An Interactive Globe

Lei Wang and Yuxi Ren

Professor Ronald Metoyer

Introduction

We want to use WebGL to design an interactive 3D globe, which will use our texture mapping, bump mapping, animation approach as well as our previous knowledge such as shaders, 3D viewing, interaction and transformation.

Solution

At first, we tend to rend a sphere for the globe with Earth texture mapped on. Then we will add mouse interact for camera moving. Thirdly, we will use phone lighting shader for it and add time parameter to make the animation. It likes the follow Figure1:

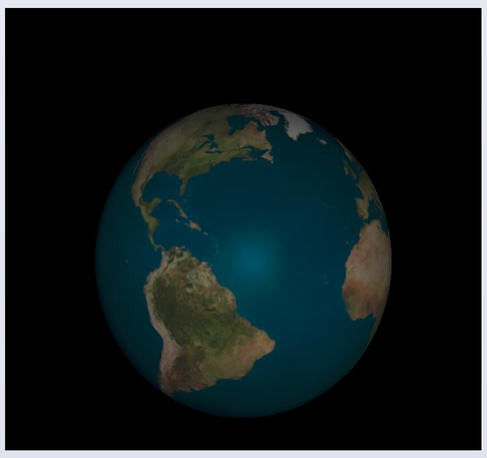


Figure1

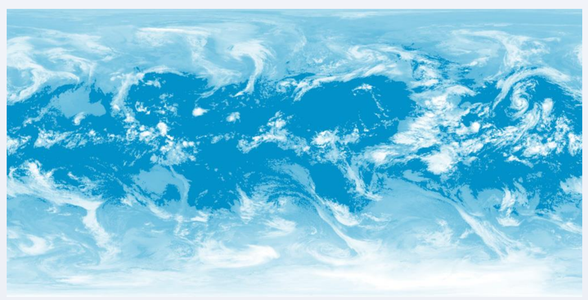
The next step is to add the night-time light. We need gamma correct the nighttime texture. The code is as follow:

```
float gammaCorrect = 1/1.2;
vec4 nightColor = pow(texture2D(u_Night, v_Texcoord), vec4(gammaCorrect));
```

The following figure2 show the globe with night lights on the dark side effect:

Figure2

The third step is add the moving clouds. Since we have day and night lights for the globe, we need two part to render. In day time, clouds should be diffuse lit. In night time, clouds should obscure city lights. In addition, we need animate the clouds by offsetting the variable which combine the time and texcoord.



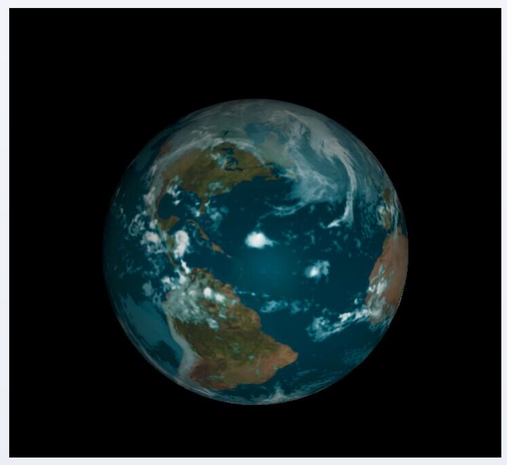
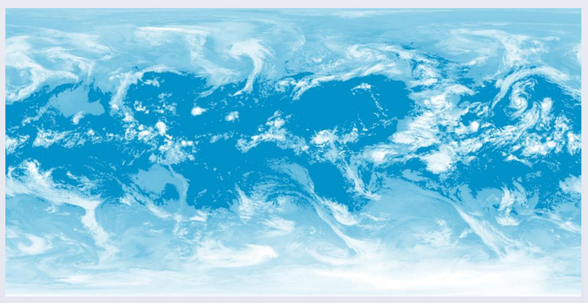


Figure3



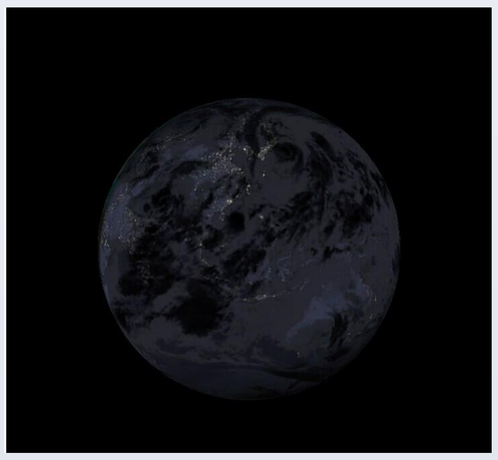


Figure4

The Figure3 and Figure4 show the effect of clouds.

The fourth step is adding the Bump Mapping. In the Figure5, is the texture of the appearance of mountains. The sea-level is black(0) and the highest mountain is white(1).We read three texels from this texture: once using v_Texcoord; once one texel to the right and once one texel above. After that, we transform the normal to eye coordinates, normalize it, then use it for diffuse lighting the ground instead of the original normal.

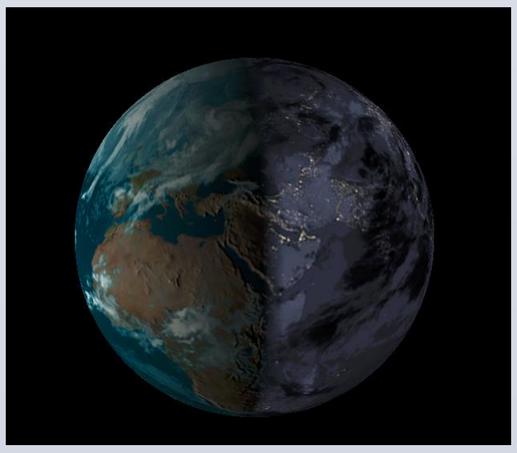


Figure5

Reference

[1] http://planetpixelemporium.com/earth.html

[2] http://cis565-fall-2014.github.io/