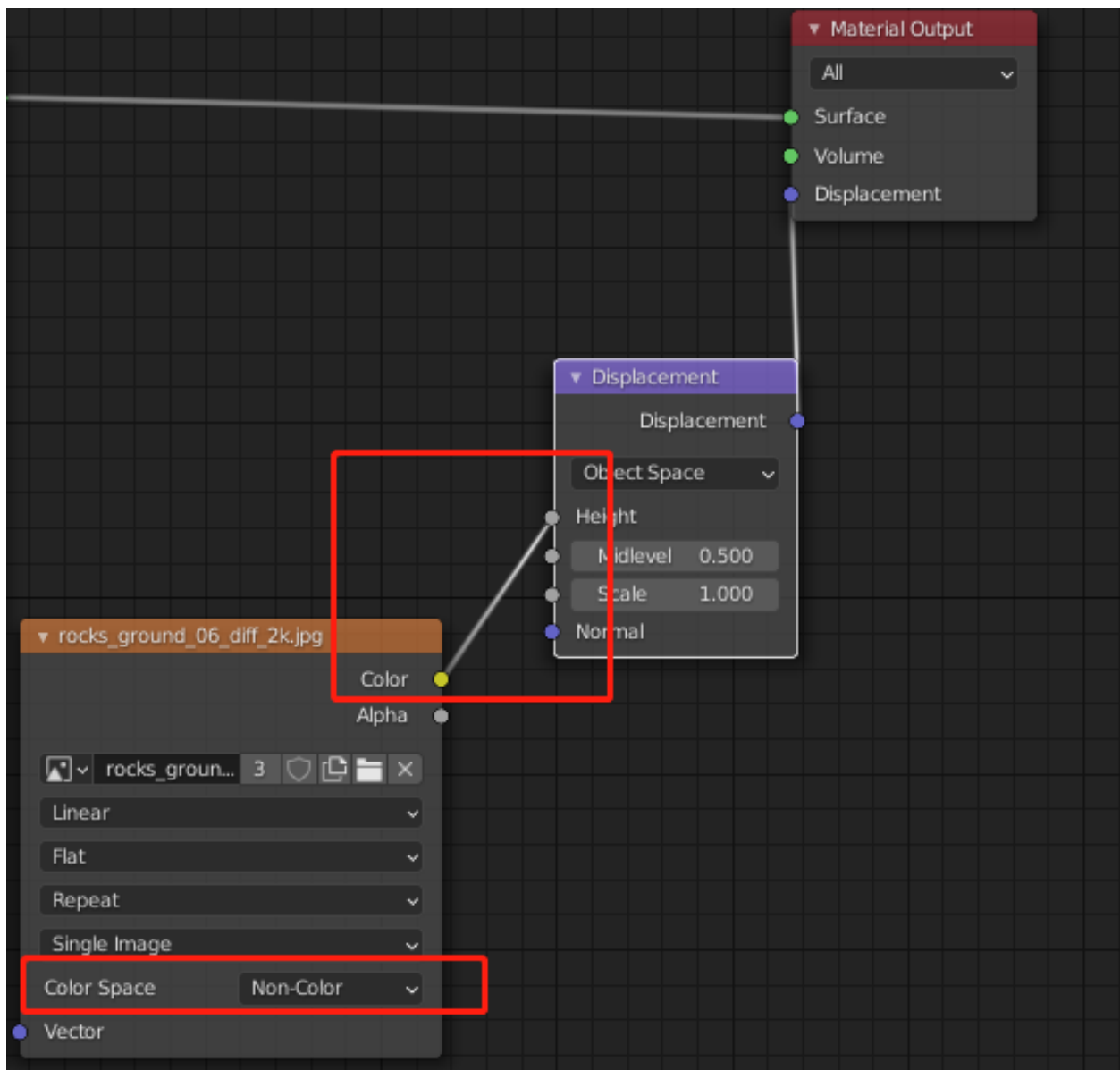
Your rocky sphere should now have a sense of depth to it:



