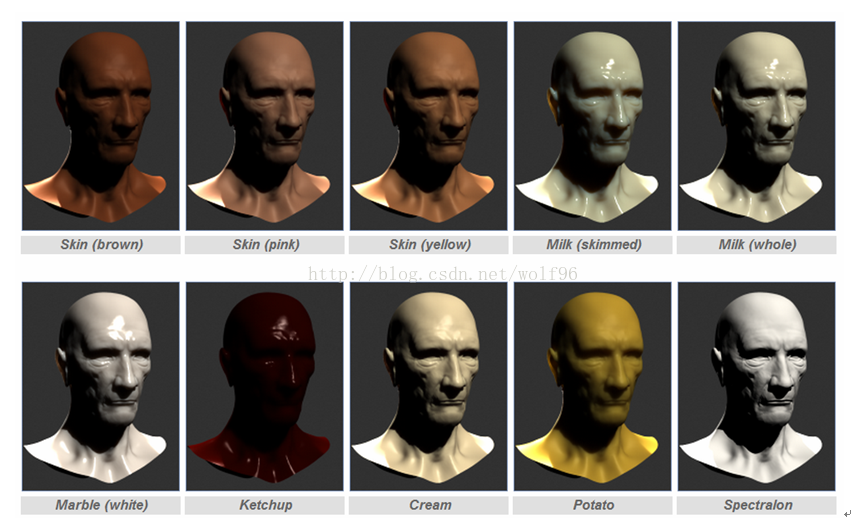
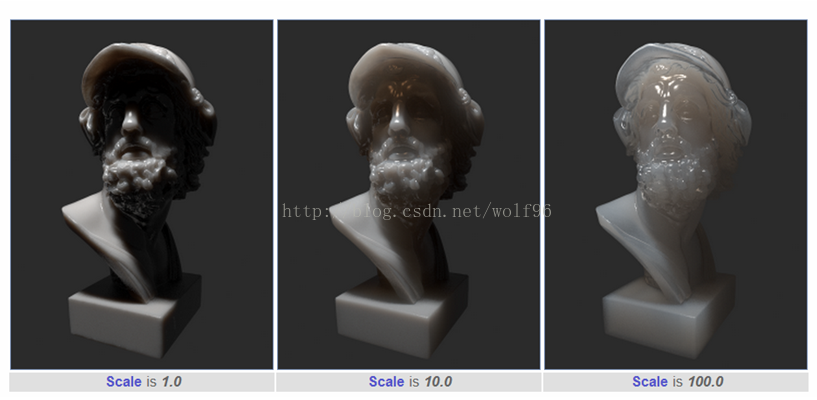
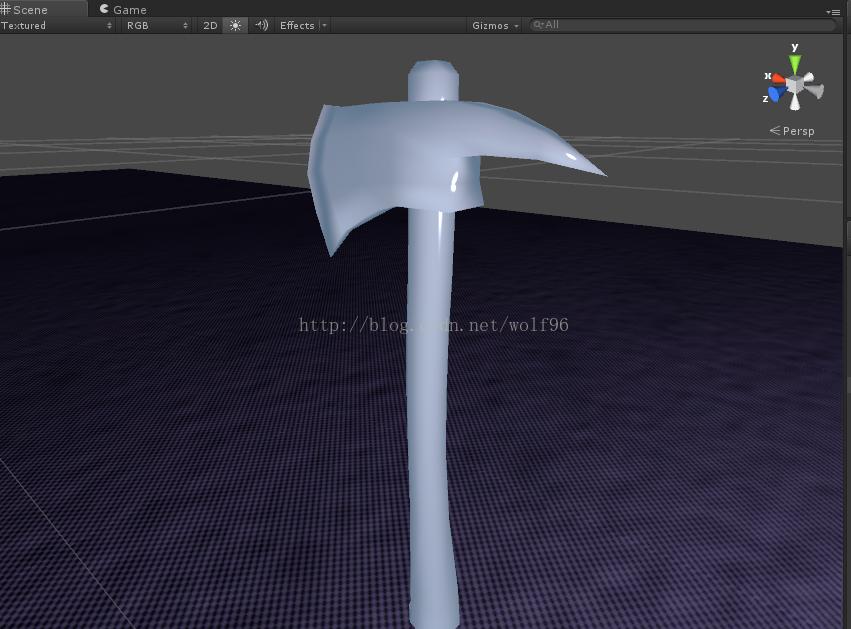
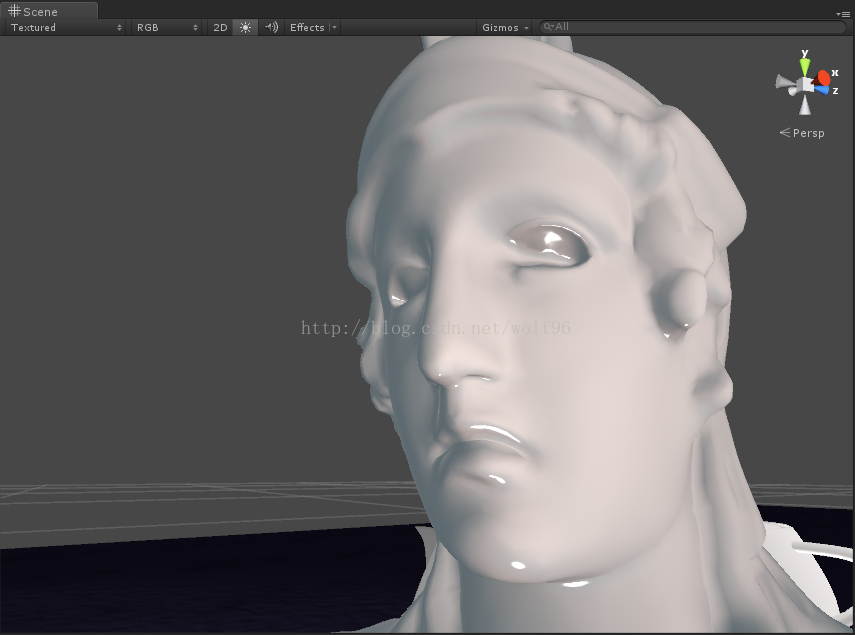
Unity3d shader之次表面散射（Subsurface Scattering）

