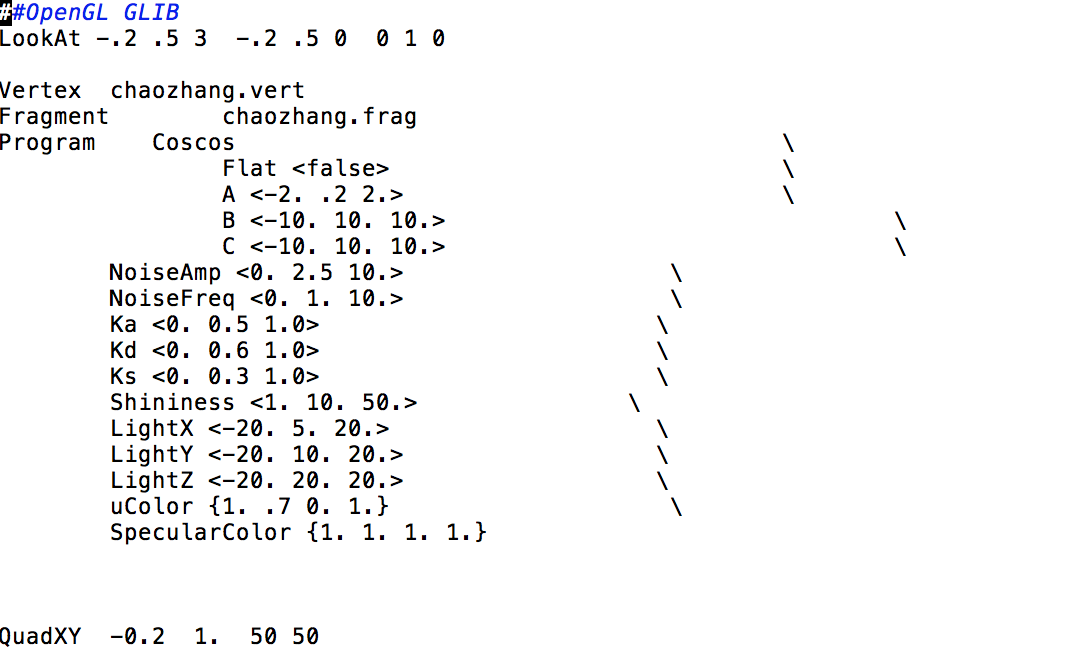
***RenderMan and OpenGL Shaders***

CS557

Project # 4

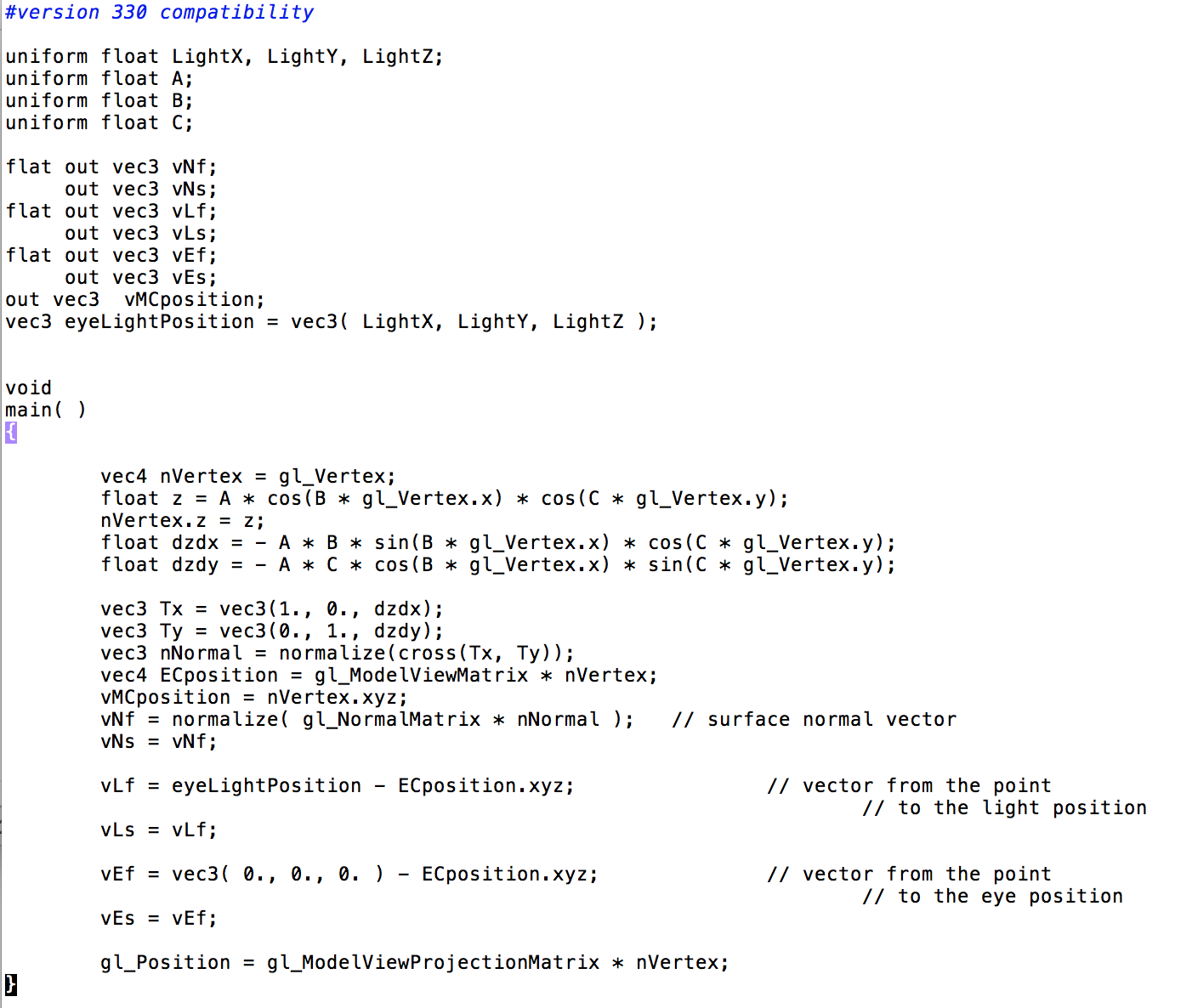
Chao Zhang

1. Source listings



chaozhang.glib

This is the glib file. This file includes all the basic information about this project. The shader type which is vertex shader and the fragment shader. The A, B, C are the changes in the x, y, z axle. The Ka, Kd and Ks are the Ambient, Diffuse, and Specular Color. The LightX, LightY and LightZ are the position of the light in the axles.



chaozhang.vert

This is the vertex shader file. This file has all the vertex shader information. The compute of the normal, position of the lights. The nVertex mean the new vertex position, it is equal to gl\_vertex in the beginning. Then I use the float z = A \* cos(B \* gl\_Vertex.x) \* cos(C \* gl\_Vertex.y); to compute the position in the z axle of the gl\_vertex and give this new z value to the nVertex. This make the change of A works.

float dzdx = - A \* B \* sin(B \* gl\_Vertex.x) \* cos(C \* gl\_Vertex.y);

float dzdy = - A \* C \* cos(B \* gl\_Vertex.x) \* sin(C \* gl\_Vertex.y);