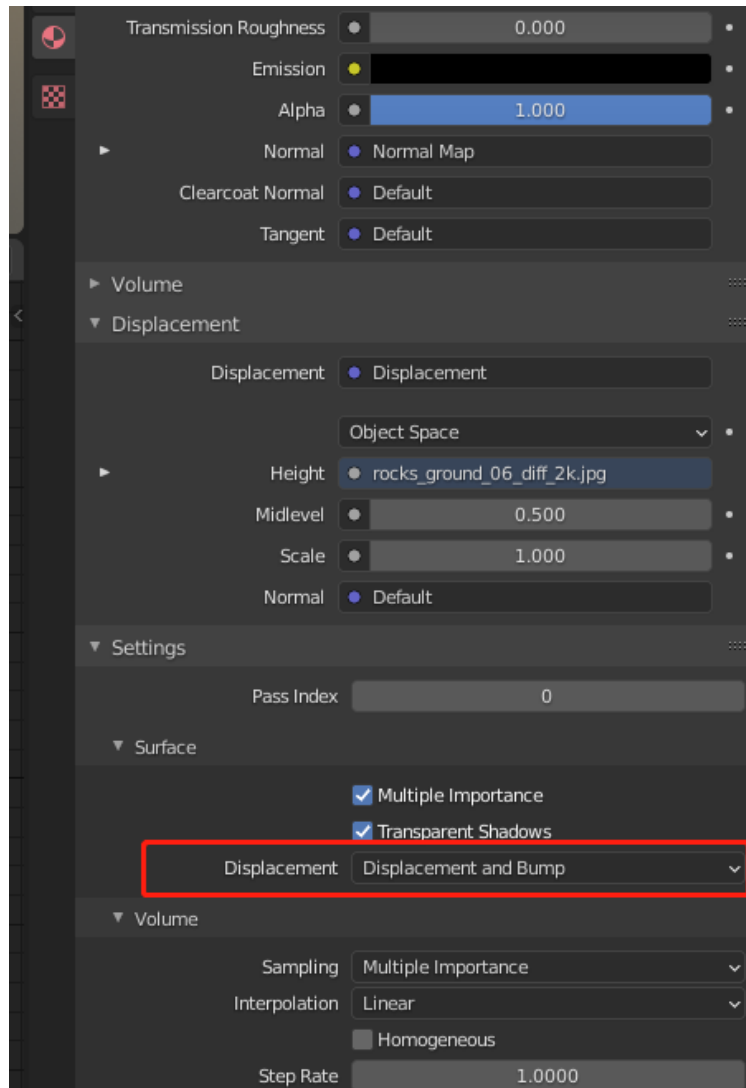
3.3 Displacement Map

Displacement mapping is an alternative texturing technique that uses a texture map to modify the actual geometry of the object in addition to its normals. Any vertex that gets covered by the displacement map texture actually has its position (and local surface normal) modified or displaced (hence “displacement” mapping) based on the values stored in the texture.

Working off the previous subsection, add another image texture node along with a displacement node (**Add** → **Vector** → **Displacement**). Select **rocks_ground_06_disp_2k.jpg** as the image for the image texture node. Set the **Color Space** to **Non-Color** as before. Connect the **Color** channel of the image texture node to the **Height** channel of the displacement node, then the output of displacement node to the **Displacement** channel of the **Material Output** node (not the Principled BSDF node!).

