Lab4\_ Part1

float3 viewDir = normalize(\_WorldSpaceCameraPos - input.positionWS);

float3 halfVector = normalize(viewDir + lightDir);

//what I did

float a = \_rough \* \_rough;

float a22 = \_rough22 \* \_rough22;

float3 blue = { 0,0,1 };

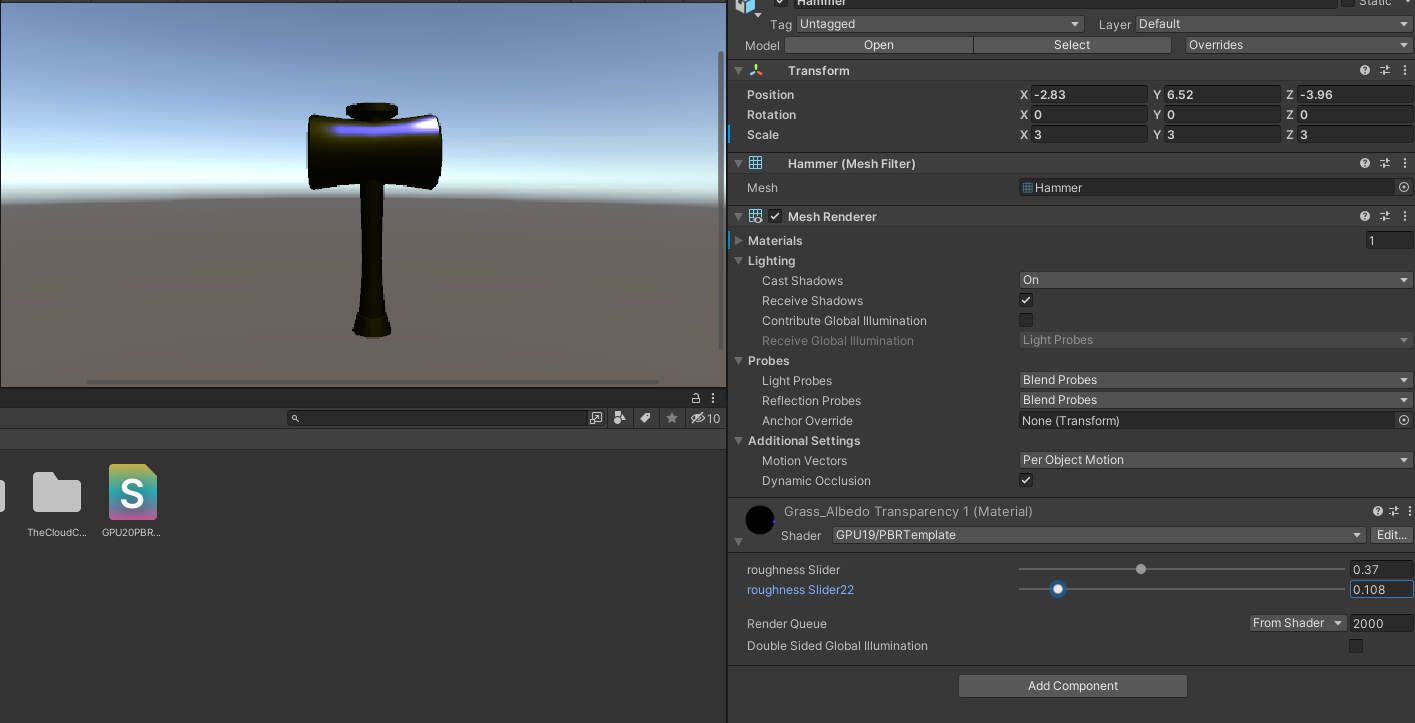
float3 yellow = { 1, 1, 0 };

