CS557 Project #3

Yunfan Li

[liyunf@oregonstate.edu](mailto:liyunf@oregonstate.edu)

1. **Source Listings**
   1. ***liyunf.glib***

##OpenGL GLIB

LookAt 0 0 3 0 0 0 0 1 0

Perspective 70

Noise3D 128

Vertex liyunf.vert

Fragment liyunf.frag

Program OvalNoise \

uAd <.01 .05 .5> \

uBd <.01 .05 .5> \

uNoiseAmp <0. 0. 10.> \

uNoiseFreq <0. .15 2.> \

uAlpha <0. 1. 1.> \

uTol <0. 0. 1.>

Color 1. 1. 1.

Sphere

* 1. ***liyunf.vert***

#version 330 compatibility

out vec4 vColor;

out float vLightIntensity;

out vec2 vST;

out vec3 vMCposition;

const vec3 LIGHTPOS = vec3( 0., 0., 10. );

void

main( )

{

vColor = gl\_Color;

vLightIntensity = abs( dot( normalize(LIGHTPOS - vec3( gl\_ModelViewMatrix \* gl\_Vertex )), normalize( gl\_NormalMatrix \* gl\_Normal ) ) );

vST = gl\_MultiTexCoord0.st;

vMCposition = gl\_Vertex.xyz;

gl\_Position = gl\_ModelViewProjectionMatrix \* gl\_Vertex;

}

* 1. ***liyunf.frag***

#version 330 compatibility

in vec4 vColor;

in float vLightIntensity;

in vec2 vST;

in vec3 vMCposition;

uniform float uAd;

uniform float uBd;

uniform float uNoiseAmp;

uniform float uNoiseFreq;

uniform float uTol;

uniform float uAlpha;

uniform sampler3D Noise3;

void

main ()

{

float sp = 2\*vST.s;

float tp = vST.t;

int numins = int( sp / (2.\*uAd) );

int numint = int( tp / (2.\*uBd) );

float sc = numins \* 2. \* uAd + uAd;

float tc = numint \* 2. \* uBd + uBd;

vec4 nv = texture3D( Noise3, uNoiseFreq \* vMCposition );

float oldrad = length(vec2(sp-sc, tp-tc));

float newrad = oldrad + (uNoiseAmp \* (nv.r + nv.g + nv.b + nv.a - 2.));

vec2 stp = vec2(sp-sc, tp-tc) \* newrad/oldrad;

float d = pow(stp.s/uAd, 2)+pow(stp.t/uBd, 2);

if (d <= 1.) {

float t = smoothstep(1-uTol, 1+uTol, d);

gl\_FragColor = mix(vec4(1, .4, 0., 1.),vec4(1., 1., 1., 1.), t);

}

else {

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

if (uAlpha == 0.)

discard;

else

gl\_FragColor.a = uAlpha;

}

gl\_FragColor.rgb \*= vLightIntensity;

}

1. **What I did & Why it worked**

***2.1 What I did***

In my liyunf.glib file, I created a sphere and connect this .glib flie with liyunf.vert vertex shader and liyunf.frag fragment shader.

In my liyunf.vert file, I take gl\_Color and passed it to vColor as one of fragment shader input value. The next line is used to calculate per-vertex lighting. After that, gl\_ MultiTexCoord0 was took as texture s and t coordinates, and gl\_Vertex was took as x, y, z coordinates at the same time.

In my liyunf.frag flie, global variable “sp” and “tp” were firstly used to calculate which box does this point belongs to. Then the center point coordinate of this box was calculated using “numins” and “numint”. Then a noise variable “nv” was calculated through Noise3D, which accept vec3 and generate a vec4 that contains 4 octave noise value. A variable named “oldrad” was calculated by typical ellipse equation, and another variable named “newrad” would be generated by oldrad plus noise value. New fake input point stp would be calculated using newrad divided by oldrad. And finally, the new “d” was generated by using stp in traditional ellipse equation. When d is less than 1, uTol would be used to generate a variable “t” in order to do color mix.

***2.2 Why it worked***

The noise is a variable that we can control. So we use texture to pass vec3 and then get a vec4. We can use this vec4 variable as a noise value and add it to old radius value to generate a new fake radius. In this way we can generate a noised object.

1. **Images**













