Introduction to Shader development

h_da WS2020/21

Paul Nasdalack

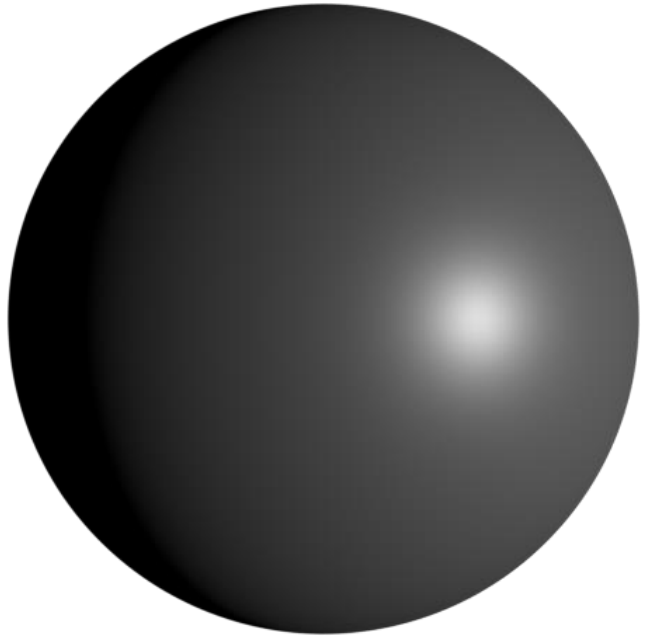
info@paul-nasdalack.com

@littleBugHunter

LIGHT

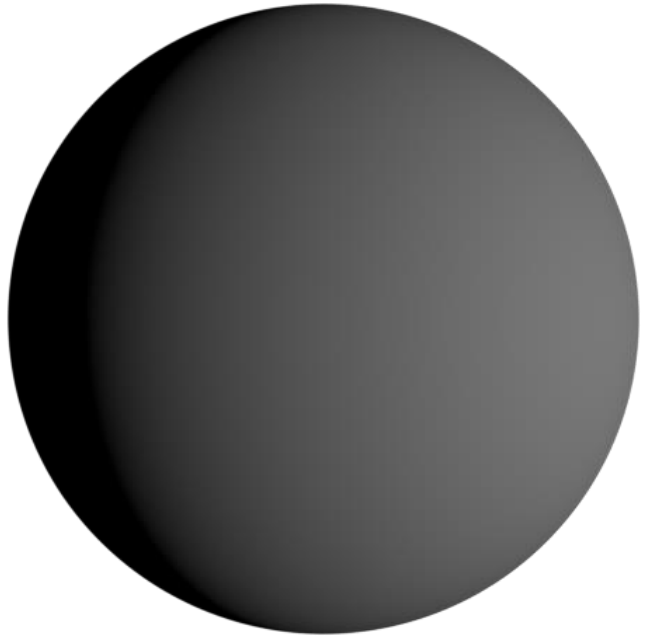
Light

- Two types of Light:
  - Diffuse Light
  - Specular Highlight



Diffuse Light

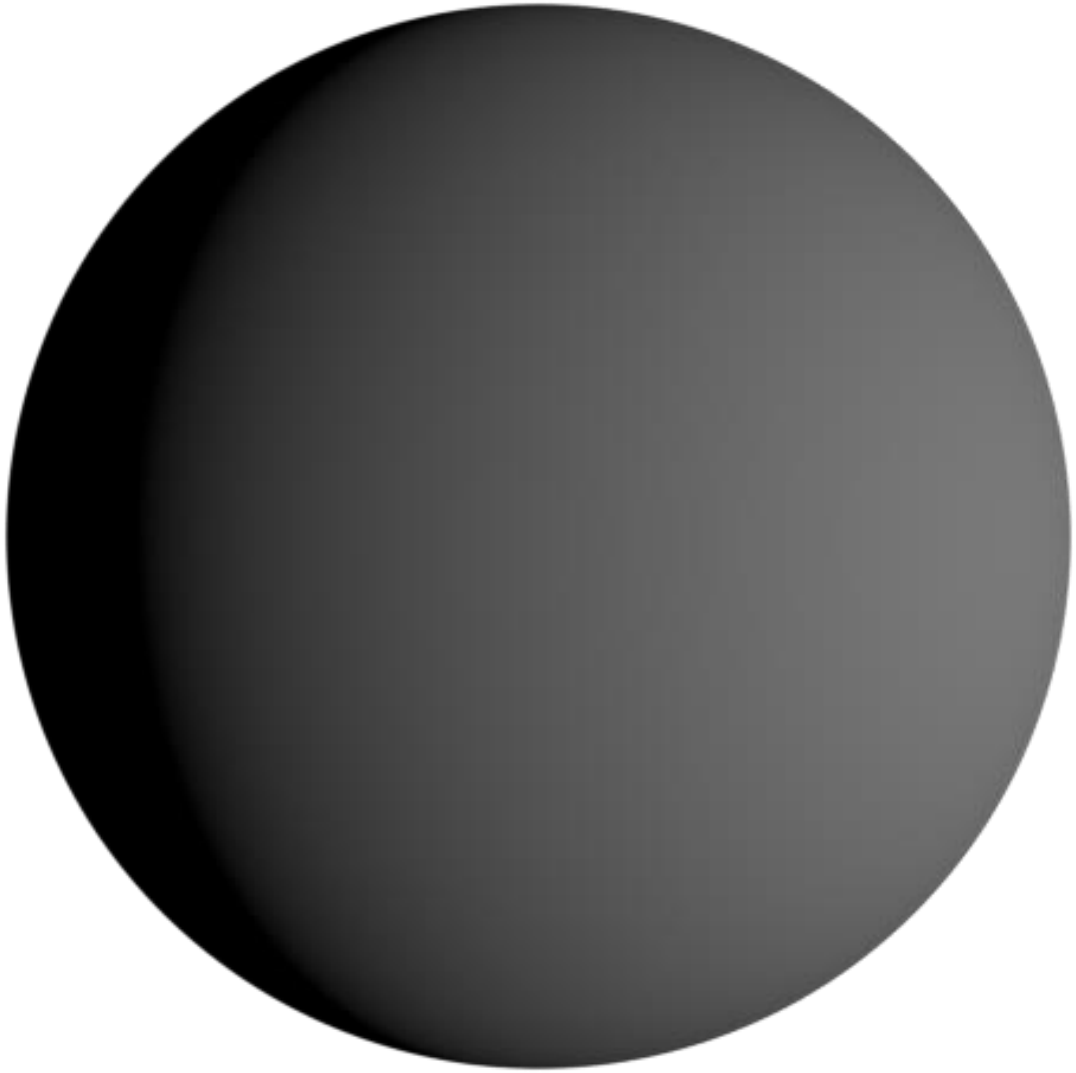
- Darkens with distance (distance Falloff)
- Lights one half of the object softly
- Colored by the Albedo of the Material