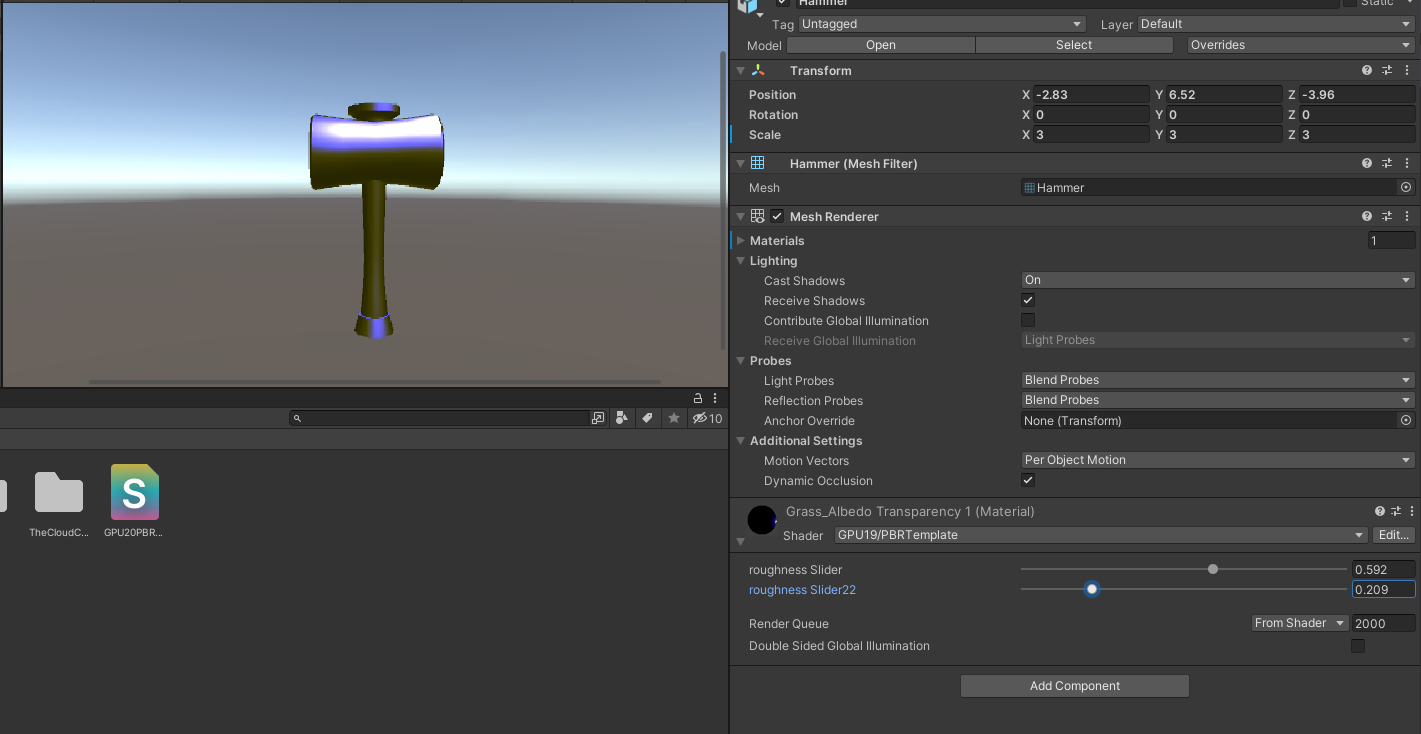
float3 beck = (1 / (3.14 \* pow(a, 2) \* pow(dot(n, halfVector), 4))) \* exp((pow(dot(n, halfVector), 2) - 1) / (pow(a, 2) \* pow(dot(n,halfVector), 2)));

float3 gcx = pow(a22, 2) / (3.14 \* pow((pow(dot(n, halfVector), 2) \* (pow(a22, 2) - 1) + 1), 2));

float3 output = (beck \* blue + gcx \* yellow) \* \_LightColor0.rgb;

//what I did

return(float4(output,1));



Lab4 part 2

clear; close all; clc;

rough = 0.8;

rough22 = 0.19;

a = rough \* rough;

a22 = rough22 \* rough22;

x = [0:0.01:10];

[row, column] = size(x);

beck = zeros(row, column);

gcx = zeros(row, column);

for i = 1 : column

r = rand(1);

beck(i) = 1/(pi \* a.^2\*(r.^4)) \* exp((r.^2 - 1)/ (a.^2 \* r.^2));

