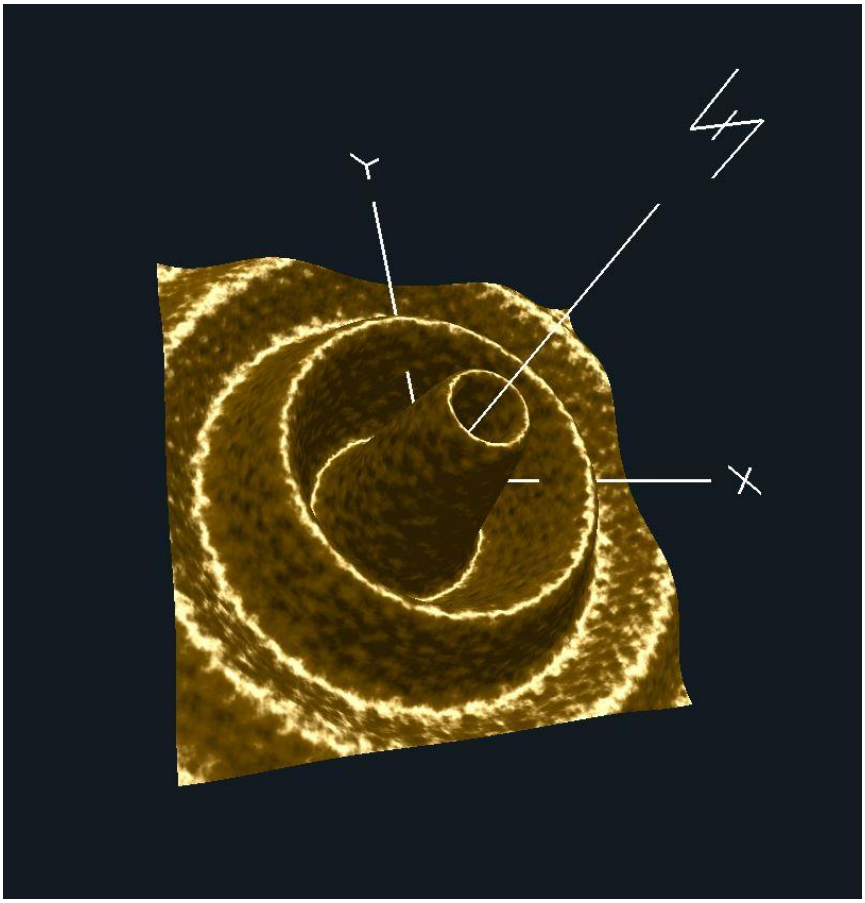

Mapping and Lighting

PROJECT 3

Megan | CS557 | 1/29/20

Images

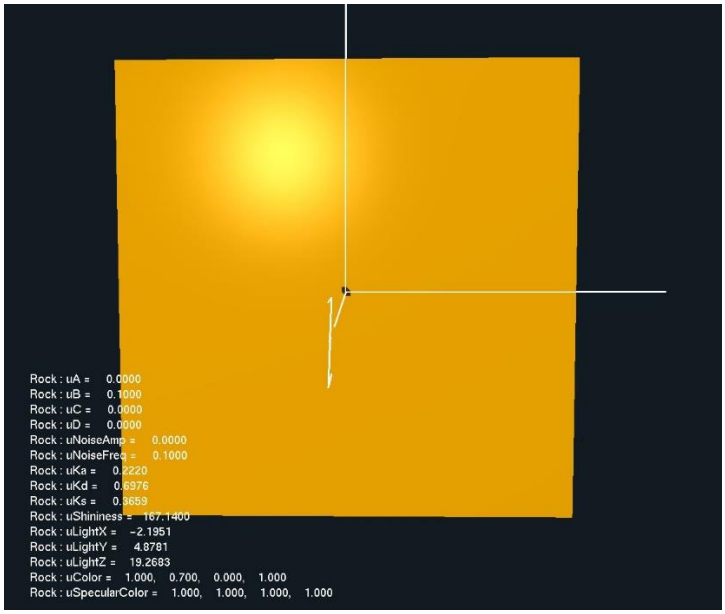

Figure 1: Start Picture

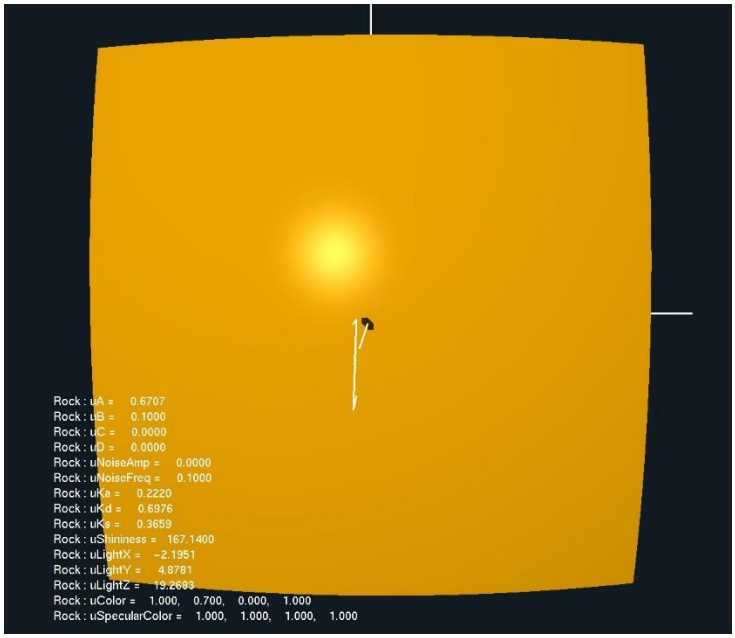

Figure 2: uA Adjusted