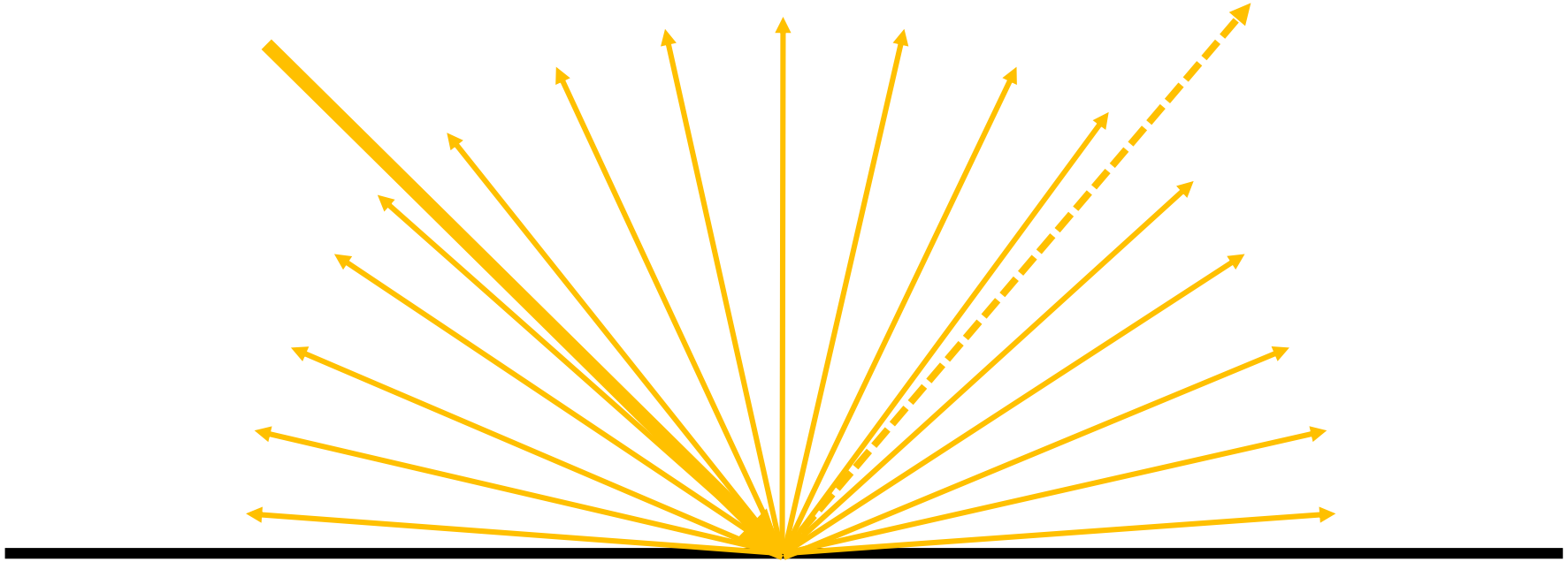
Specular Light

- Darkens with distance (distance Falloff)
- One Highlight
- View direction dependent
- Colored by the Specular Color of the Material