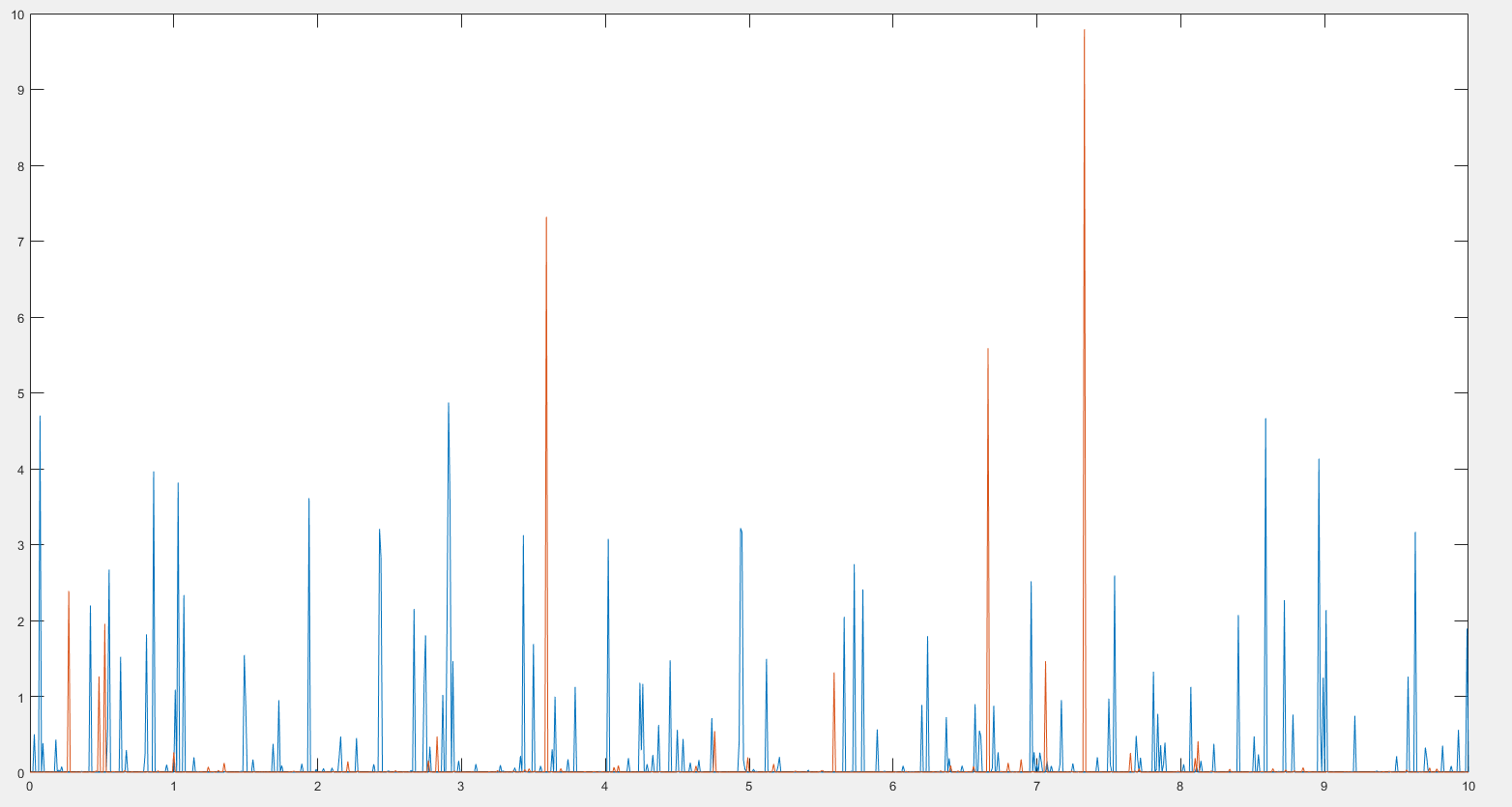
gcx(i) = (a22.^2)/ (pi \* (r.^2) \* (a22.^2 - 1) + 1).^2;

end

plot(x, beck);

hold on;

plot(x, gcx);



