Dhvanil Shah
ECE 462
Prof. Sable
8 May 2019

3D Globe Displaying Geographical Data

The goal of this project is to create a 3-Dimensional Earth for use in visualization of geographical data. The data used for this project is global light pollution data provided by National Oceanic and Atmospheric Administration (NOAA). The nature of this dataset will influence the design of this project.

Project Breakdown

This project was completed in three parts:

1. Making the Earth
2. Making the Earth Realistic
3. Displaying Data

Making the Earth

This part of the project was relatively simple compared to the next two parts, therefore extensive detail will be skipped. The first step of the project was to simply render a sphere. This was followed by reading and loading textures. And finally, the loaded textures were mapped onto the sphere using the fragment and vertex shaders. To gain control over the model mouse interaction was added where:

- left click + drag moves the camera around
- right click + drag zooms the model in and out

This resulted in a very flat cartoonish Earth. Therefore, it was determined that the next step was to make the Earth look realistic. The aforementioned steps described were done with the aid of the *Interactive Computer Graphics* textbook by Angel and Shreiner and online resources provided by learningwebgl.com.

Making the Earth Realistic

As suggested by Professor Sable, to add complexity and realness to the globe, a differentiation between day and night should be added. To do this, one side of the model was decided to be daytime with an artificial sun illuminating it and the opposite side (the backside) was determined to be night. To display the nighttime, a night lights texture was used. The diffuse part of the lighting component was used to detect if a fragment was on the backside of the globe, and, if so, the globe was colored using the night lights texture. To avoid a stark transition from night to day, the GLSL *mix* function was used to smoothly transition between the two textures. The gamma correction technique was used to brighten the nighttime texture.

Second, a specular map was used to add some reflection in the oceans when light was shone on them. This distinguishing between land and water was very important in adding reality to the globe. In the fragment shader, *u_EarthSpec* is used determine if a fragment is on ocean or land, and only the specular component is only included if it is in ocean. Furthermore, a wave texture map and bump map are applied at the locations where the water is present as denoted by the specular map.

The next step was to add the atmosphere around the Earth. This was done by adding a layer of clouds that rotated around the Earth. The layer of clouds was made possible by the user of a clouds texture map and a cloud transparency map. The cloud transparency map is similar to the Earth specular map, except it helps distinguish the transparency of the clouds throughout the clouds texture map. In the fragment shader, *u_Cloud* mixed with *u_CloudTrans* was used to determine the color of the clouds. The clouds are animated by offsetting the *v_Texcoords.s* by *u_time* (time in seconds since load) when reading *u_Cloud* and *u_CloudTrans*. Furthermore, a technique called rim lighting is used to apply a lighting effect which makes the Earth appear as if it has an atmospheric layer which catches light from the sun. Implementing rim lighting was slightly complicated and it was done using a tutorial titled Edge Effects by Malcolm Kesson. First, a rim factor was found by adding one to the dot product of *v_Normal* and *v_Position.* If the rim factor was greater than 0, then a blue color based on the rim factor to the current fragment color (as suggested by the tutorial, *vec4(rim/4, rim/2, rim/2, 1)*). If the rim factor was not greater than 0, then the fragment color was left as is.

Finally, a bump map was applied to add geographic by perturbing the normal used for diffuse lighting the ground. The texture map, *u_Bump*, is zero when the fragment is at sea-level, and one when the fragment is on the highest mountain. A perturbed normal was created using the *normal* function. This new normal was then input into the *transformBump* function to transform this normal to eye coordinates, normalize it, then use it for diffuse lighting the ground instead of the original normal. A similar technique was used to add the water bump map mentioned previously.

<u>Displaying Data</u>

Surprisingly, the easiest part was displaying the data. This was thanks to NOAA for providing a plethora of Earth texture maps reporting a variety of geographical data. The light pollution dataset was chosen due to its parallels with using the night lights texture map. Adding data was done simply by replacing the source of the night light texture map with the source of the city lights texture map.

<u>Extras</u>

This portion of the project was not required but done for fun. Environment mapping was attempted with partial success. A skybox, cube, was placed around the object. Using separate fragment and vertex shaders for the cube, six (of the same) images were mapped onto the inner surfaces of the cube to simulate a galaxy. This was considered a partial success because the galaxy looks terrible. This is because the same image was used on each of the six faces causing obvious appearance of repetition. In true environment mapping six different images must be used to cause the appearance of one seamless environment. This was no done due to an inability to find a open source environment map of the galaxy.

Instructions
Use URL parameters to select views:
Example: path/to/file/index.html?data=lightpollution&sky=galaxy
Default (no parameters): path/to/filex/index.html

- Globe Data: use the URL parameter "data" to select the data
  - Default: No data
  - Light Pollution: "data=lightpollution"
- Skybox: use the URL parameter "sky" to select the skybox
  - Default: No skybox
  - Galaxy: "sky=galaxy"

Movement

- left click + drag moves the camera around
- right click + drag zooms the model in and out

Sources Cited

- Data
  - Light Pollution Data: https://sos.noaa.gov/datasets/light-pollution-artificial-sky-brightness/
  - All Earth Texture Maps: http://planetpixelemporium.com/earth.html
- Tutorials
  - Rim Effects Tutorial: http://www.fundza.com/rman_shaders/surface/rim_effects/index.html
  - Using Shaders to Move Clouds: https://github.com/mattdesl/lwjgl-basics/wiki/ShaderLesson2
  - Gamma Correction (dawn/dusk): https://learnopengl.com/Advanced-Lighting/Gamma-Correction