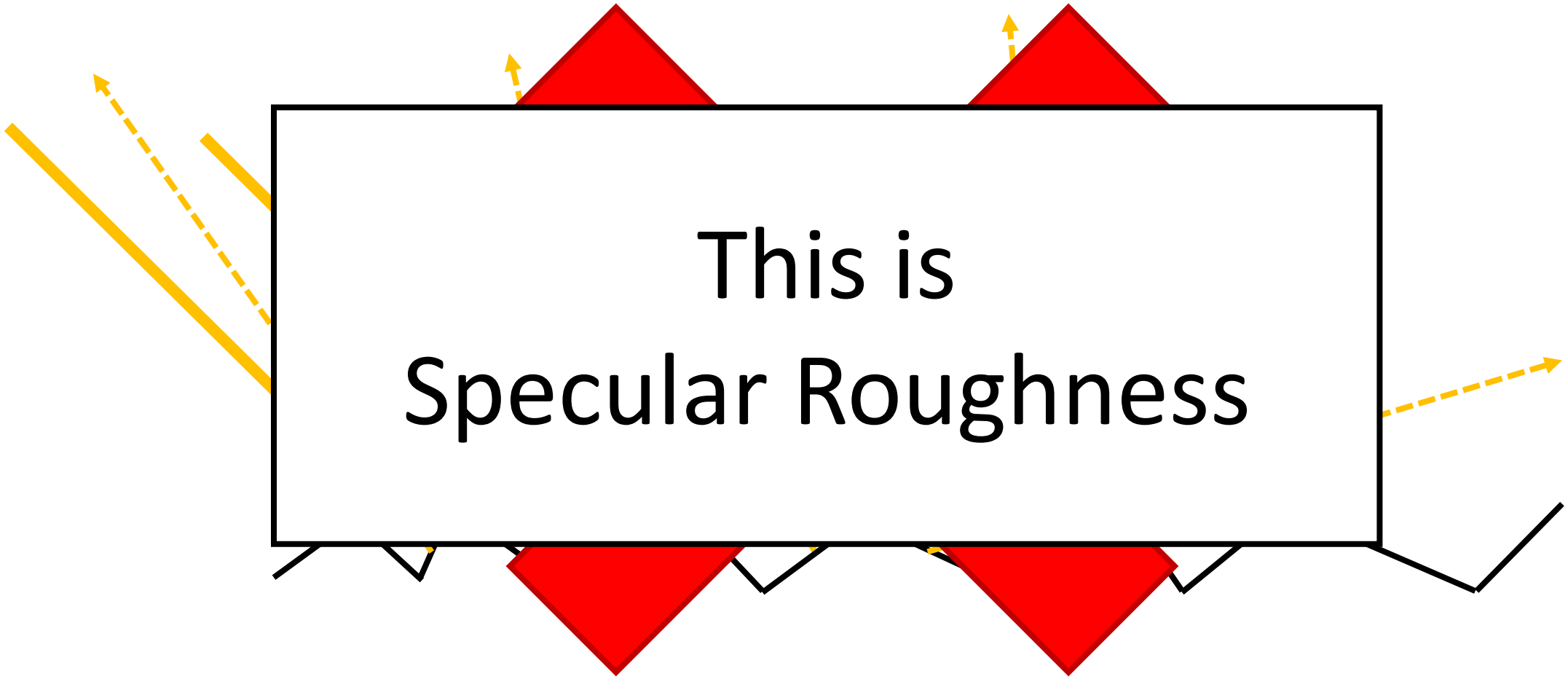
Diffuse Light