This two line can get the tangent and then I use

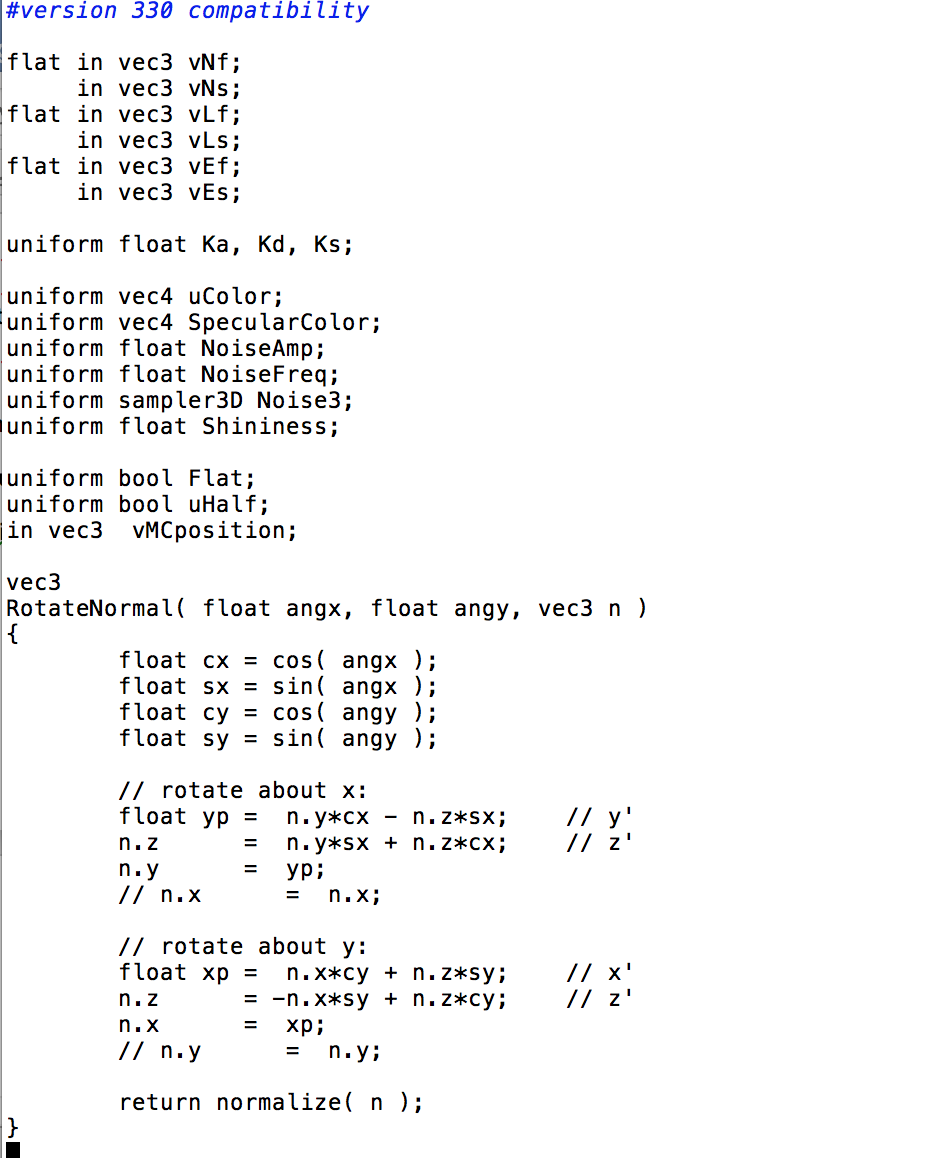
vec3 Tx = vec3(1., 0., dzdx);

vec3 Ty = vec3(0., 1., dzdy); to get the tangent vectors. The Tx and Ty can help us to get the new normal. vec3 nNormal = normalize(cross(Tx, Ty));

Because of the change of the A, I need to change the light position.