Lab 4 part 3

float3 viewDir = normalize(\_WorldSpaceCameraPos - input.positionWS);

float3 halfVector = normalize(viewDir + lightDir);

//what I did

//replace n dot l with n dot V

float a = \_rough \* \_rough;

float3 blue = float3(0,0,1);

float3 yellow = float3(1, 1, 0);

float3 cook = min((2 \* dot(n, halfVector) \* dot(n, viewDir))

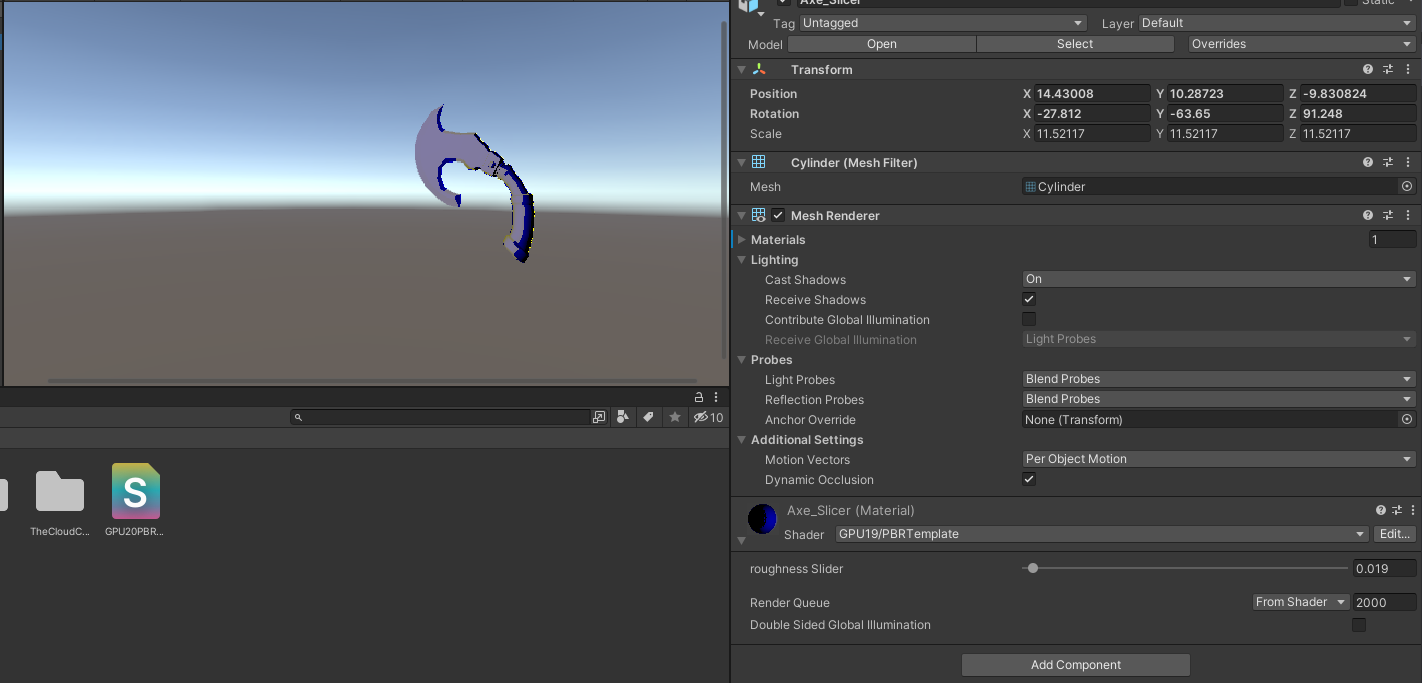
/ dot(viewDir, halfVector), (2 \* dot(n, halfVector) \* dot(n, viewDir)) / dot(viewDir, halfVector));