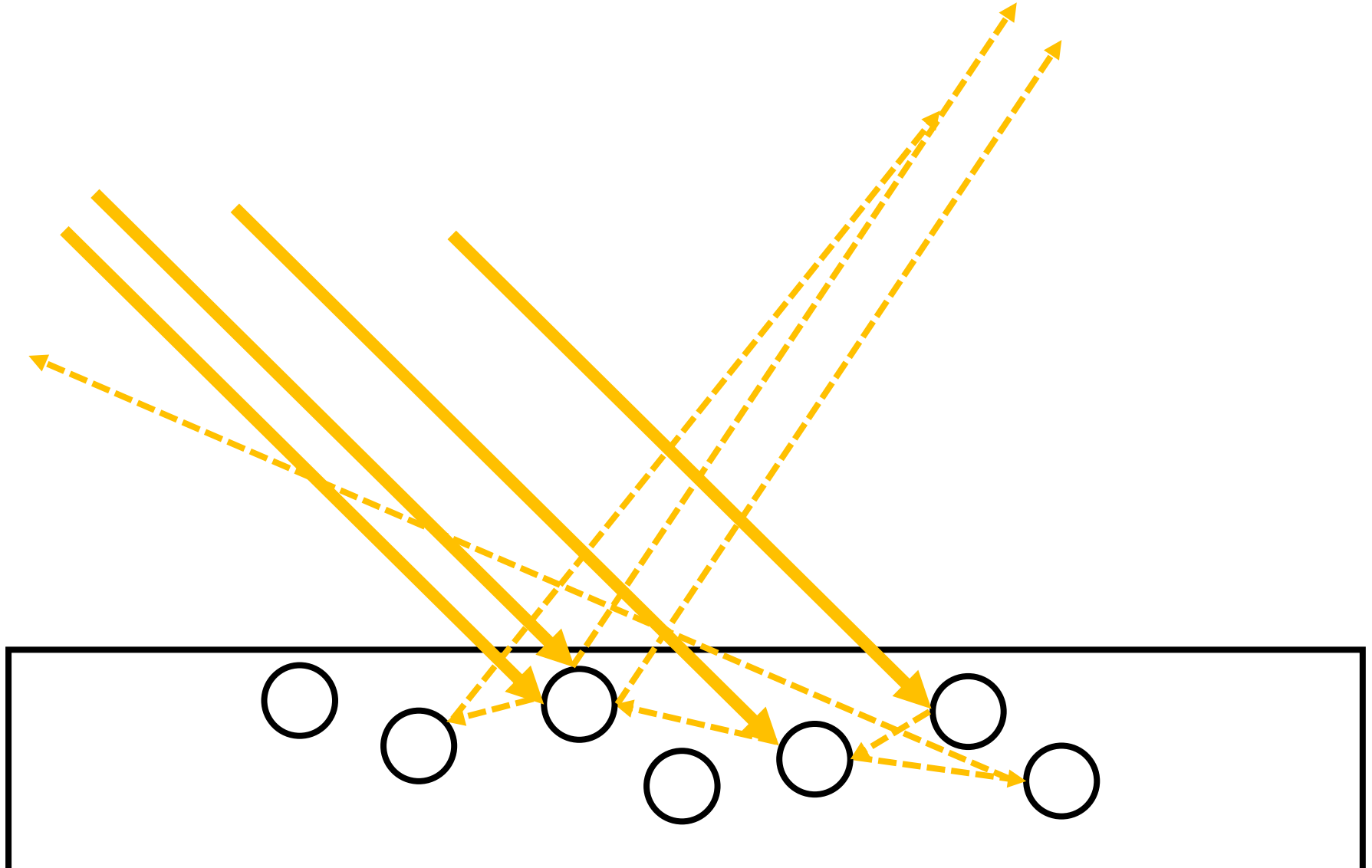
Diffuse Light (how it works)