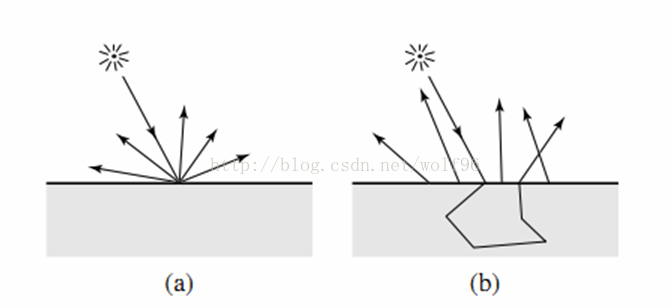
次表面散射是一种非常常用的效果，可以用在很多材质上  
如皮肤，牛奶，奶油奶酪，番茄酱，土豆等等  
  
初衷是想做一个牛奶shader的，但后来就干脆研究了sss  
这是在vray上的次表面散射效果



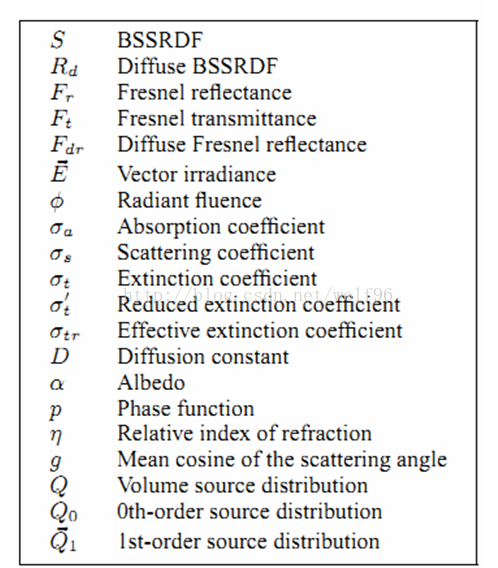
这是本文在unity中实现了的次表面散射效果：







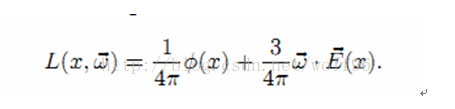
左侧为BRDF（双向反射分布），右侧为BSSRDF（双向次表面散射反射分布）



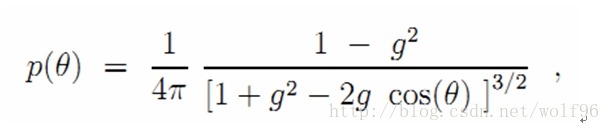
各参数如上为：  
S:BSSRDF结果  
Rd: BSSRDF的漫反射  
Fr:: 菲涅尔反射  
Ft: 菲涅尔透明度，透射比  
Fdr: 菲涅尔漫反射的反射率  
E:辐照方向  
Wiki 中给出：radiant fluence is the radiant energy received by a surface per unit area, or equivalently the irradiance of a surface integrated over time of irradiation

（辐射通量是每单位面积表面所接受的辐射能量，或者等效为随着照射时间积分的表面的辐照度。）  
Phi:每单位表面受到的辐照能  
Sigma A: 吸收率  
Sigma S: 散射率  
Sigma T: 消散率  
Sigma T’ : 减少消散率  
Sigma TR : 有效消散率  
D:漫反射常量  
Alpha: 反射率  
P:相函数  
Eta:反射的相关指数  
g:散射角的平均cos值  
Q:源分布值  
Q0:第0个源分布  
Q1:第1个源分布  


**漫反射近似**  
漫反射近似是基于光线高分散媒介倾向于各向异性的观察，光源的分布与相函数是各向异性的。每次散射都模糊了光线的分布，随着散射的次数增多导致光线的分布更加均匀。  
这种辐照类似于一个二项式涉及单位表面受到的辐照能和辐照方向





使用了Henyey-Greenstein的相函数：  
 

  
常量决定于单位表面受到的辐照能和辐照方向。  
对于一个无穷小的光线进入了一个媒介，入射能量将随着进入深度s呈指数性减小  
减小强度：

  float Lri(float3 w\_P, float phi\_x, float p\_L\_Dist, float D)

{

float \_Sigma\_t = \_Sigma\_A + \_Sigma\_S;

float L = 1 / (4 \* PIE) \* phi\_x + 3 / (4 \* PIE) \* dot(w\_P, -D\*\_Nabla \* phi\_x);

float Lri = L \* pow(E, -\_Sigma\_t\* p\_L\_Dist);

return Lri;

}

第一次散射减小强度，被作为体积来源处理

for (int i = 0; i < 30; i++)

{

w\_P = normalize(float3(N.x + rand(fixed2(i\*0.05, i\*0.05)), N.y + rand(fixed2(-i\*0.05, i\*0.05)), N.z + rand(fixed2(i\*0.05, -i\*0.05))));

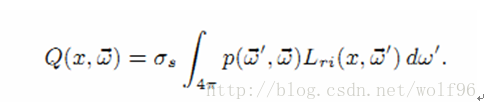
// float3 w\_P = normalize(float3(lightDir.x + rand(i.uv\_MainTex + fixed2(i\*0.01, i\*0.01)), lightDir.y + rand(i.uv\_MainTex + fixed2(-i\*0.01, i\*0.01)), lightDir.z + rand(i.uv\_MainTex + fixed2(i\*0.01, -i\*0.01))));

Q += phase(dot(lightDir, w\_P))\*Lri(w\_P, phi\_x, p\_L\_Dist, D);

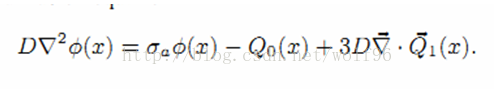
Q \*= \_Sigma\_S;