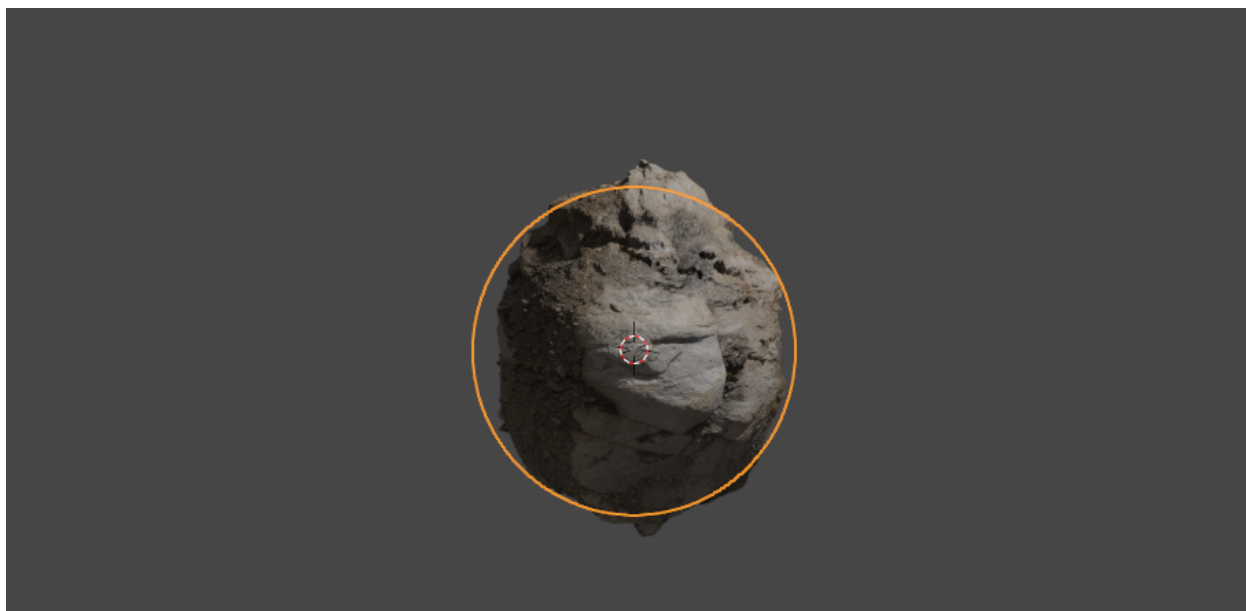


Then in **Material Properties**, set **Displacement** to **Displacement and Bump**.



Since displacement mapping perturbs the actual mesh vertices, and the default UV sphere has a low resolution, we need to add a subdivision modifier. As a reminder from HW7, go to the Properties Editor, then **Modifier Properties** and add a **Subdivision Surface** modifier. Set the **Levels Viewport** to 3.

You should see that the sphere looks very bumpy now due to the displacement map being applied to the subdivided surface. You can change the **Scale** value of the displacement node to fine-tune the amount of displacement. With a bit of scaling, you can get an image like the following.



4 Texture Resources

As you might have noticed, to get realistic materials, you need to apply multiple types of texture maps. Base color aka diffuse maps can often just be any high resolution image that you find online, but getting the other maps (like normal and displacement maps) can be tricky. Here are some websites that provide free textures sets:

- [TextureHaven](#)
- [Poliigon](#)
- [CC0 Textures](#)
- [3D Textures](#)
- [texture.ninja](#)

If you don't find anything that fits what you're looking for, then you can also create your own maps from a diffuse map. You can use image editing software such as Photoshop, Gimp, or Krita to create the other maps. There are also dedicated tools, such as [CrazyBump](#), [ShaderMap](#), and [Materialize](#). There is also this nice [web app](#) as well as a way to create these maps in [Blender itself](#).

Some other topics:

- If you want to start from scratch, then you might find useful [this tutorial](#) on how to make all the maps from a photograph.
- For tiling textures without repetition in Blender, there's a handy node group created by Blender Guru, explained in [this video](#).
- If you want to explore advanced texturing, then you can try out the industry-standard texturing software [Substance Painter](#), which has a free educational license.

Here's a comprehensive list of all the Blender-related resources: [awesome-blender](#).