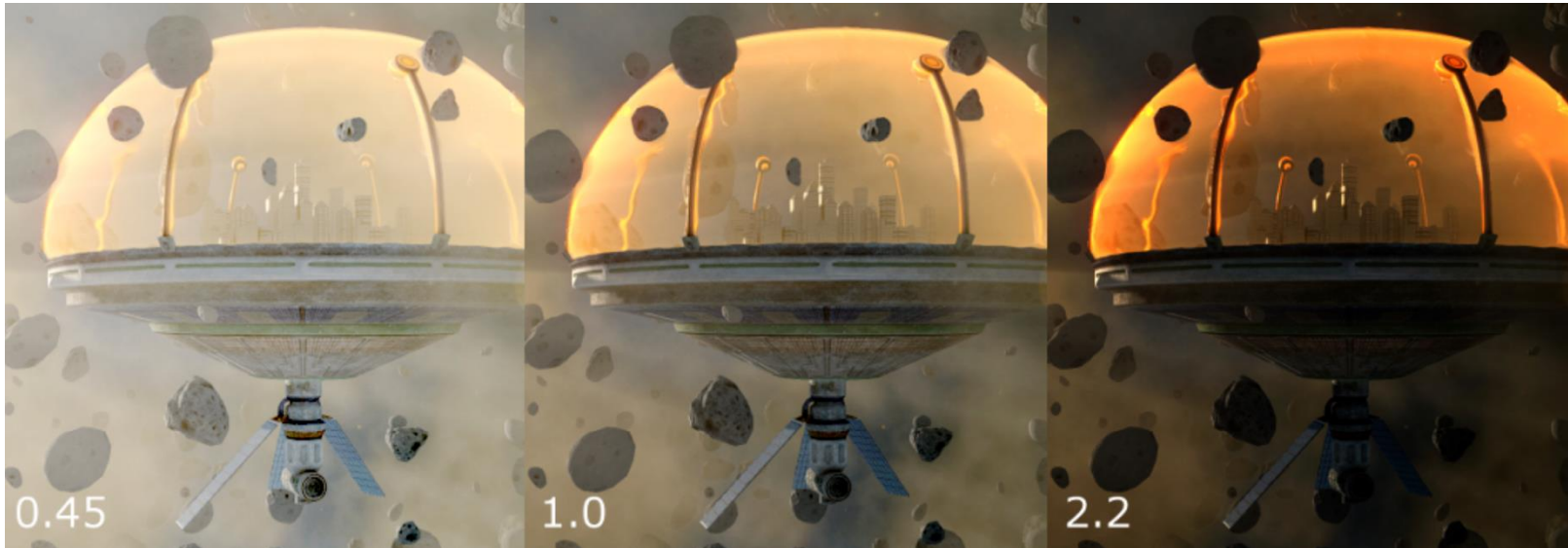


# Computação Gráfica

Raytracing 2

# Gamma space vs linear space

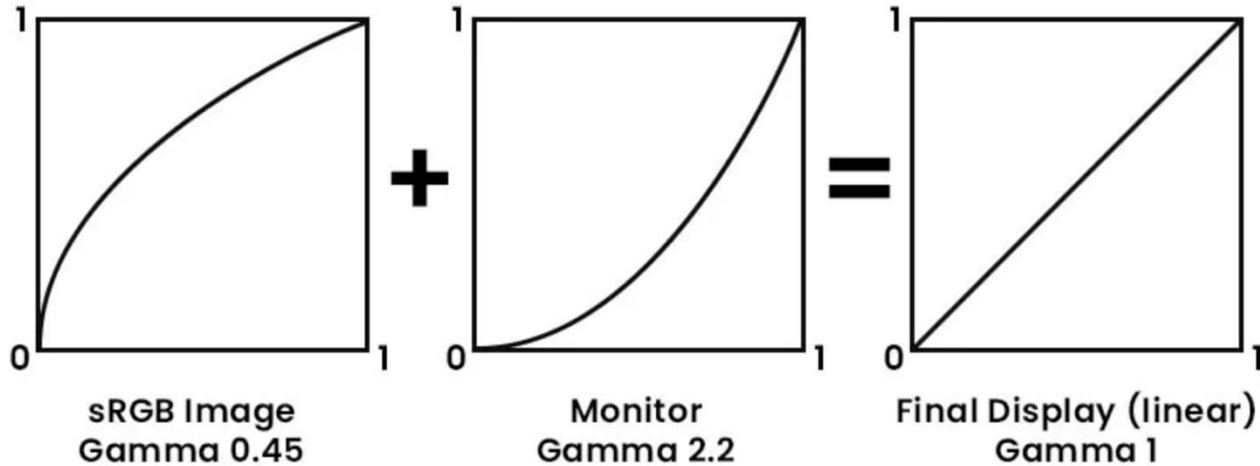
- **Armazenamento de iluminação em computadores:** Geralmente feito no formato sRGB, com correção gama.
- **Câmeras:** Capturam iluminação de forma linear, onde mais luz resulta em números proporcionalmente maiores.
- **Percepção humana:** Segue uma curva logarítmica, percebendo mais mudanças em áreas escuras e menos em áreas claras (vantagem evolutiva para detectar predadores no escuro).
- **Espaço de cores linear:** Armazena as informações de iluminação de forma mais realista, permitindo a aplicação de efeitos físicos realistas antes de converter para exibição final.



# Gamma space vs linear space

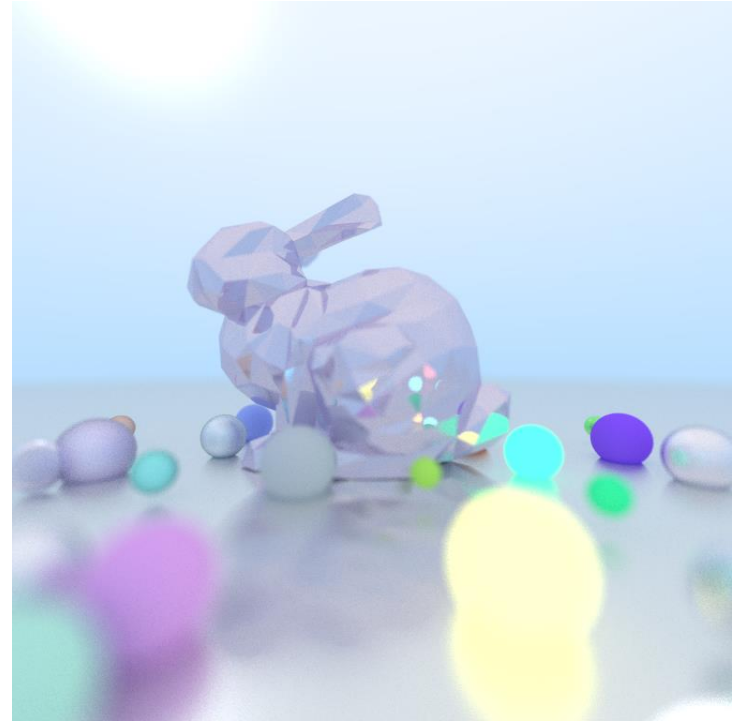
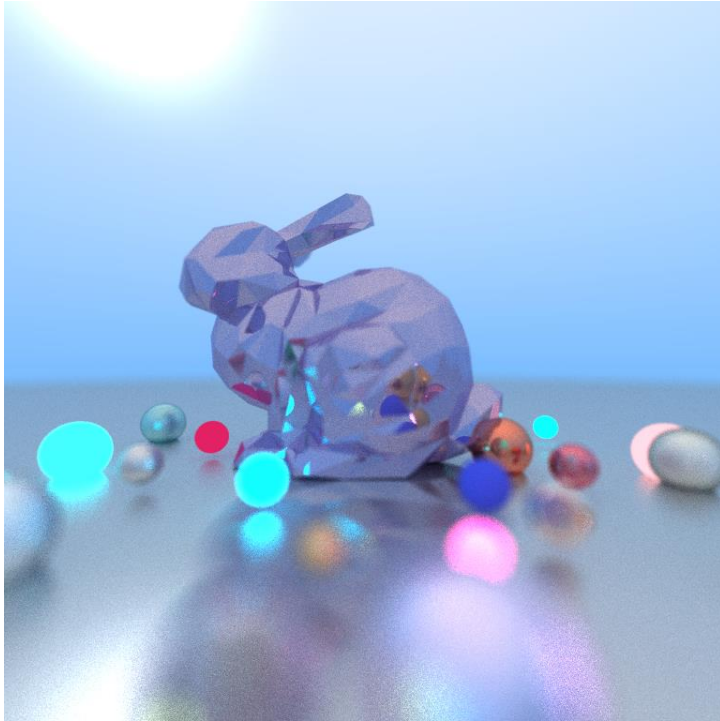
```
fn linear_to_gamma_channel(channel: f32) -> f32
{
    return pow(channel, 0.4545);
}

fn linear_to_gamma(color: vec3f) -> vec3f
{
    return vec3f(linear_to_gamma_channel(color.x),
                 linear_to_gamma_channel(color.y),
                 linear_to_gamma_channel(color.z));
}
```



