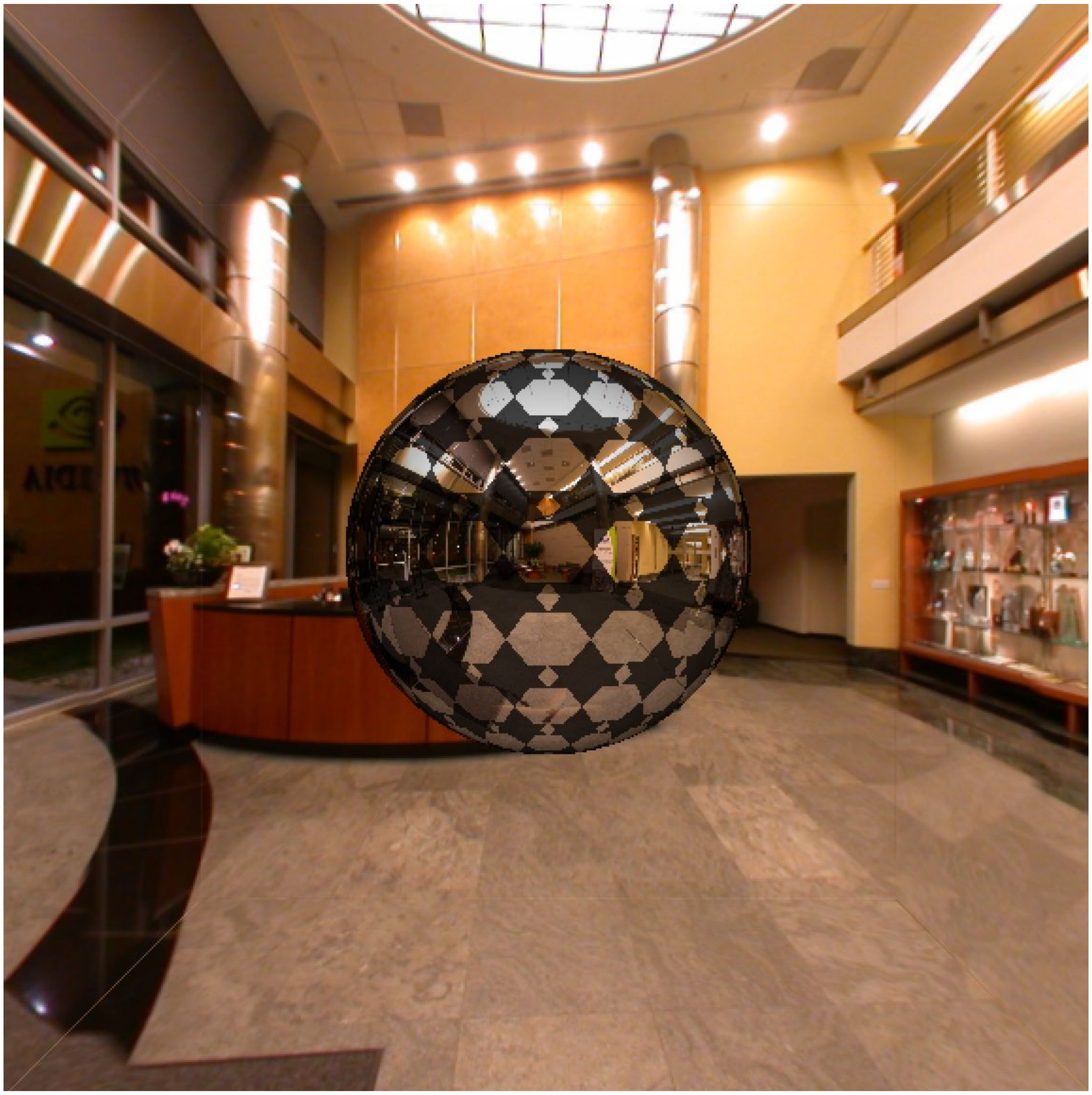
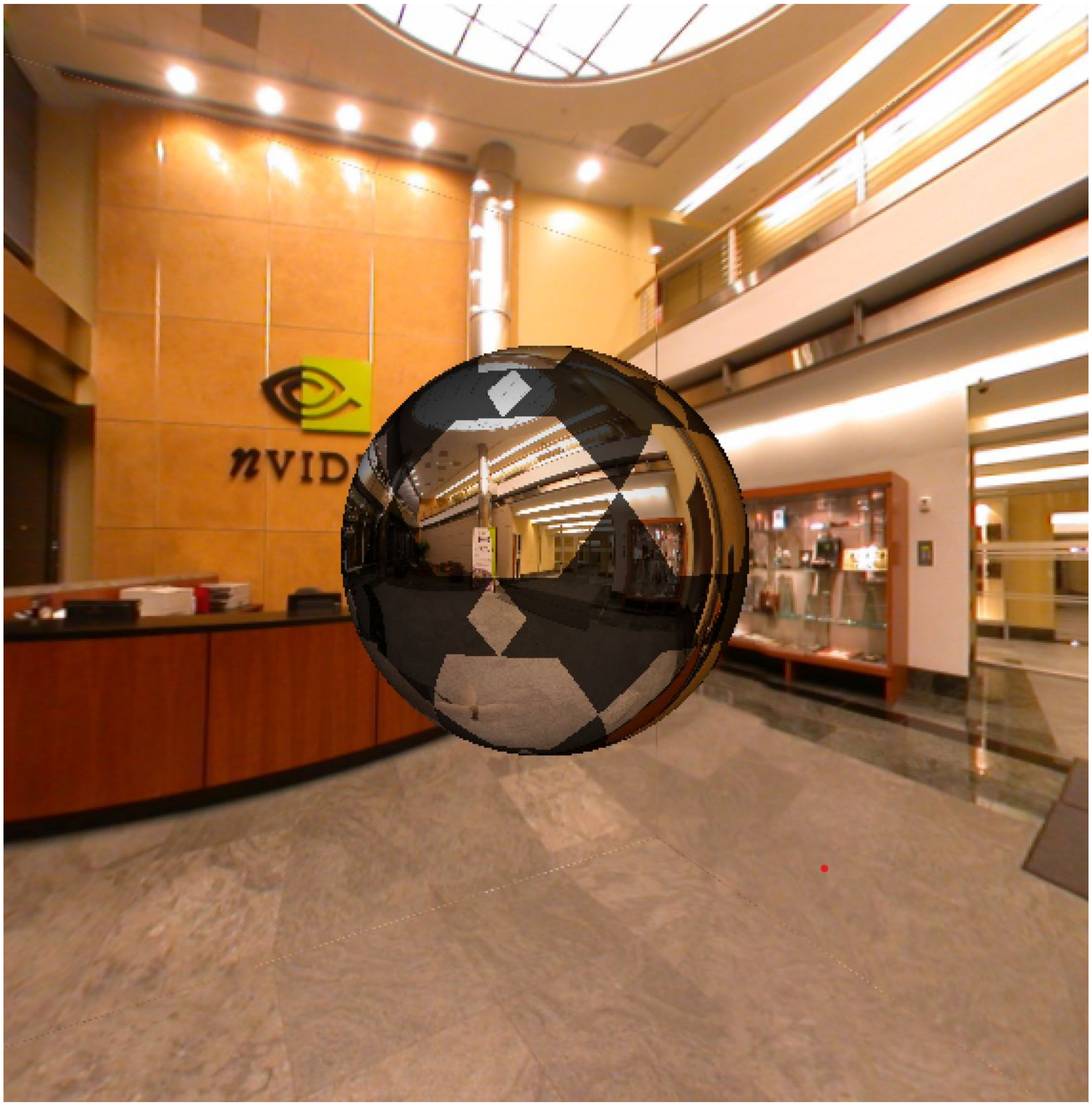
Project Title : Final Project
Name : Si Thu Lin
ID : 933-957-884
Video Link : https://media.oregonstate.edu/media/t/0_2l19oau3
Email : linsi@oregonstate.edu