Diffuse Light (how it works NOT)

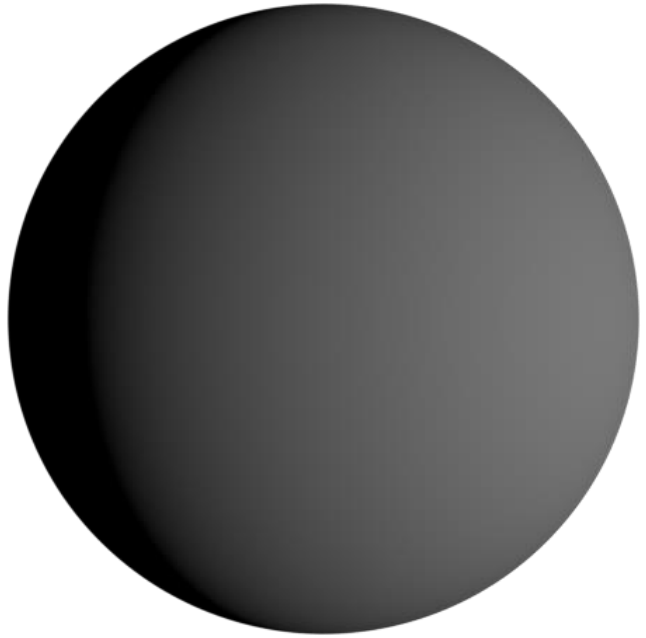


Diffuse Light (how it works)



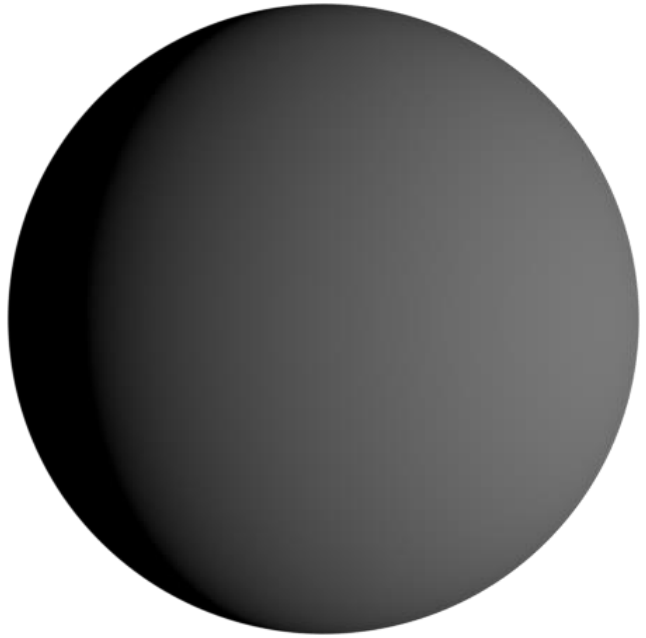
Diffuse Light (how to fake it)

