

リスト 9.5

film.frag

```
varying vec3 P;
varying vec3 N;
uniform float refract;//相対屈折率
uniform float filmSpace ;//nano meter
uniform float blendFactor;

const float ultraviolet = 380.0;
const float infrared = 780.0;

vec3 getRGB(float lambda)
{
    //中略(リスト9.4に同じ)
}

void main(void)
{
    vec3 L = normalize(gl_LightSource[0].position.xyz - P);
    N = normalize(N);

    vec4 ambient = gl_FrontLightProduct[0].ambient;
    float dotNL = dot(N, L);
    vec3 V = normalize(-P);
    float dotNV = dot(N, V);
    vec3 H = normalize(L + V);
    float dotNH = max(dot(N, H), 0.0);
    vec4 diffuse = gl_FrontLightProduct[0].diffuse * max(0.0, dotNL);

    float powNH = pow(dotNH, gl_FrontMaterial.shininess);
    if(dotNL <= 0.0) powNH = 0.0;
    vec4 specular = gl_FrontLightProduct[0].specular * powNH;

    //干渉による色の計算
    float cosThetaR = sqrt(1.0 - (1.0 - dotNV*dotNV)/(refract * refract));
    float a = 2.0 * filmSpace * cosThetaR * refract;
    vec4 colorInter = vec4(0.0);
    for(int i = 0; i <= 5; i++)
    {
        float lambda = a / (float(i) + 0.5);
        if(lambda < ultraviolet || lambda > infrared) continue;
        colorInter.rgb += getRGB(lambda);
    }

    //統合
    if( colorInter.rgb == vec3(0.0) )
        gl_FragColor = ambient + diffuse + specular;
    else
        gl_FragColor = (ambient + diffuse + specular) * mix(vec4(1.0), colorInter,
blendFactor);
    gl_FragColor.a = 1.0;
}
```