Introduction to Shader development

h_da WS2020/21

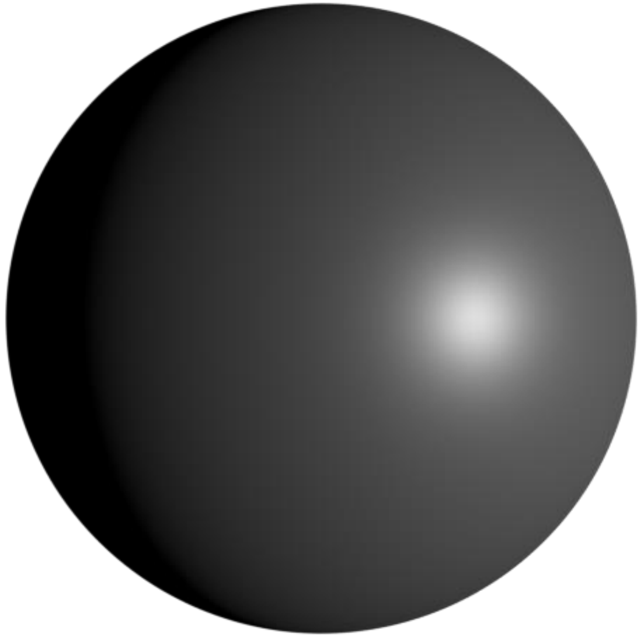
Paul Nasdalack

info@paul-nasdalack.com

@littleBugHunter

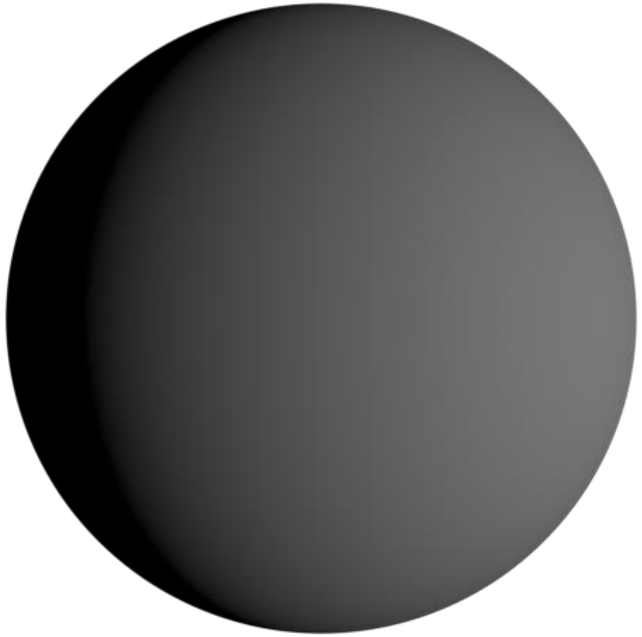
Light

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



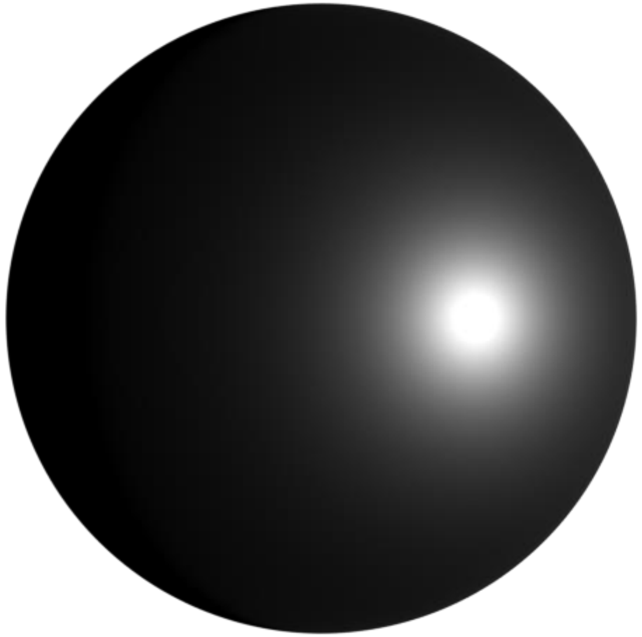
Diffuse Light

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

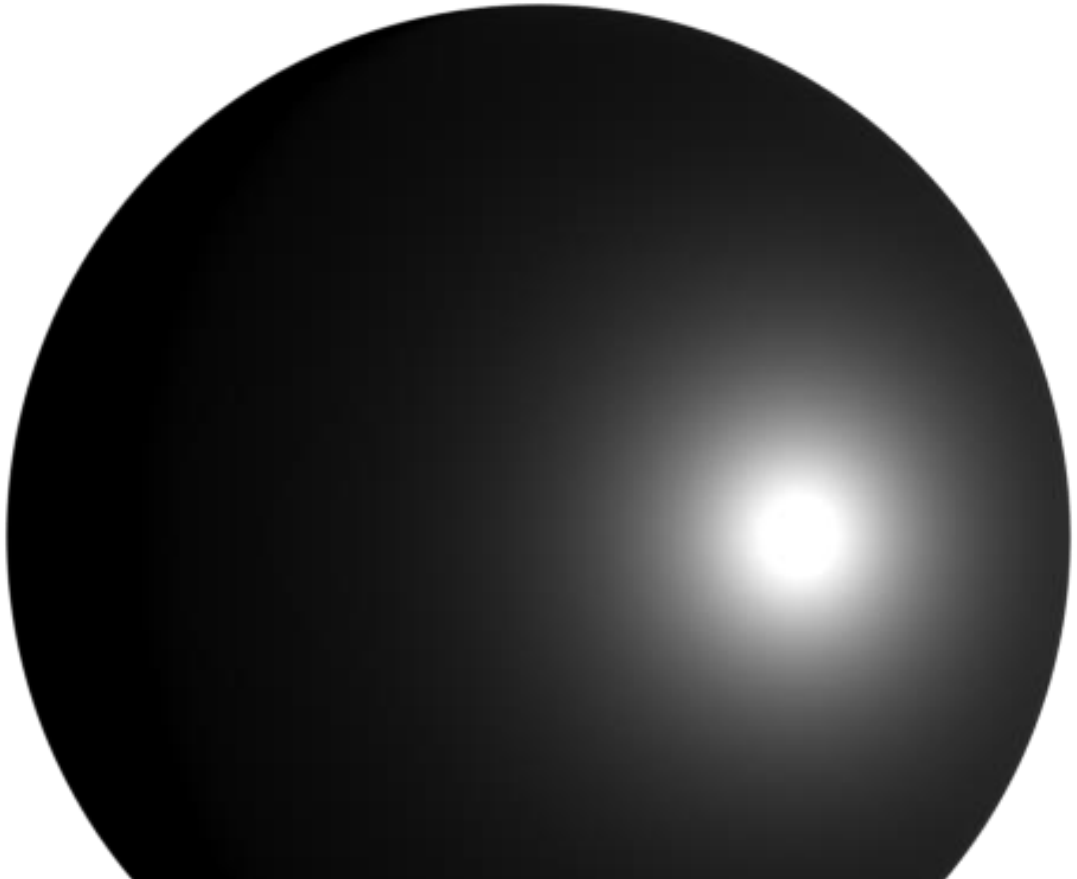


Specular Light

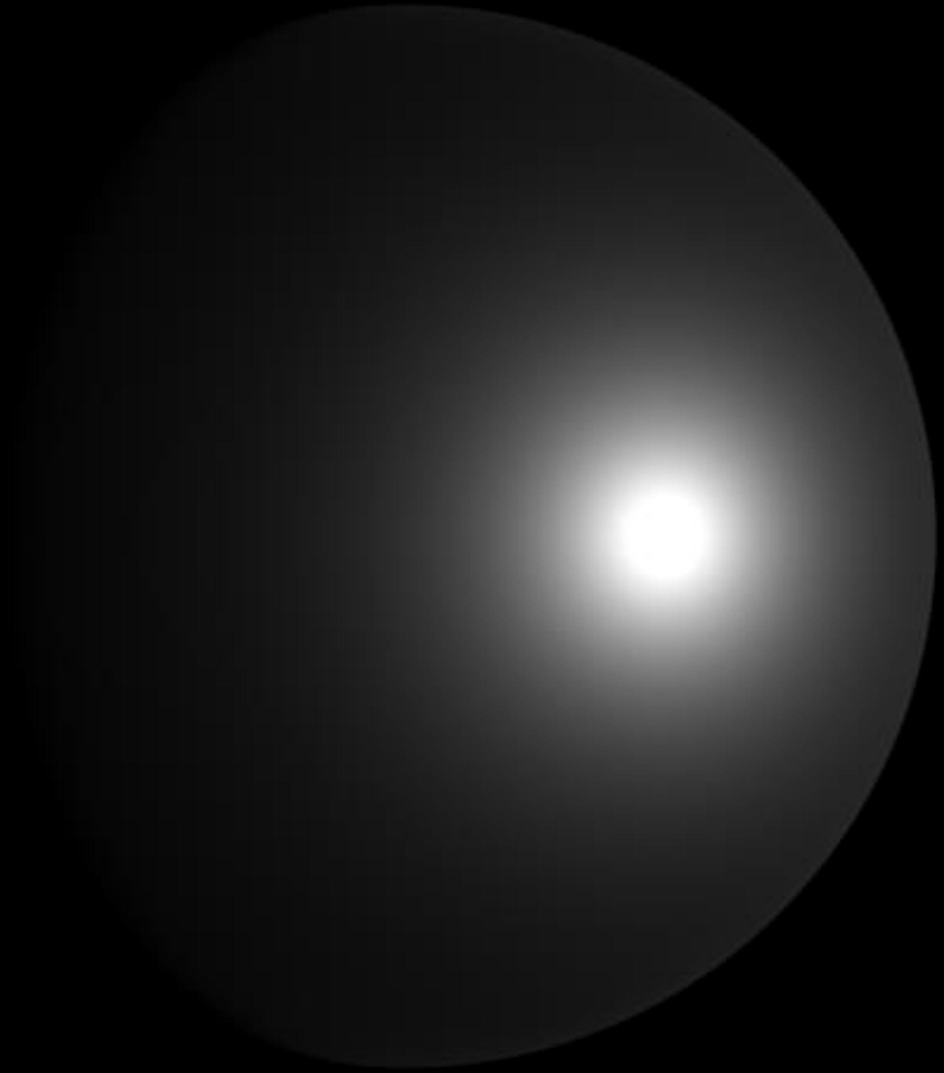
- Darkens with distance (distance Falloff)
- One Highlight
- View direction dependent
- Colored by the Specular Color of the Material



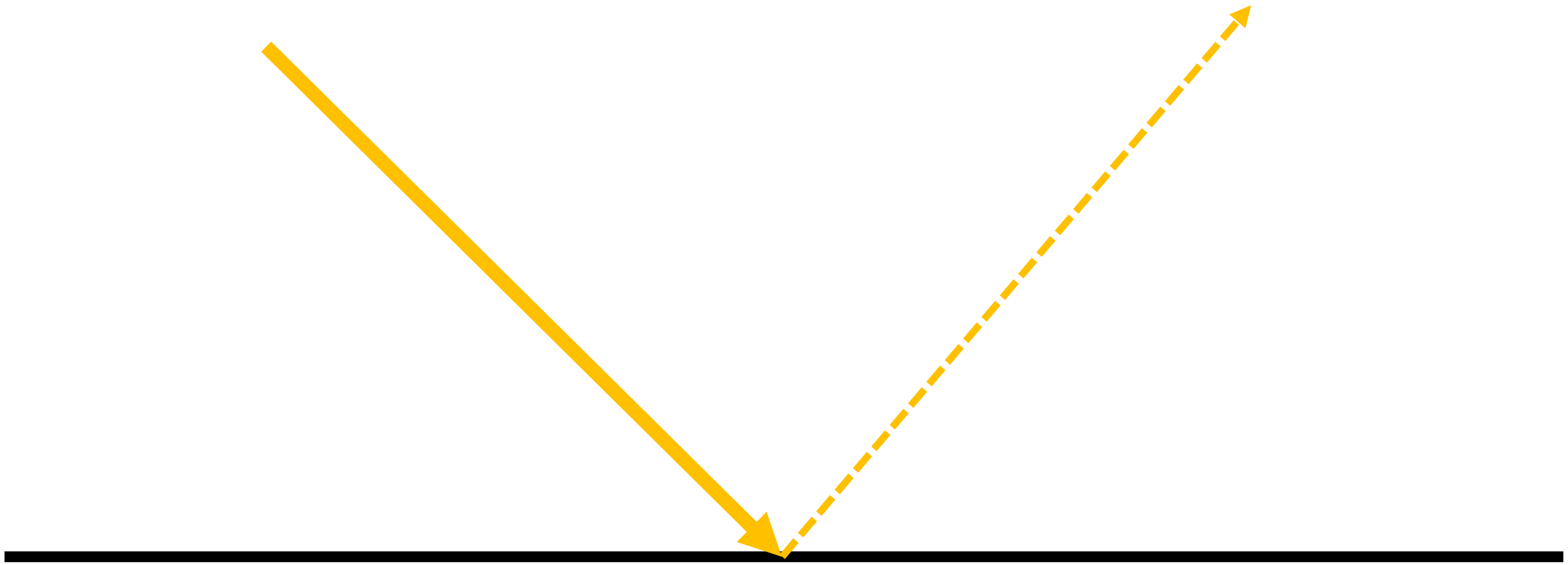
Specular Light



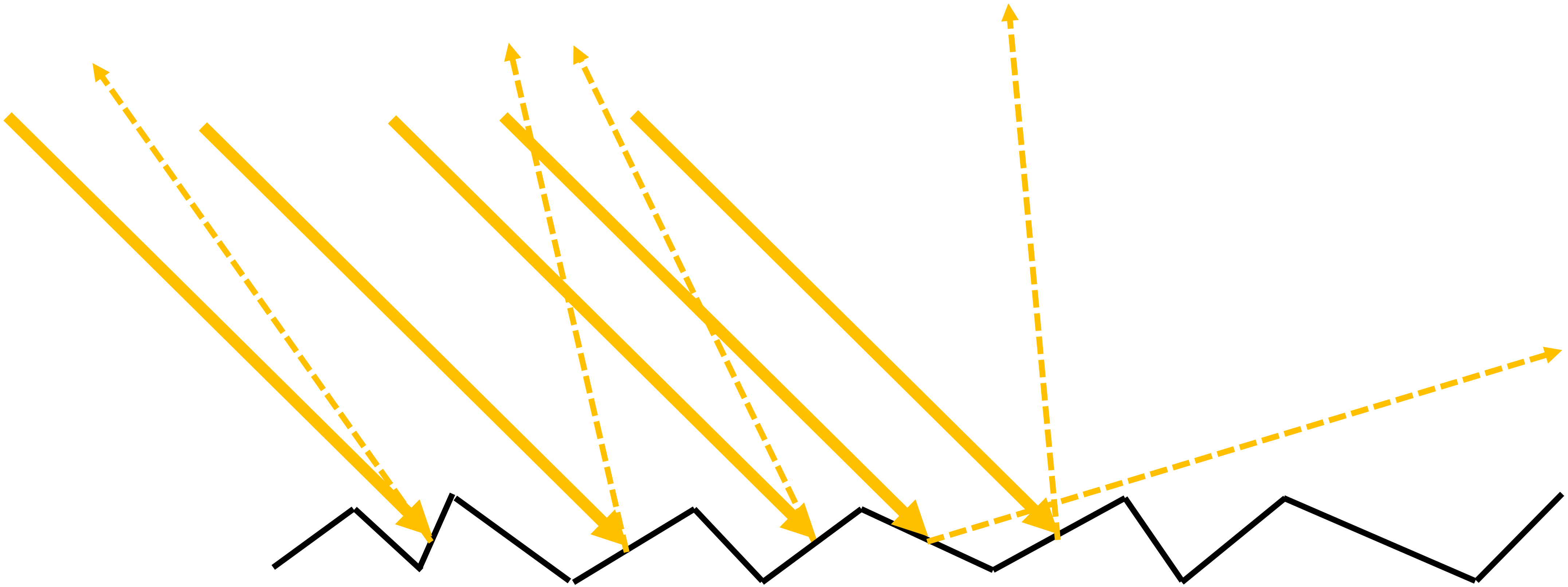
Light



Specular Light (how it works)