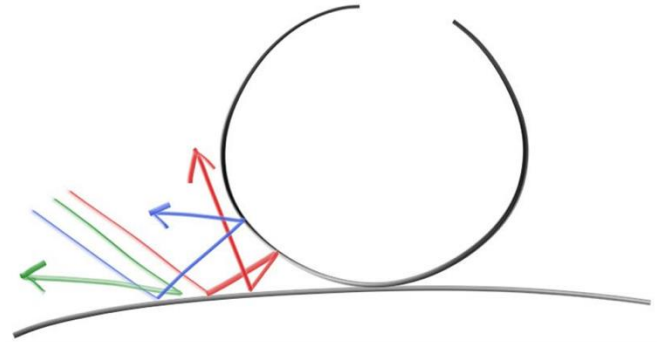
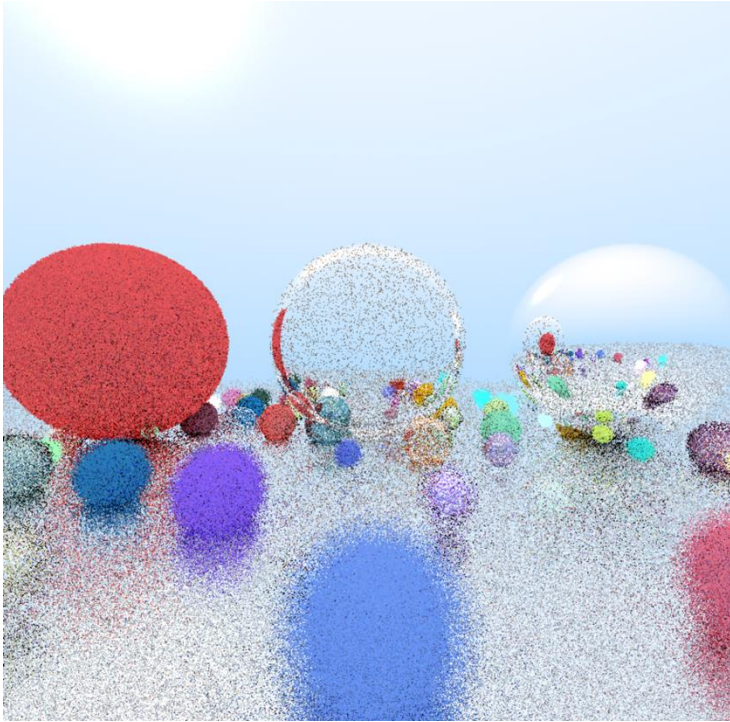
# Gamma space vs linear space

Resultado no projeto:



# Accumulate frames

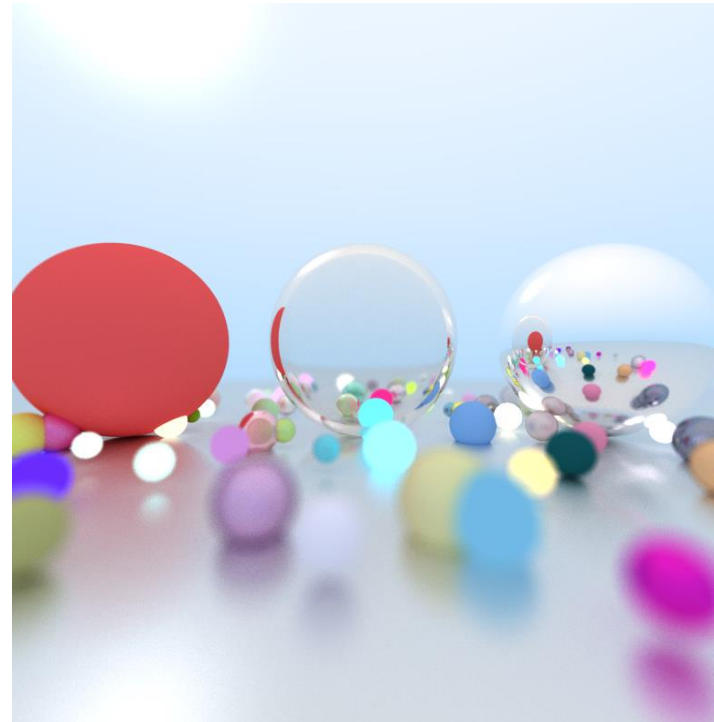
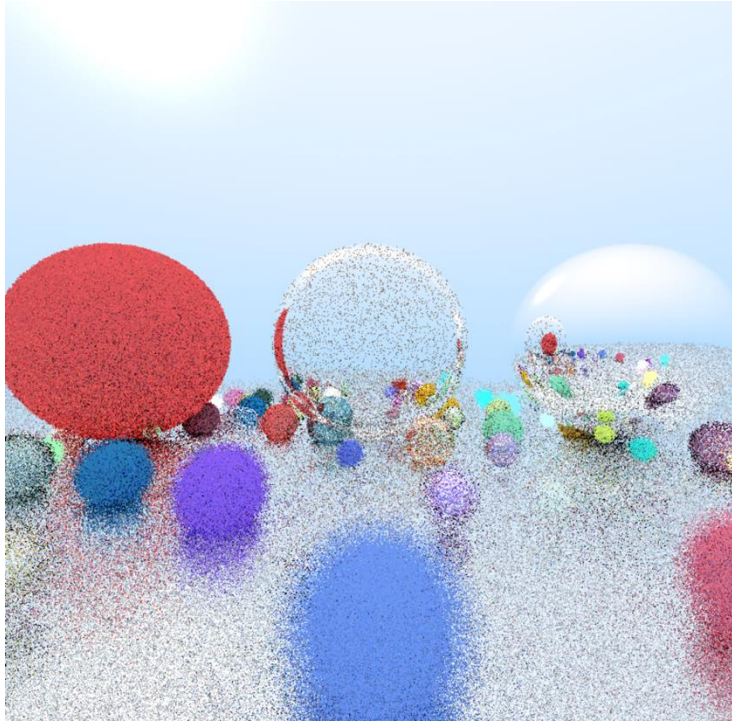
Porque isso acontece?





