

## リスト 2.8

## spotlight.frag

```
varying vec3 P;  
varying vec3 N;  
int numLight = 2;  
  
void main(void)  
{  
    gl_FragColor = vec4(0.0);  
    float attenuation = 1.0;  
    for(int i = 0; i < numLight; i++)  
    {  
        vec3 L = gl_LightSource[i].position.xyz - P; //光源ベクトル  
        if(gl_LightSource[i].spotCutoff == 180.0) //通常の光源  
        {  
            float d = length(L); //光源までの距離  
            //減衰計数  
            attenuation = 1.0 / (gl_LightSource[i].constantAttenuation  
                                + gl_LightSource[i].linearAttenuation * d  
                                + gl_LightSource[i].quadraticAttenuation * d * d);  
        }  
  
        L = normalize(L);  
        if(gl_LightSource[i].spotCutoff <= 90.0) //スポットライト  
        {  
            float LS = dot(-L, normalize(gl_LightSource[i].spotDirection));  
            if(LS < gl_LightSource[i].spotCosCutoff)  
                attenuation = 0.0; //スポットライトの外側  
            else  
                attenuation = pow(LS, gl_LightSource[i].spotExponent);  
        }  
  
        N = normalize(N);  
        vec4 ambient = gl_FrontLightProduct[i].ambient;  
        float dotNL = dot(N, L);  
        vec4 diffuse = gl_FrontLightProduct[i].diffuse * max(0.0, dotNL);  
        vec3 V = normalize(-P);  
        vec3 H = normalize(L + V);  
        float powNH = pow(max(dot(N, H), 0.0), gl_FrontMaterial.shininess);  
        if(dotNL <= 0.0) powNH = 0.0;  
        vec4 specular = gl_FrontLightProduct[i].specular * powNH;  
        vec4 col = ambient + diffuse + specular;  
        col *= attenuation;  
        gl_FragColor += col;  
    }  
}
```