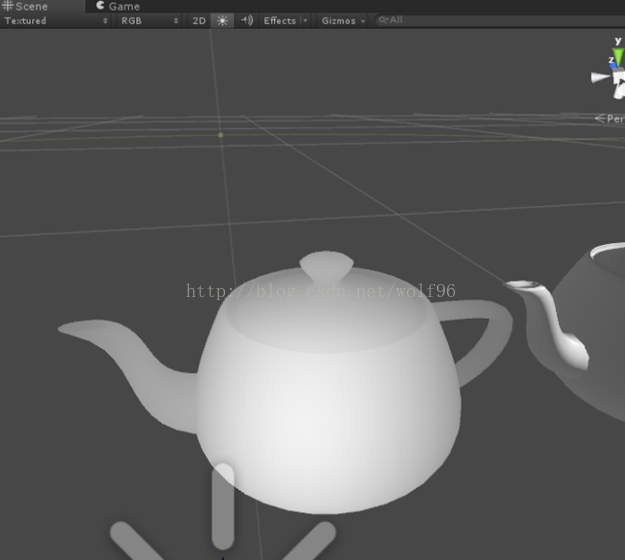
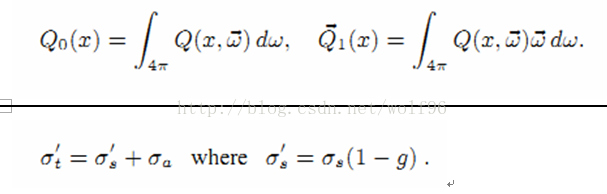
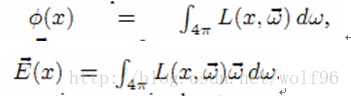
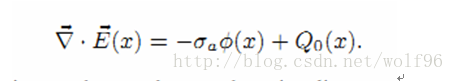
Q1 += Q\*w\_P;

}

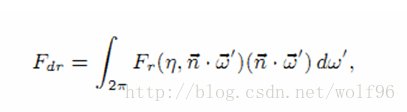
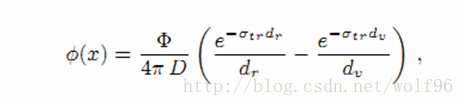


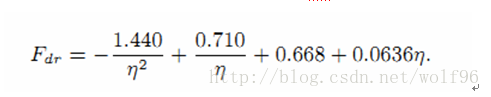
30次随机光线散射方向

观察光在体积内部传播行为，这个方程式很有用  
   
这个方程式与辐照度标量或通量相关  
   
第0个与第一个源分布公式  
   
Sigma参数之间的互相推倒，  
光线变成各向异性的，后向散射关系改变了净通量，前向散射与无散射是没有区别的。  
此处D = 1/(3\* sigma\_T’);是漫反射常量  
最终我们得到了漫反射公式  
   
漫反射部分的推导公式，得到如下结果  
  
   
在做定积分时进行叠加了30次随机光线散射方向，效果还算不错。



**漫反射的反射部分**

然后就是求漫反射的反射部分  
菲涅尔反射公式，在可传导介质的菲涅耳漫反射的反射：  
   
媒介本身的性质不同反射器情况也不同，Eta为这种性质的相关指数  
   
这是经过精确测量的反射率，我们可以用这个公式来免去计算消耗  
  
通量公式：  
  
   
Dr = ||x - xr||为当前点与光源的距离  
Dv = ||x-xv||为当前点与眼睛（相机）的距离



if (\_WorldSpaceLightPos0.w != 0)

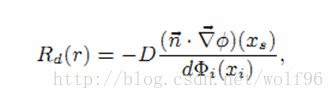
{

p\_L\_Dist = distance(\_WorldSpaceLightPos0, i.worldPos);

}

float v\_C\_Dist = distance(\_WorldSpaceCameraPos, i.worldPos)\*0.3;