# Accumulate frames

Como podemos melhorar?



# Accumulate frames

O que acontece se não tivermos "pesos" nos frames?



```
// Solucao:  
// pra que serve?  
var should_accumulate = uniforms[3];  
  
var accumulated_color = cor_anterior * should_accumulate + color_saida;  
  
rtb[map_fb] = accumulated_color;  
frameb[map_fb] = accumulated_color / peso;
```

# Reflexão em Materiais Espelhados

Para calcular o vetor de reflexão (vermelho) somaremos o vetor de entrada ( $v$ ) por duas vezes  $B$ .

$B$  tem a direção e sentido de  $N$  com um comprimento de  $|v \cdot N|$ .

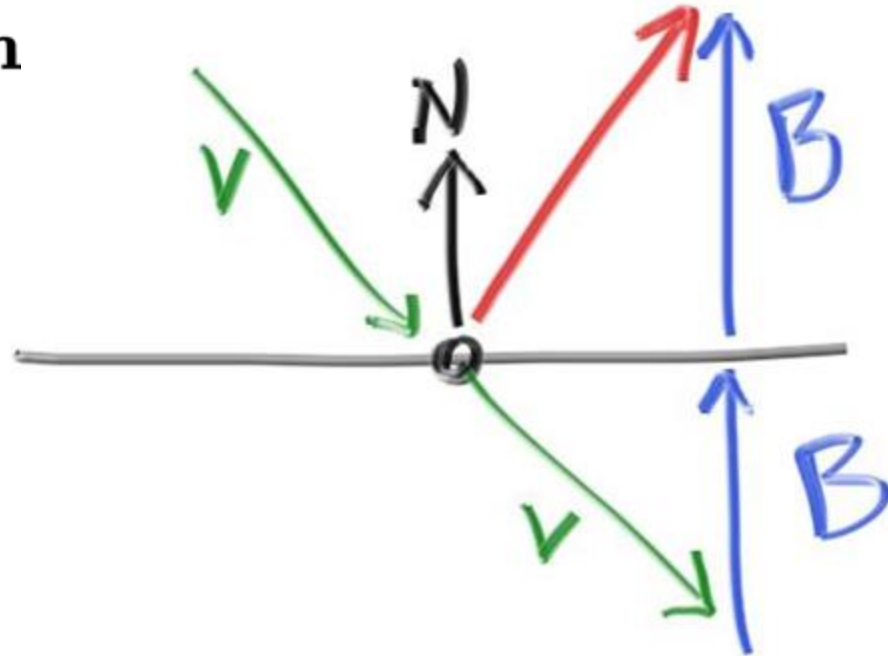
$$\mathbf{R} = \mathbf{v} - 2 \cdot (\mathbf{v} \cdot \mathbf{n}) \cdot \mathbf{n}$$

Em WGSL: `reflect()`

A variável `smoothness` controla a componente metálica no projeto.

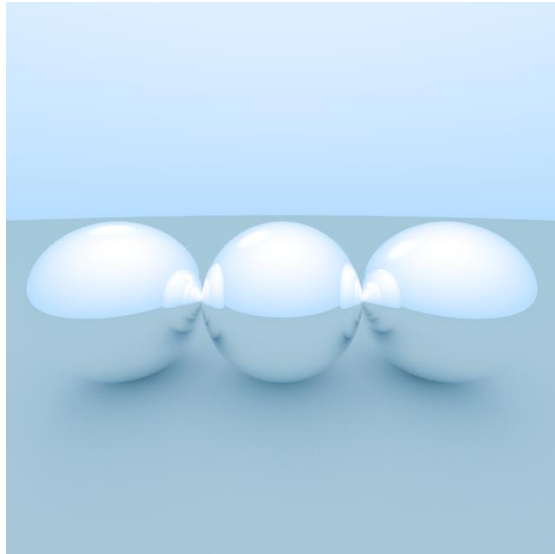
0 - Lambertiano

1 - Metálico

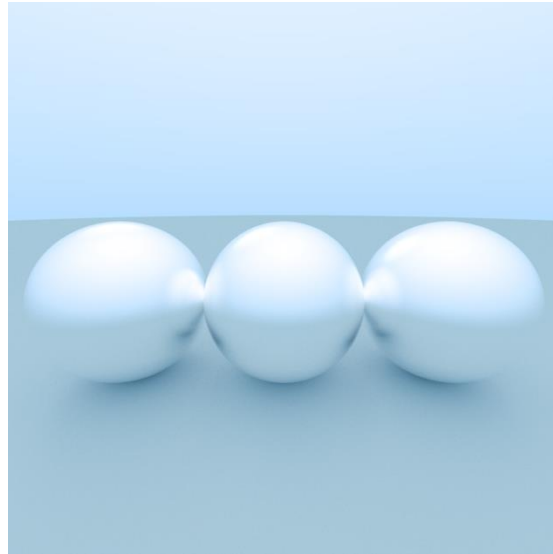




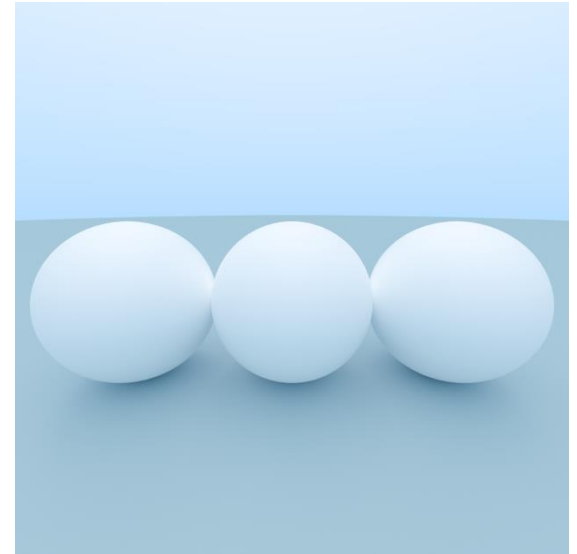
# Reflexão em Materiais Espelhados - Resultados