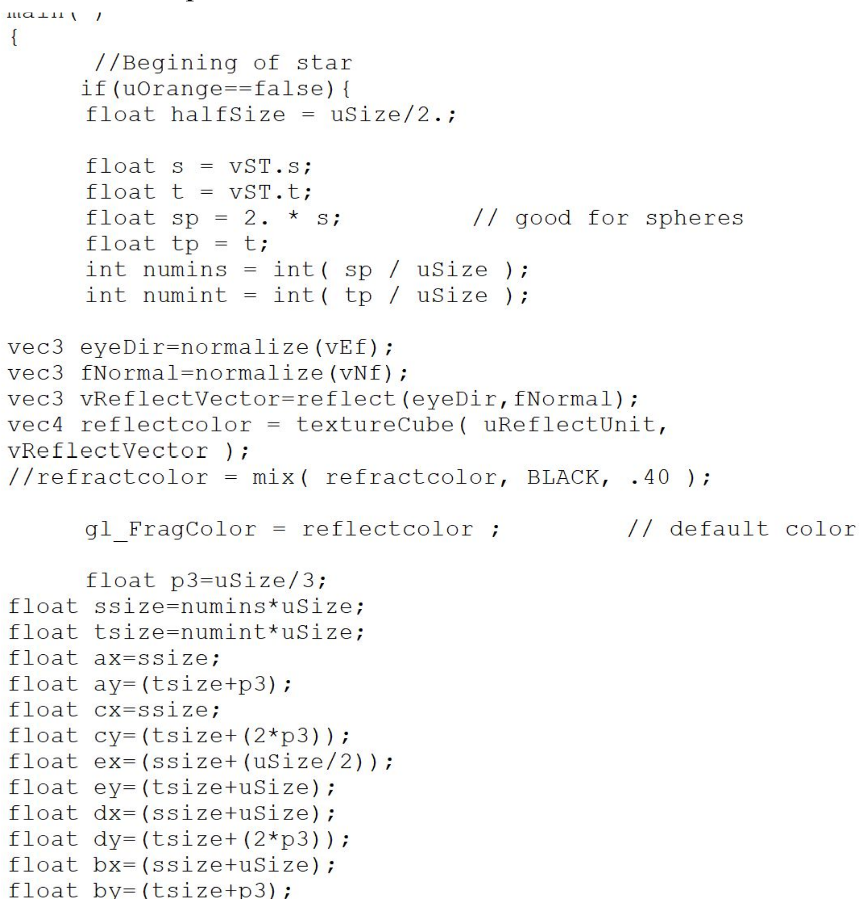
My final project contains two parts. Although both of them look simple, they are not so easy. The first one is star patterns. I created this pattern using the slope equation(y2-y1)/(x2-x1). I got this idea after knowing how to create elliptical dots by using the ellipse equation. It may look simple ,but it is not easy to create.



The size can be changed.



This is the code for the star pattern.

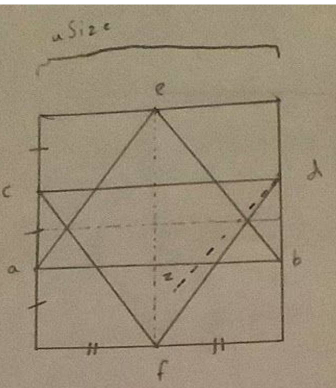


```
main( )
{
      //Begining of star
     if(uOrange==false){
     float halfSize = uSize/2.;

     float s = vST.s;
     float t = vST.t;
     float sp = 2. * s;         // good for spheres
     float tp = t;
     int numins = int( sp / uSize );
     int numint = int( tp / uSize );

vec3 eyeDir=normalize(vEf);
vec3 fNormal=normalize(vNf);
vec3 vReflectVector=reflect(eyeDir,fNormal);
vec4 reflectcolor = textureCube( uReflectUnit,
vReflectVector );
//refractcolor = mix( refractcolor, BLACK, .40 );

     gl_FragColor = reflectcolor ;        // default color

     float p3=uSize/3;
float ssize=numins*uSize;
float tsize=numint*uSize;
float ax=ssize;
float ay=(tsize+p3);
float cx=ssize;
float cy=(tsize+(2*p3));
float ex=(ssize+(uSize/2));
float ey=(tsize+uSize);
float dx=(ssize+uSize);
float dy=(tsize+(2*p3));
float bx=(ssize+uSize);
float by=(tsize+p3);
```

```
     float p3=uSize/3;
float ssize=numins*uSize;
float tsize=numint*uSize;
float ax=ssize;
float ay=(tsize+p3);
float cx=ssize;
float cy=(tsize+(2*p3));
float ex=(ssize+(uSize/2));
float ey=(tsize+uSize);
float dx=(ssize+uSize);
float dy=(tsize+(2*p3));
float bx=(ssize+uSize);
float by=(tsize+p3);
float fx=(ssize+(uSize/2));
float fy=tsize;

if(tp>(tsize+(2*p3))&& sp<(ssize+(uSize/2))){
float ms=((ey-ay)/(ex-ax));
float cs=((tp-ay)/(sp-ax));
if(cs<ms){
gl_FragColor=mix(BLACK,reflectcolor,0.3) ;
}           //1
}
else if(tp>(tsize+(2*p3)) && sp>(ssize+(uSize/2))){
float ms=((ey-by)/(ex-bx));
float cs=((tp-by)/(sp-bx));
if(cs>ms){
gl_FragColor=mix(BLACK,reflectcolor,0.3);
}          //2
}
else if(tp<(tsize+p3) && sp<(ssize+(uSize/2))){
float ms=((fy-cy)/(fx-cx));
float cs=((tp-cy)/(sp-cx));
if(cs>ms){
gl_FragColor=mix(BLACK,reflectcolor,0.3);
gl_FragColor=mix(BLACK,reflectcolor,0.3);
}          //2
}
else if(tp<(tsize+p3) && sp<(ssize+(uSize/2))){
float ms=((fy-cy)/(fx-cx));
float cs=((tp-cy)/(sp-cx));
if(cs>ms){
gl_FragColor=mix(BLACK,reflectcolor,0.3);
}        //3
}
else if(tp<(tsize+p3) && sp>(ssize+(uSize/2))){
float ms=((fy-dy)/(fx-dx));
float cs=((tp-dy)/(sp-dx));
if(cs<ms){
gl_FragColor =mix(BLACK,reflectcolor,0.3);
}          //4
}
else if(tp>(tsize+p3+(p3/2)) && tp<(tsize+(2*p3))&&
sp<(ssize+(uSize/2))){
float ms=((fy-cy)/(fx-cx));
float cs=((tp-cy)/(sp-cx));
if(cs>ms)
{gl_FragColor =mix(BLACK,reflectcolor,0.3);}         //5
}
else if(tp>(tsize+p3) && tp<(tsize+p3+(p3/2)) &&
sp<(ssize+(uSize/2))){
float ms=((ey-ay)/(ex-ax));
float cs=((tp-ay)/(sp-ax));
if(cs<ms)
{gl_FragColor =mix(BLACK,reflectcolor,0.3);}           //6
}
else if(tp>(tsize+p3+(p3/2)) && tp<(tsize+(p3*2)) && sp>
(ssize+(uSize/2))){
float ms=((fy-dy)/(fx-dx));
float cs=((tp-dy)/(sp-dx));
```

```
else if(tp>(tsize+p3) && tp<(tsize+p3+(p3/2)) &&
sp<(ssize+(uSize/2))){
float ms=((ey-ay)/(ex-ax));
float cs=((tp-ay)/(sp-ax));
if(cs<ms)
{gl_FragColor =mix(BLACK,reflectcolor,0.3);}           //6
}
else if(tp>(tsize+p3+(p3/2)) && tp<(tsize+(p3*2)) && sp>
(ssize+(uSize/2))){
float ms=((fy-dy)/(fx-dx));
float cs=((tp-dy)/(sp-dx));
if(cs<ms){
gl_FragColor =mix(BLACK,reflectcolor,0.3);
}        //7
}
else if(tp>(tsize+p3) && tp<(tsize+p3+(p3/2)) && sp>
(ssize+(uSize/2))){
float ms=((ey-by)/(ex-bx));
float cs=((tp-by)/(sp-bx));
if(cs>ms){
gl_FragColor =mix(BLACK,reflectcolor,0.3);
}          //8
}



     gl_FragColor.rgb *= vLightIntensity;  // apply lighting
model
}         //end of star
```

