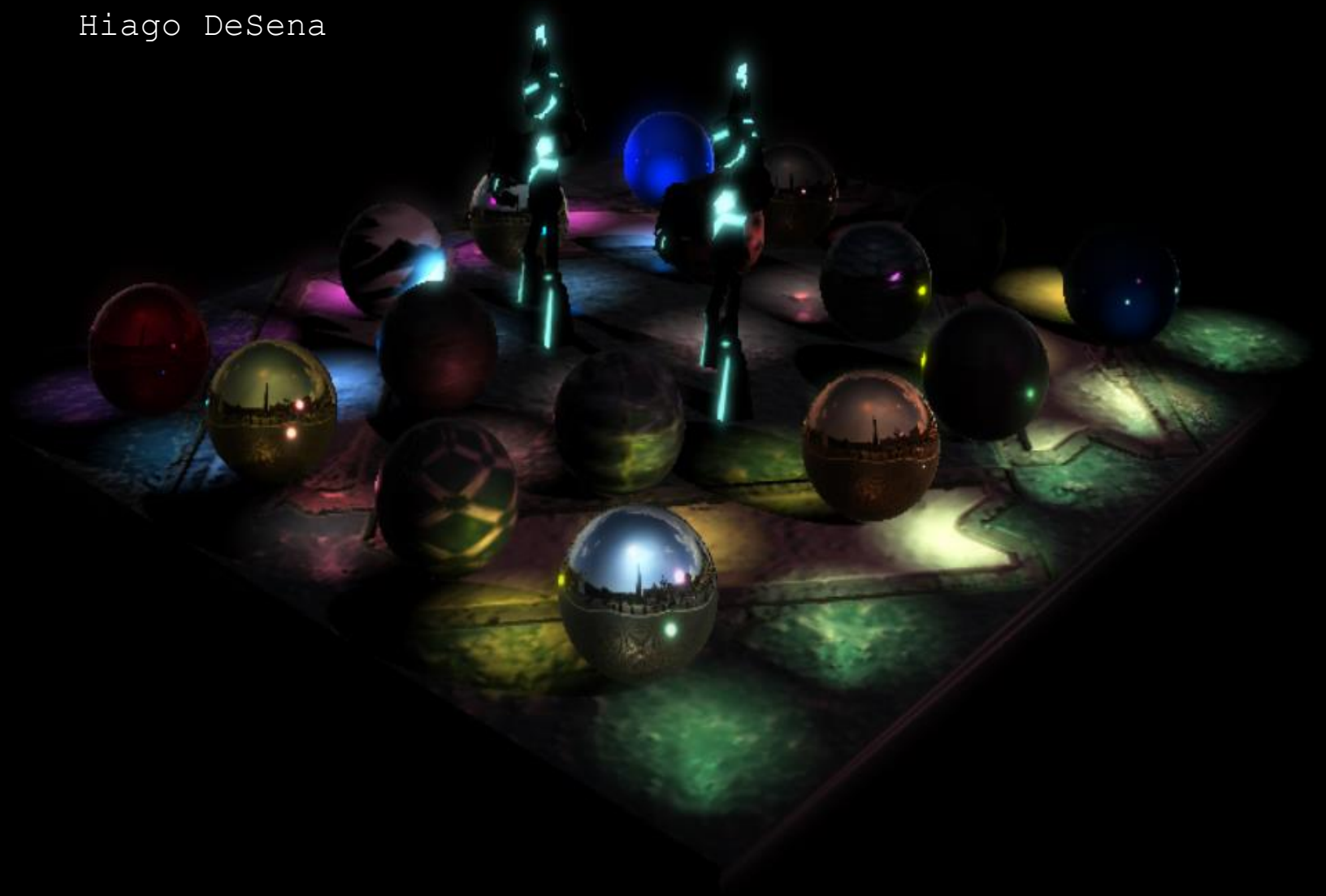
Deferred Shading

Hiago DeSena



Introduction:

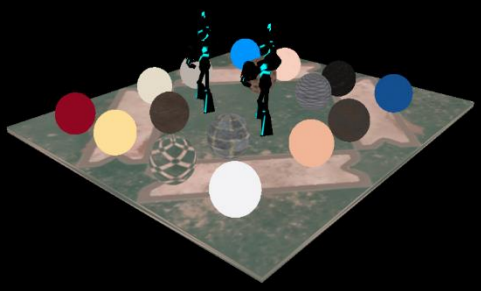
Deferred shading is a rendering technique that processes all opaque geometry in a scene and renderers important information needed for lighting calculation to different render targets simultaneously. The deferred shading pipeline is a two-step process. The first step is known as the **GBuffer** stage which processes all geometry in the scene, building the depth buffer and saving out geometry material information to multiple render targets simultaneously to be used for lighting calculation. Finally when all the data has been collected we bind those resources as textures and sample from them and calculate our lighting information based on the light type.

GBuffer Stage:

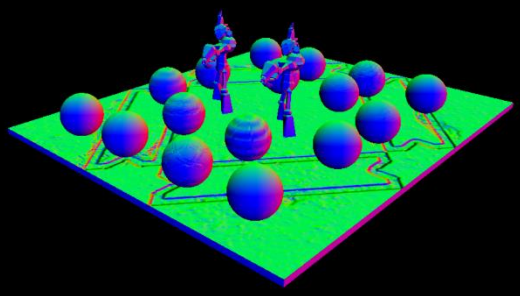
For the GBuffer stage we will allocate two render targets with the format R8G8B8A8 and a depth stencil view with the comparison state of COMPARE_LESS. We will process all opaque geometry in the scene, binding and mapping their material information in a const buffer and writing out the material information in the channels of the render target. The table below shows the material information we will saving in their respected render target channels.

Render Target #	Red	Green	Blue	Alpha
RT0	DiffuseColor.R	DiffuseColor.G	DiffuseColor.B	Metallic
RT1	Normal.x	Normal.y	Normal.z	Roughness

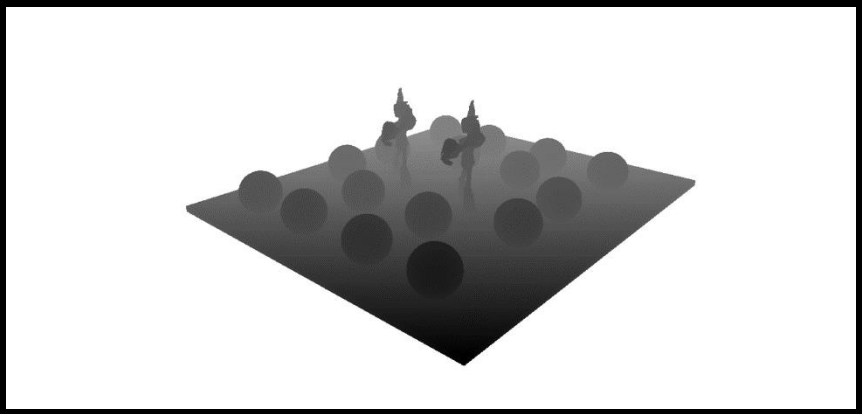
DiffuseColor + Metallic



Normals + Roughness

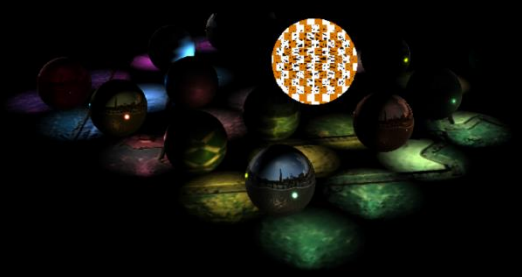


32-bit Depth Buffer



Lighting Stage:

After all our information has been gathered in our GBuffer we can process our lighting objects in the scene. Each light is represented by different geometry as shown below.