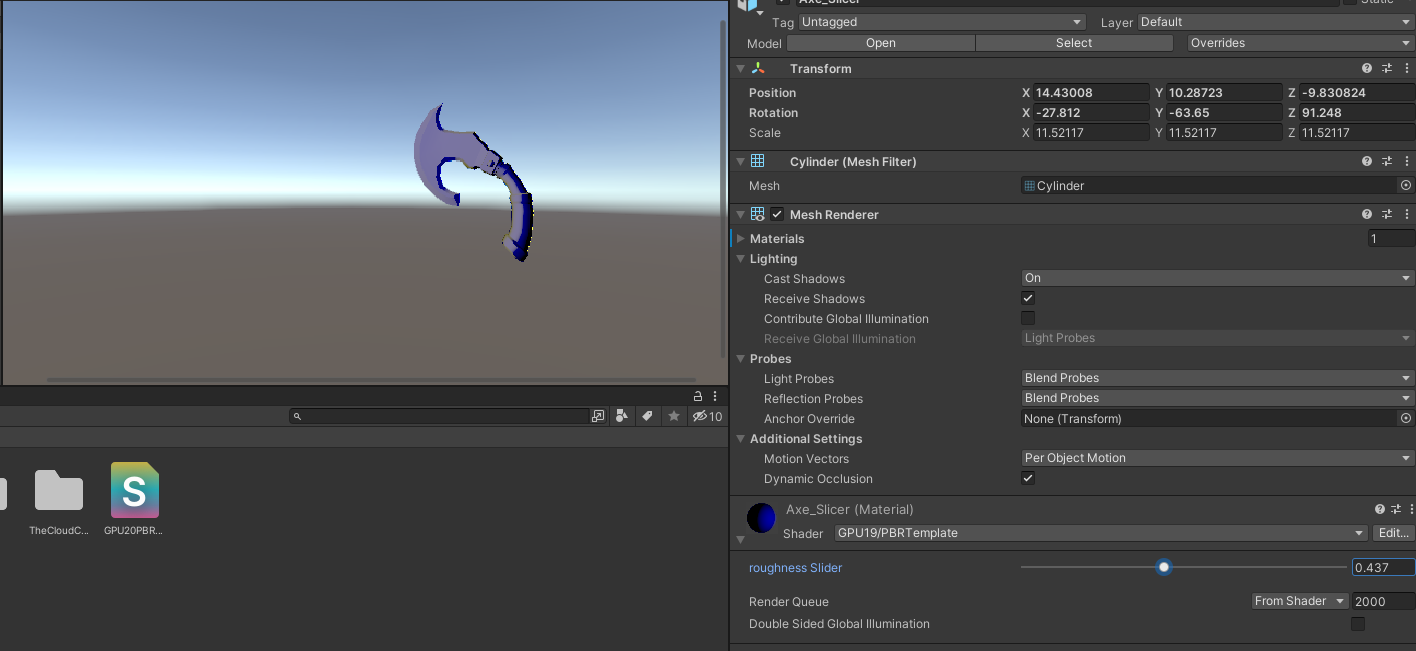
float3 ggx = (2 \* dot(n, viewDir)) / (dot(n, viewDir) + sqrt(pow(a, 2) + (1 - pow(a, 2)) \* pow(dot(n, viewDir), 2)));