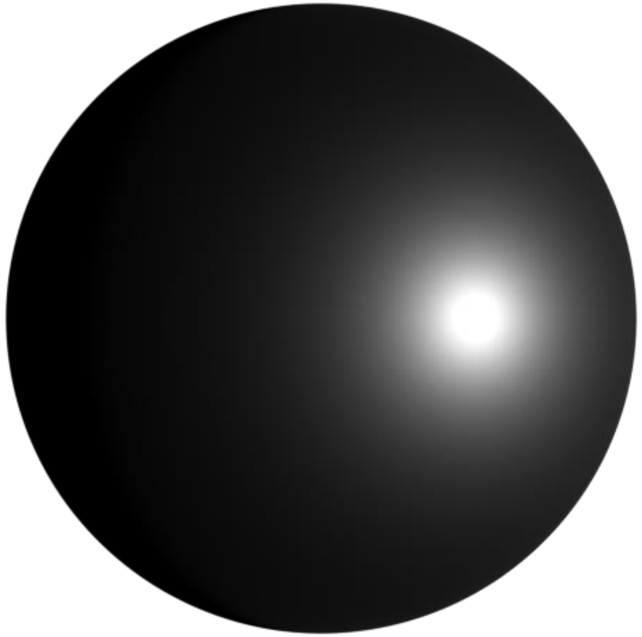


Specular Light (how it works)



Specular Light (how to fake it)

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

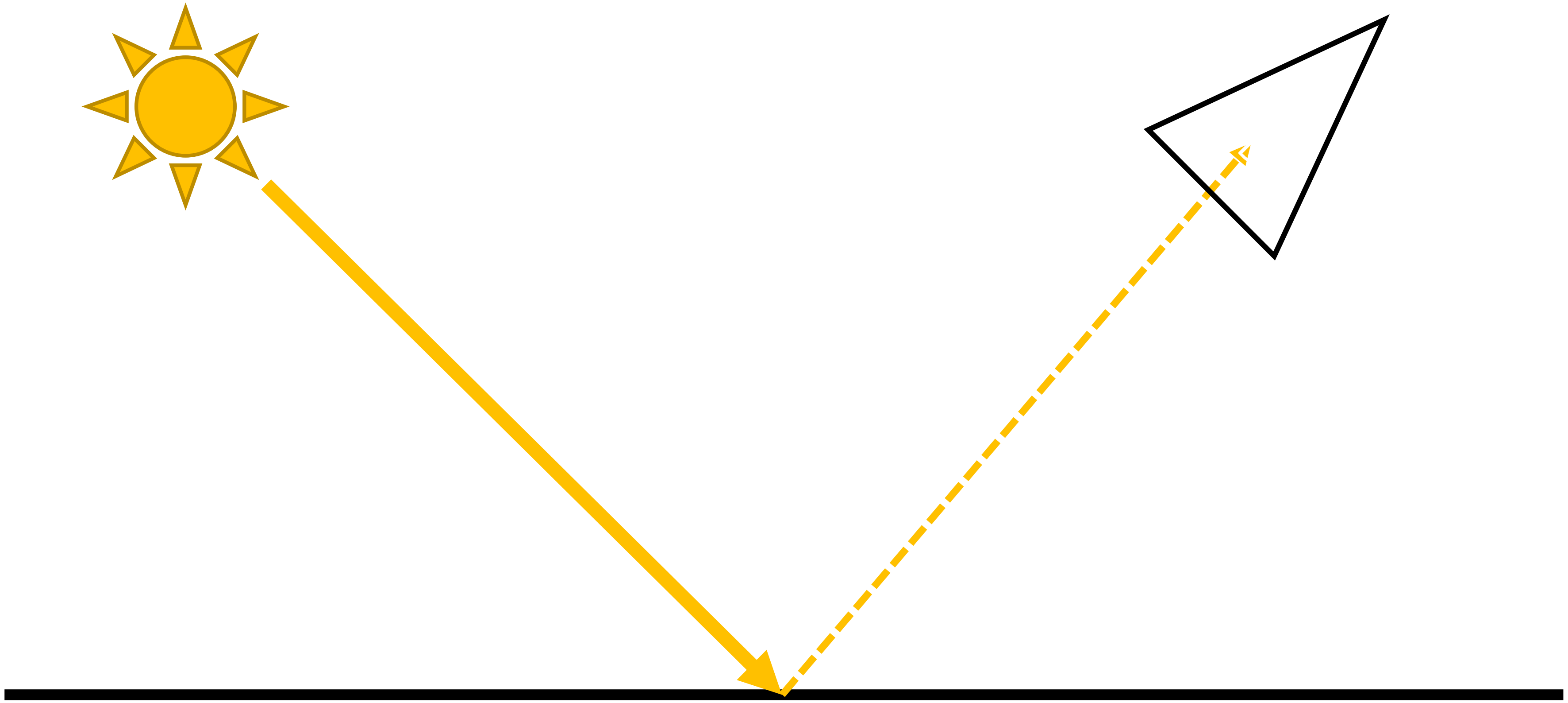


Specular Light (how to fake it)

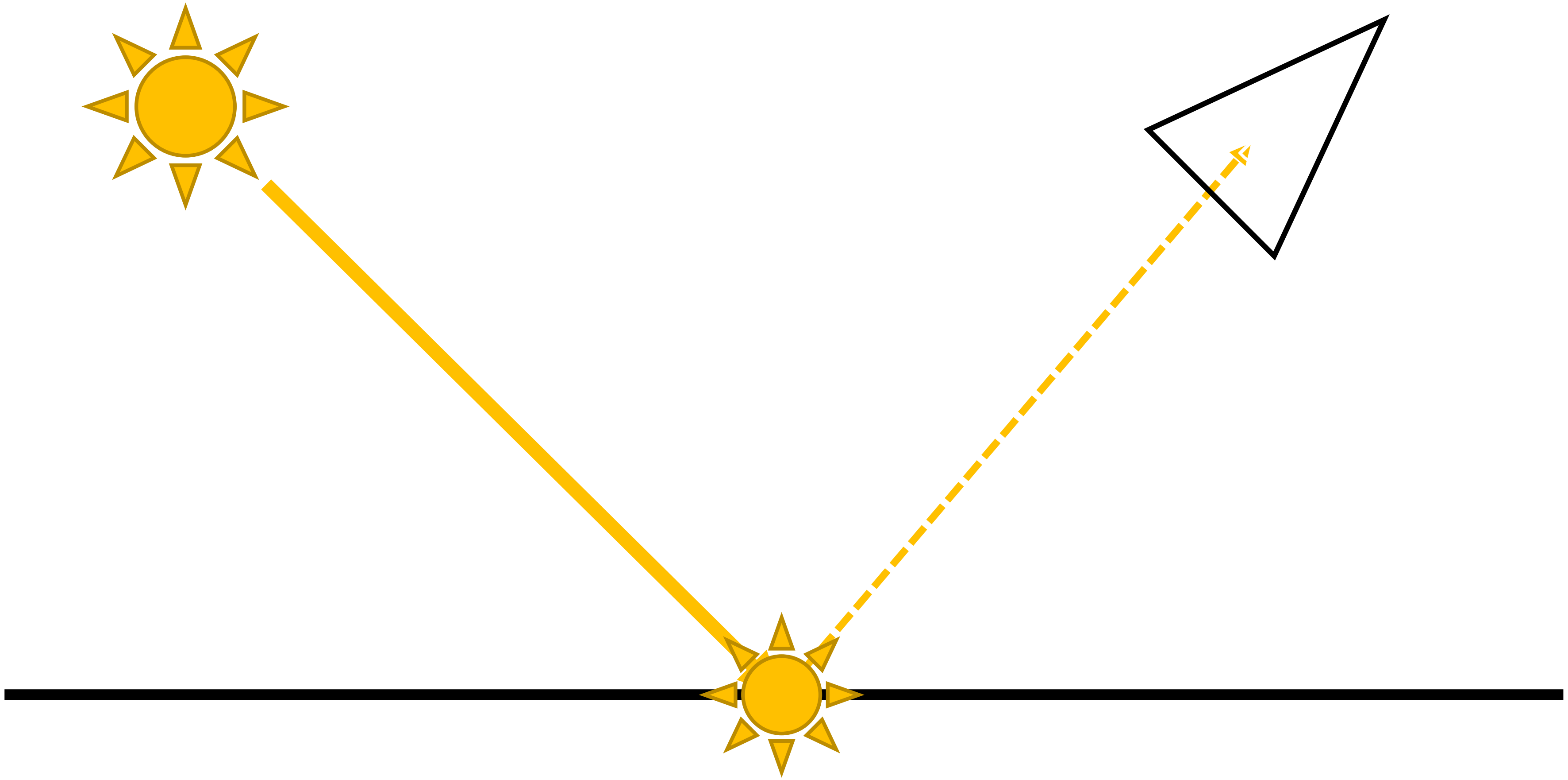
- Darkens with distance (distance Falloff)
  - The further away a Vertex/Fragment is from the Light source, the darker it gets

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

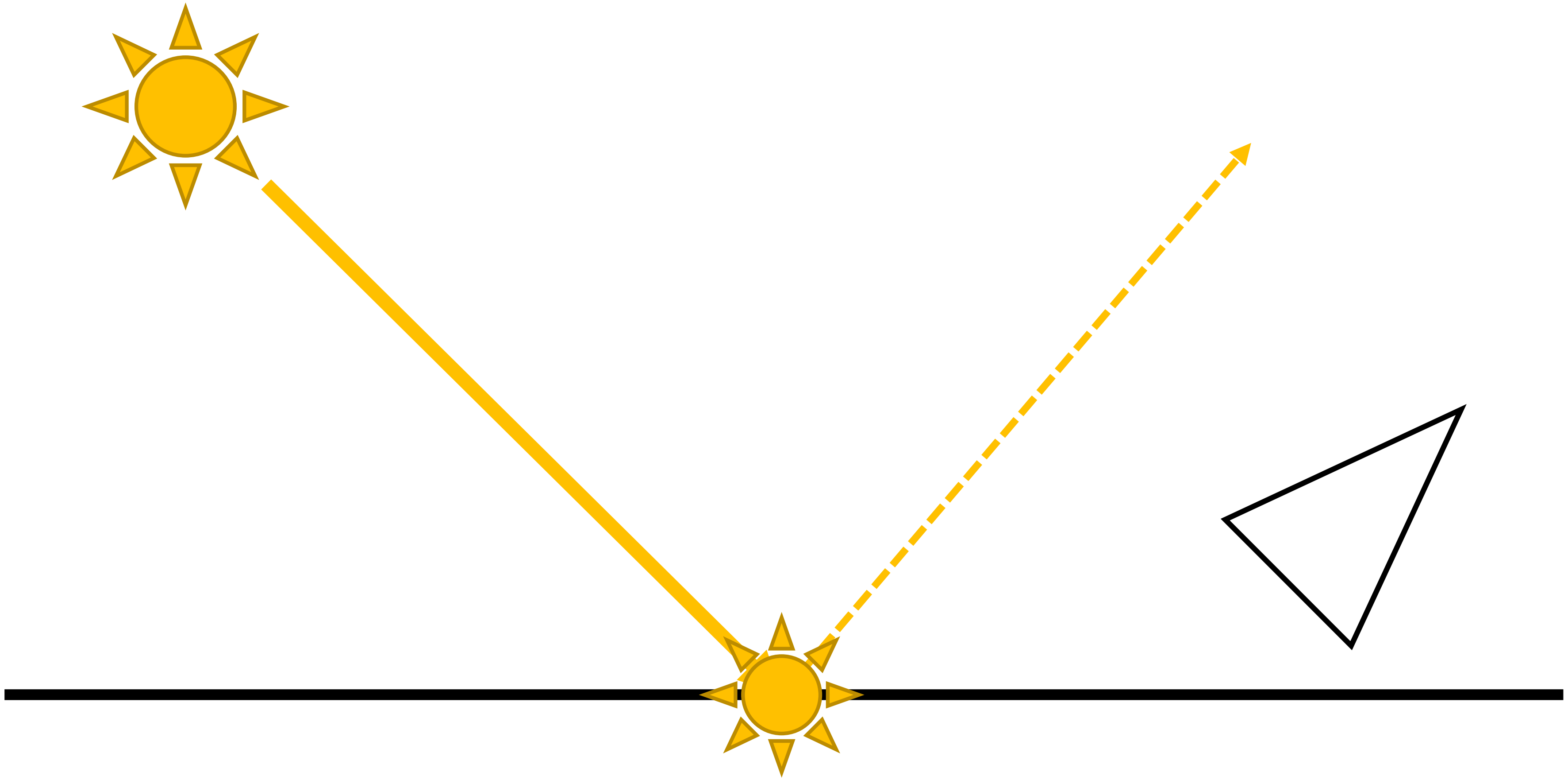
Specular Light (how to fake it)