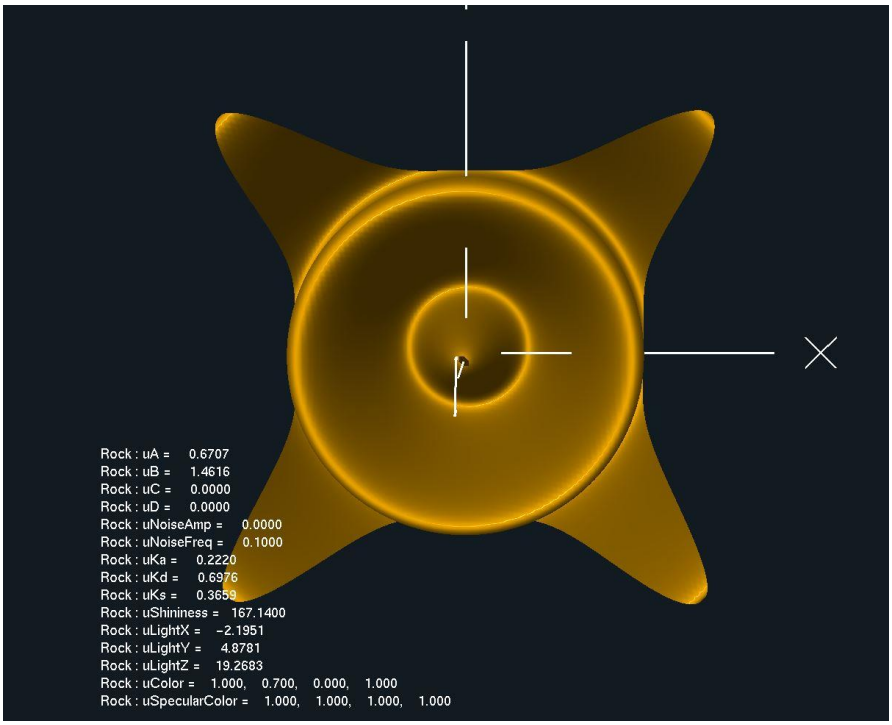

Figure 3: uB Adjusted

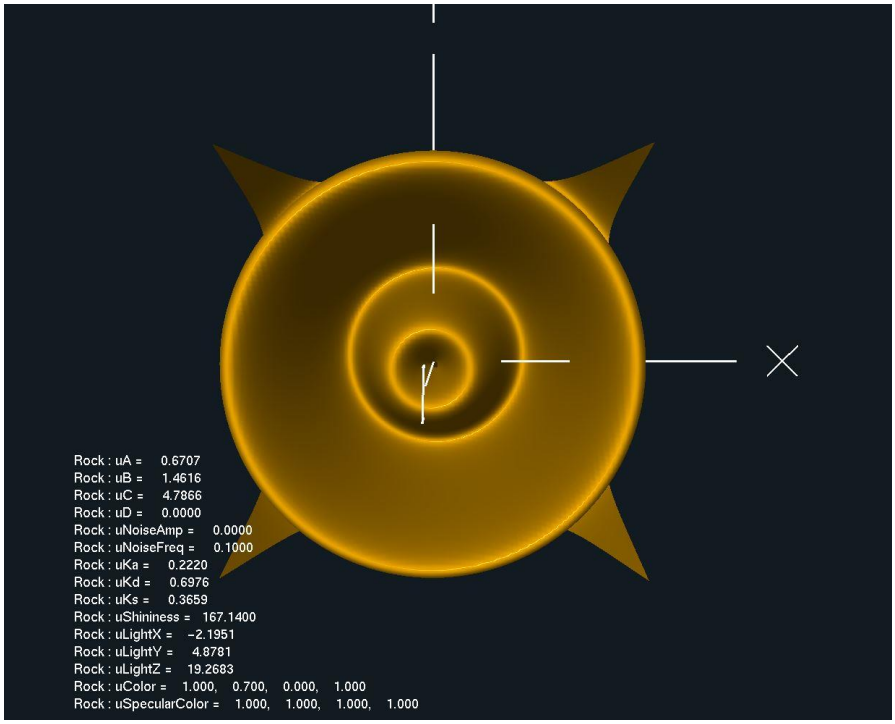

Figure 3: uC Adjusted

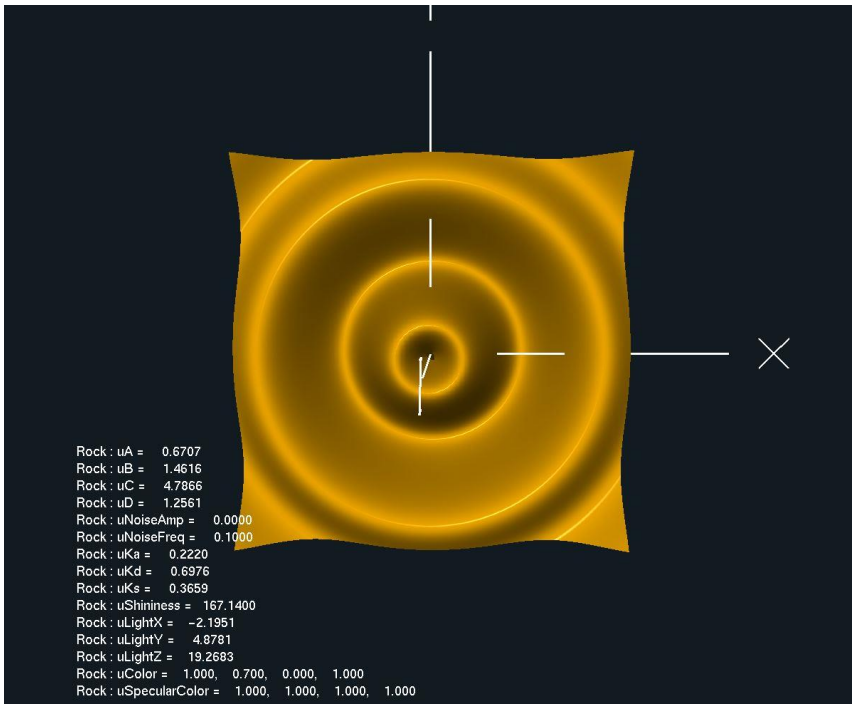

Figure 5: uD Adjusted

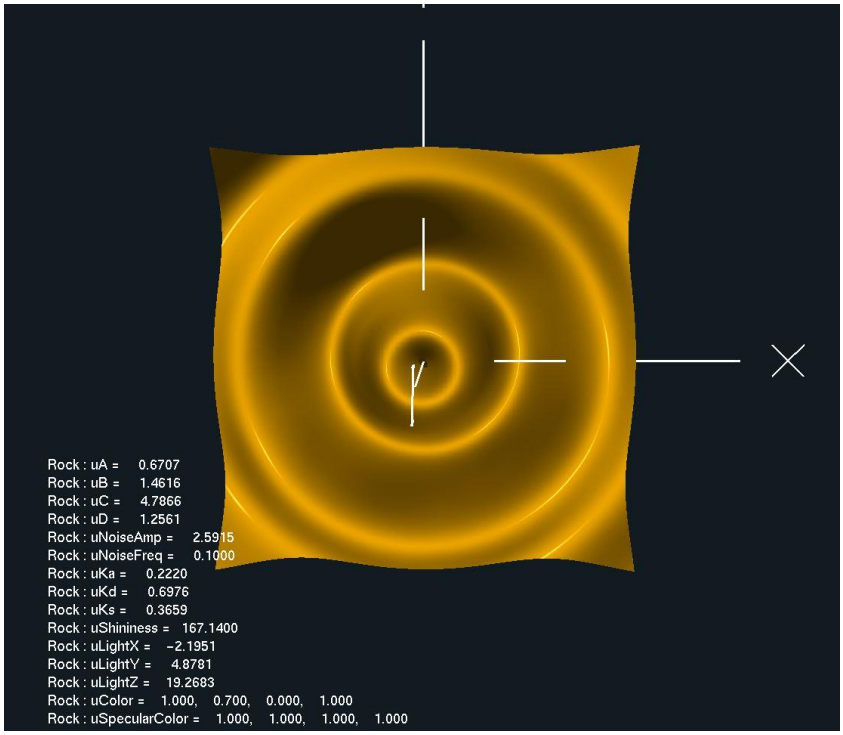

Figure 6: uNoiseAmp Adjusted