Smoothness 1.0

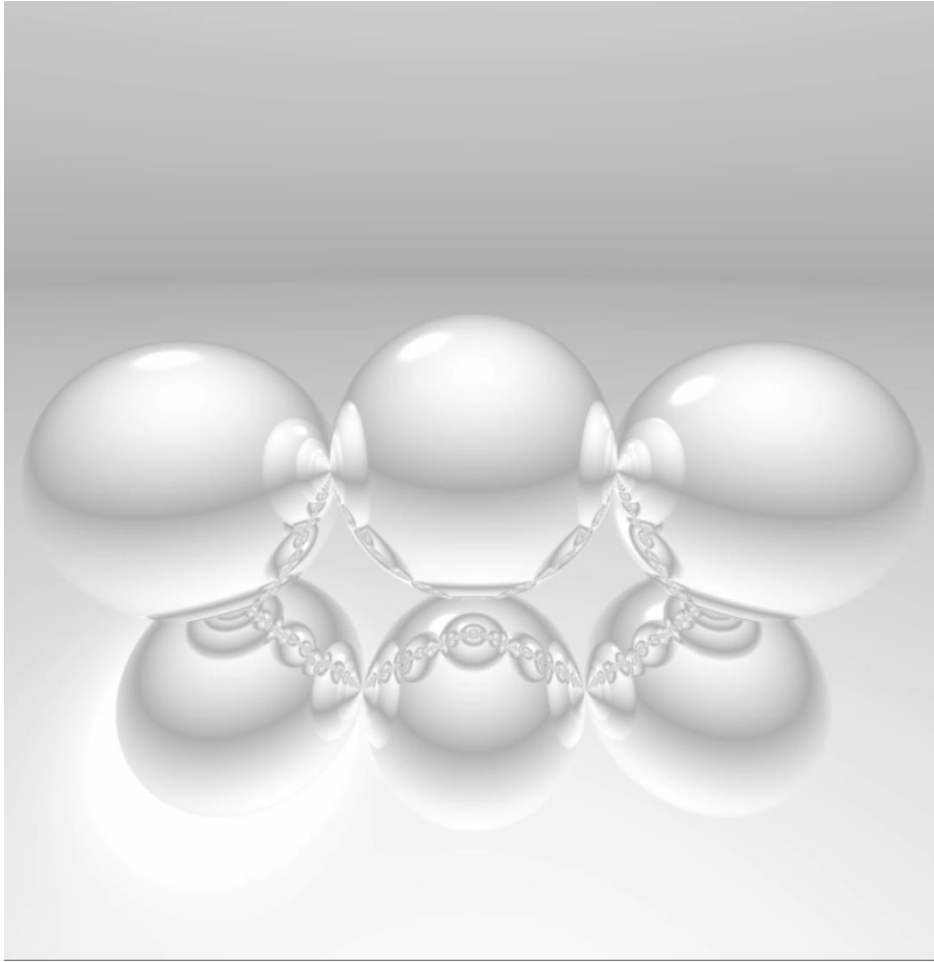


Smoothness 0.8



Smoothness 0.2

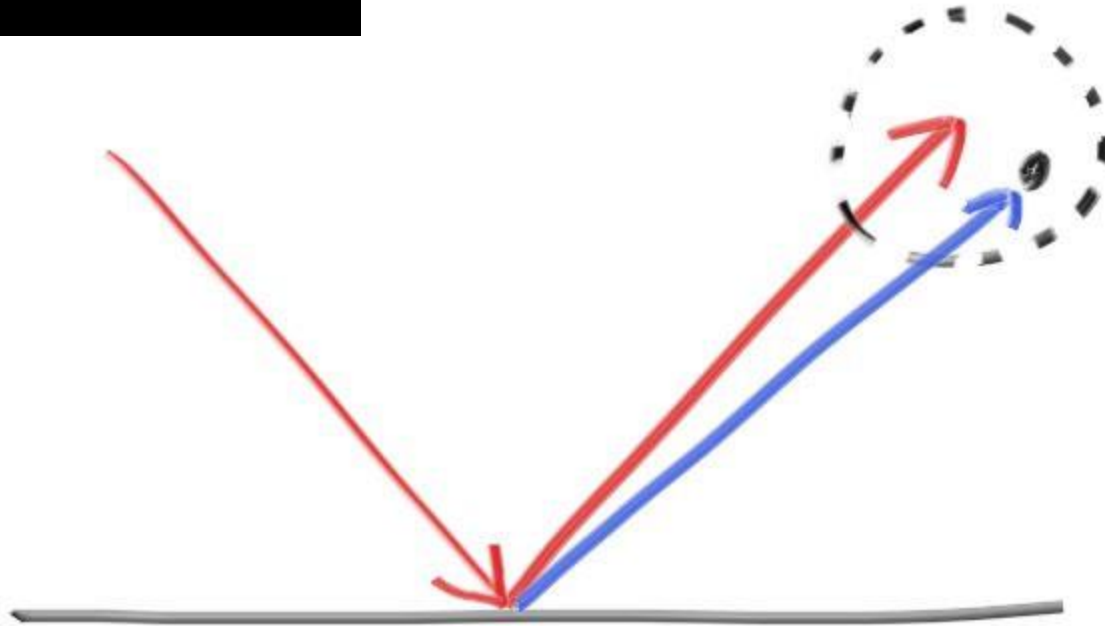
# Reflexão em Materiais Espelhados – Mundo real



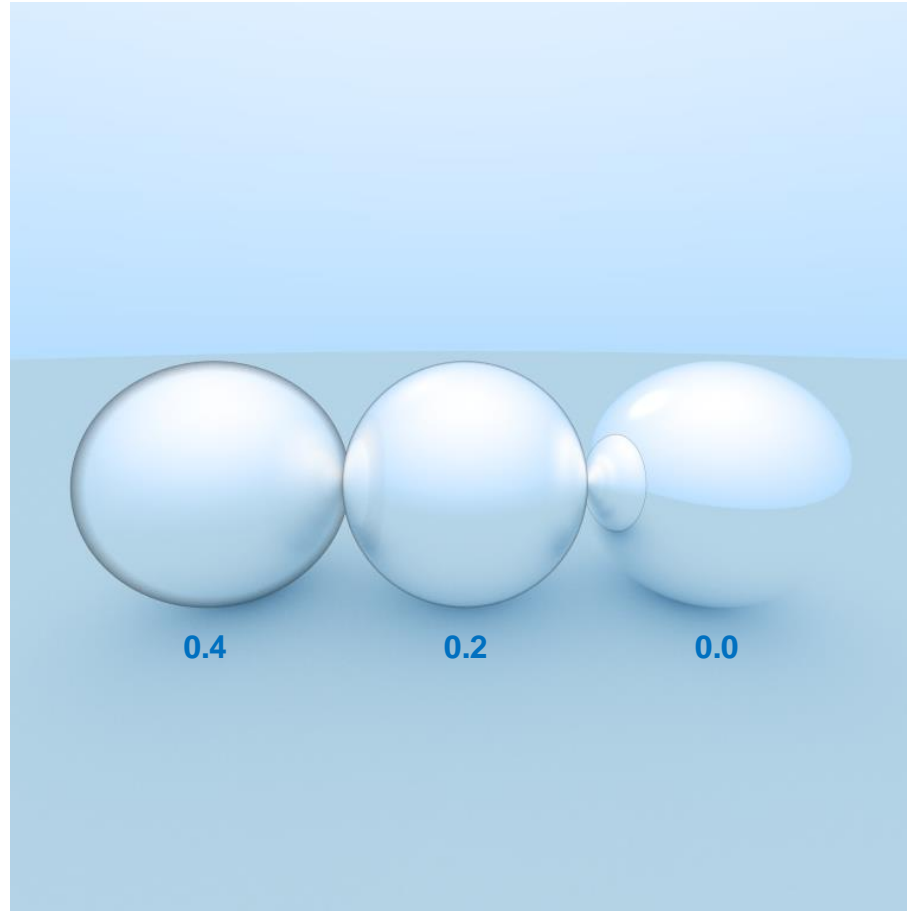
# Reflexão Fuzzy

Podemos colocar alguma aleatoriedade na reflexão, deslocando o destino do raio pelo deslocamento unitário radial.

```
reflect + fuzz * random_sphere;
```