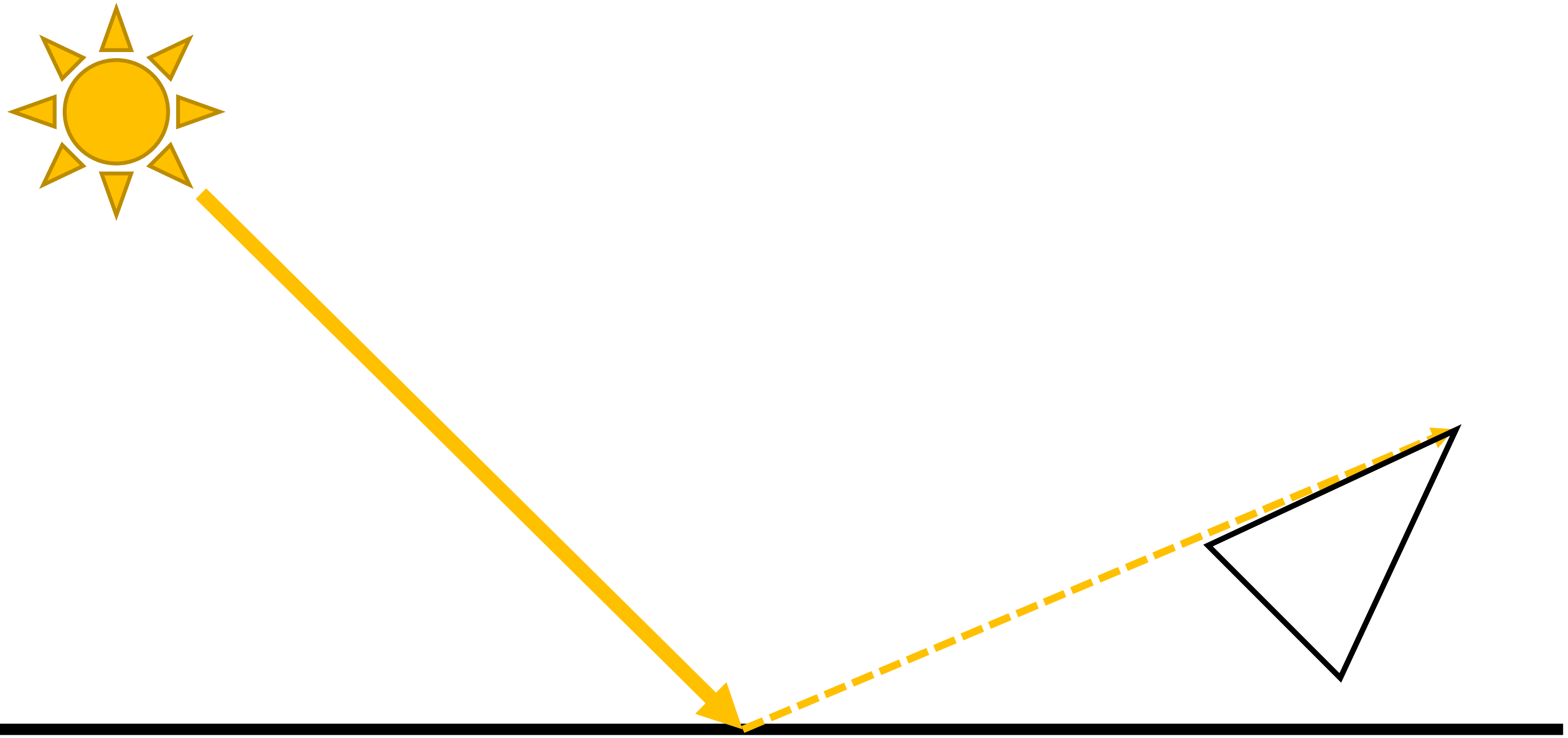
Specular Light (how to fake it)



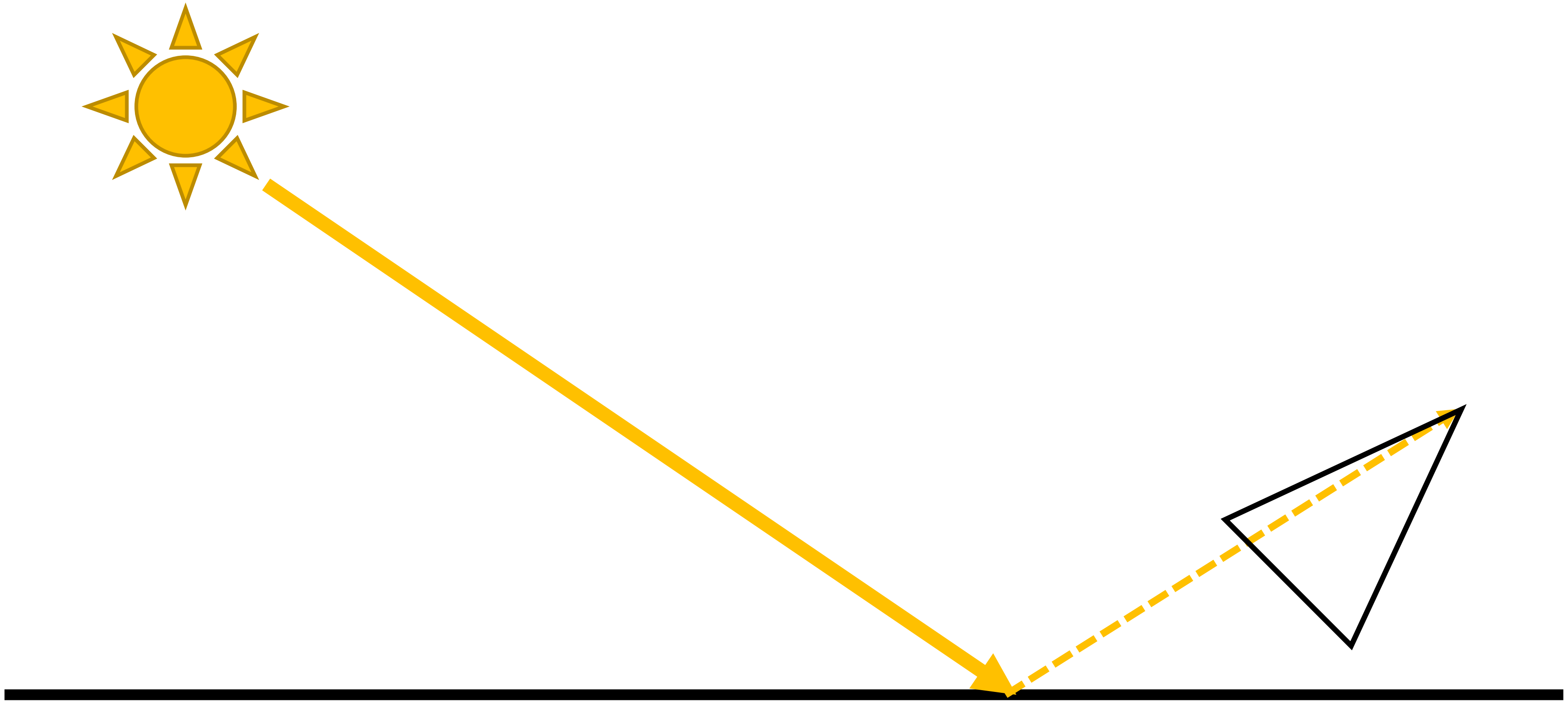
Specular Light (how to fake it)



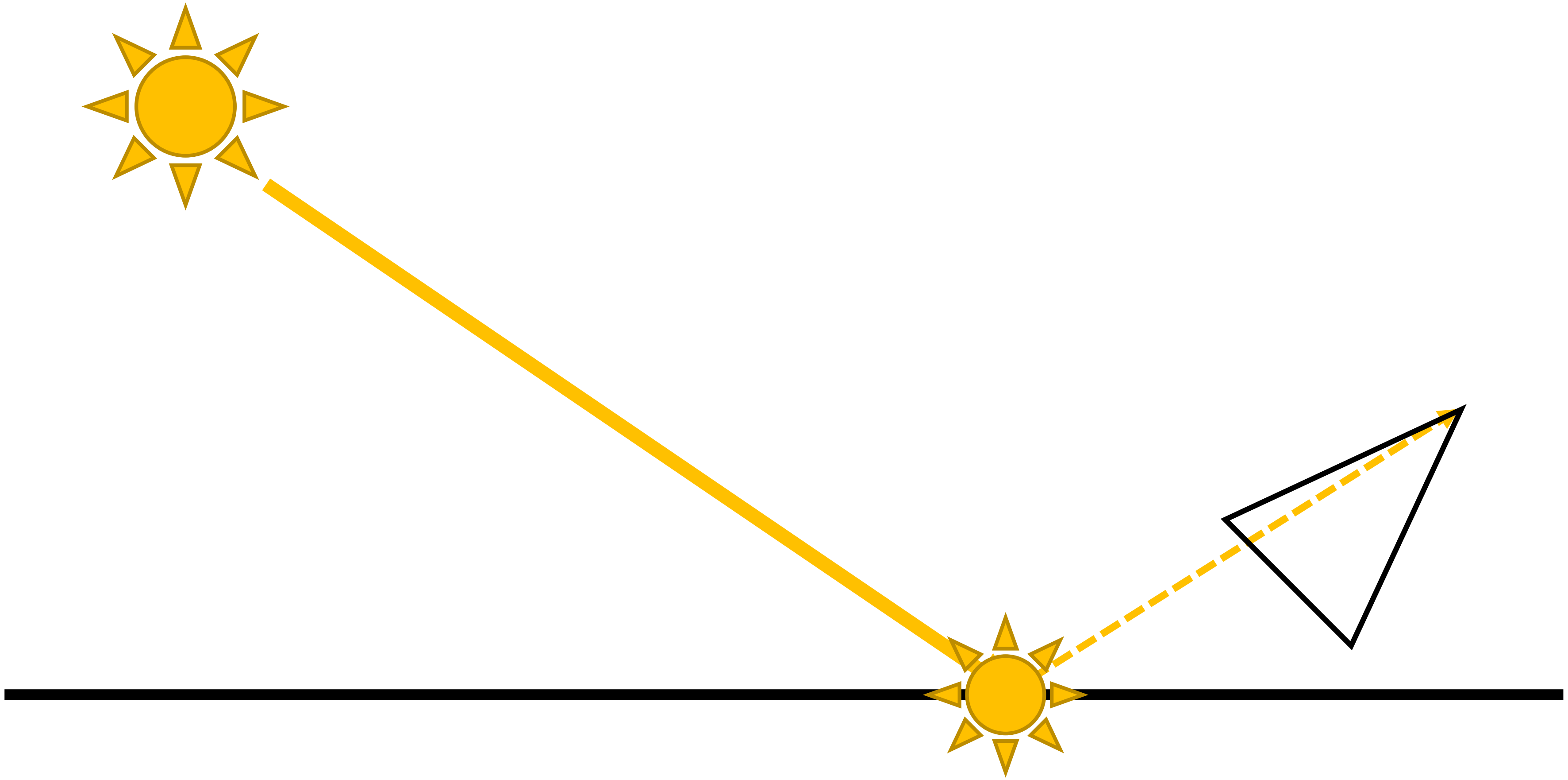
Specular Light (how to fake it)



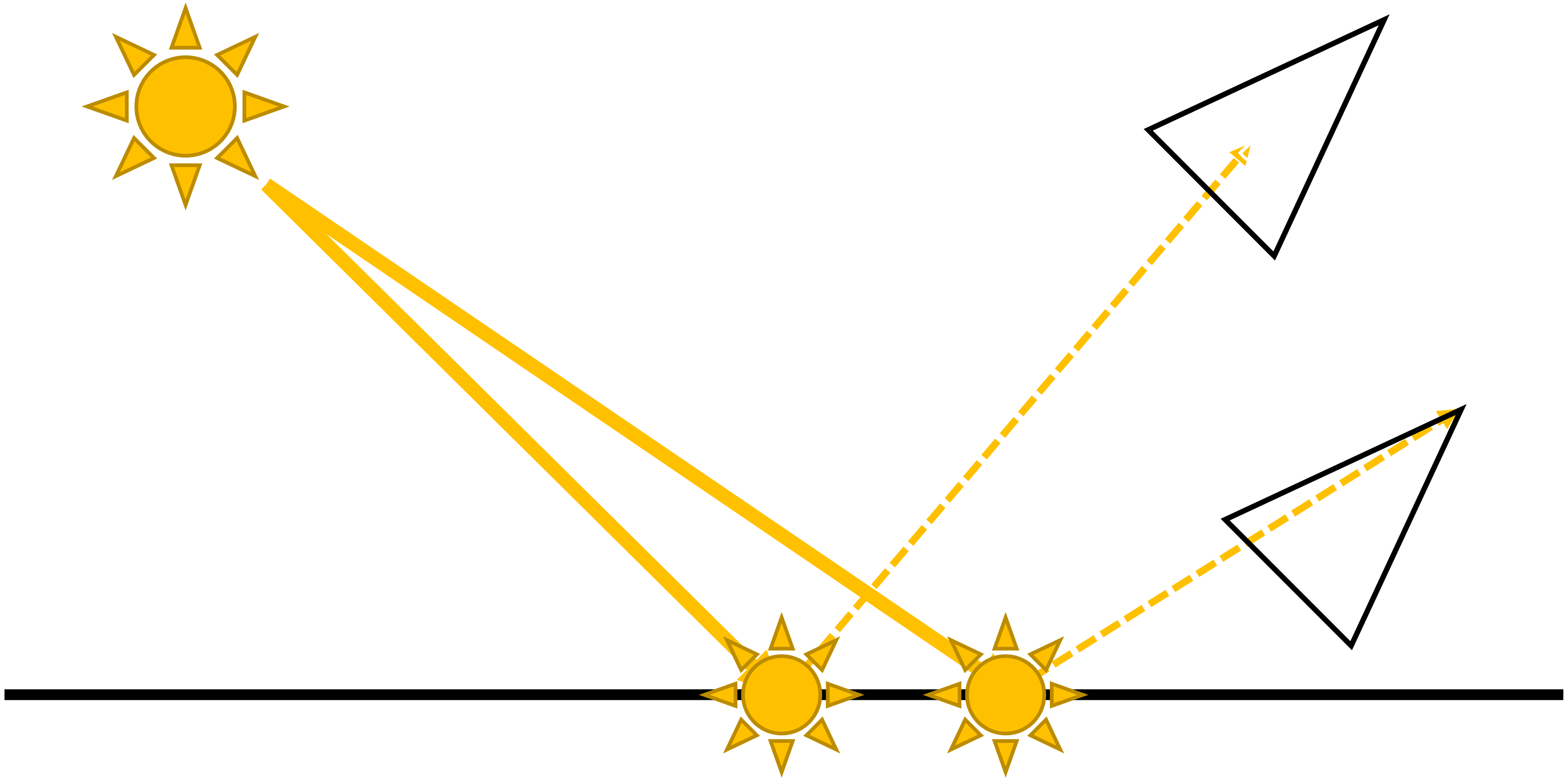
Specular Light (how to fake it)