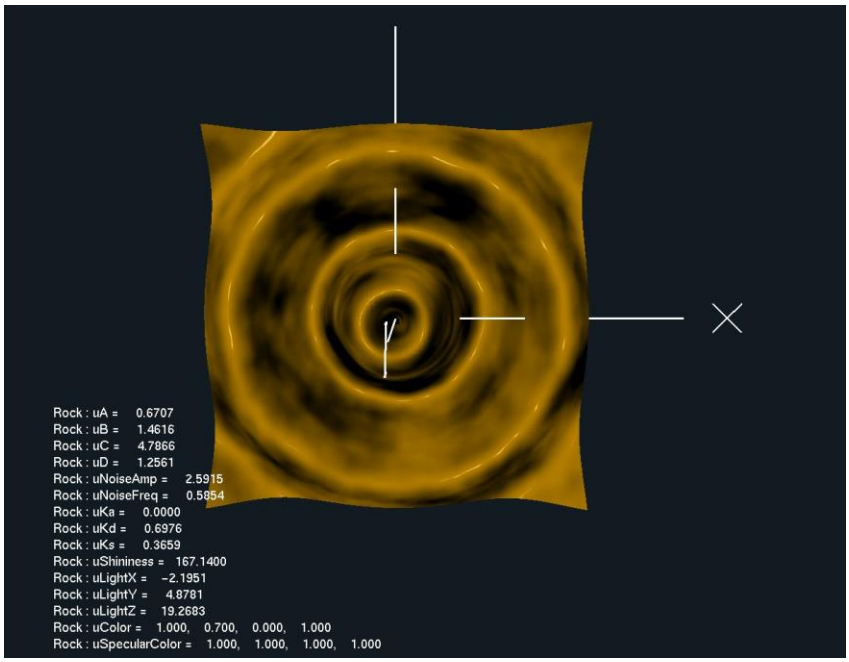

Figure 6: uNoiseFreq Adjusted

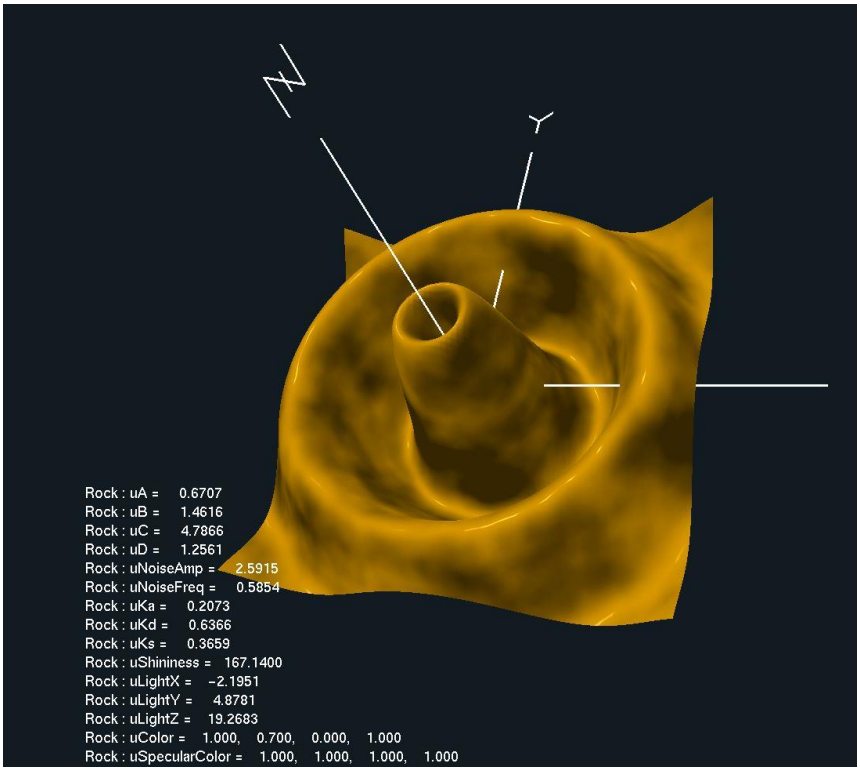

Figure 7: uKa Adjusted

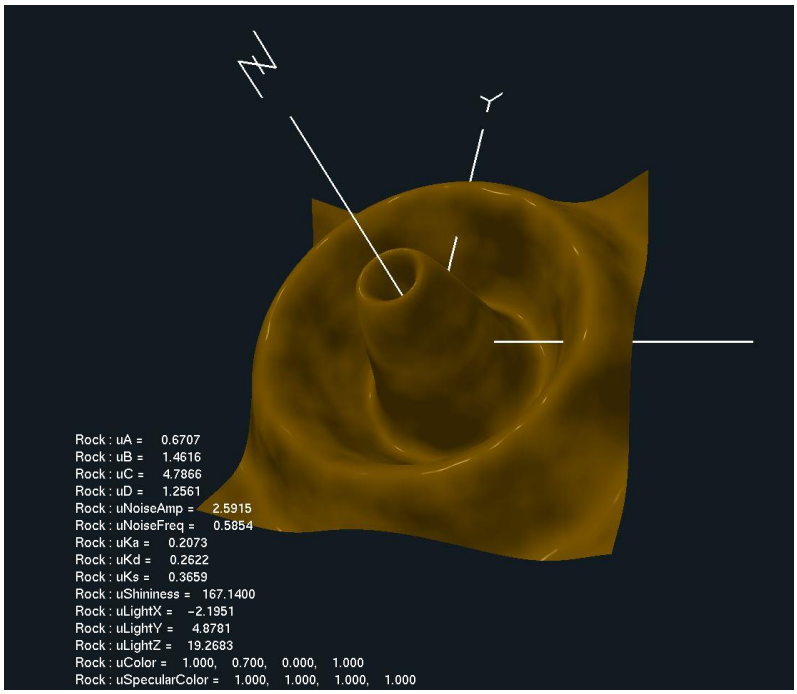

Figure 8: uKd Adjusted

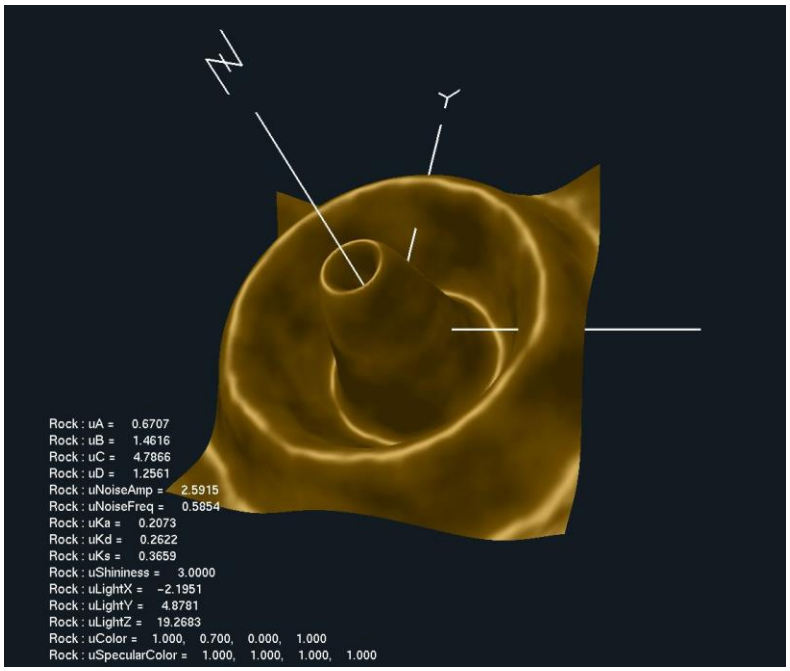