# Reflexão Fuzzy - Resultados



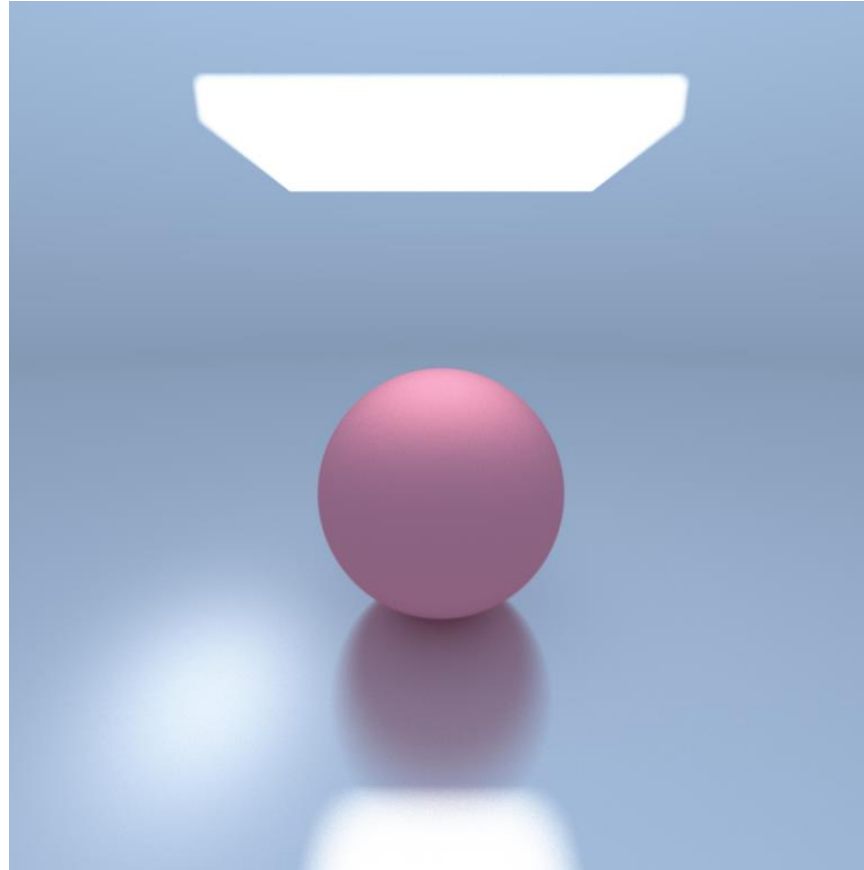
# Tomate

Como o tomate reflete a luz?



# Tomate

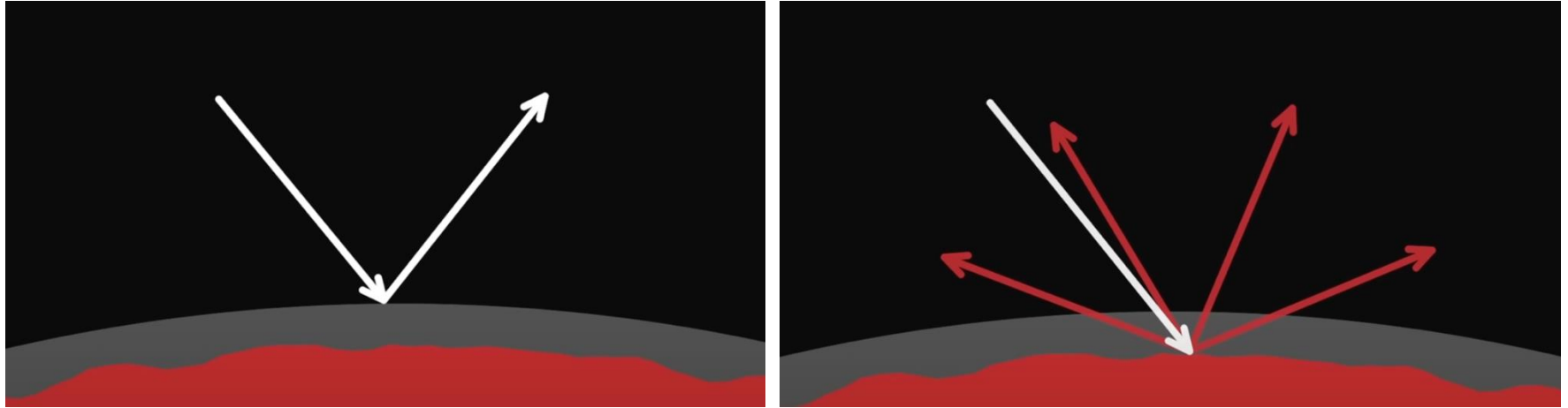
Smoothness mais baixo?





# Reflexão especular

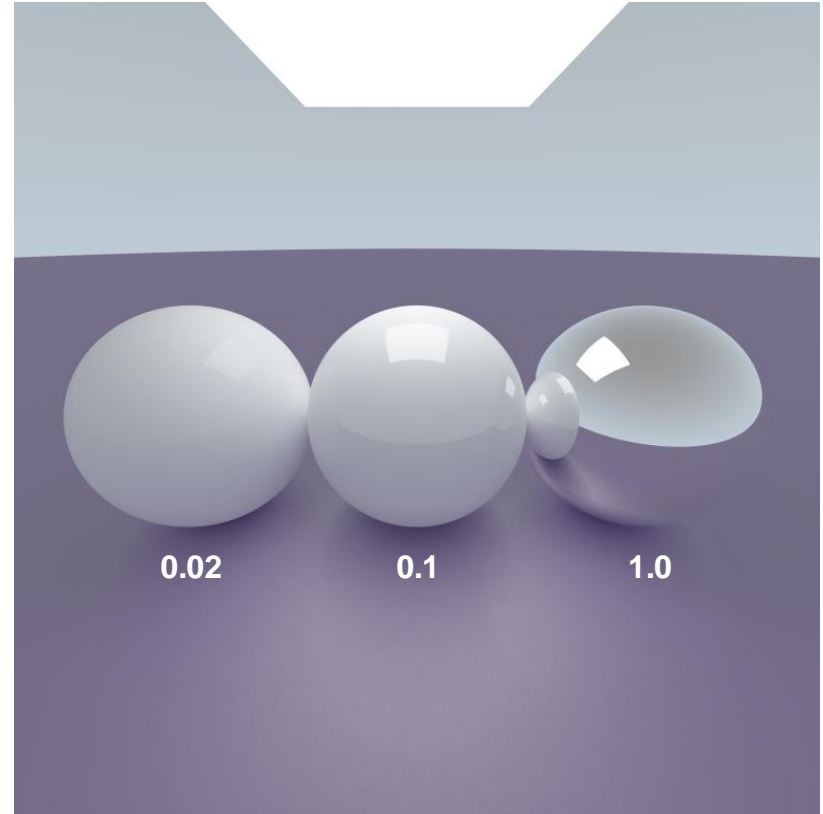
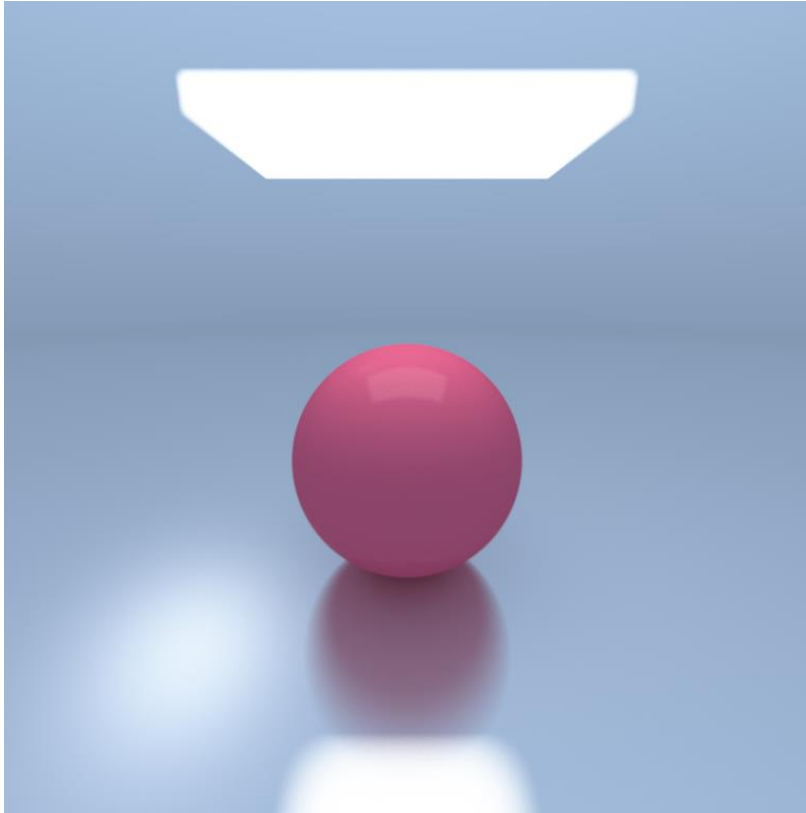
Alguns raios são refletidos e outros se espalham igual um material lambertiano