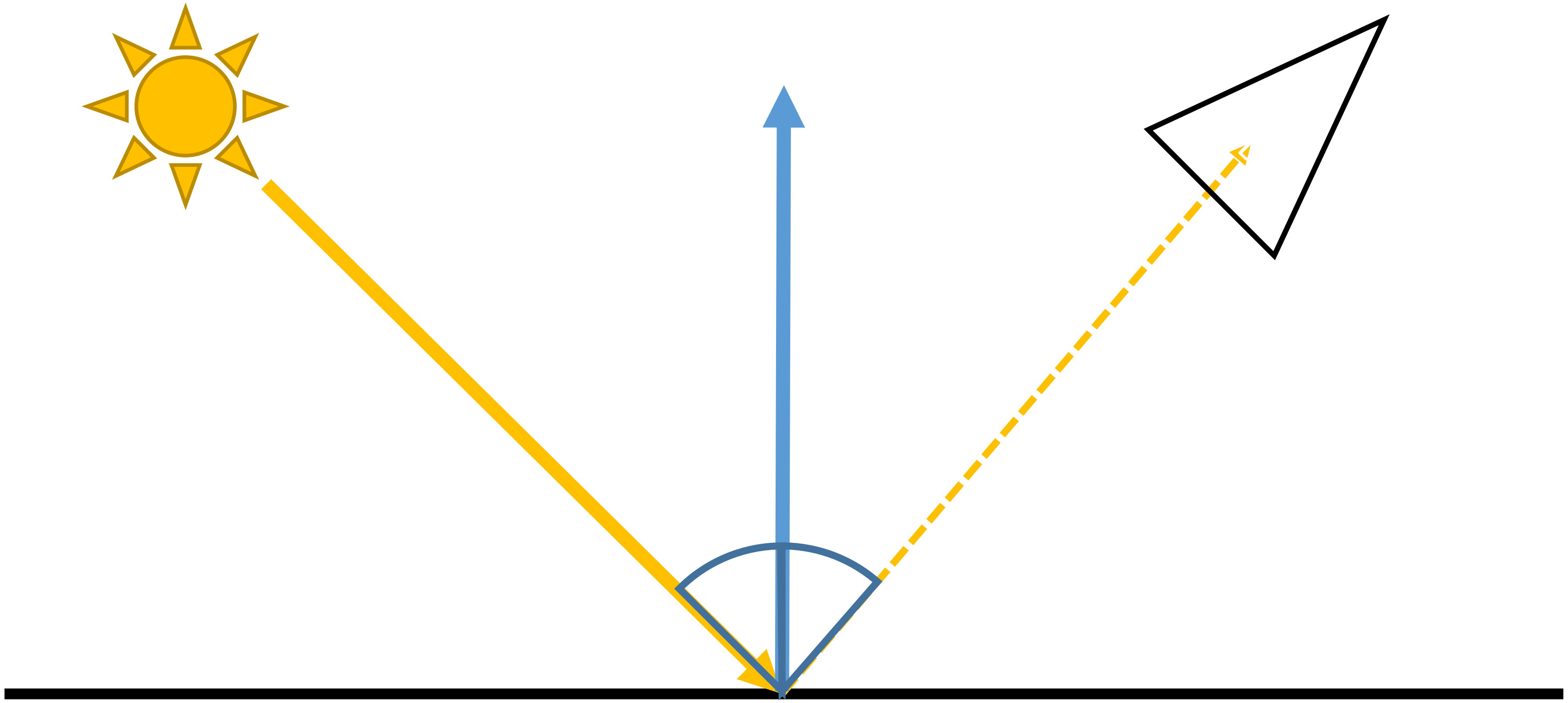
Specular Light (how to fake it)



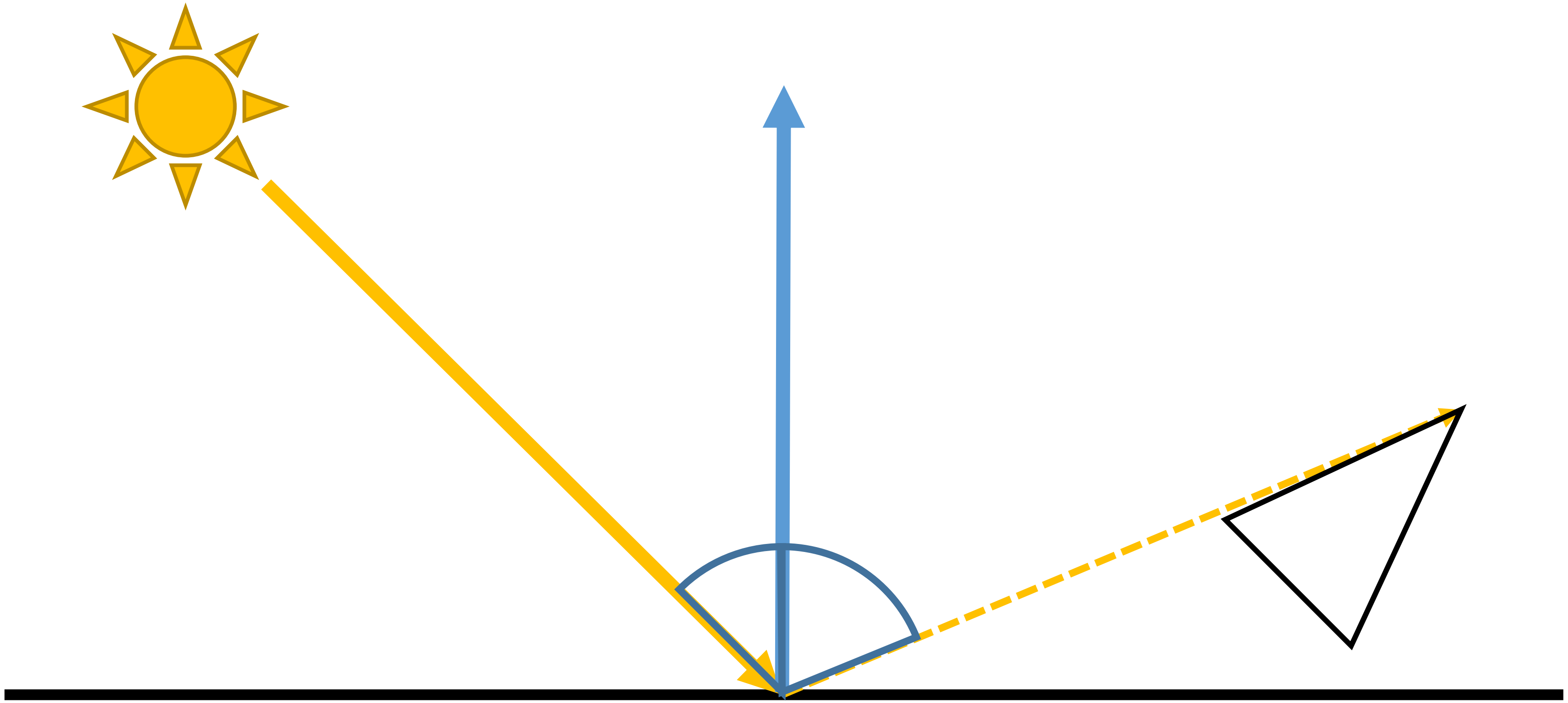
Specular Light (how to fake it)



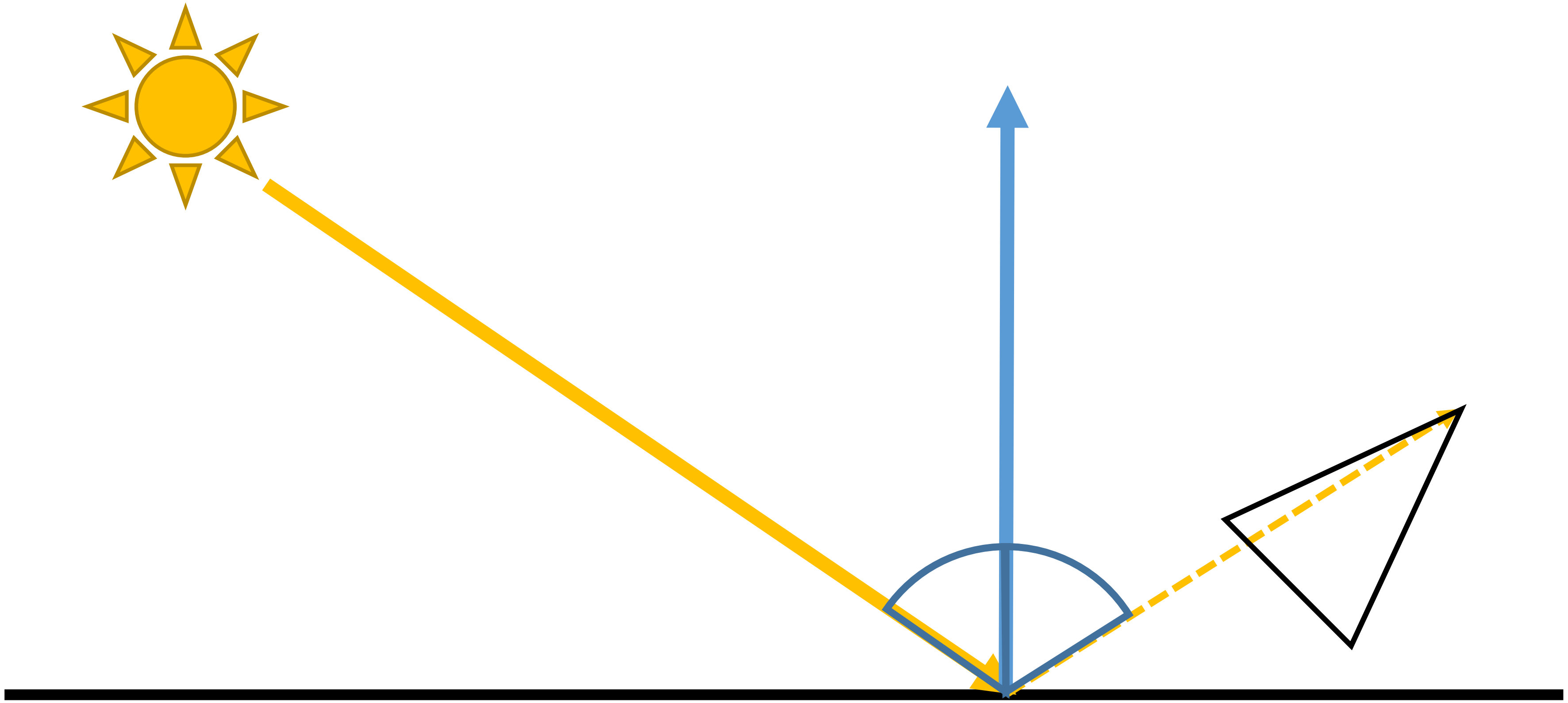
Specular Light (how to fake it)



Specular Light (how to fake it)



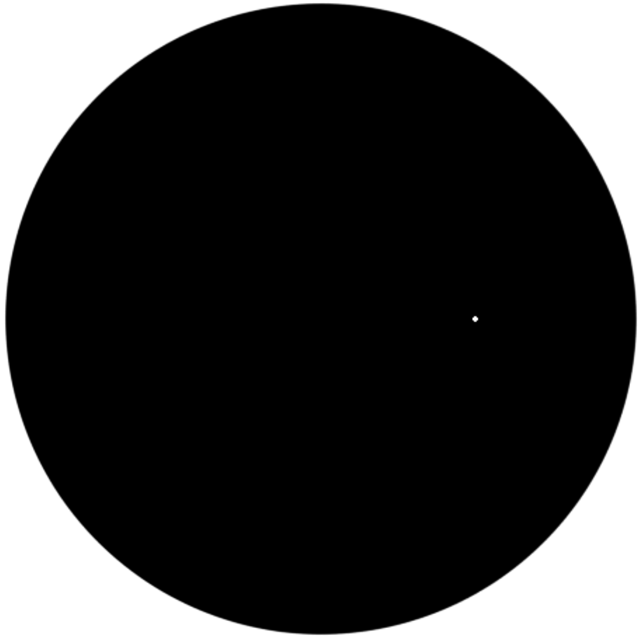
Specular Light (how to fake it)



Specular Light (how to fake it)

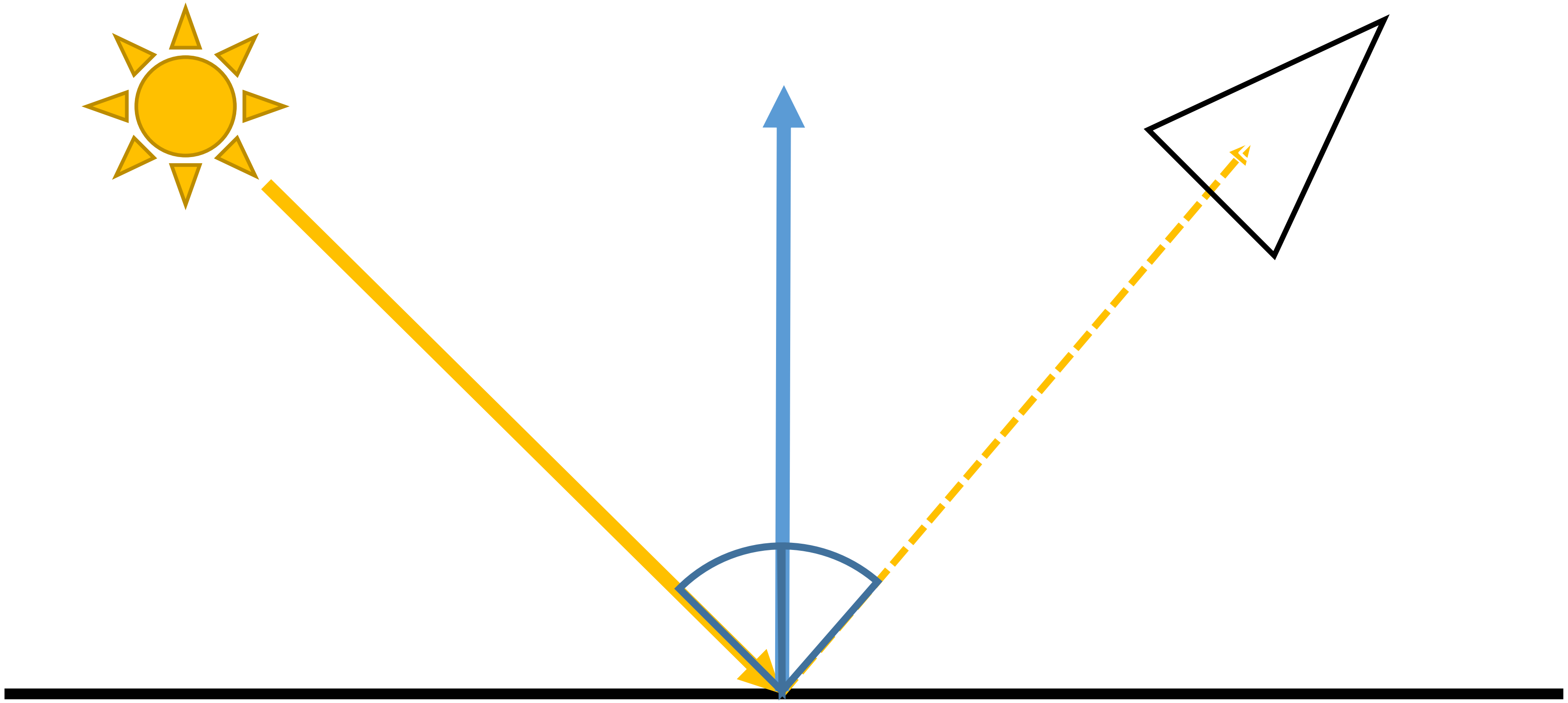
- One Highlight

```
float3 lightDir = normalize(i.pos - lightPos);
float3 viewDir = normalize(i.pos - cameraPos);

float lightAngle = dot(normal, lightDir);
float viewAngle = dot(normal,  viewDir);
bool isReflected = lightAngle == viewAngle;

if(isReflected)
{
        //ADD SPECULAR STUFF
}
```

