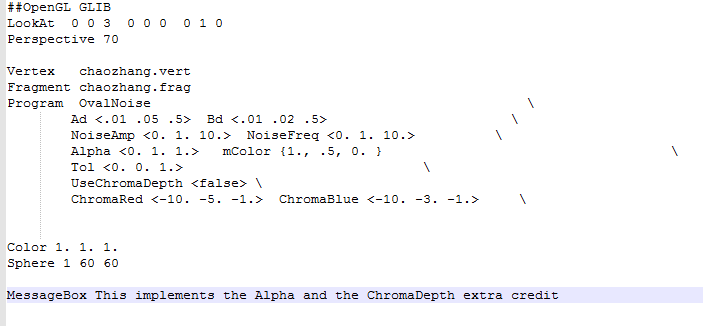
***RenderMan and OpenGL Shaders***

CS557

Project # 3

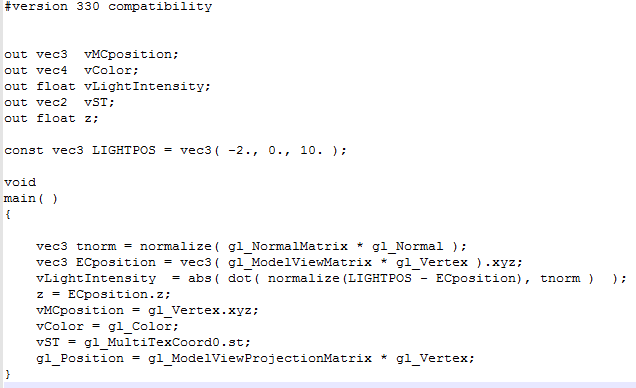
Chao Zhang

1. Source listings



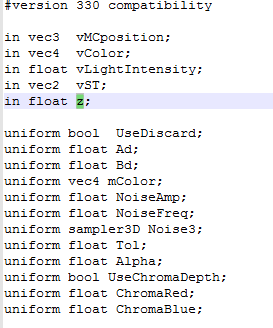
chaozhang.glib

This file include all the definition I need to use. The ChromaRed and ChromaBlue are both in the range of -10 to -1, and the default is -5. mColor is still the beaver orange. This time I didn’t use the teal as background color because it is not clear to see the tol. So I use the write as background color.



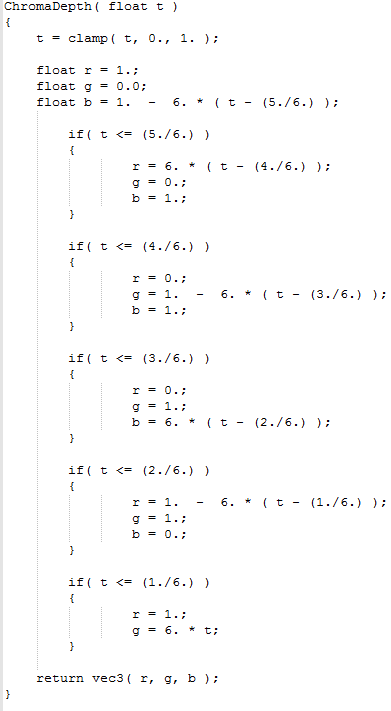
chaozhang.vert

Z= ECposition.z is the vertex position in eye coordinates. The vLightIntensity is the computed ligthingin the vertex shader. tnormfor storing the transformed normal. The vMCposition, vcolor, cLightIntensity, VST and z will be trans to the .freq file.



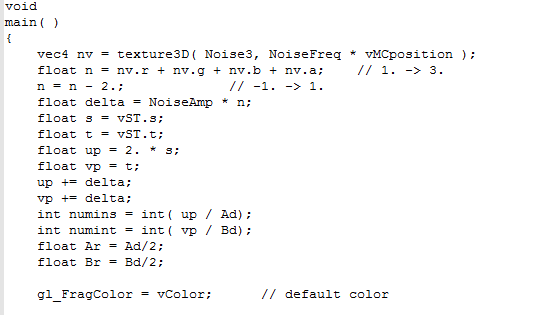
Part1 chaozhang.freq

This is part of the .freq file. This part will get the data from the vertex shader and define the variable.



Part2 chanzhang.freq

This part is to implement the different color for the ChromaDepth. Use different color for different range of t. This can change the sphere into rainbow sphere.