This is the idea for how to create the star pattern.



slope equation $= \dfrac{y_2 - y_1}{x_2 - x_1}$

Assume $d\ (0.9, 0.9)$

$f\ (0.4, 0.4)$

$z\ (0.5, 0.6)$

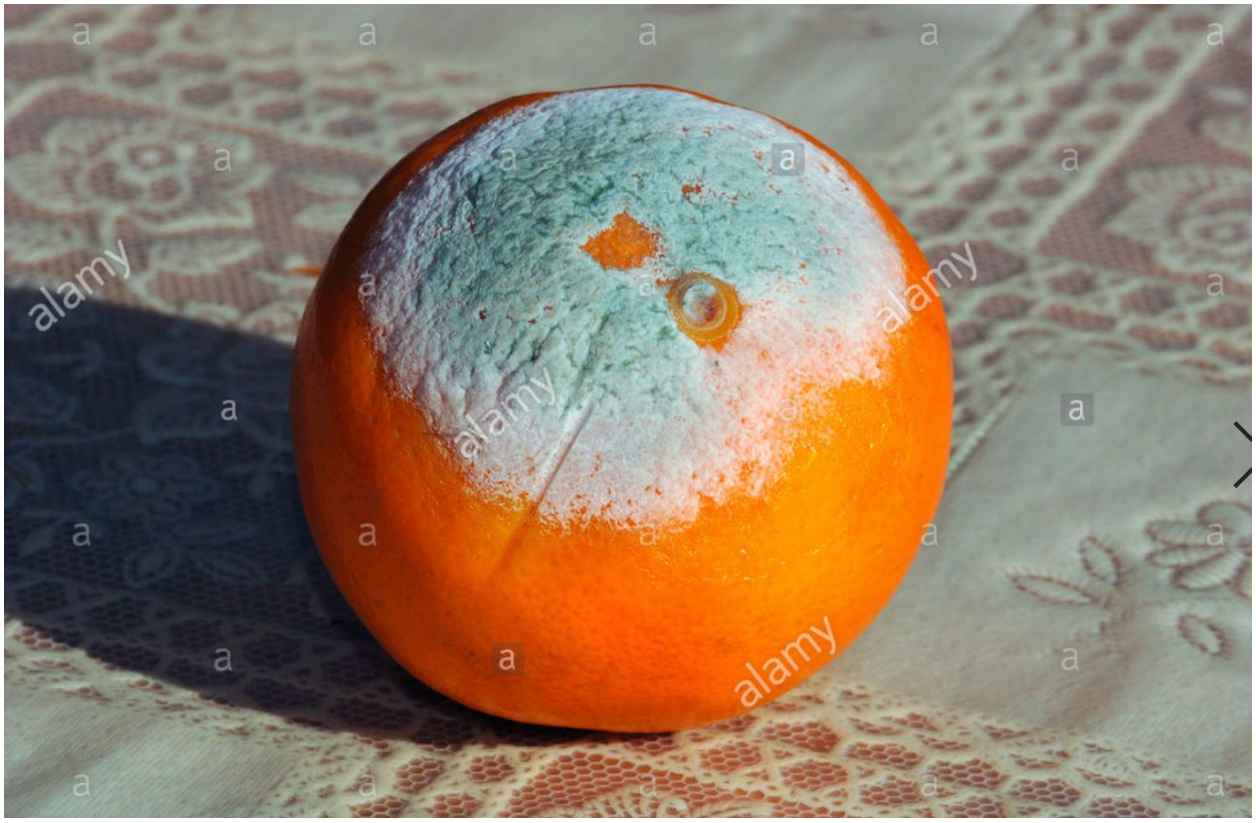
s_{fd} = slope of line $fd = \dfrac{0.4 - 0.9}{0.4 - 0.9} = \dfrac{0.5}{0.5} = 1$

s_{zd} = slope of line $zd = \dfrac{0.6 - 0.9}{0.5 - 0.9} = \dfrac{0.3}{0.4} = 0.75$