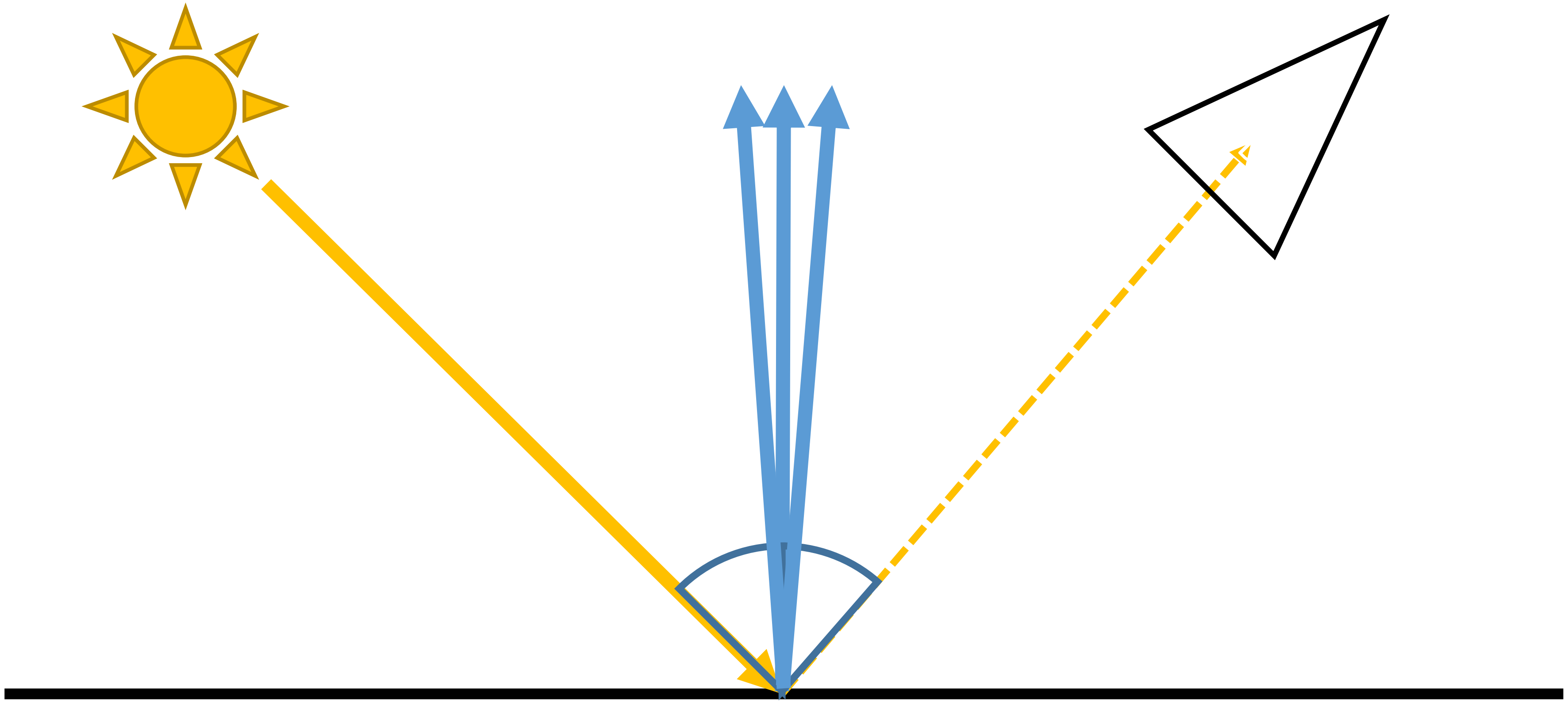


ROUGHNESS!

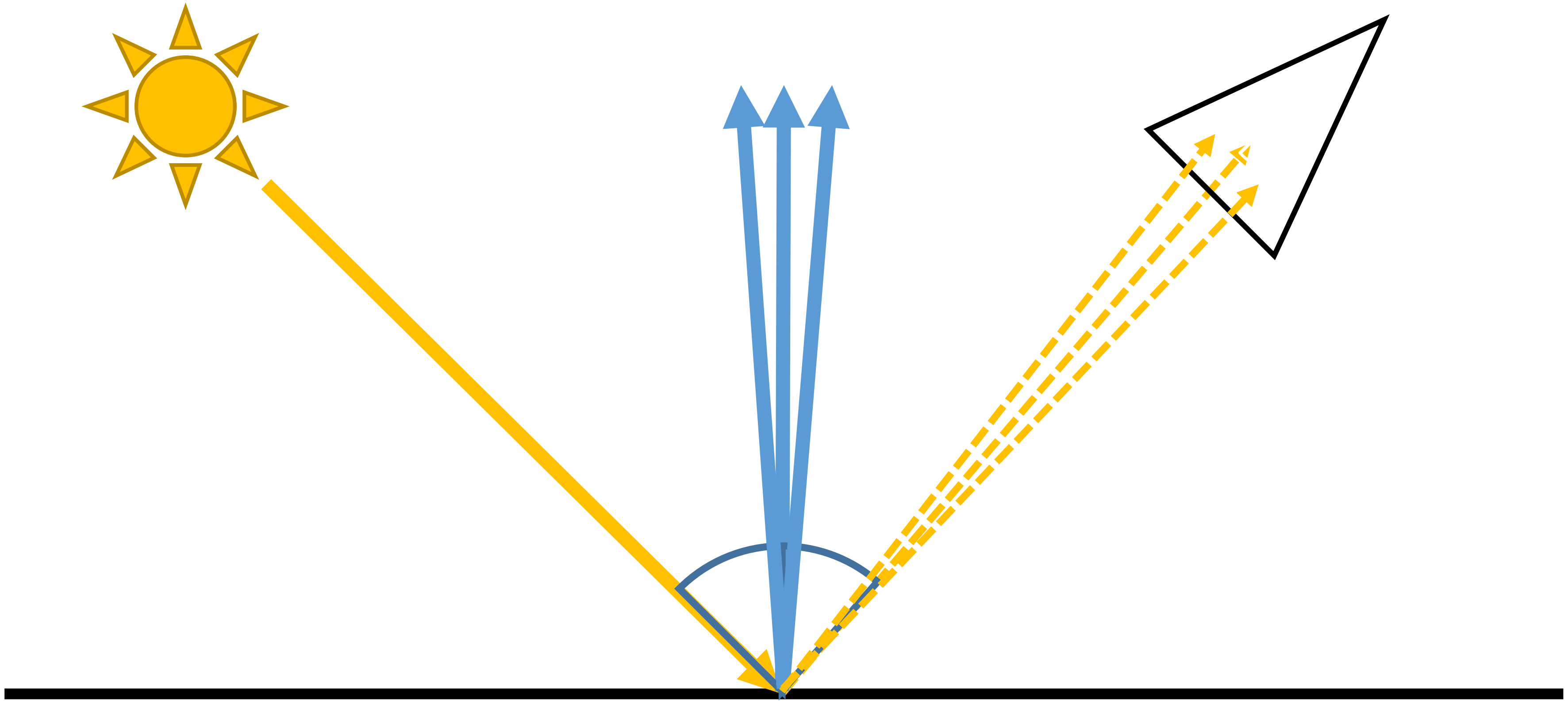
Specular Light (how to fake it)



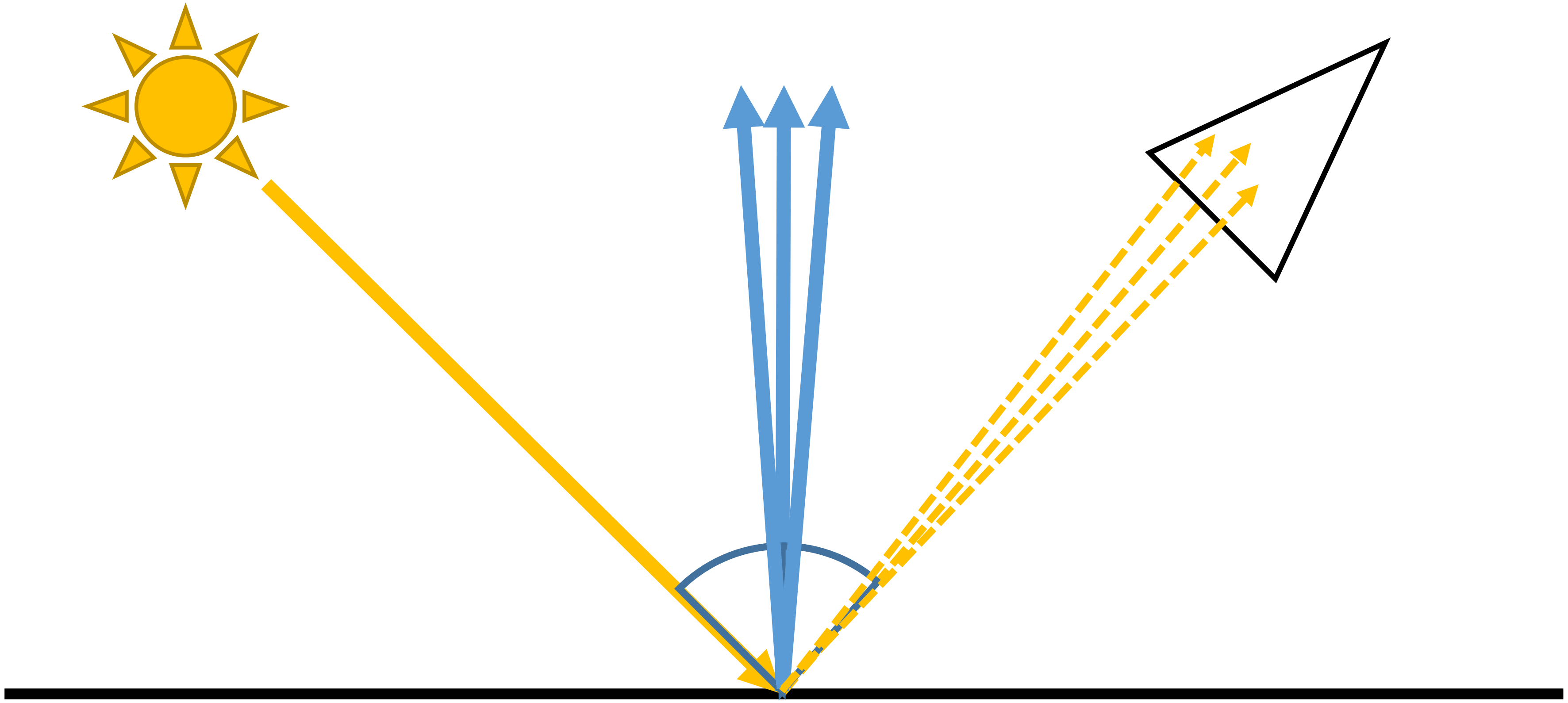
Specular Light (how to fake it)



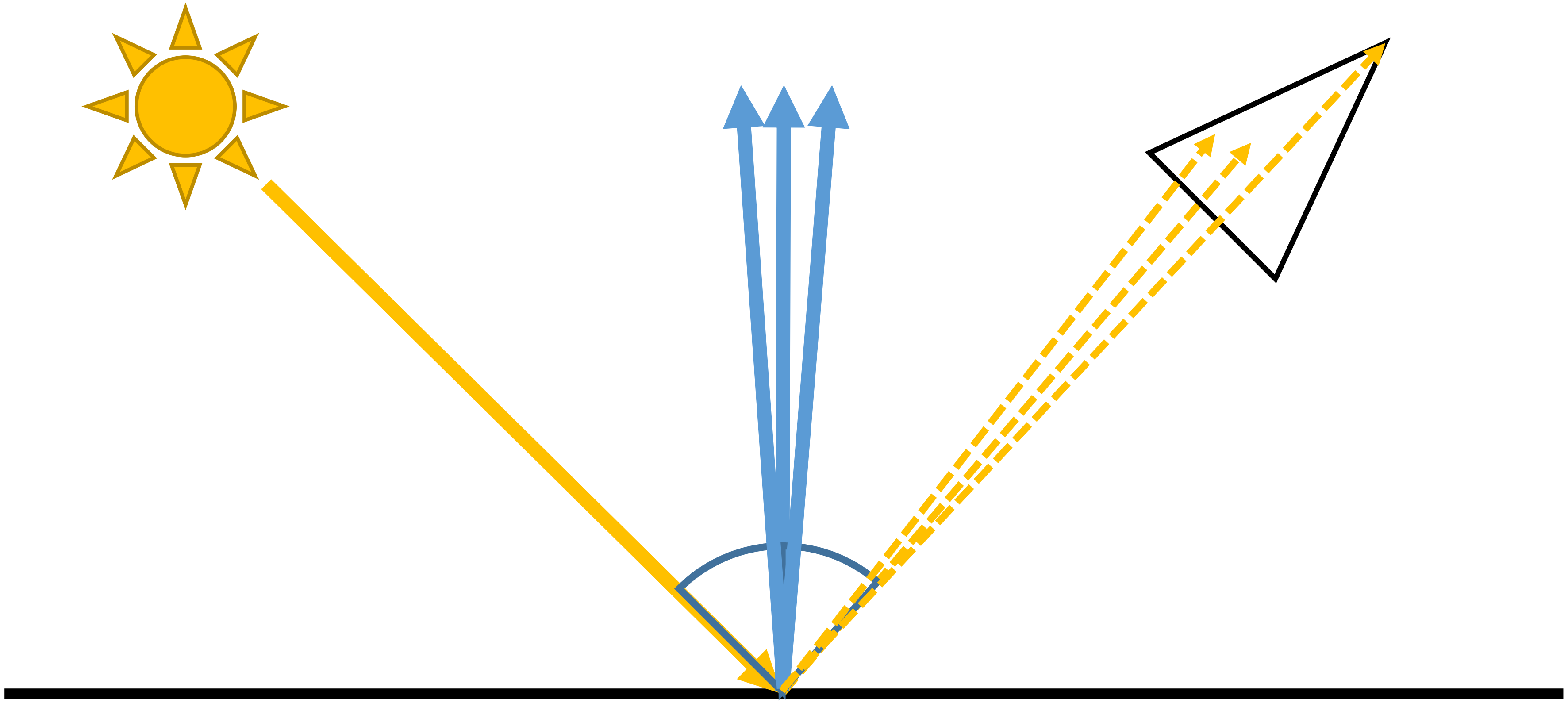
Specular Light (how to fake it)



