

Kevin King
Professor Zhu
COSC 77: Computer Graphics
Assignment 3: Technical Implementation

Step 0: Generating UV Coordinates For a Sphere

I implemented the `Update_UV_Using_Spherical_Coordinates` function in the `main.cpp` file for each vertex `'vtx'` in the `'vertices'` array. The radius is the square root of $\text{vtx.x}^2 + \text{vtx.y}^2 + \text{vtx.z}^2$. The angle theta is equal to $\arcsin(-\text{vtx.y}/\text{radius})$, and the angle phi is equal to the arctan of $-\text{vtx.z}$ and vtx.x . I used the uv coordinate calculations using theta and phi to get both u and v, storing those values in the `'uv'` array. For the checkerboard, I implemented these same steps directly in the fragment shader to fix the seam issue for the sphere.

Step 1: Color Texture

For this step, I passed the vector `'uv'` from the vertex shader into the fragment shader. I then used the `'texture'` function to load the texture color `'tex_color'` from `'tex_albedo'` and set that as the final output color. In the `normal_mapping.frag` file, I implemented the uv spherical coordinate calculations to get rid of the seam issue for the globe, but left it in comments and used the `'uv'` vertex for this step.

Step 2: Texturing + Shading

I added constant vectors for the `Ia` and `Id` values for Lambertian shading, then used the `'tex_color'` for the shading model. The final output color was the sum of the diffuse and ambient lighting.

Step 3: Normal Mapping

I loaded in `'tex_color'` the same way as Step 1, then for the `'tex_nml'` vector I used `'tex_normal'` to load. To remap each component from $[0,1]$ to $[-1,1]$, I multiplied `'tex_nml'` by 2 and subtracted 1. For the TBN matrix, I used the `'tangent'` and `'normal'` vectors from the vertex shader to calculate the bitangent, normalized each of them, then created the TBN matrix with the three vectors. I used the TBN matrix to transform `'tex_nml'` from the local tangent space to the global world space. Finally, I used `'tex_nml'` for the normal vector in the diffuse lighting calculation and followed similar steps as in Step 2 to get the final output color.

Step 4: Texturing a Different Mesh

I tested out the different textures on the `bunny.obj` file. I used the `dartmouth.jpg`, `earth_albedo.png`, and `bunny.jpg` files for `'tex_albedo.'` I also found a jukebox mesh online that provided its own jukebox texture that I used for `'tex_albedo,'` and I found a wood texture that I used for `'tex_normal'` to create a realistic-looking old jukebox.