$s_{fd} > s_{zd}$

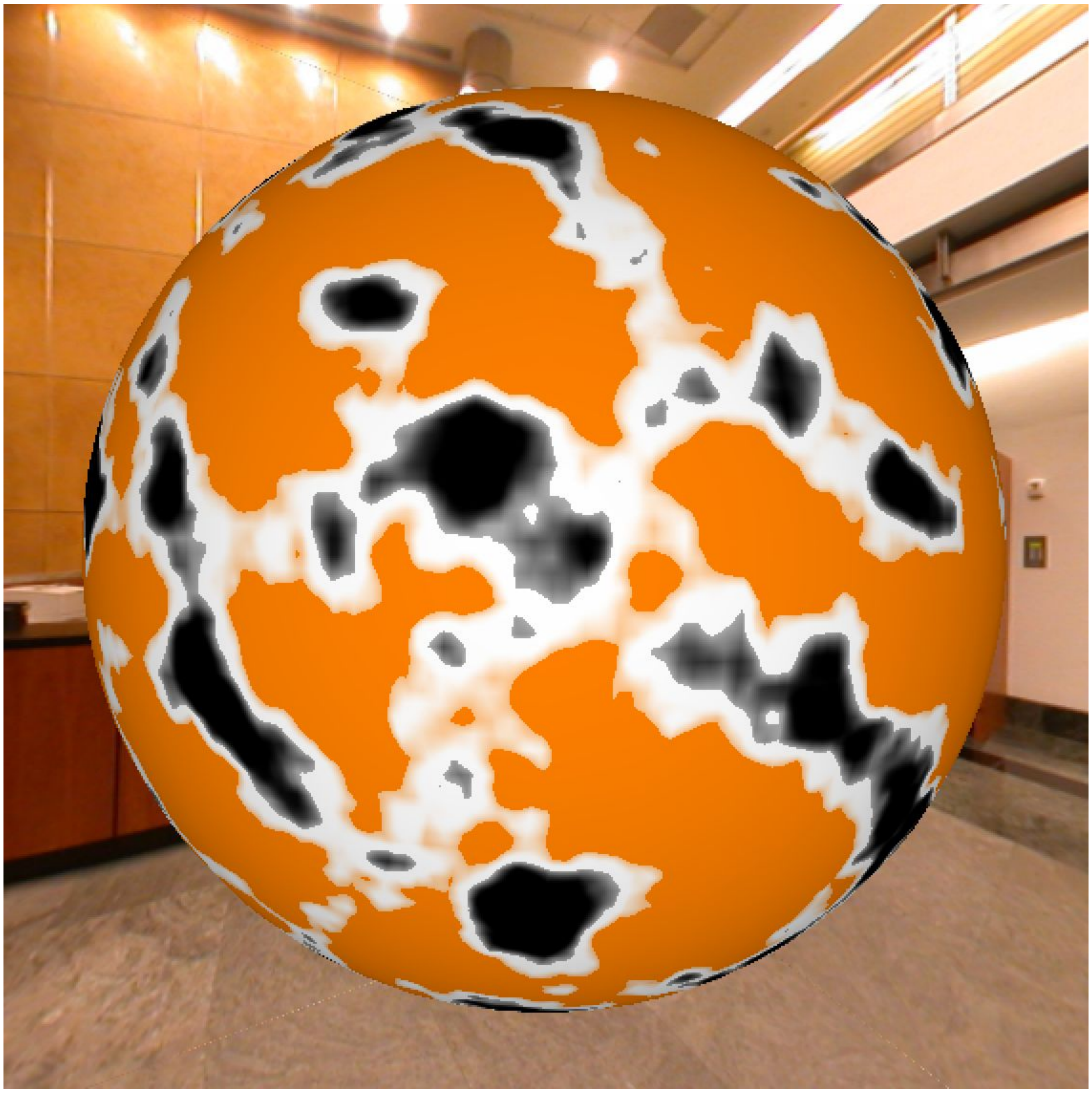
Line zd exists in the clockwise direction of line fd.

The second part is creating fungus patterns. The one I try to create is like the one on orange. They are like this.

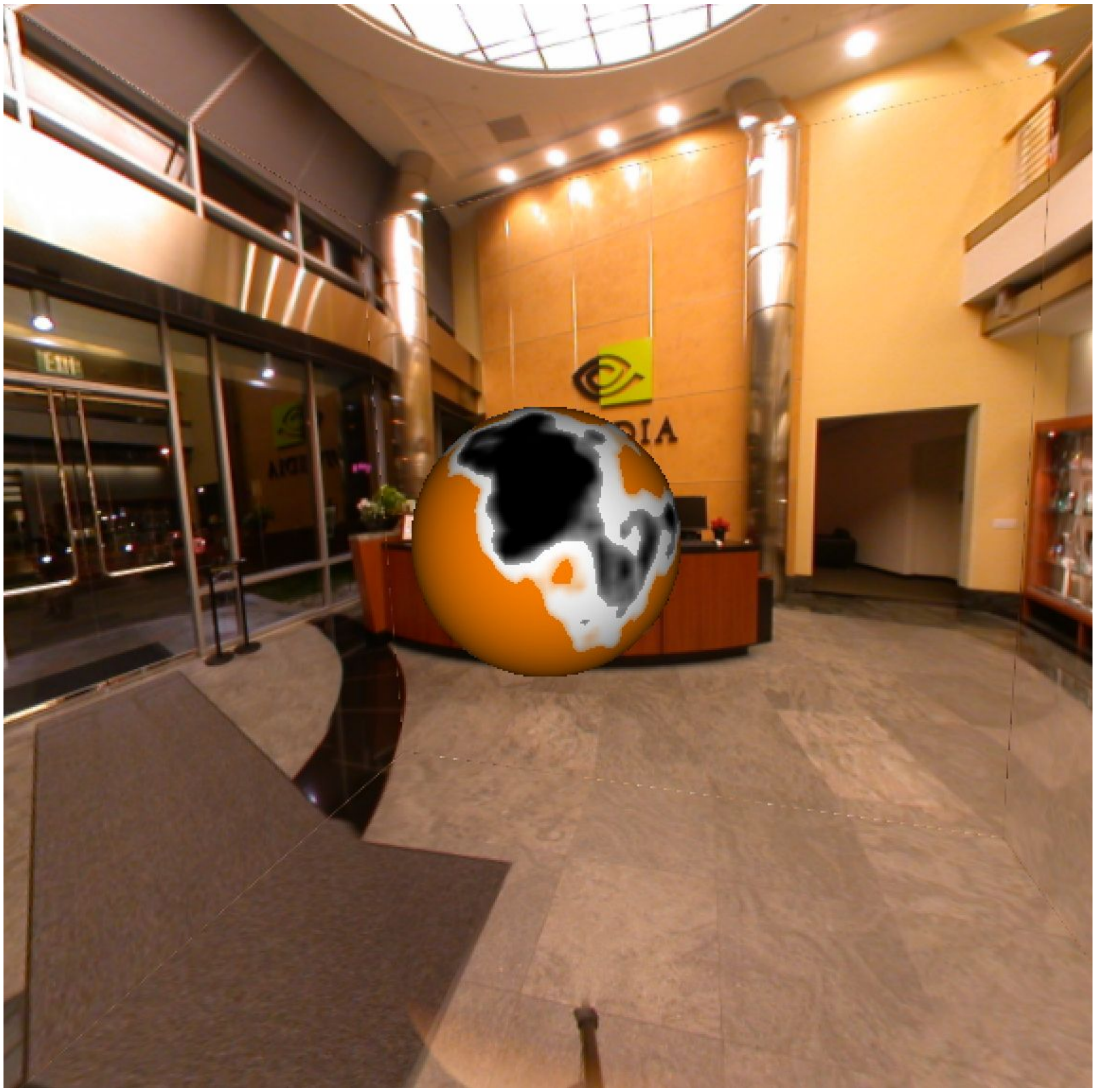


If the pattern is looked at carefully, it has two different colors, the inner one and the outer one. Since there are two colors, there must be two boundaries and the boundaries should not have sharp changes. They should blend with the color of the neighbours.