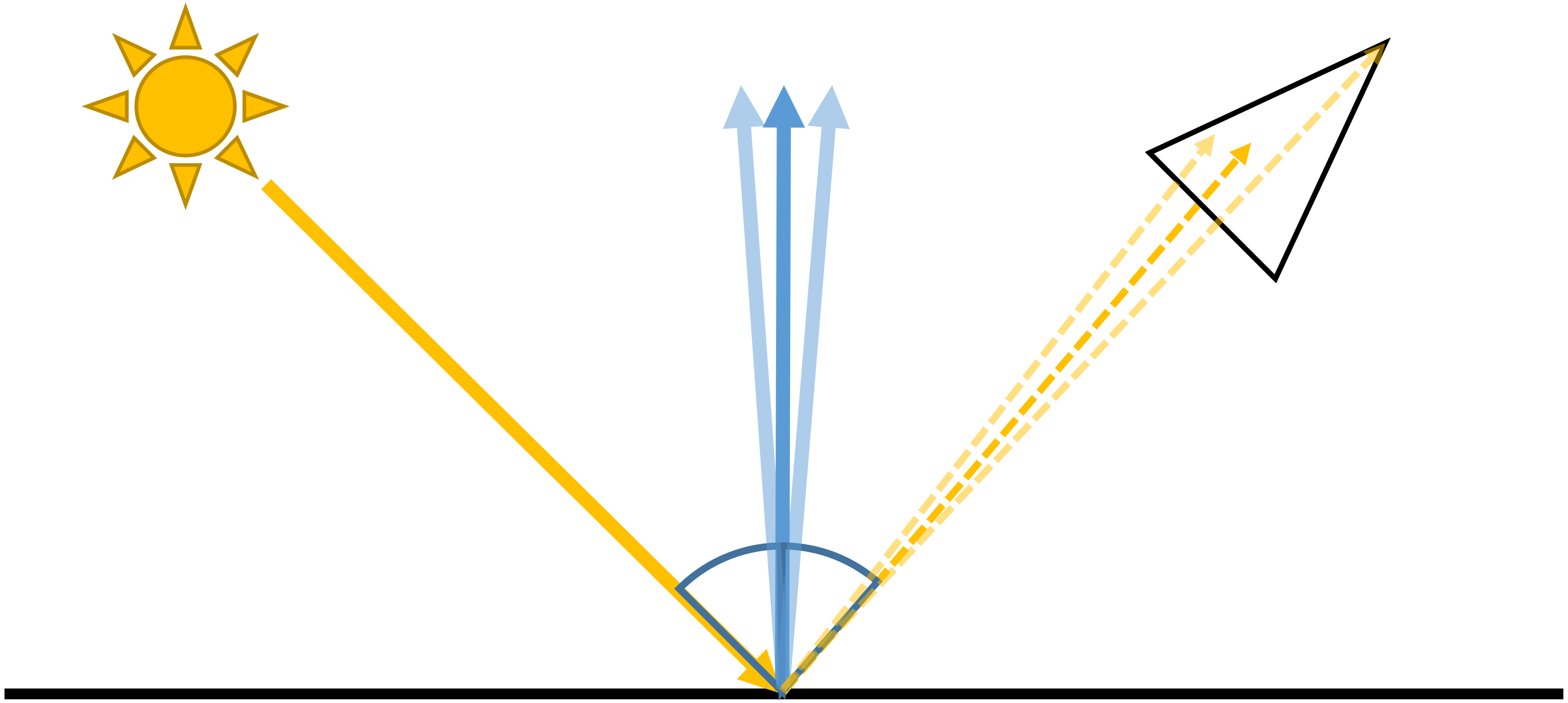
Specular Light (how to fake it)



Specular Light (how to fake it)



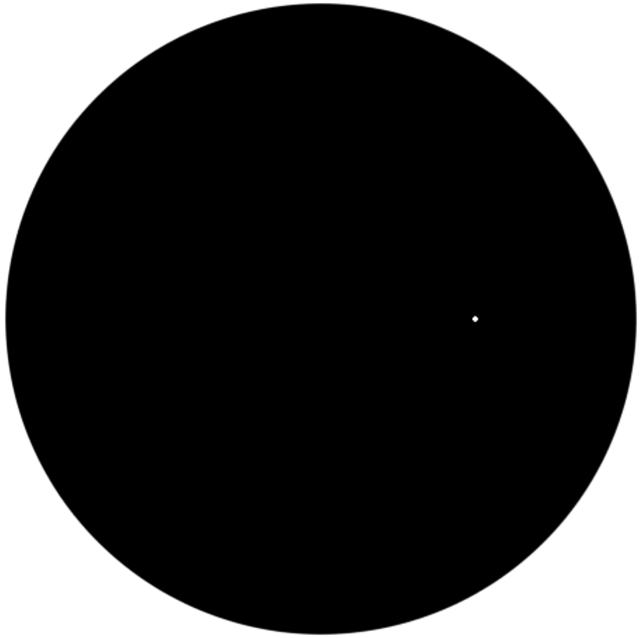
Specular Light (how to fake it)



Specular Light (how to fake it)

- One Highlight

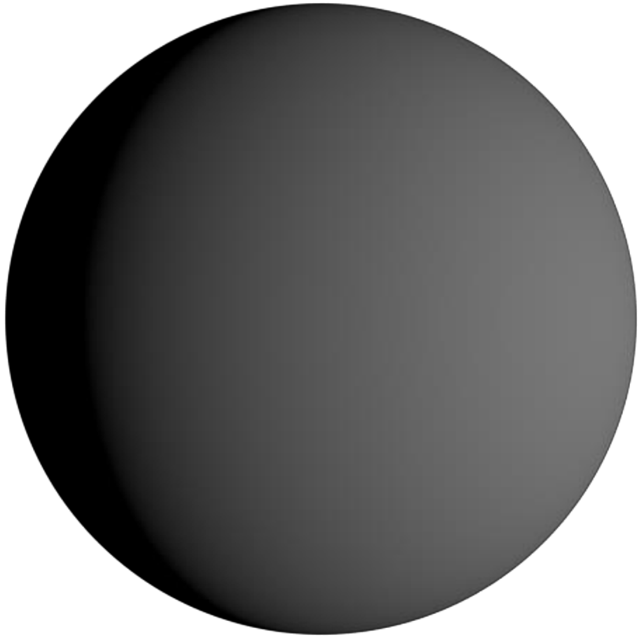
```
float3 lightDir = normalize(i.pos - lightPos);
float3 viewDir = normalize(i.pos - cameraPos);

float lightAngle = dot(normal, lightDir);
float viewAngle = dot(normal,  viewDir);
bool isReflected = lightAngle == viewAngle;

if(isReflected)
{
        //ADD SPECULAR STUFF
}
```



Specular Light (how to fake it)

- One Highlight

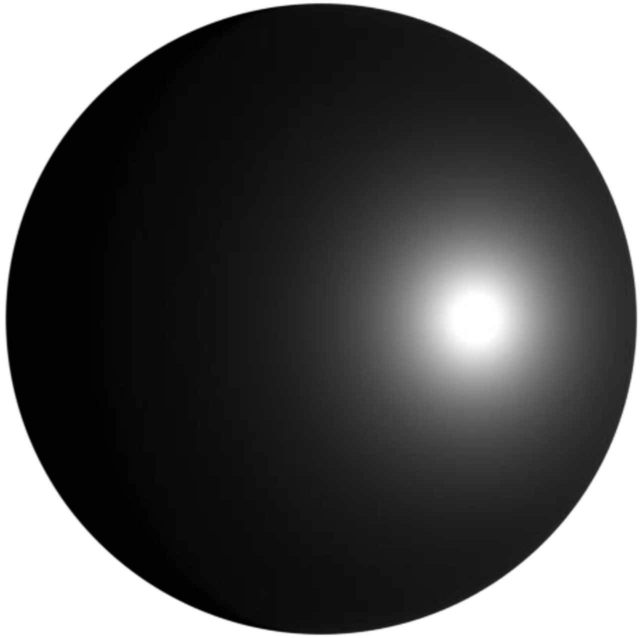
```
float3 lightDir = normalize(i.pos - lightPos);
float3 viewDir  = normalize(i.pos - cameraPos);

float lightAngle = dot(normal, lightDir);
float viewAngle = dot(normal,  viewDir);
float angleDiff = 1-abs(lightAngle - viewAngle);

float specular = angleDiff;
```



Specular Light (how to fake it)

- One Highlight

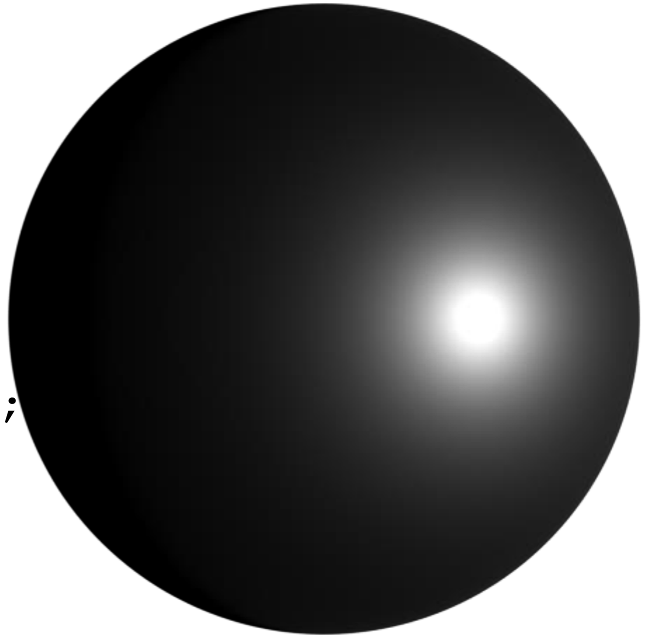
```
float3 lightDir = normalize(i.pos - lightPos);
float3 viewDir  = normalize(i.pos - cameraPos);

float lightAngle = dot(normal, lightDir);
float viewAngle = dot(normal,  viewDir);
float angleDiff = 1-abs(lightAngle - viewAngle);

float specular = pow(angleDiff, glossiness);
```



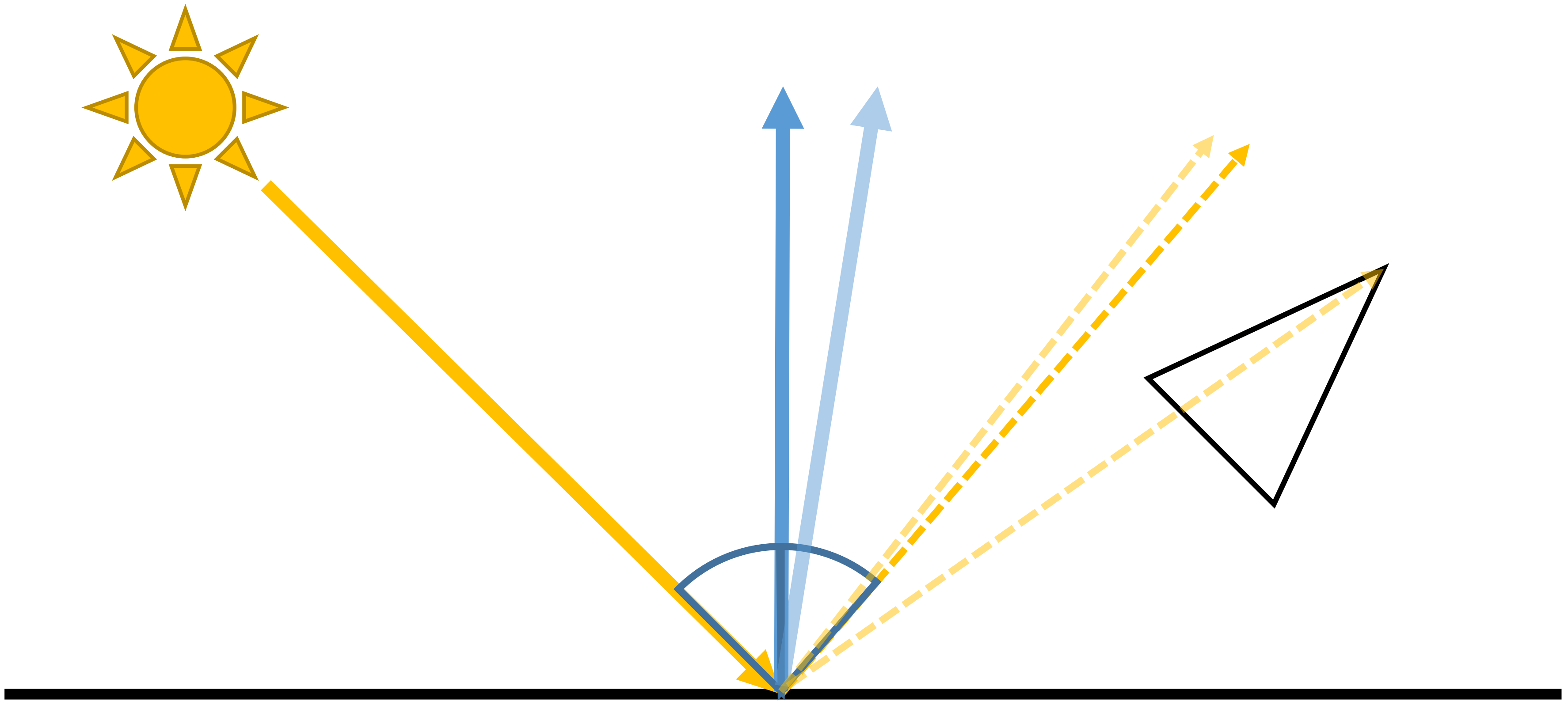
Specular Light (how to fake it)

