

Cube Mapping Reflective and Refractive Bump-mapped Surfaces

PROJECT 4

Megan McCormick | CS557 | 2/11/20

Images

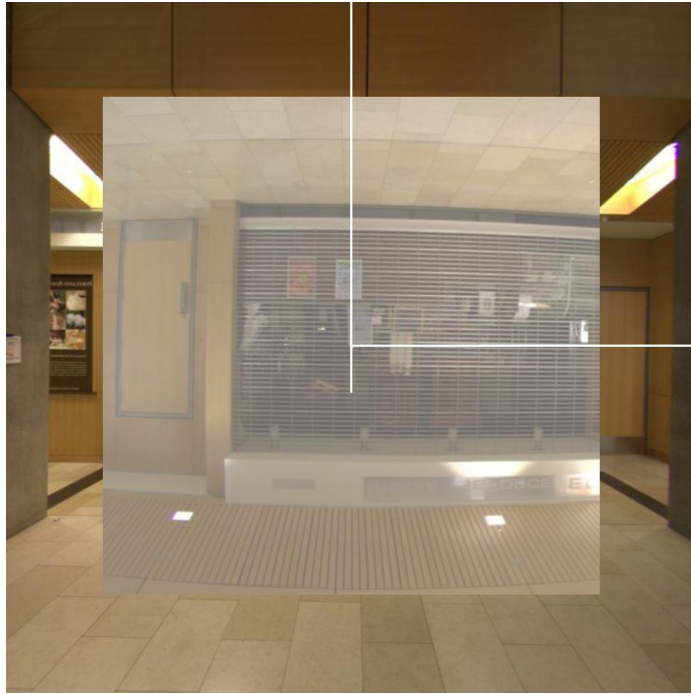


Figure 1: Start Picture



Figure 2: uA Adjusted



Figure 3: uB Adjusted

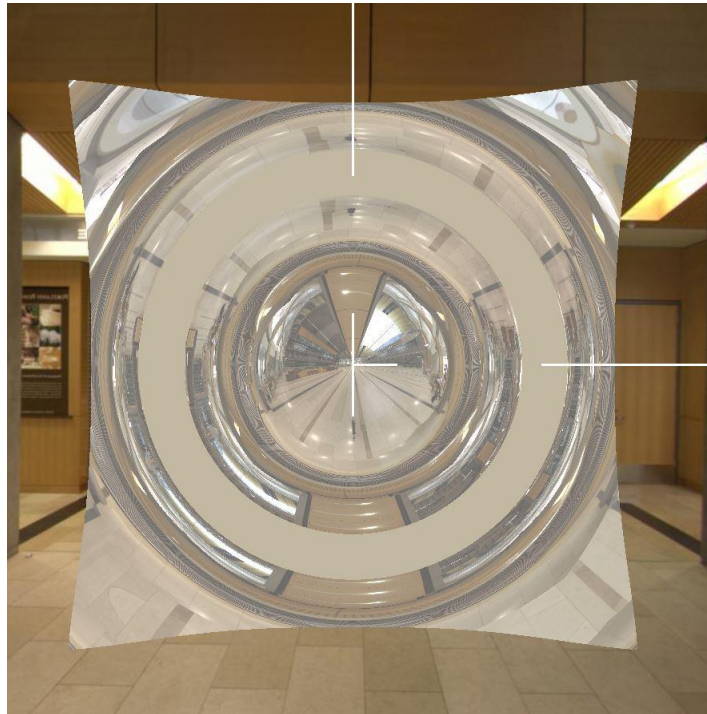


Figure 3: uC Adjusted



Figure 5: uD Adjusted



Figure 6: uNoiseAmp Adjusted



Figure 6: uNoiseFreq Adjusted



Figure 7: uETA Adjusted

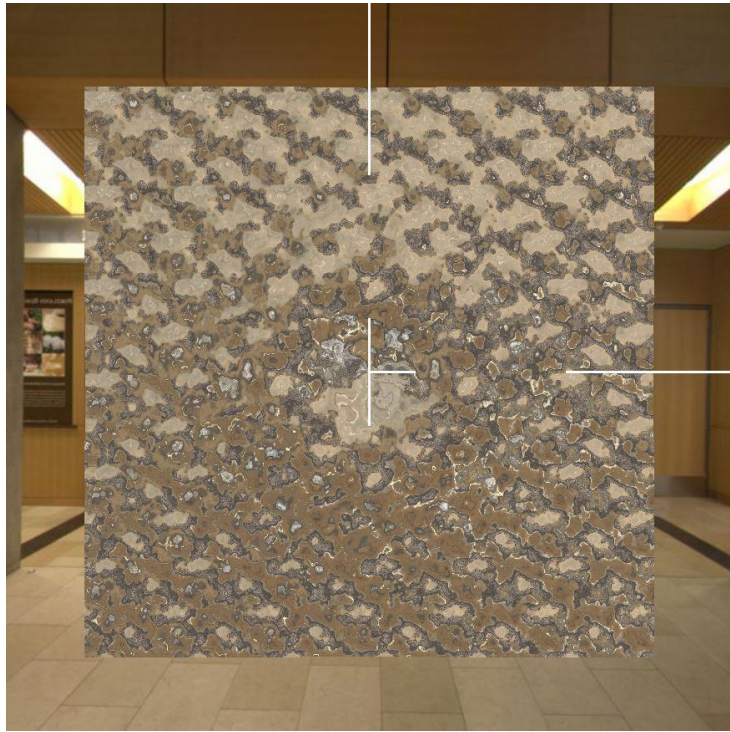


Figure 8: uMix Adjusted (halfway)

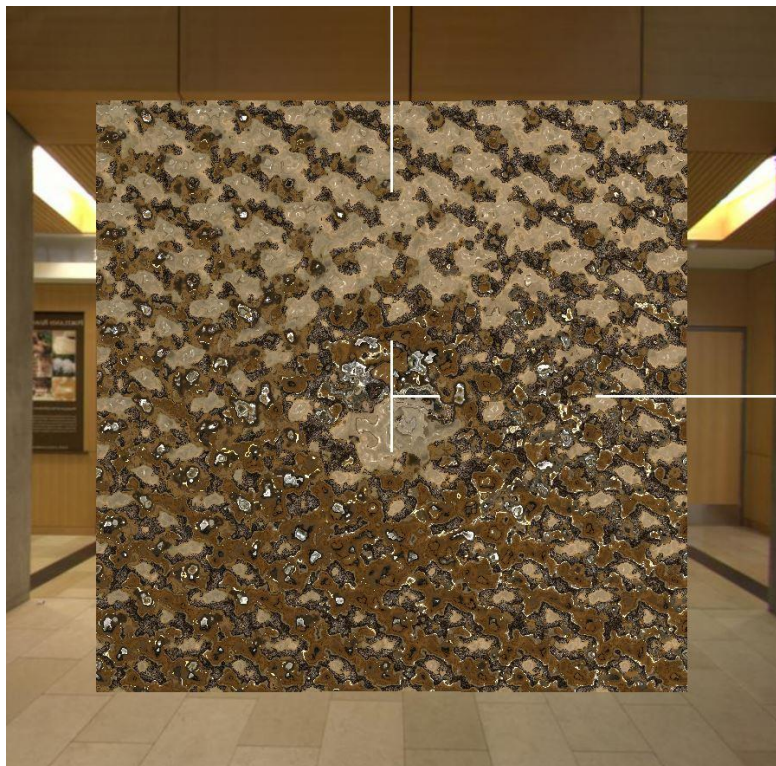


Figure 9: uMix Adjusted (All the way)

KEY CODE

```
vec3 vRefractVector = refract( Eye, Normal, uEta );
vec3 vReflectVector = reflect( Eye, Normal );

vec4 refractcolor = textureCube( uRefractUnit, vRefractVector );
vec4 reflectcolor = textureCube( uReflectUnit, vReflectVector );
refractcolor = mix( refractcolor, WHITE, .40 );

GL_FragColor = vec4( mix( refractcolor, reflectcolor, uMix ).rgb, 1. );
```

VIDEO LINK

https://media.oregonstate.edu/media/t/o_cs3c8pq2

COMMENTS

I computed the normal using the same method as project 3. These equations were from both the assignment documentation and the lecture notes. It works because we take an x and y vector on the plane, compute their cross product, and take the cross product. This will result in the normal vector. Using that normal, the built-in refraction and reflection functions we can get refraction and reflection vectors. Those are mixed together to result in the program seen in the video.