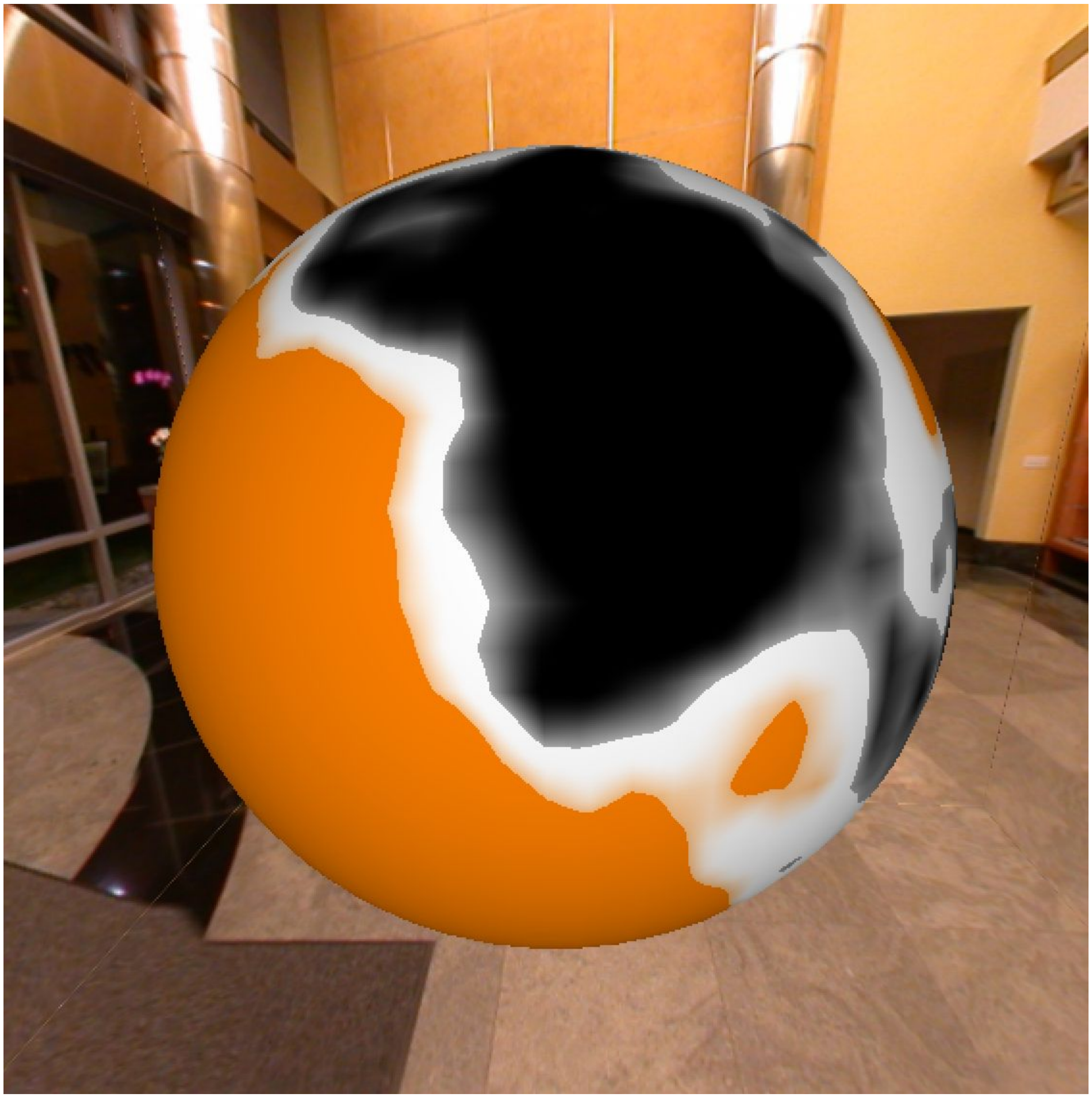
I got these patterns.



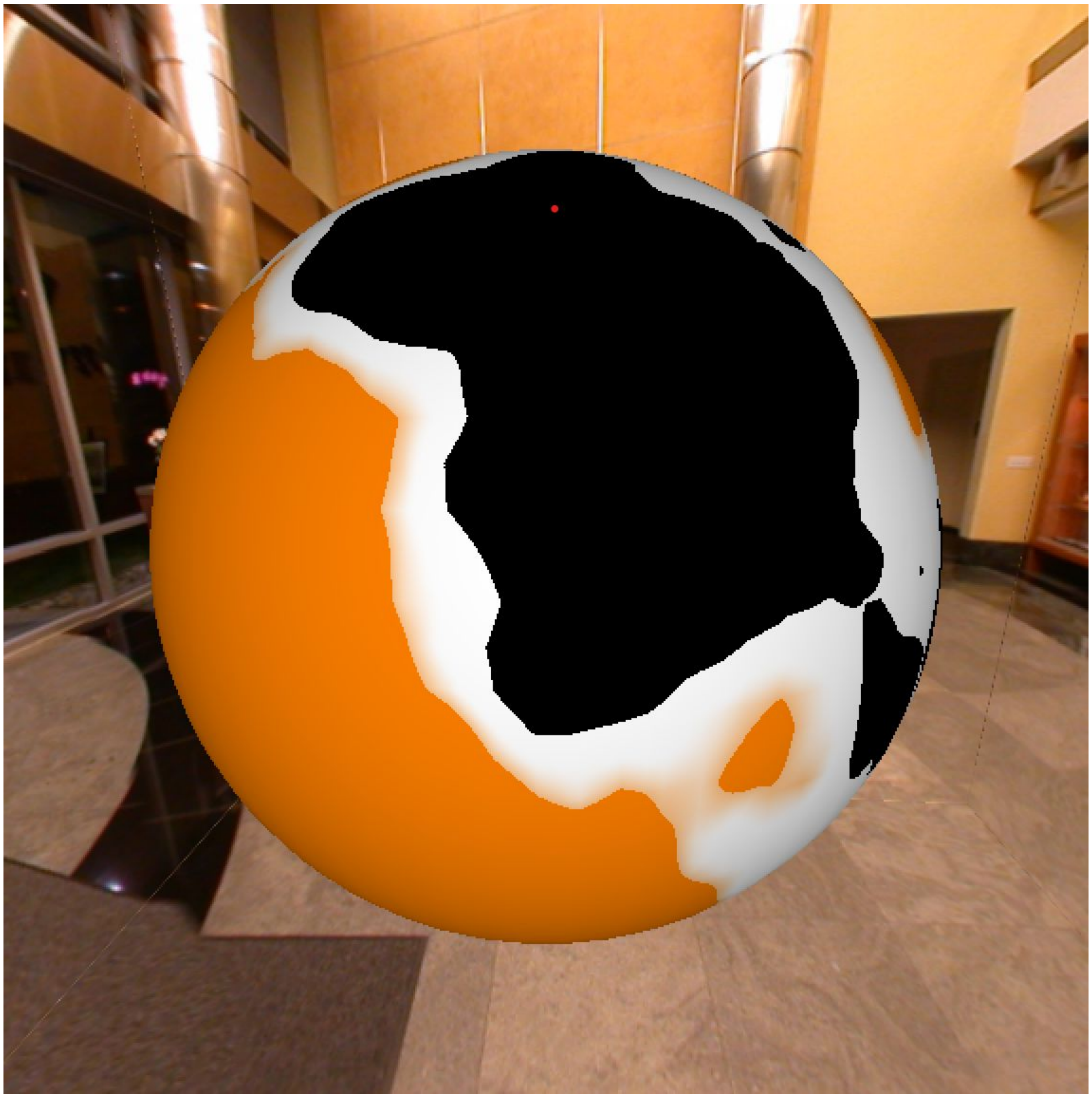
The fungus size can also be changed.