Lighting Type	Geometry
Directional Light	Full-Screen Quad
Point Light	Sphere
Spot Light	Shere/Cone

Lighting Equation:

We are using a BRDF lighting model for our light calculation. A BRDF lighting model is a good representation for physically-based lighting. To be physically plausible our equation must have the property of reciprocity and energy conservation.

The equation below is our BRDF diffuse term:

$$Diffuse = \frac{albedo}{\pi}$$

The equation below is our BRDF specular term:

$$f(l,v) = \frac{D(h)F(v,h)G(l,v,h)}{4(n*l)(n*v)}$$

Note: A surface can not reflect more than 100% of the incoming light energy.

Equation Explained:

The BRDF lighting model has a diffuse and specular contribution. The diffuse part of the equation is just a Lambertian model. The Lambertian model states that all diffuse light is distributed equally about a hemi-sphere. The specular portion of the BRDF can be broken down into three functions. The functions are the micro geometry distribution function D(h), Fresnel reflectance function F(v, h) and Geometry function G(l, v, h). The denominator is a correction factor that accounts for quantities being transformed from local to overall micro facet surface.

For our Distribution function we chose the GGX/Trowbridge approximation.

$alpha = roughness^2$

$$D(h) = \frac{alpha^2}{\pi((n*h)^2(alpha^2-1)+1)}$$

The D(h) function determines the size, brightness and shape of the specular highlight. D(h) is a scalar value that as a surface roughness decreases the concentration of micro geometry normal (m) around the overall surface normal (n) increases.

For our Fresnel function we chose the Schlick approximation.

$$F0 = \ Specular\ Color$$

$$F(v,h,F0) = \mathrm{F0} + (1 - \mathrm{F0})(saturate(1 - (v * h)))^5$$

The Fresnel reflectance function computes the fraction of light reflected from an optically flat surface. The function is dependent on the incoming angle between (v) and (h). Note (h = halfway vector).

For our Geometry function we chose a Shlick approximation.

$$G(l,v,h) = G1(v) * G(l)$$

$$G1(v) = \frac{n * v}{(n * v)(1 - k) + k}$$

The geometry function is the probability that a surface's points with a given micro geometry normal (m) will be visible from both the light direction (l) and the view direction (v). The function is a scalar value from 0 to 1. This function is essential for the energy conservation part of the BRDF and also applies shadowing and masking at the microscopic level.

$$\frac{active\ surface\ area}{total\ surface\ area}$$

Using our Lighting Equation:

We can now use the BRDF lighting model explained above to calculate our light contribution in our scene depending on the properties of our geometry material information and light type. Our main goal with this deferred rendering technique is to be able to draw three different ranges of materials. Metallic materials such as steel, gold, copper and silver. Rough materials such as concrete, wood and dirt. Finally we want to be able to render anything in between fully metallic and fully rough materials such as plastic, leather, cotton and skin.

Before applying our lighting calculation we need to recalculate our albedo color and our specular color of our object based on the metallic contribution of the material.

$$realAlbedo = diffuseColor - diffuseColor * metallic$$

This equation states that as a material becomes more metallic it loses the diffuse color contribution of the object and becomes fully reflected.

Next we calculate the real specular color of the material.

$$realSpecularColor = lerp(0.03f, diffuseColor, metallic)$$

To calculate the specular color we just lerp from diffuse color to metallic by 0.03f (this is a good value for starting range of dielectrics).

Finally we use all the information stored in our GBuffer and our lighting equations from above to calculate the final color of our lit scene.

$$albedoDiffuse = Diffuse(realAlebdo)$$

$$specular = Specular(realSpecularColor, alpha, dotNL, dotNV, dotVH, dotNH)$$

$$finalColor = lightColor * attn\ * dotNL * (albedoDiffuse * (1.0 - specular) + specular)$$

Conclusion:

In conclusion the deferred rendering technique presented creates a GBuffer and uses all the stored information to apply lighting to the whole scene using a BRDF rendering model with support for physically based shading. The picture below is the final Render of the deferred rendering technique.