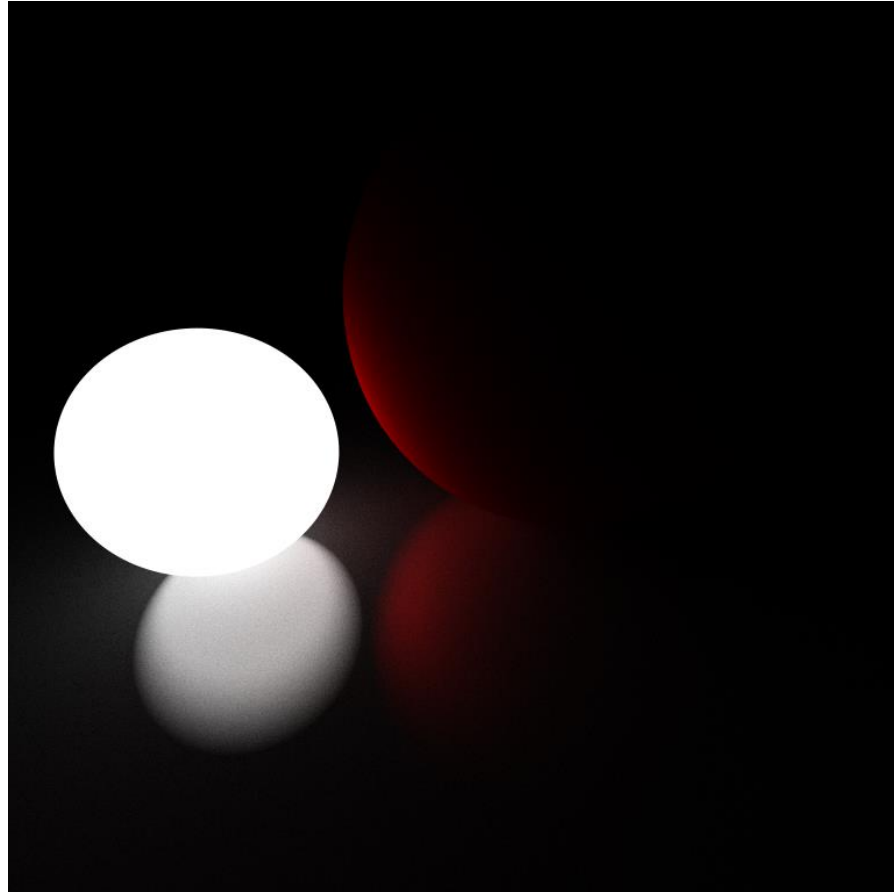
**Como fazer esse efeito?**

Specular probability > random()

# Reflexão especular - Resultados



# Materiais emissivos



**Cor multiplicada pela intensidade**

# Materiais Dielétricos

Materiais transparentes como água, vidro e diamantes são dielétricos.

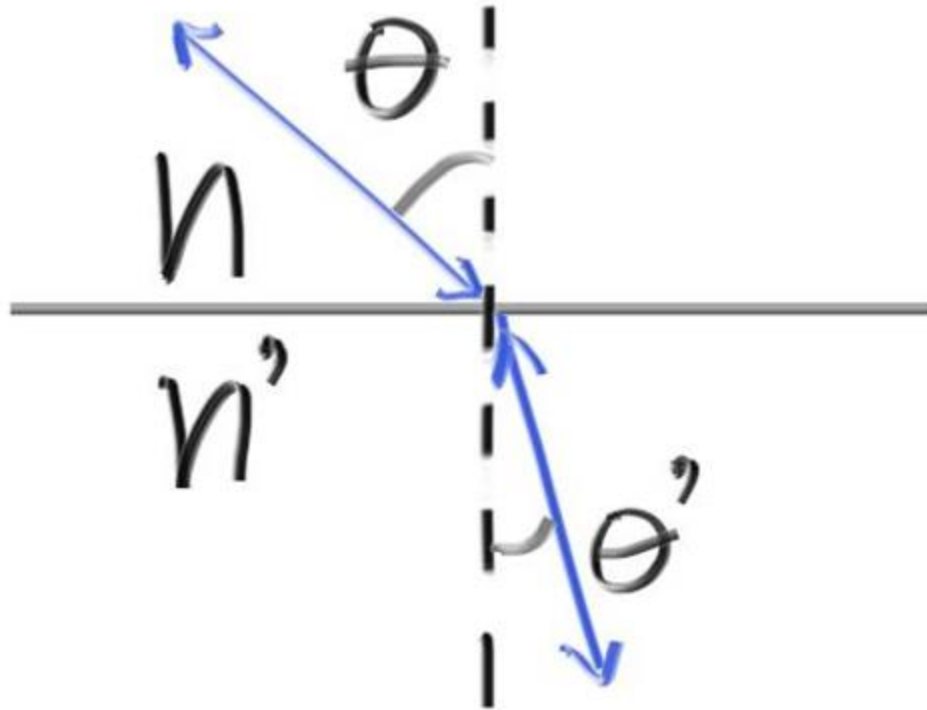
Quando um raio de luz os atinge, ele se divide em um raio refletido e um raio refratado (transmitido).

Vamos lidar com isso escolhendo aleatoriamente entre reflexão ou refração, e apenas gerando um raio de espalhamento por interação.

