

# Make Your Own Scene with Texture Mapping

20150860 김태윤

## 1 Texture mapping with external texture images

### 1.1 Texture mapping of objects

Texture mapping was applied to two objects in the scene : the Earth and the Moon. The textures used are shown on the right. Because of an unknown issue in the texture coordinates, the images had to be flipped to create a correct mapping.

### 1.2 Skybox mapping

Skybox mapping was done using 6 images of the Milky Way cropped out from a panoramic image of the Milky Way. The skybox images of the skeleton code was simply replaced with the Milky Way images.

## 2 Normal Mapping

Normal mapping was achieved using normal map images and shaders that utilizes normal values obtained from the images. Normal map images are shown on the right. The normal map images are imported using the texture loaders. Then the normal map texture was sent to the fragment shader as sampler2D values. In the fragment, the normal values corresponding to the fragment position was used as the normal vector for intensity calculation. To use the normal value correctly, the TBN matrix was used to transform the value to the tangent space, as normal mapping should

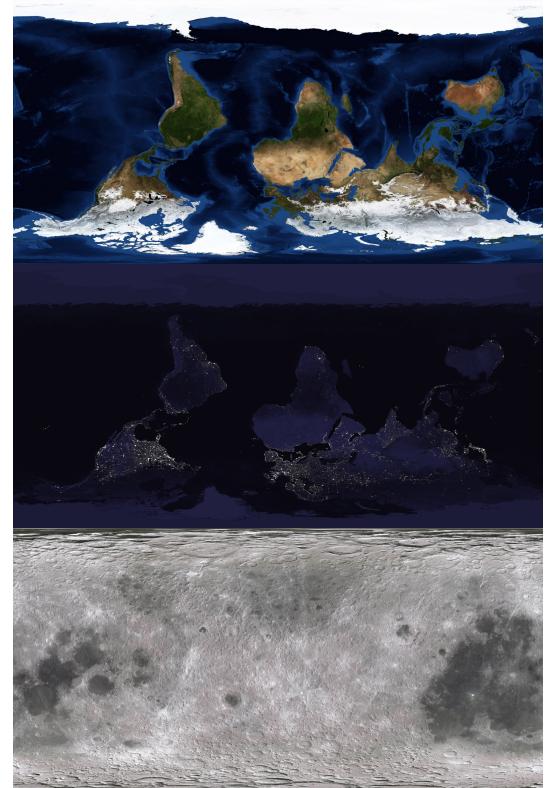


Figure 1 Object Textures

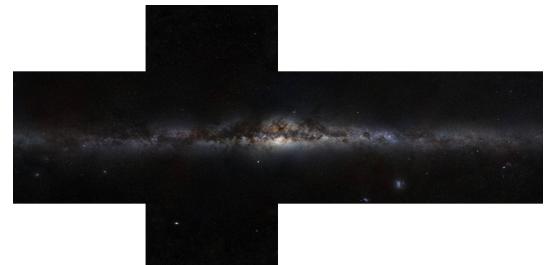


Figure 2 Milky Way

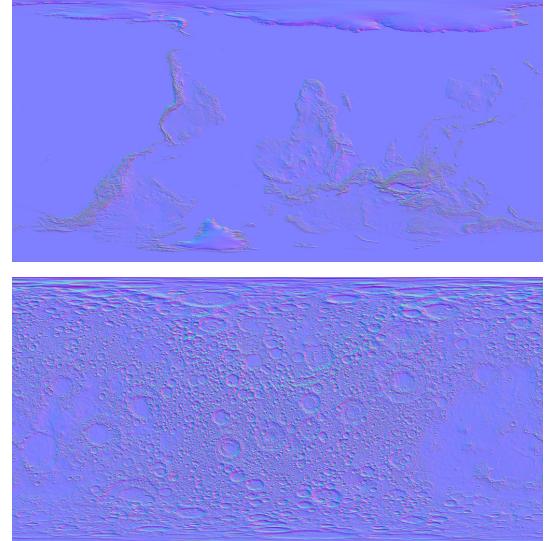
be performed in the tangent space. This resulted in a bumpy effect on the otherwise smooth surfaces of the Earth and the Moon.