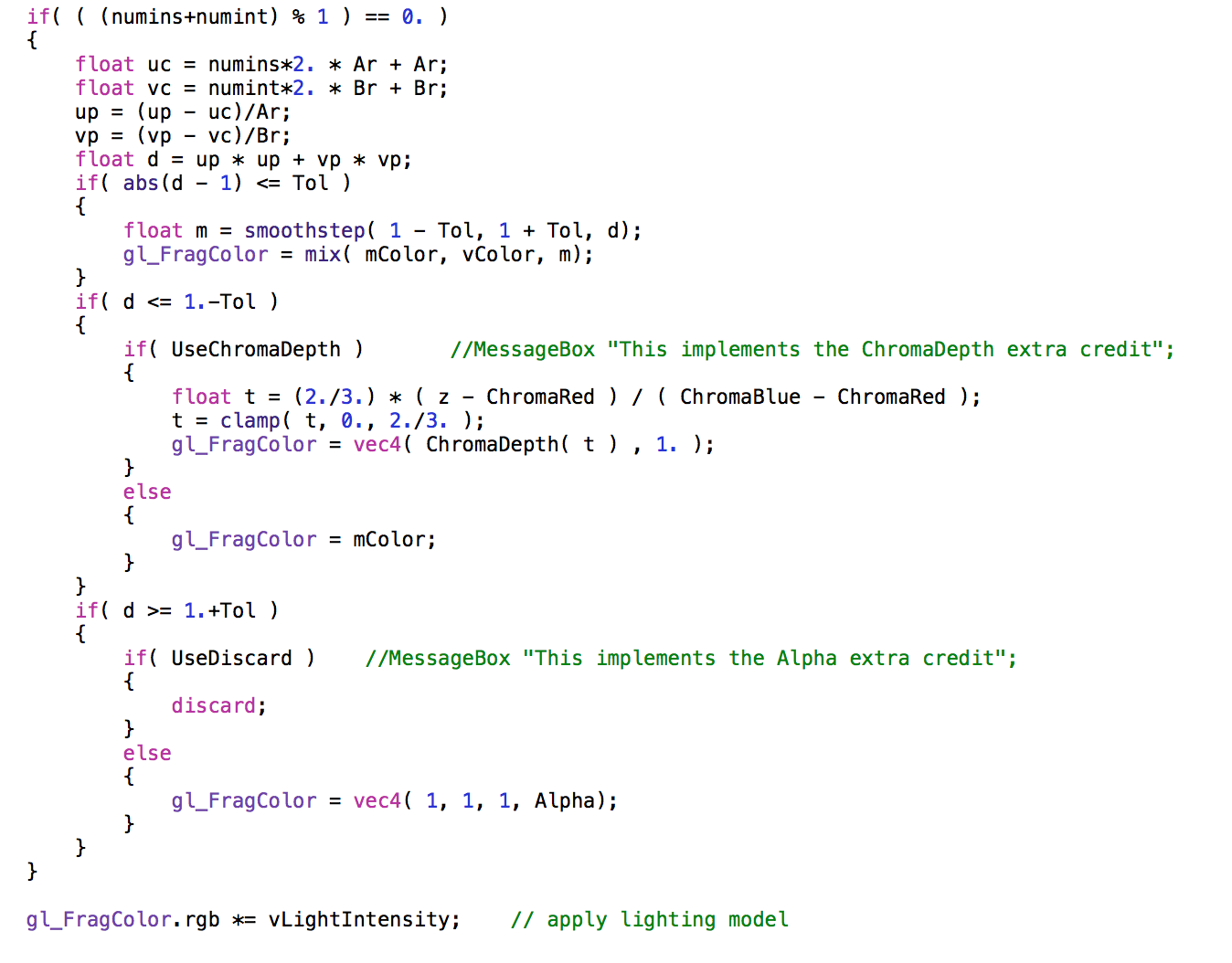
Part3 chaozhang.freq

This part use the Noise3 to create the moise. There are four values in this, the r,g,b and a. The “noise vector” texture nv is a vec4 whose components have separate meanings.

float n = nv.r + nv.g + nv.b + nv.a;

n = n – 2;

Those two lines made the range of the four-octave function from 0 to 1. The rest of the code is like the project1 and 2. It is for the eclipse.



part4 chaozhang.freq

This part is the most important part of this project.

if( abs(d - 1) <= Tol )

{

float m = smoothstep( 1 - Tol, 1 + Tol, d);

gl\_FragColor = mix( mColor, vColor, m);

}

Those lines for the Tol function. With the increase of tol, the edge of the eclipse smooth. I mix the color to achieve the smooth goal.

if( d <= 1.-Tol )

{

if( UseChromaDepth ) //MessageBox "This implements the ChromaDepth extra credit";

{

float t = (2./3.) \* ( z - ChromaRed ) / ( ChromaBlue - ChromaRed );

t = clamp( t, 0., 2./3. );

gl\_FragColor = vec4( ChromaDepth( t ) , 1. );

}

else

{

gl\_FragColor = mColor;

}

}

the inside loop is for the chromadepth. The reason it is a inside loop is because I only turn the eclipse color to rainbow not include the background color. The range of t is from 0 to 2/3 is because we want the red and blue.

1. Results