- One Highlight

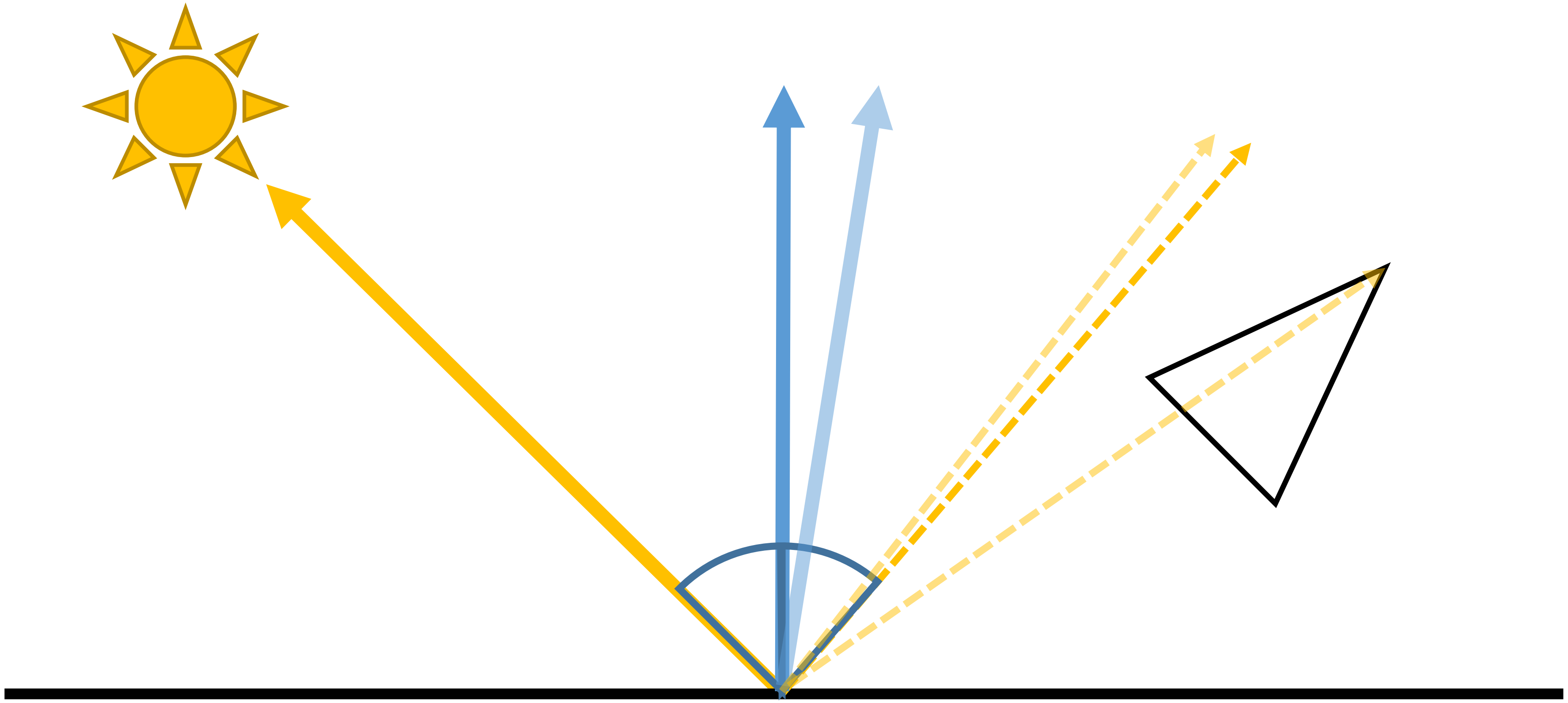
```
float3 lightDir = normalize(lightPos - i.pos);
float3 viewDir  = normalize(i.pos - cameraPos);

float3 halfLightView = normalize(lightDir + viewDir);

float halfAngle = dot(normal, halfLightView);

float blinnPhong = pow(halfAngle, glossiness);
```



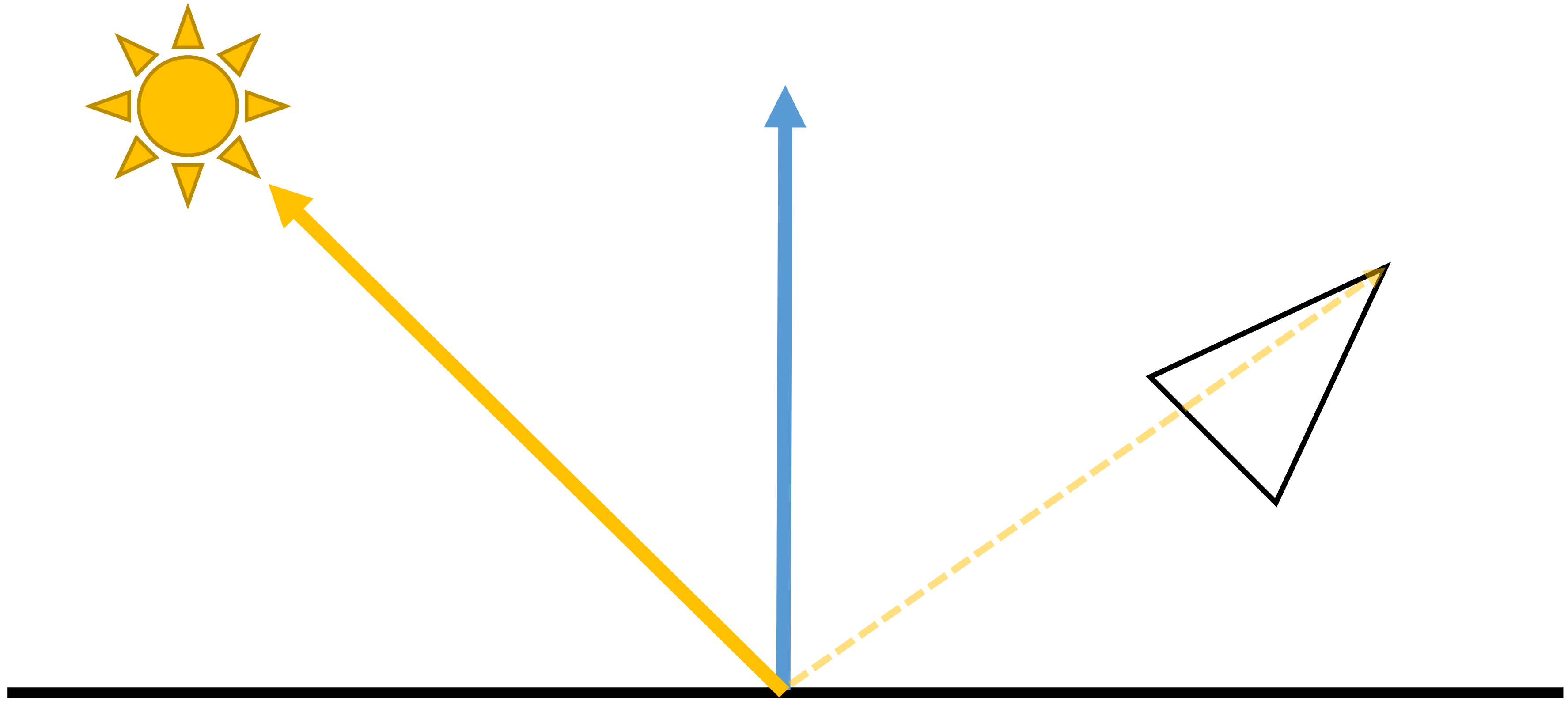
Specular Light (how to fake it)



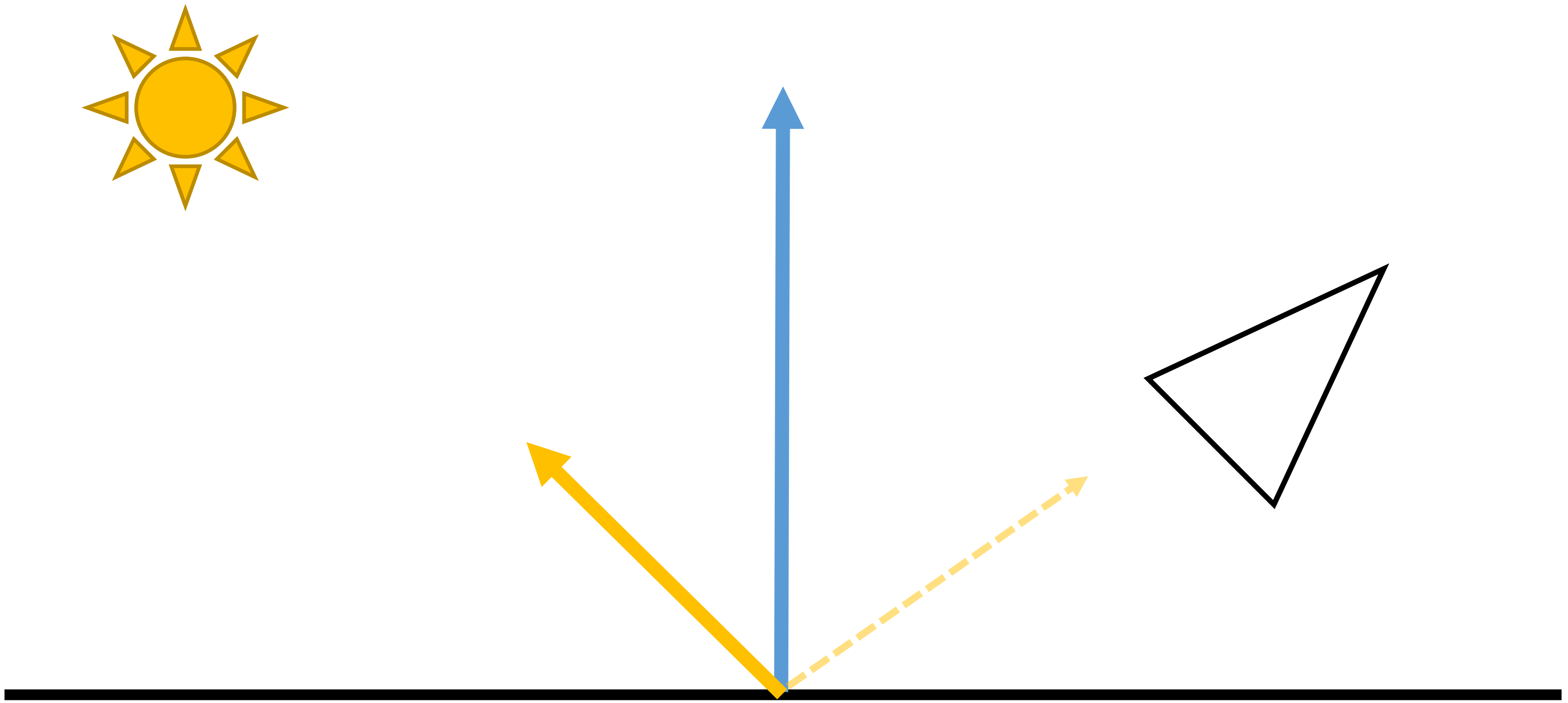
Specular Light (how to fake it)



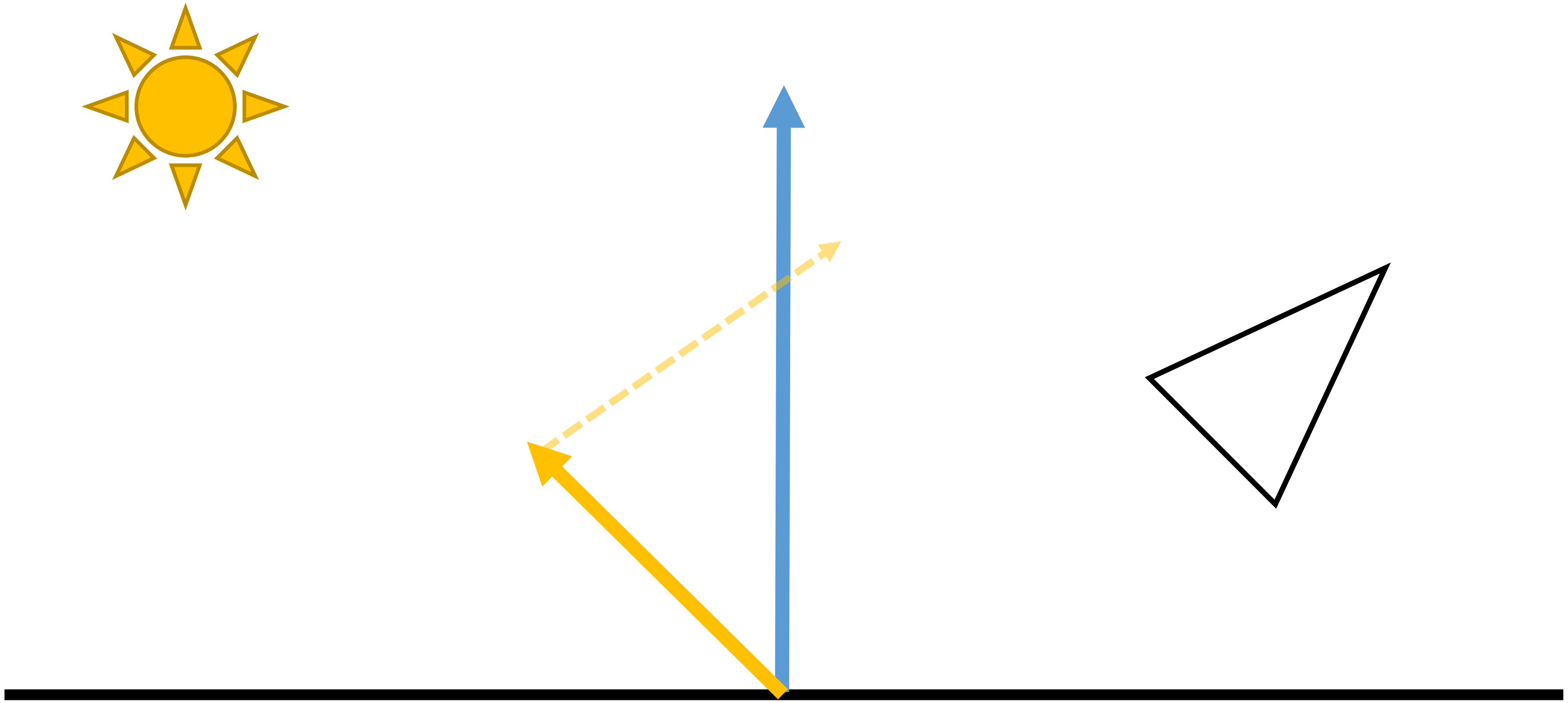
Specular Light (how to fake it)