- Darkens with distance (distance Falloff)
- Darkens with angle (Lambertian reflection)
- Colored by the Albedo of the Material



Diffuse Light (how to fake it)

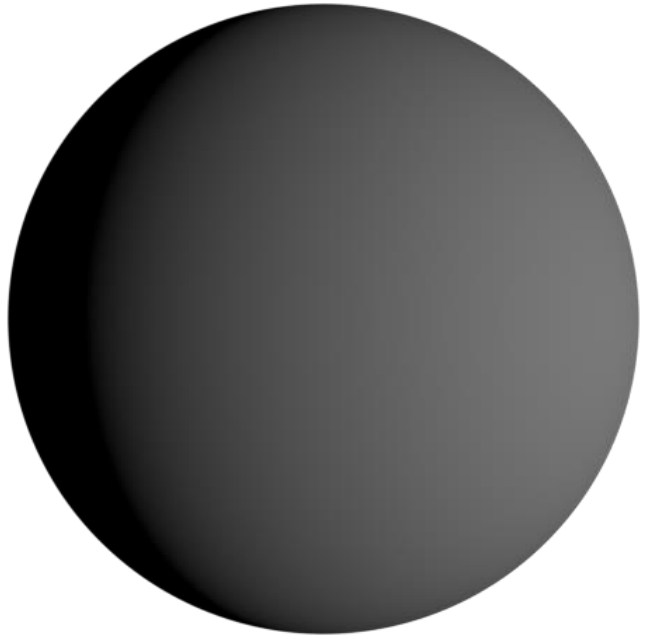
- Darkens with distance (distance Falloff)
  - The further away a Vertex/Fragment is from the Light source, the darker it gets



```
float distance = length(_LightPos-i.worldPos);
float falloff = 1/(distance*distance); //inverse square falloff
```

Diffuse Light (how to fake it)

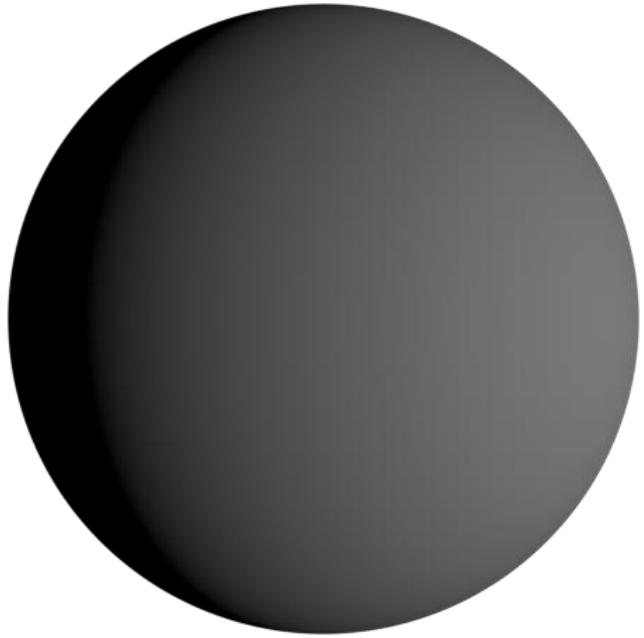
- Darkens with angle (Lambertian reflection)
  - Compare the Light direction with the surface normal, the steeper the angle, the less light is reflected



```
float distance = length(_LightPos-i.worldPos);
float falloff = 1/(distance*distance); //inverse square falloff
float lightDir = normalize(i.worldPos-_LightPos);
float lambert = saturate(dot(i.worldNormal, - lightDir));
```

Diffuse Light (how to fake it)

- Colored by the Albedo of the Material
  - Simply multiply the light with the Albedo Color



```
float distance = length(_LightPos-i.worldPos);
float falloff = 1/(distance*distance); //inverse square falloff
float lightDir = normalize(i.worldPos-_LightPos);
float lambert = saturate(dot(i.worldNormal, - lightDir));
float4 diffuse = falloff * lambert * _LightColor * _AlbedoColor;
```