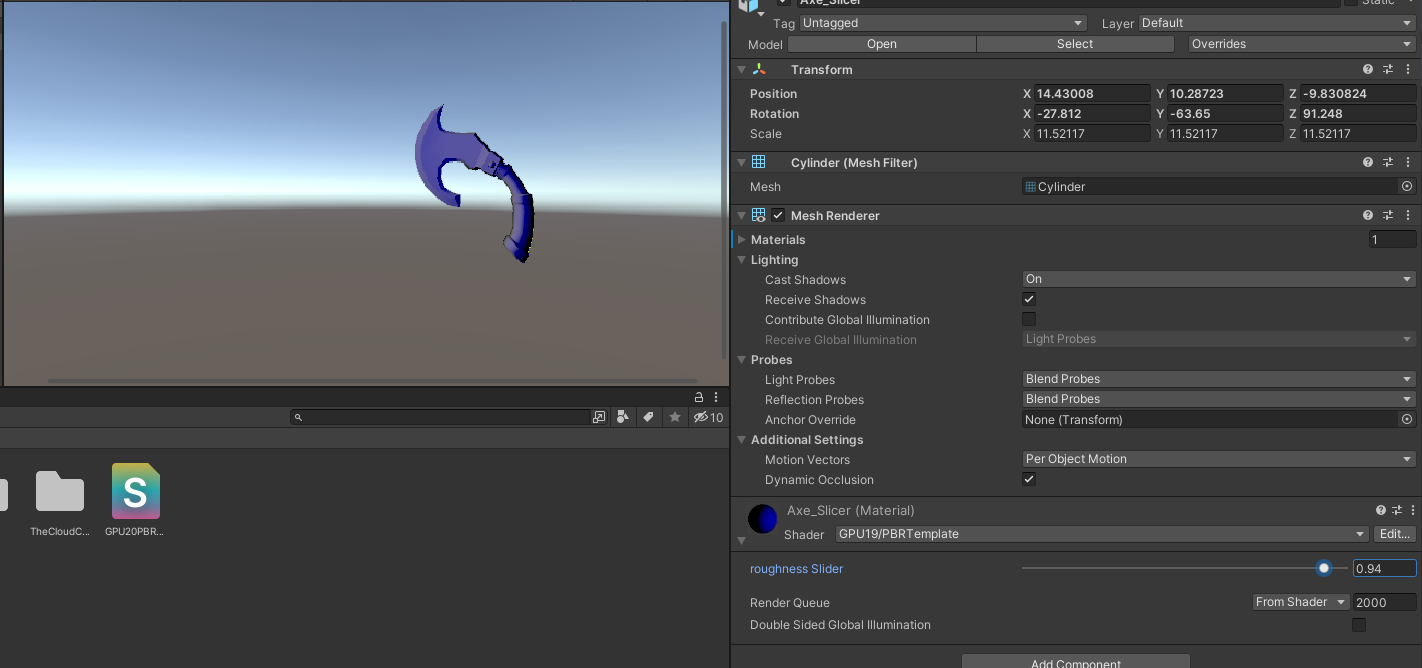
ggx = (2 \* dot(n, lightDir)) / (dot(n, lightDir) + sqrt(pow(a, 2) + (1 - pow(a, 2)) \* pow(dot(n, lightDir), 2))) \* ggx;

float3 output = (ggx \* yellow + cook \* blue) \* \_LightColor0.rgb;

//what I did

return(float4(output,1)); 





Lab4\_part4

There is a difference between models when geometric term is applied. On the Axe slider you can clear see the contrast and it looks like there is some form of light coming from the one side. On the other hand the box doesn’t look realistic because the light is shining from the front of the box, and the surface of the box is very flat so it doesn’t create much of an illusion. I think it is definitely worth costing more computation power to get a better results

Lab4\_part5

//what I did

//replace n dot l with n dot V

float a = \_rough \* \_rough;

float3 blue = float3(0,0,1);

float3 yellow = float3(1, 1, 0);

//float3 cook = min((2 \* dot(n, halfVector) \* dot(n, viewDir))

// / dot(viewDir, halfVector), (2 \* dot(n, halfVector) \* dot(n, viewDir)) / dot(viewDir, halfVector));

float3 fres = 0.971519 + (1 - 0.971519) \* pow((1 - dot(viewDir, halfVector)), 5);

float3 beck = (1 / (3.14 \* pow(a, 2) \* pow(dot(n, halfVector), 4))) \* exp((pow(dot(n, halfVector), 2) - 1) / (pow(a, 2) \* pow(dot(n,halfVector), 2)));

float3 ggx = (2 \* dot(n, viewDir)) / (dot(n, viewDir) + sqrt(pow(a, 2) + (1 - pow(a, 2)) \* pow(dot(n, viewDir), 2)));

ggx = (2 \* dot(n, lightDir)) / (dot(n, lightDir) + sqrt(pow(a, 2) + (1 - pow(a, 2)) \* pow(dot(n, lightDir), 2))) \* ggx;

//float3 output = (ggx \* yellow + beck \* blue) \* \_LightColor0.rgb;

float3 output = fres \* beck \* ggx \* \_LightColor0.rgb;

//what I did

return(float4(output,1));