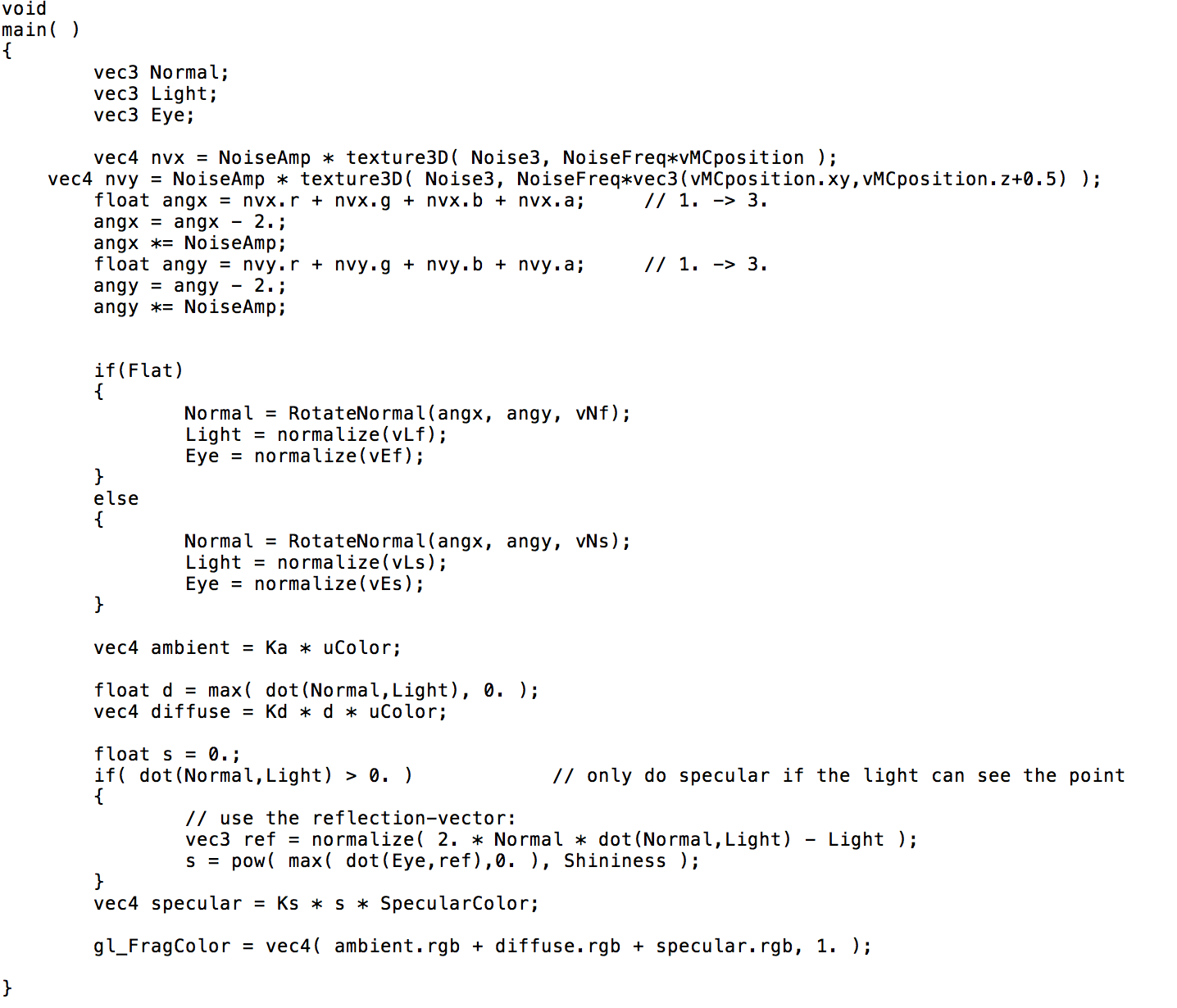
vNf = normalize( gl\_NormalMatrix \* nNormal );

gl\_Position = gl\_ModelViewProjectionMatrix \* nVertex;



part1 chaozhang.frag

This is the first part of the fragment shader. This part do a rotate of the normal. The reason I do this is to implement the bump-mapping. This function is to rotate the normal in two ways, the angle with x axle and the angle with y axle.



part2 chaozhang.frag

This part is to implement the rotate normal.

vec4 nvx = uNoiseAmp \* texture3D( Noise3, uNoiseFreq\*vMC );

vec4 nvy = uNoiseAmp \* texture3D( Noise3, uNoiseFreq\*vec3(vMC.xy,vMC.z+0.5) );

this two line is to calculate the noise vectors. This part is the same as project 3 noise part. After that, I make the Normal = RotateNormal(angx, angy, vNf); to get the result.

1. Result