The level of the boundaries blending with neighbours can be adjusted.(Although it should be constant, I use the uniform variables in the place of the constants so that I can show how the changes look like.)
This is the pattern before making any change.



This is the result after assigning 0.0 to the blending level of the inner boundary.

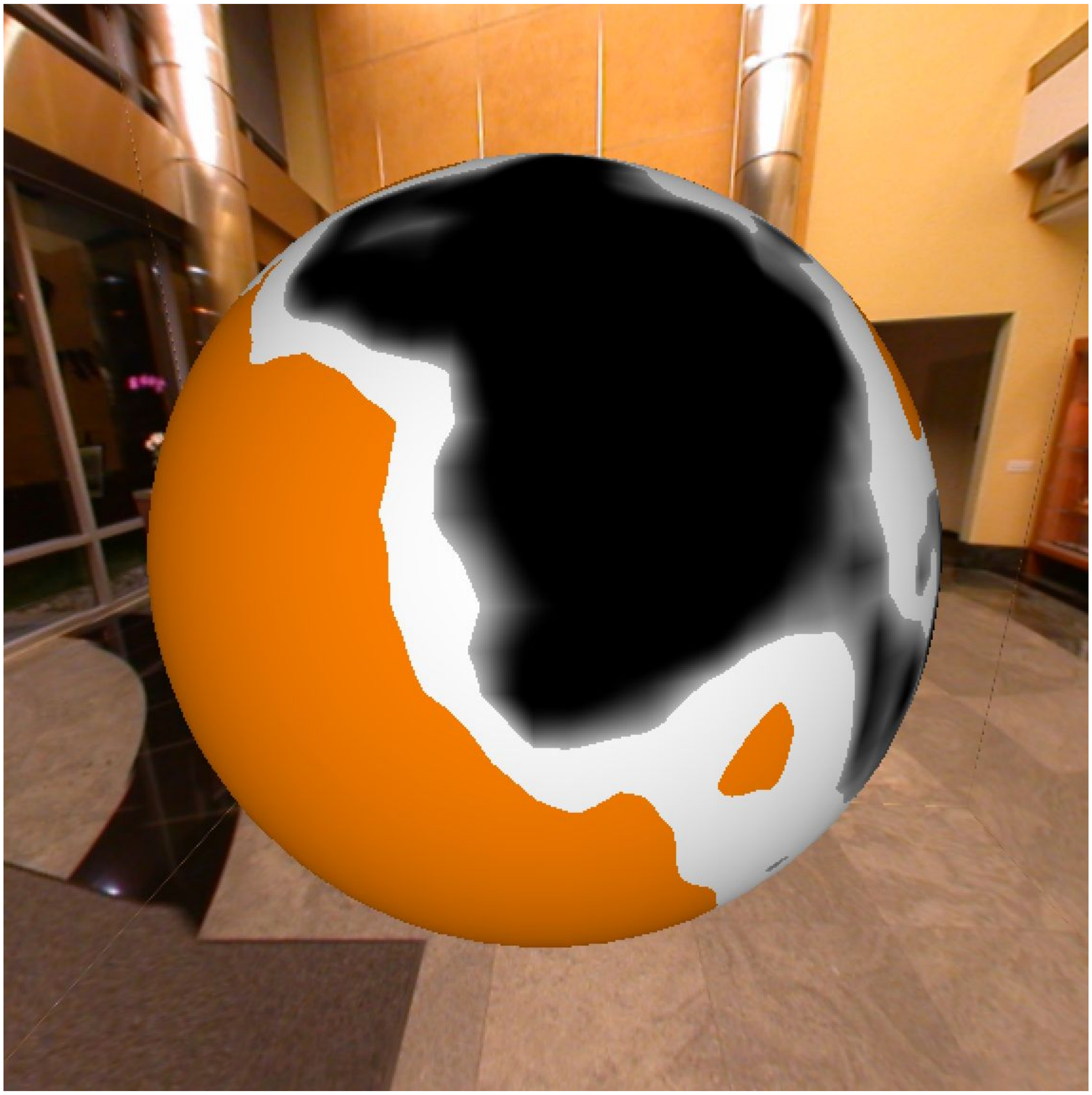


The inner boundary has sharp change now.

The corresponding code for the blending level of the inner boundary is

```
      if(n>uFungus){
rcolor=vec4(1.,1.,1.,1.);
t= smoothstep( (uFungus+0.06)-uTol, (uFungus+0.06)+uTol, n );
fc=mix(rcolor,vec4(0.,0.,0.,1.),t);
```

This is the result after assigning 0.0 to the blending level of the outer boundary.