Figure 9: uKs Adjusted

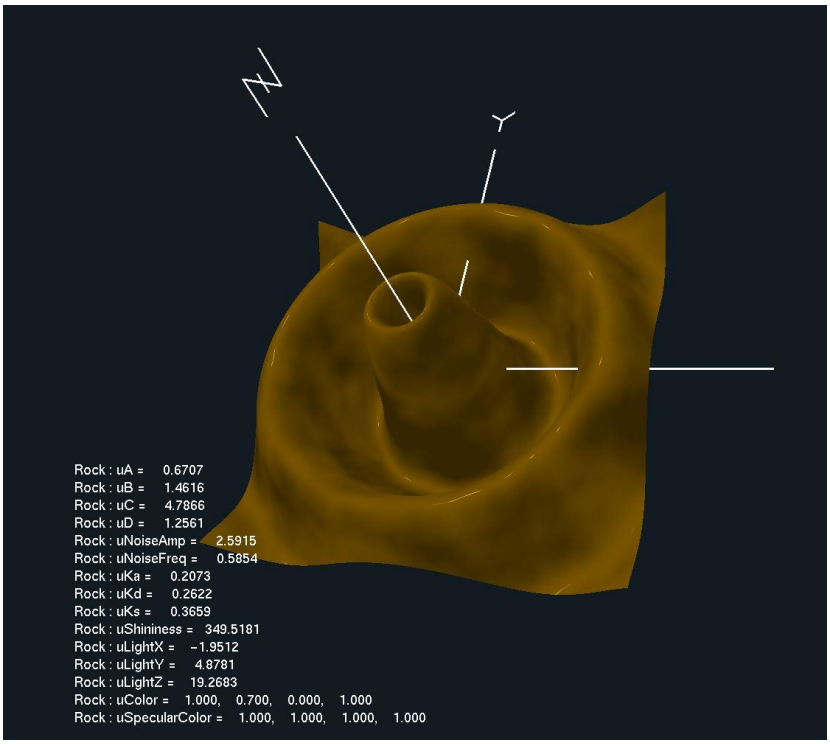

Figure 10: uLightX Adjusted

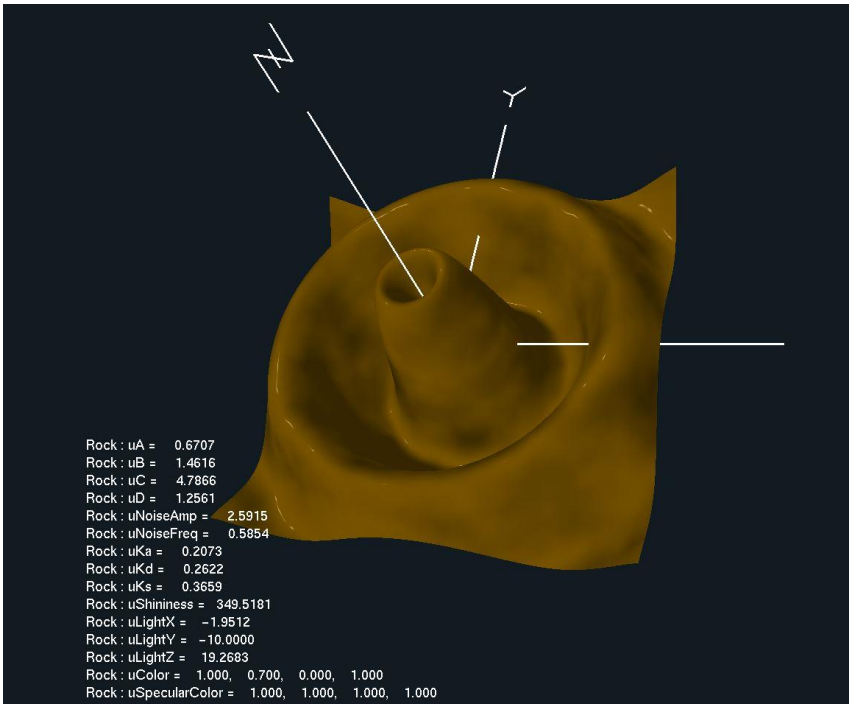

Figure 11: uLightY Adjusted

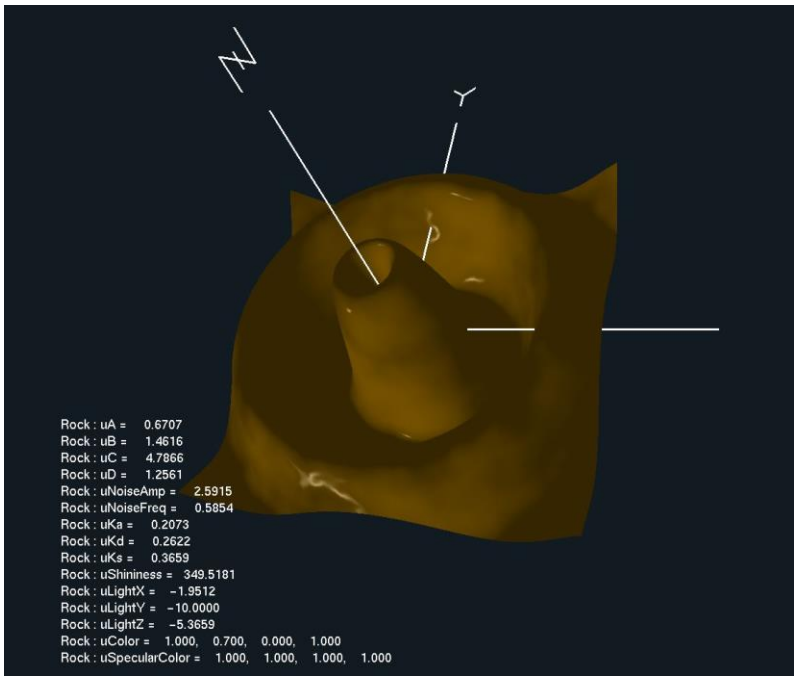

Figure 12: uLightZ Adjusted