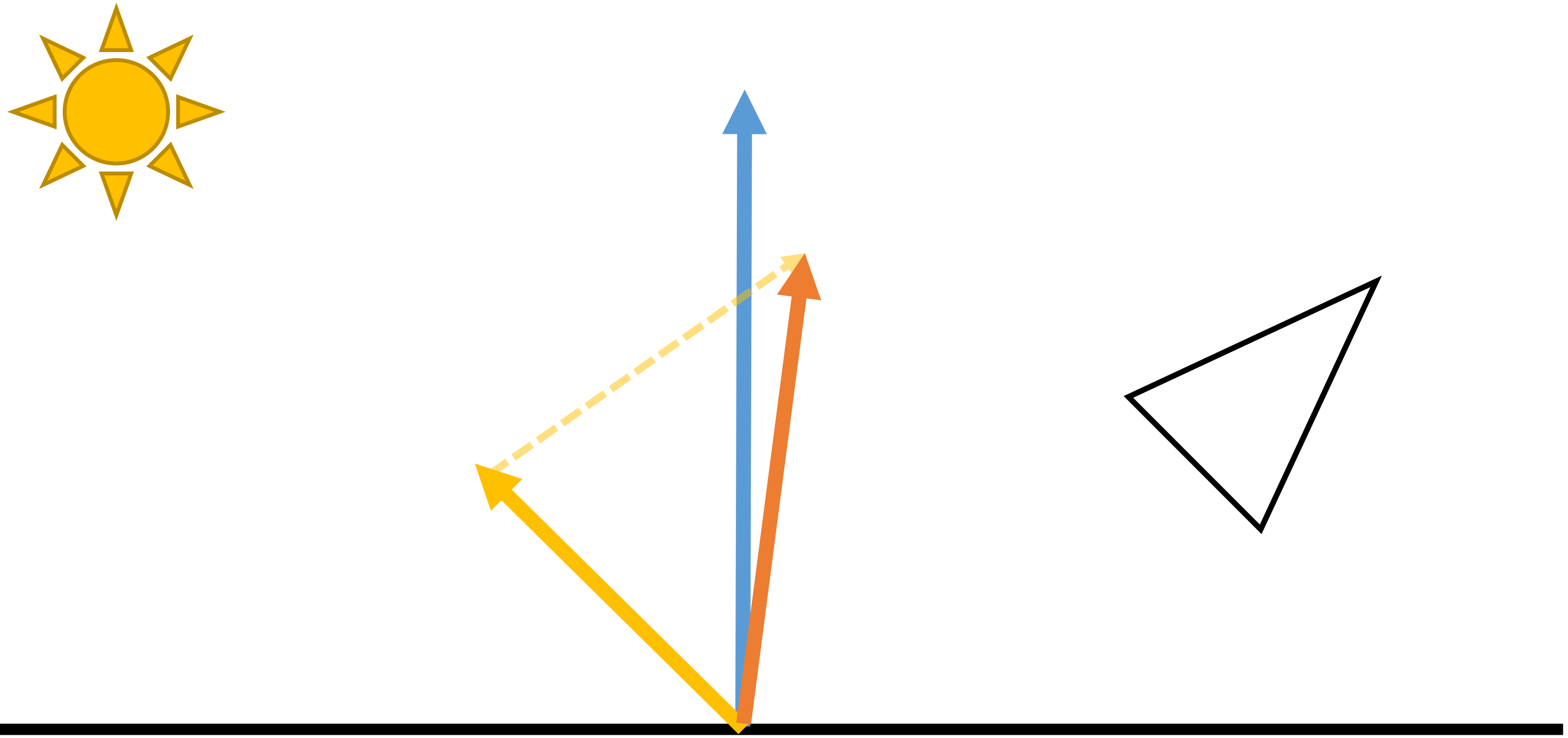
Specular Light (how to fake it)



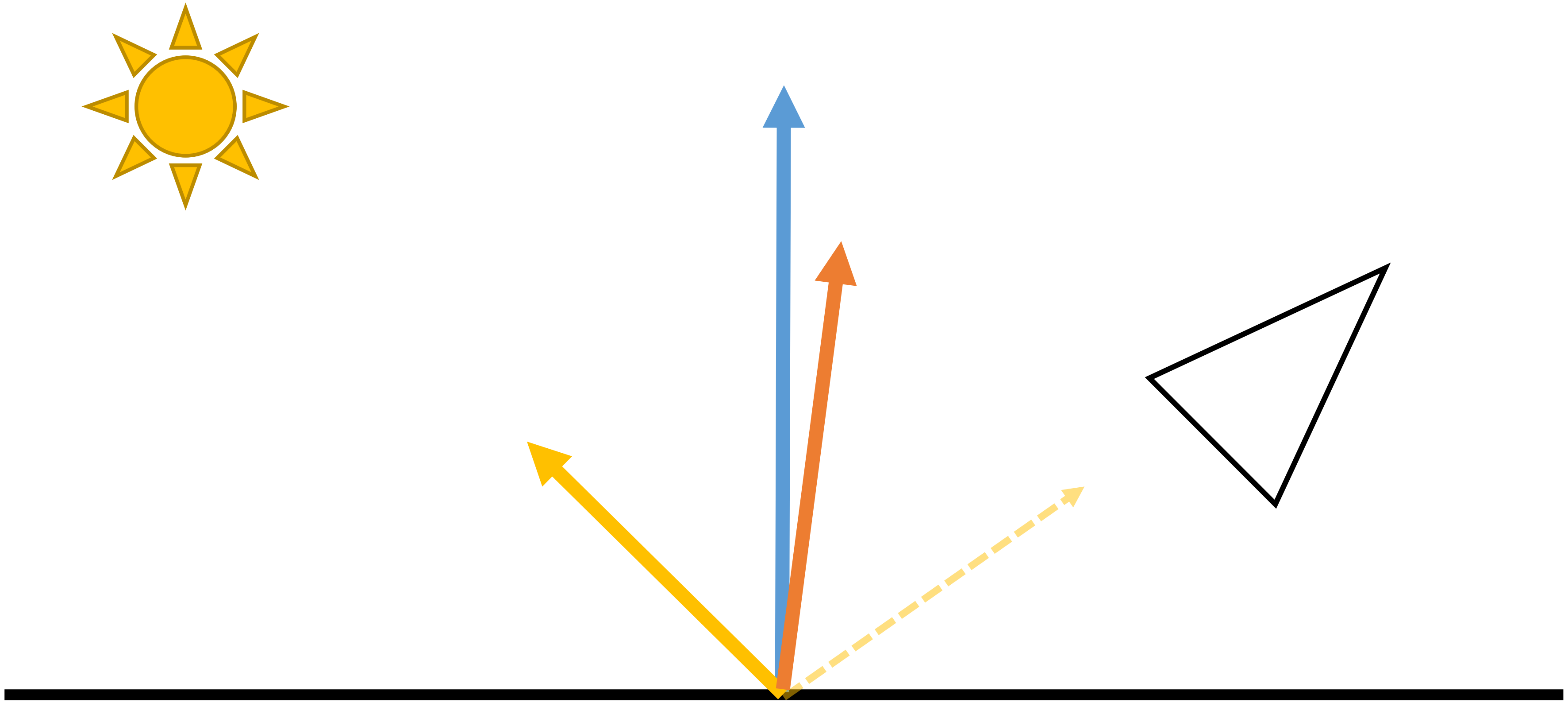
Specular Light (how to fake it)



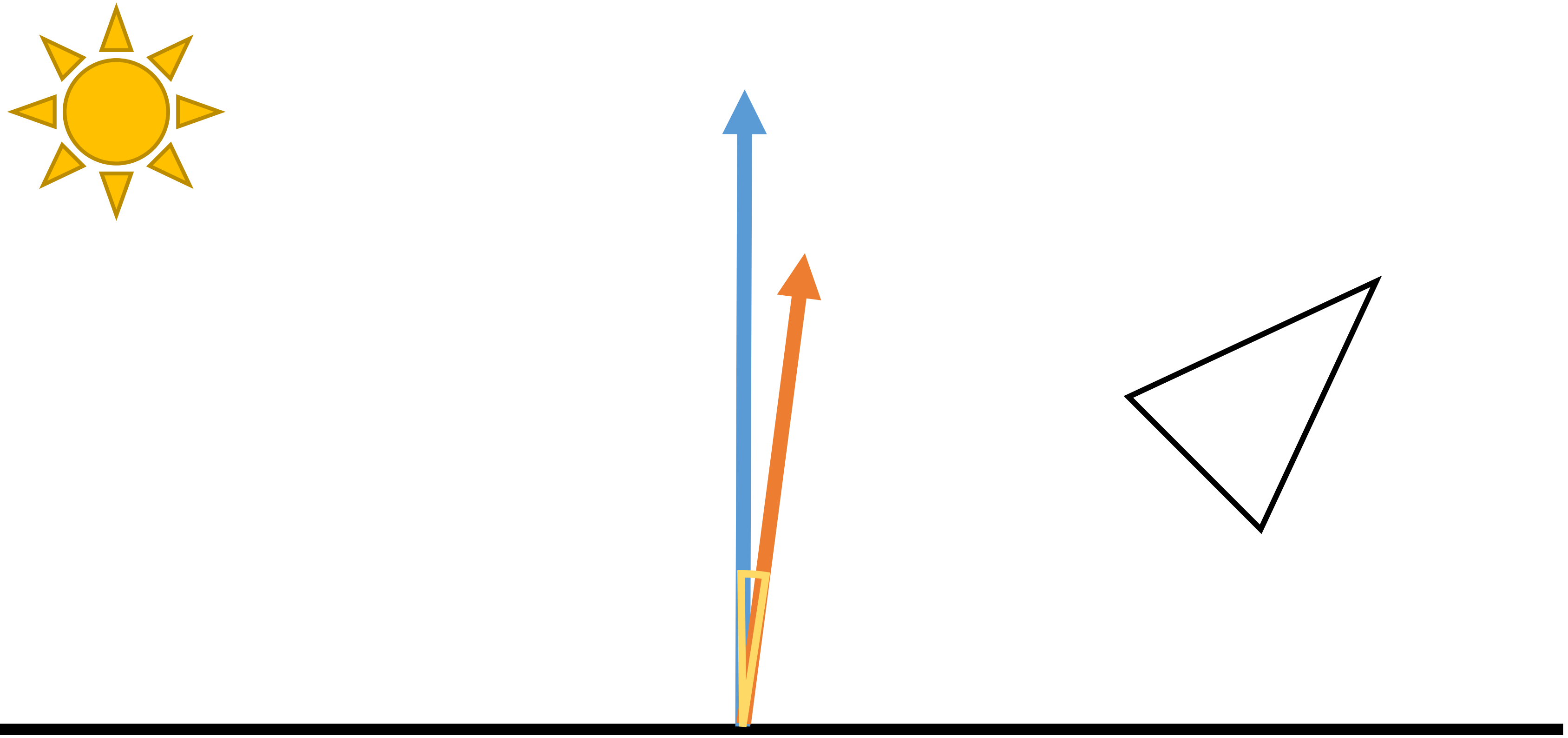
Specular Light (how to fake it)



Specular Light (how to fake it)



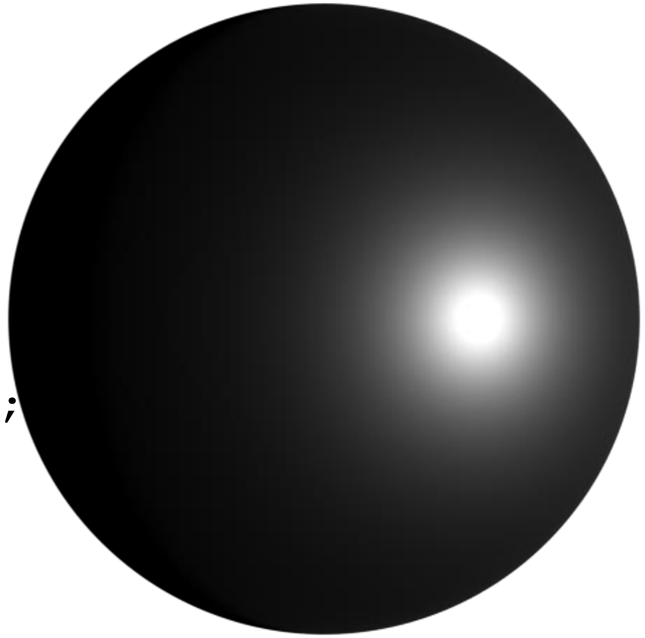
Specular Light (how to fake it)



Specular Light (how to fake it)

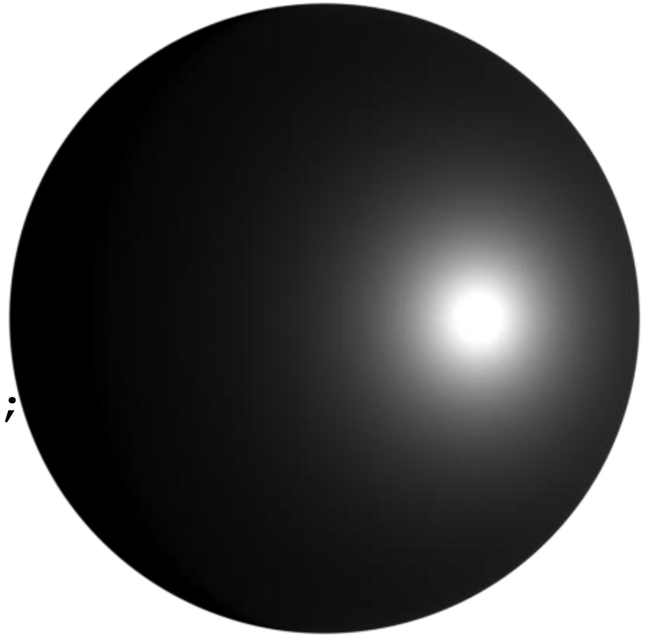
- One Highlight

```
float3 lightDir = normalize(lightPos - i.pos);
float3 viewDir  = normalize(i.pos - cameraPos);

float3 halfLightView = normalize(lightDir + viewDir);

float halfAngle = dot(normal, halfLightView);

float blinnPhong = pow(halfAngle, glossiness);
```



Specular Light (how to fake it)

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



```
float3 lightDir = normalize(lightPos - i.pos);
float3 viewDir  = i.pos - cameraPos);

float3 halfLightView = normalize(lightDir + viewDir);

float halfAngle = dot(normal, halfLightView);

float blinnPhong = pow(halfAngle, glossiness);

float4 specular = falloff * blinnPhong * _LightColor * _SpecularColor;
```