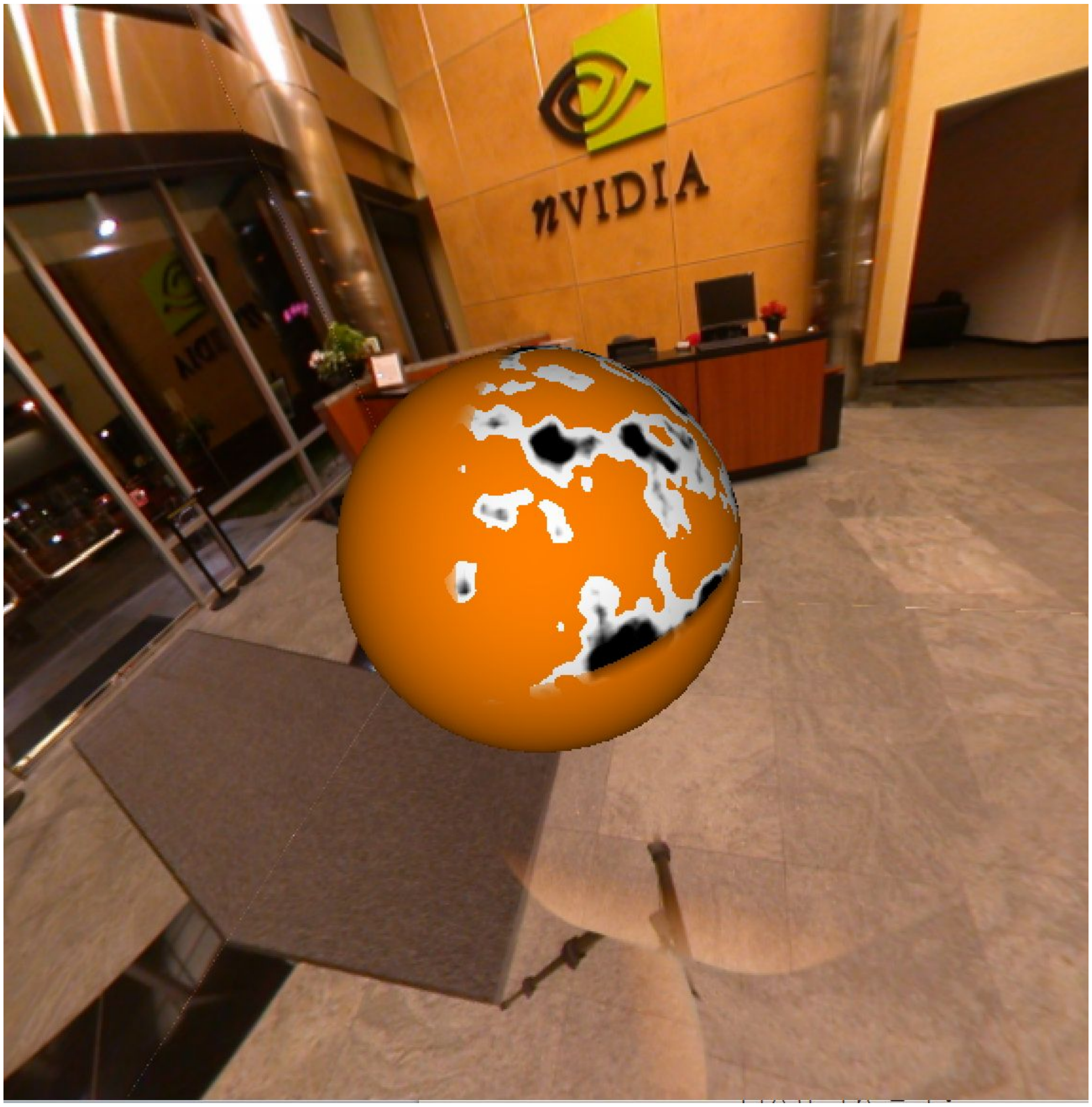


The outer boundary is with the sharp change.

The corresponding code for the blending level of the outer boundary is

```
if((uFungus-0.08)<n && ((uFungus+0.05)>n))
{float a=smoothstep(uFungus-uTol1,uFungus+uTol1,n);
fc=mix(orange,vec4(1.,1.,1.,1.),a);

}
```

The orange can have the fungus only on the part of the surface. Therefore, I created and included the function "uPart". When the function is on, the result will look like this.