Φ为光源强度  
最终，我们的反射公式为

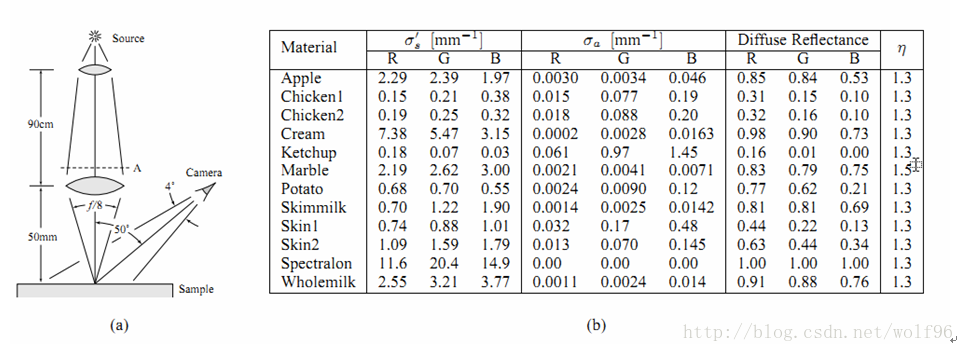


float3 ref = -D \* (dot(N, \_Nabla\*phi\_x\_S)) / (diff\*\_LumPow\_D);