### 3 Shadow Effect

The shadow effect was achieved first by making the depth map, which is the depth information as seen by the light. This was done by the `ShadowMaterial` and the shadow shaders after the `shadowInitialize` function was called. The information in the depth map was used to determine if a fragment was in shadow or not. If it was found to be in the shadow, corresponding fragment was colored with darker intensity. For the comparison, the depth obtained from the depth map for a fragment was compared with the depth of that fragment from the light. If the fragment was closer to light than the value stored in the depth map, that fragment is not in the shadow. To perform this calculation, we should know the position of the fragment in the light space. This was calculated using a matrix called `lightSpaceMatrix` to convert fragment position to light space. The matrix was obtained using the light's position and the lookat direction. The calculation was sent to the `NormalVertexShader` to convert the fragment position to light space. So to generate shadows two kinds of shaders were used: shadow shaders to create depth map and normal shaders to apply shadowed colors. So the objects were first rendered with `ShadowMaterial` for generating depth map and later the material was switched to `NormalMaterial` for the actual rendering. So the rendering was done in multi-pass manner.

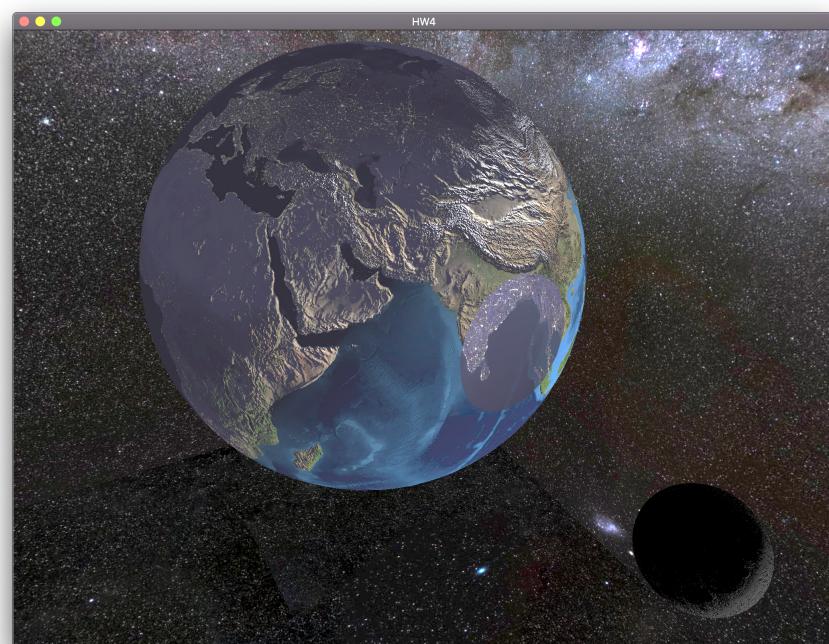
### 4 Creativity

#### 4.1 Other technique in texturing



**Figure 3 Normal Maps**

I added another texture for the night to the Earth object because I felt that the parts of the Earth that are night time would have lights that the people use. It should not be a darker color of texture of the day. Instead, it should have specks of bright parts such as cities. To create this effect, I used the intensity value to combine the color of day texture and night texture. Because the intensity value was simply used, the two textures do not exactly match the lit and unlit sides but it is a good approximation. Also, this scene is a solar eclipse. Parts of the earth that are in shadow of the moon should also have night texture. So I used the shadow value to combine night texture color.



#### 4.2 Interesting and attractive

The project is a model of a solar eclipse. With the normal mapping and shadows, the scene is quite realistic. I wanted to observe the scene more carefully, and help others observe better as well. That's why I added some keyboard interactions. Pressing C on the keyboard enables and disables rotation of the camera around

the earth. Pressing Q, W, A, S, Z, and X translates the camera in the Z, Y, and X axes, respectively. Using this feature, one can observe the solar eclipse from different angles. Also, the Earth and the Moon should both keep rotating even when the solar eclipse happens. To emulate this feature, I added animations for them. Pressing M on the keyboard starts and pauses the rotation of the Earth and the orbiting of the Moon.