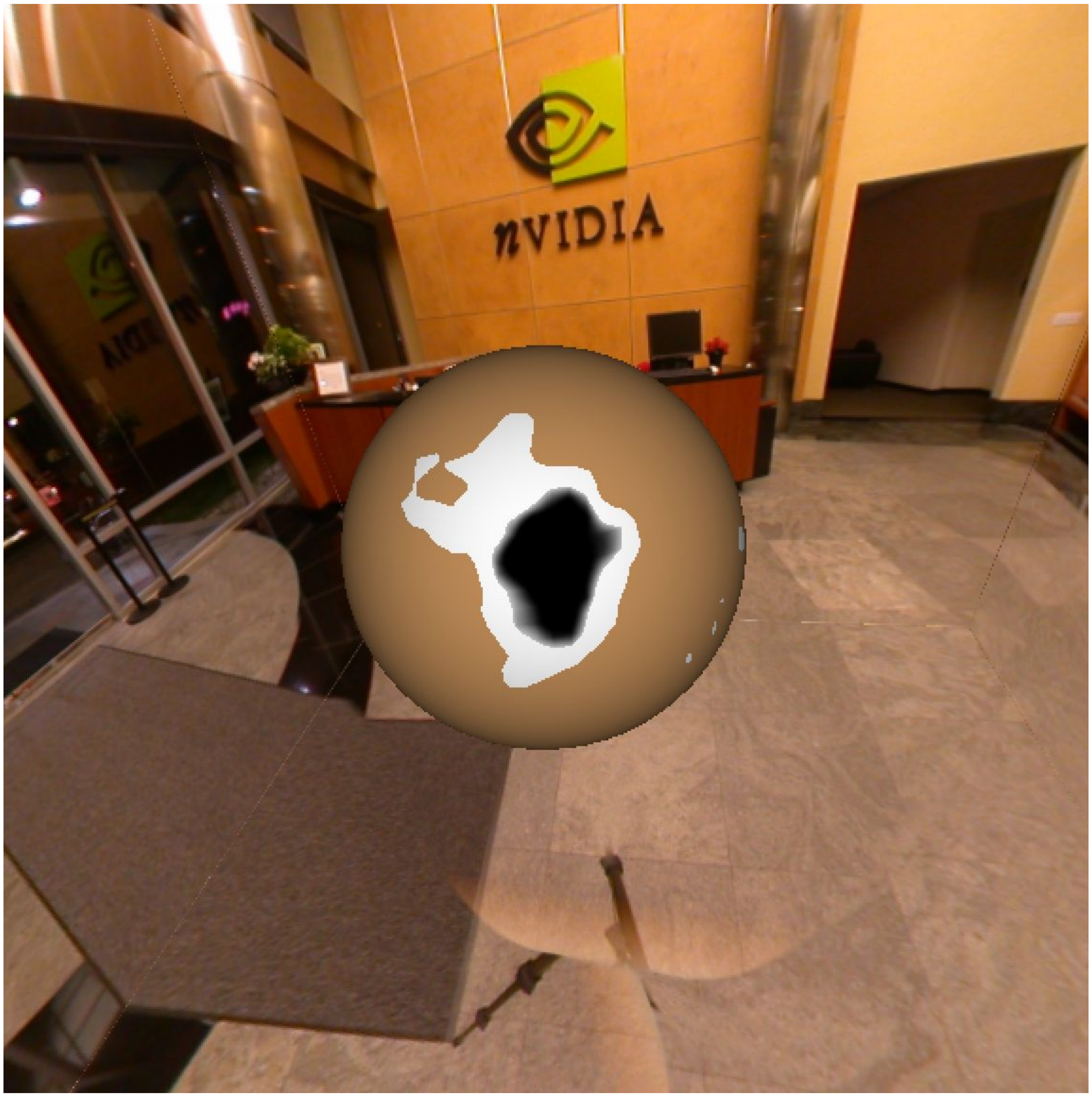


It is noticeable that the boundary of the part on which the fungi exists also blends with the neighbours. I use the x and y coordinates in the smoothstep functions.

The corresponding code for the fungi being on the specific part of the surface is

```
if(uPart==true){
     if(n>uFungus && vMCposition.x>.0 && vMCposition.y>.0 ){
rcolor=vec4(1.,1.,1.,1.);
t= smoothstep( (uFungus+0.06)-uTol, (uFungus+0.06)+uTol, n );
fc=mix(rcolor,vec4(0.,0.,0.,1.),t);

if((uFungus-0.08)<n && ((uFungus+0.03)>n))
{float a=smoothstep(uFungus-uTol1,uFungus+uTol1,n);
fc=mix(orange,vec4(1.,1.,1.,1.),a);

}

if(vMCposition.y<0.1 && vMCposition.y>0.){
float b=smoothstep(0.,0.2,vMCposition.y);
fc=mix(orange,fc,b);

}
if(vMCposition.x<0.1 && vMCposition.x>0.){
float b=smoothstep(0.,0.2,vMCposition.x);
fc=mix(orange,fc,b);

}

gl_FragColor=fc;
}
}
```

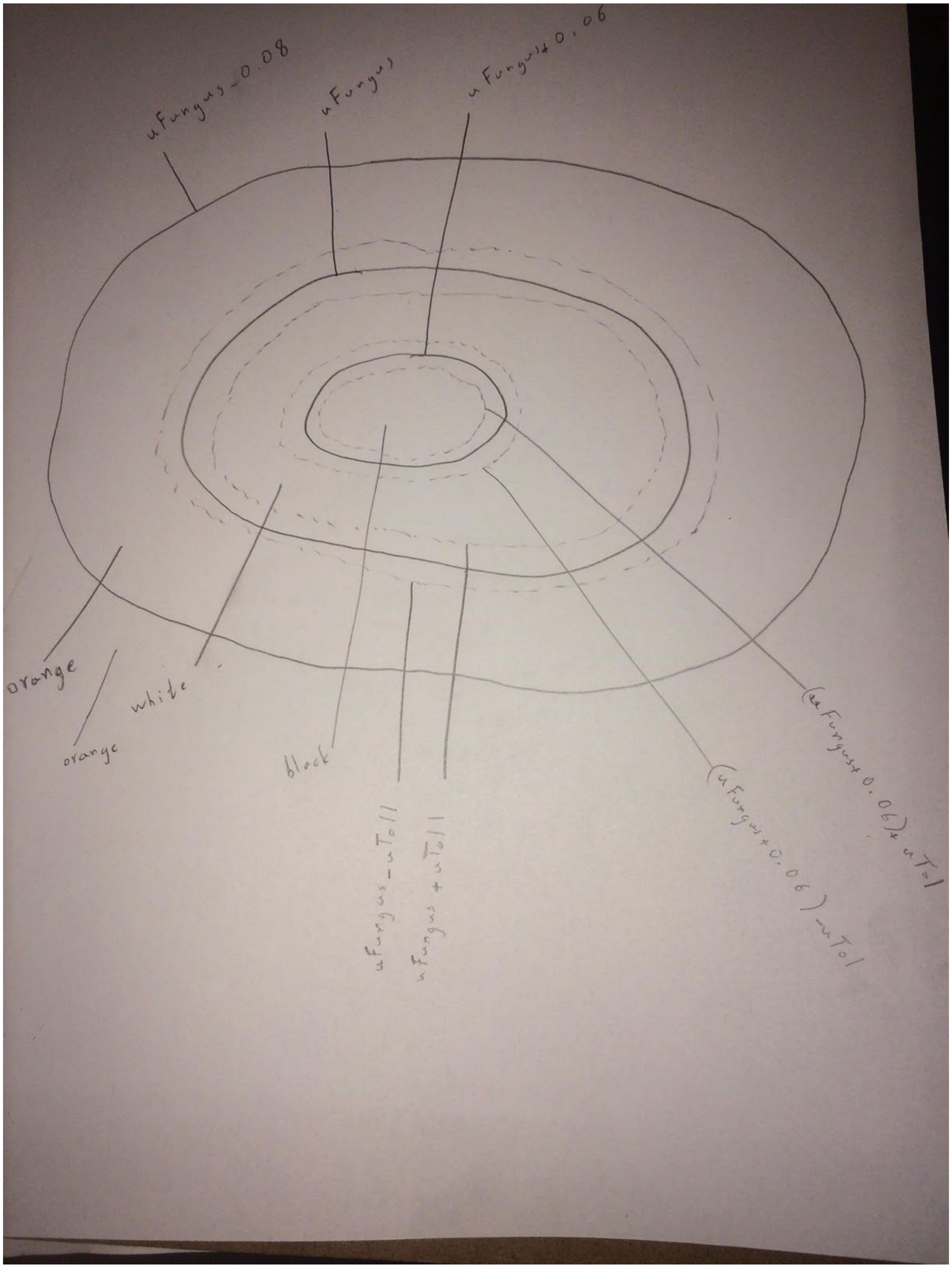
In real life, the more fungi an orange has, the darker the color of the rest surface of the orange becomes.



The corresponding code is

```
vec4 nv = texture3D( Noise3, uFungussize * vMCposition );
     float n = nv.r + nv.g + nv.b + nv.a;//range is 1. -> 3.
float rat=(2.2-uFungus)/0.2;
vec4 rcolor;
vec4 fc;
float t=0.;
rcolor=vec4(1.,1.,1.,1.);
if(uR==true){
gl_FragColor=mix(orange,vec4(0.6,0.6,0.6,1.),rat);
}
else
{gl_FragColor=orange;}
```

This is the idea for how to create the fungus pattern.


uFungus_0.08
uFungus
uFungus+0.06
orange
orange
white
black
uFungus - uToll
uFungus + uToll
(uFungus+0.06) - uTol
(uFungus+0.06)+uTol