在最后加入Physically-Based Rendering的specular，大功告成

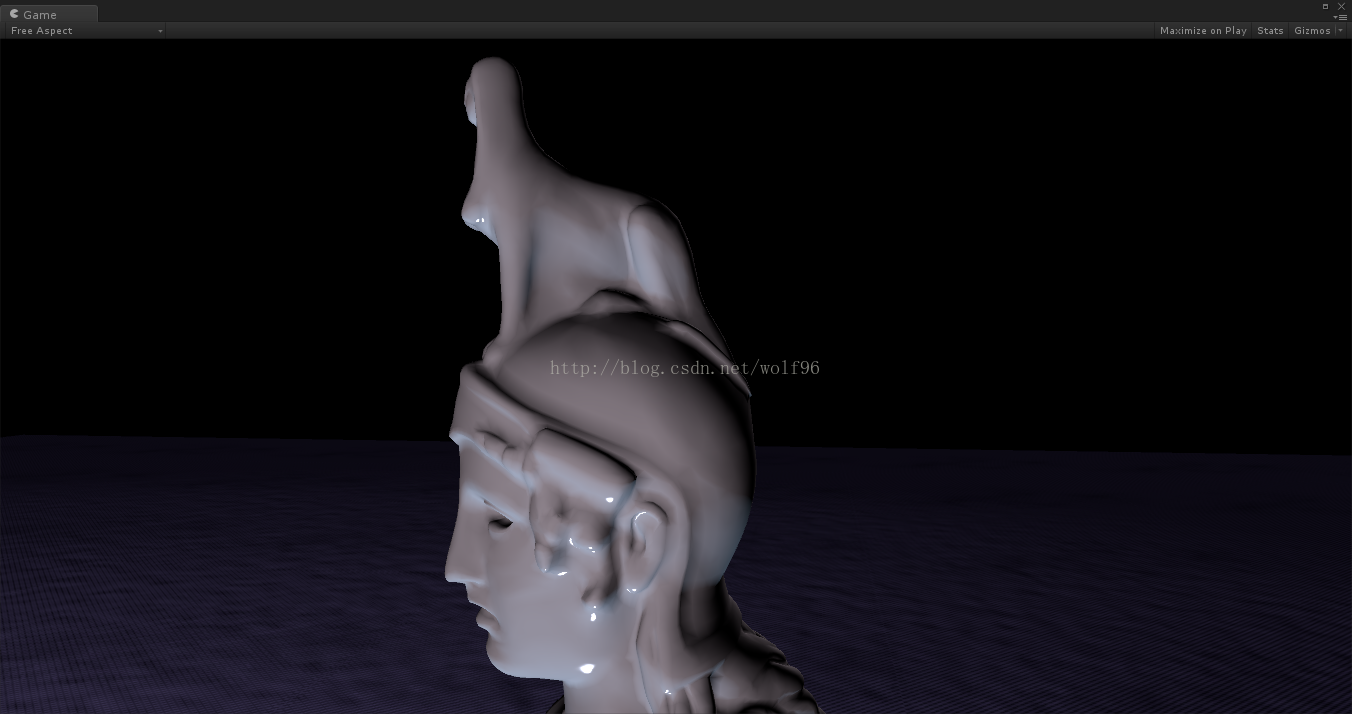
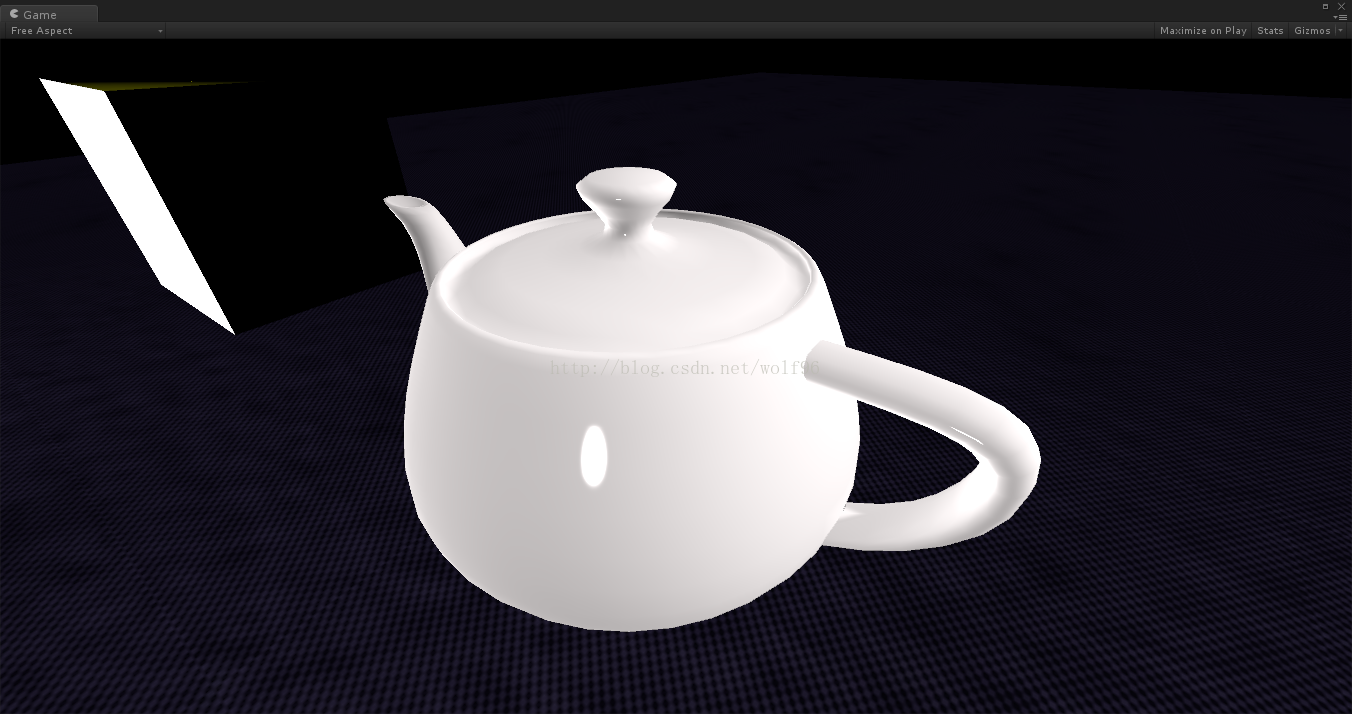
**参数调节**

关于参数调节，参数非常不好调，pdf上和自己弄得参数不搭，只能自己调了





最终效果：



参考：

1.  A Practical Model for Subsurface Light Transport  
2.  A Measurement-Based Skin Reﬂectance Model for Face Rendering and Editing