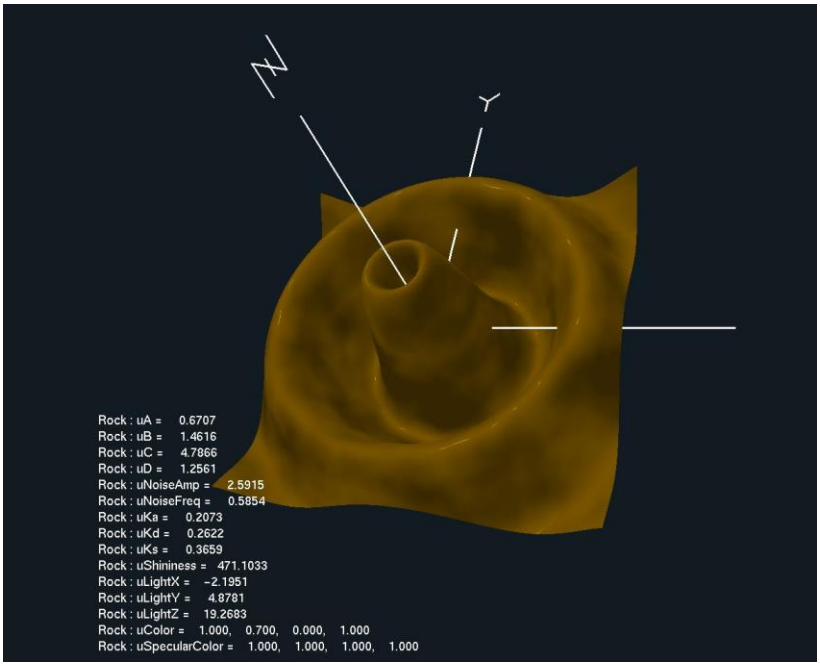

Figure 13: uShininess Adjusted

KEY CODE

```
float r = sqrt( (gl_Vertex.x*gl_Vertex.x)+ (gl_Vertex.y * gl_Vertex.y));
       float z = uA * (cos(2*PI*uB*r+uC)*exp(-uD*r));
       float dzdr = uA * (-sin(2.*PI*uB*r+uC) * 2.*PI*uB * exp(-uD*r) +
cos(2.*PI*uB*r+uC) * -uD * exp(-uD*r) );
       float drdx = gl_Vertex.x/r;
       float drdy = gl_Vertex.y/r;
       float dzdx = dzdr * drdx;
       float dzdy = dzdr * drdy;

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

       vec4 ECposition = gl_ModelViewMatrix * gl_Vertex;


       NF = NORMALIZE(CROSS( TX, TY ) );  // SURFACE NORMAL VECTOR
```

VIDEO LINK

https://media.oregonstate.edu/media/t/0_7ze05tks

COMMENTS

I computed the normal using the equations above in the code. These equations were from both the assignment documentation and the lecture notes. It works because we take and x and y vector on the plane compute or math and take the cross product. This will result in the normal vector.