# Lei de Snell-Descartes

A refração é descrita por:

$$\eta \cdot \sin \theta = \eta' \cdot \sin \theta'$$



# Determinando o raio refratado

O raio de entrada ( $\mathbf{R}$ ) vai ser refratado em um raio ( $\mathbf{R}'$ ). Para efetuar os cálculos vamos dividir o raio refratado nos componentes perpendicular e paralelos a normal.

$$\mathbf{R}' = \mathbf{R}'_{\perp} + \mathbf{R}'_{\parallel}$$

A solução para o  $\mathbf{R}$  perpendicular e paralelo é:

$$\mathbf{R}'_{\perp} = \frac{\eta}{\eta'} (\mathbf{R} + \cos \theta \mathbf{n})$$

$$\mathbf{R}'_{\parallel} = -\sqrt{1 - |\mathbf{R}'_{\perp}|^2} \cdot \mathbf{n}$$



# Determinando o raio refratado

Para calcular o valor do cosseno de  $\theta$  podemos usar o produto escalar dos vetores. Ou seja:

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta$$

Restringindo  $\mathbf{a}$  e  $\mathbf{b}$  a vetores unitários:

$$\mathbf{a} \cdot \mathbf{b} = \cos \theta$$

Dessa forma a componente perpendicular do raio refratado fica:

$$\mathbf{R}'_{\perp} = \frac{\eta}{\eta'} (\mathbf{R} + (-\mathbf{R} \cdot \mathbf{n})\mathbf{n})$$

# Resultado das Esferas de Vidro



# Reflexão Total / Total Internal Reflection

Uma situação curiosa pode acontecer quando o raio de luz sai de um meio de maior índice de refração para um índice menor.

Lembrando que podemos calcular o ângulo do raio refratado:

$$\sin \theta' = \frac{\eta}{\eta'} \cdot \sin \theta$$

Suponha agora o raio indo do vidro ( $n=1.5$ ) para o ar ( $n = 1.0$ ):

$$\sin \theta' = \frac{1.5}{1.0} \cdot \sin \theta$$

Você vê algum problema?

# Reflexão Total Interna / Total Internal Reflection

O valor de seno  $\theta$  não pode ser maior que 1, e conforme a incidência o ângulo a fórmula pode ficar sem solução.

$$\frac{1.5}{1.0} \cdot \sin \theta > 1 \text{ ??????}$$

Ou seja, se a razão dos índices de refração vezes o seno  $\theta$  for maior que 1, o que acontece na verdade é uma reflexão.

# senos e cossenos

Precisamos achar a razão dos índices de refração vezes o seno  $\theta$

Para isso vamos fazer algumas contas. Os índices de refração são dados, e o cosseno  $\theta$  pode ser calculado pelo produto escalar.

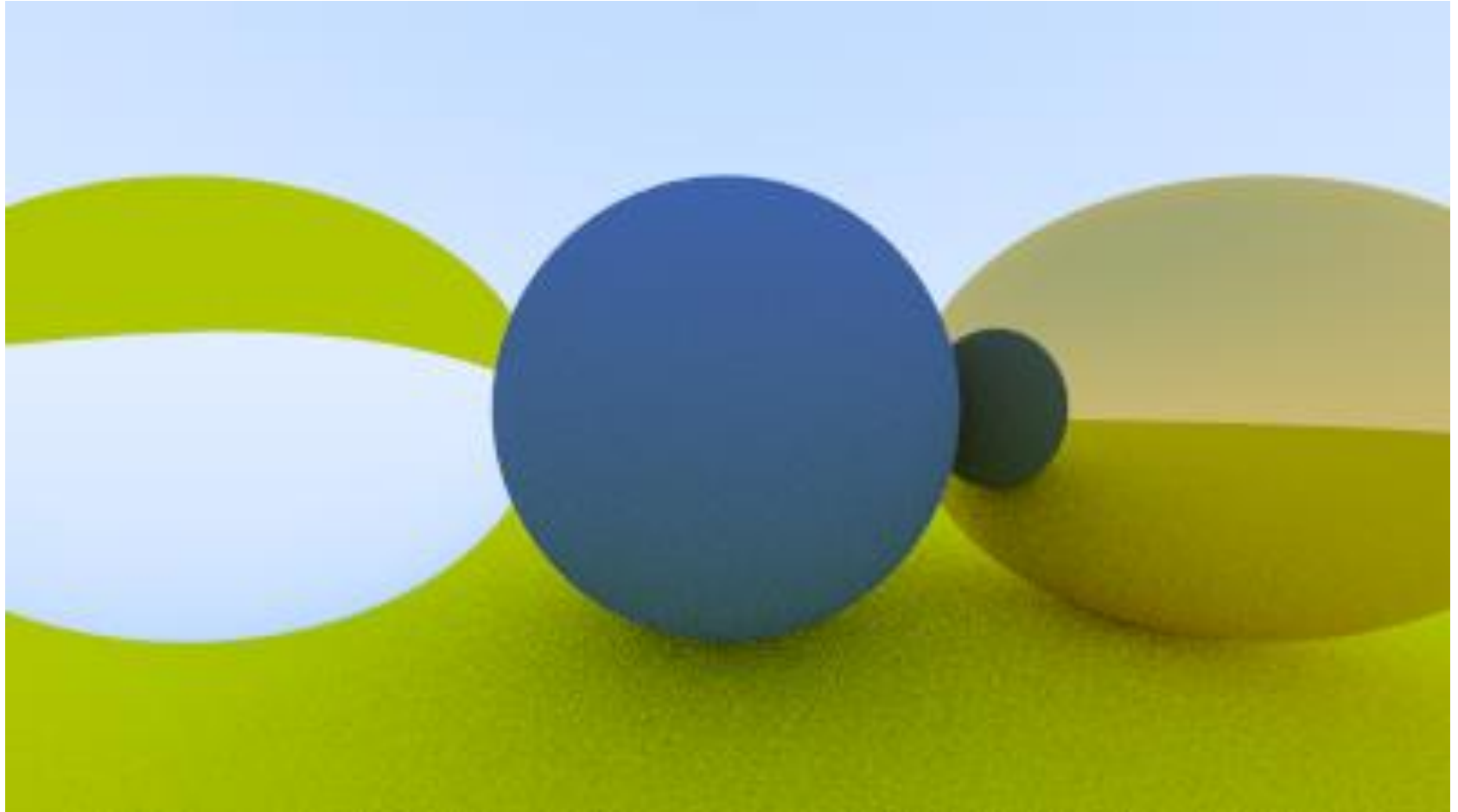
$$\sin^2 \theta + \cos^2 \theta = 1$$

$$\sin \theta = \sqrt{1 - \cos^2 \theta}$$

E o cosseno temos como:

$$\cos \theta = \mathbf{R} \cdot \mathbf{n}$$

# Resultado com novos materiais





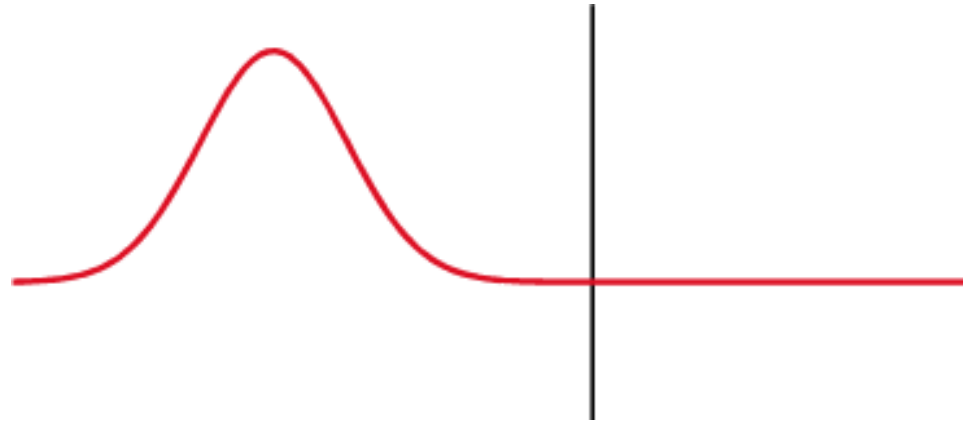
# Aproximação de Schlick

Permite calcular o coeficiente de reflexão:

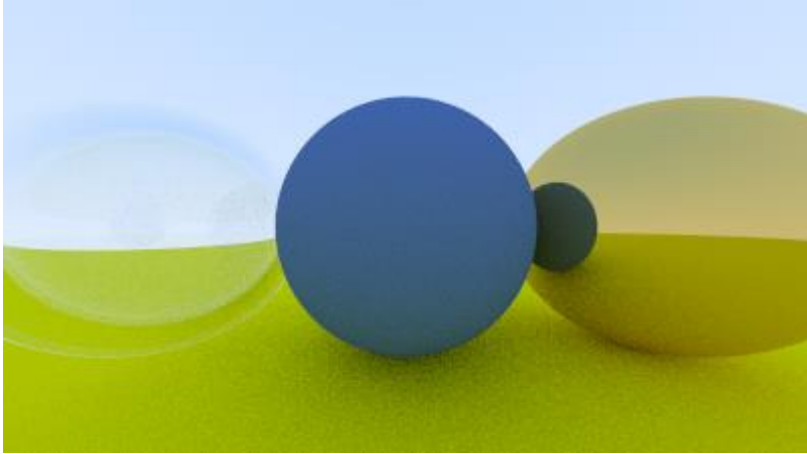
$$R(\theta) = R_0 + (1 - R_0)(1 - \cos \theta)^5$$

*where*

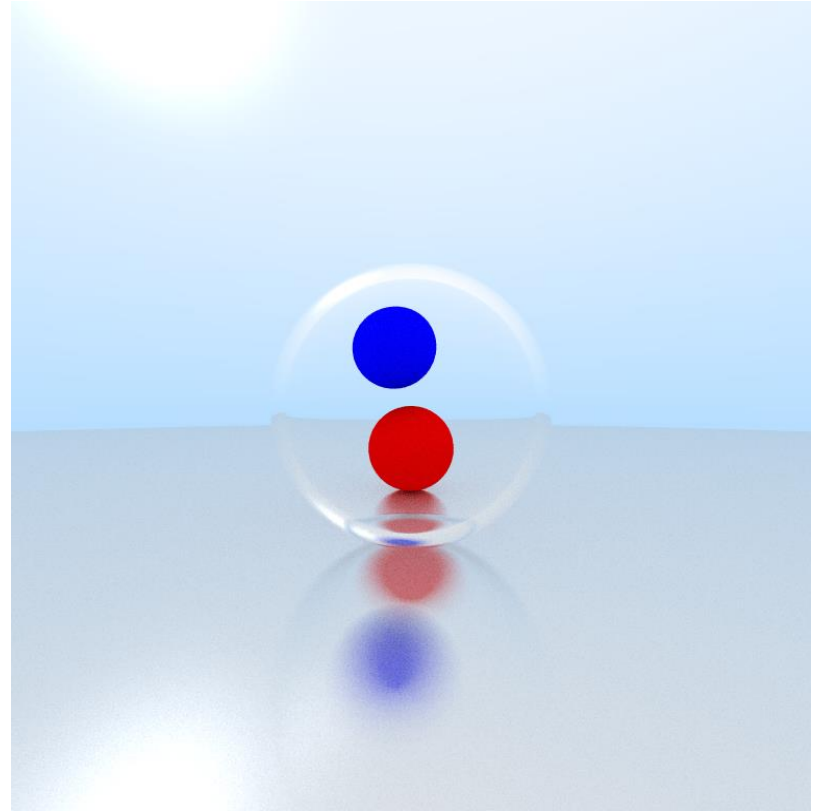
$$R_0 = \left( \frac{n_1 - n_2}{n_1 + n_2} \right)^2$$



# Resultado das reflexões atualizadas

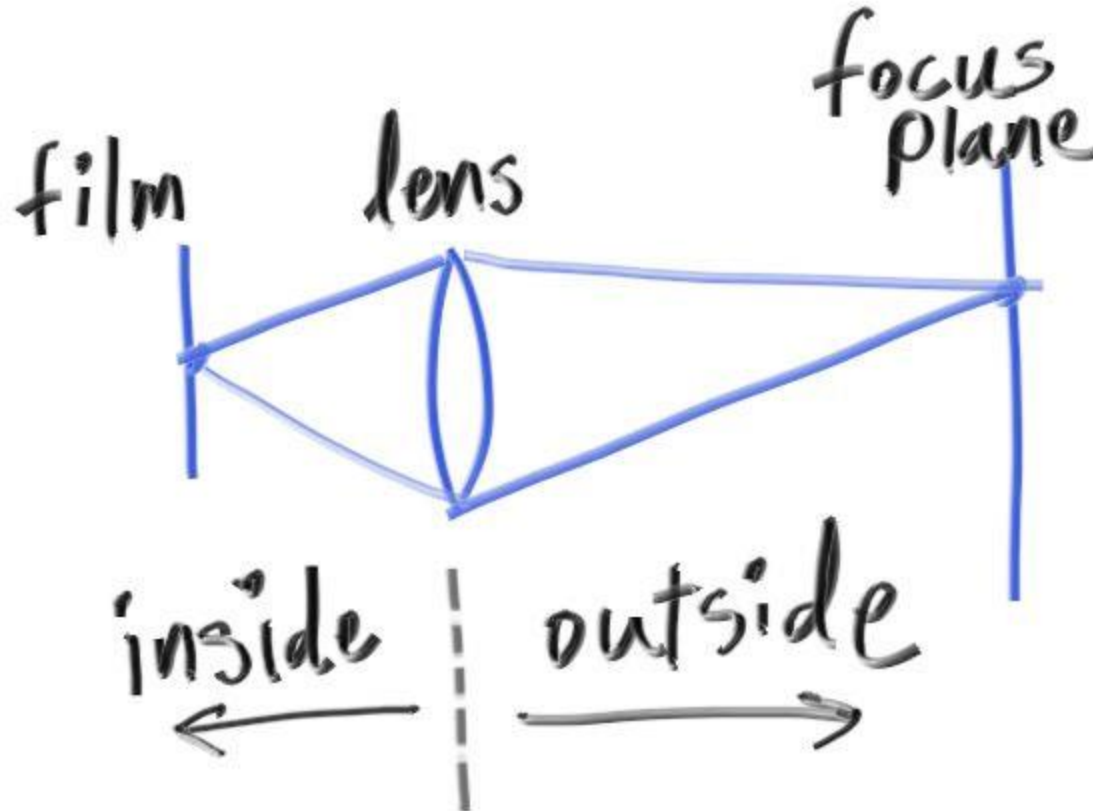


No Projeto:



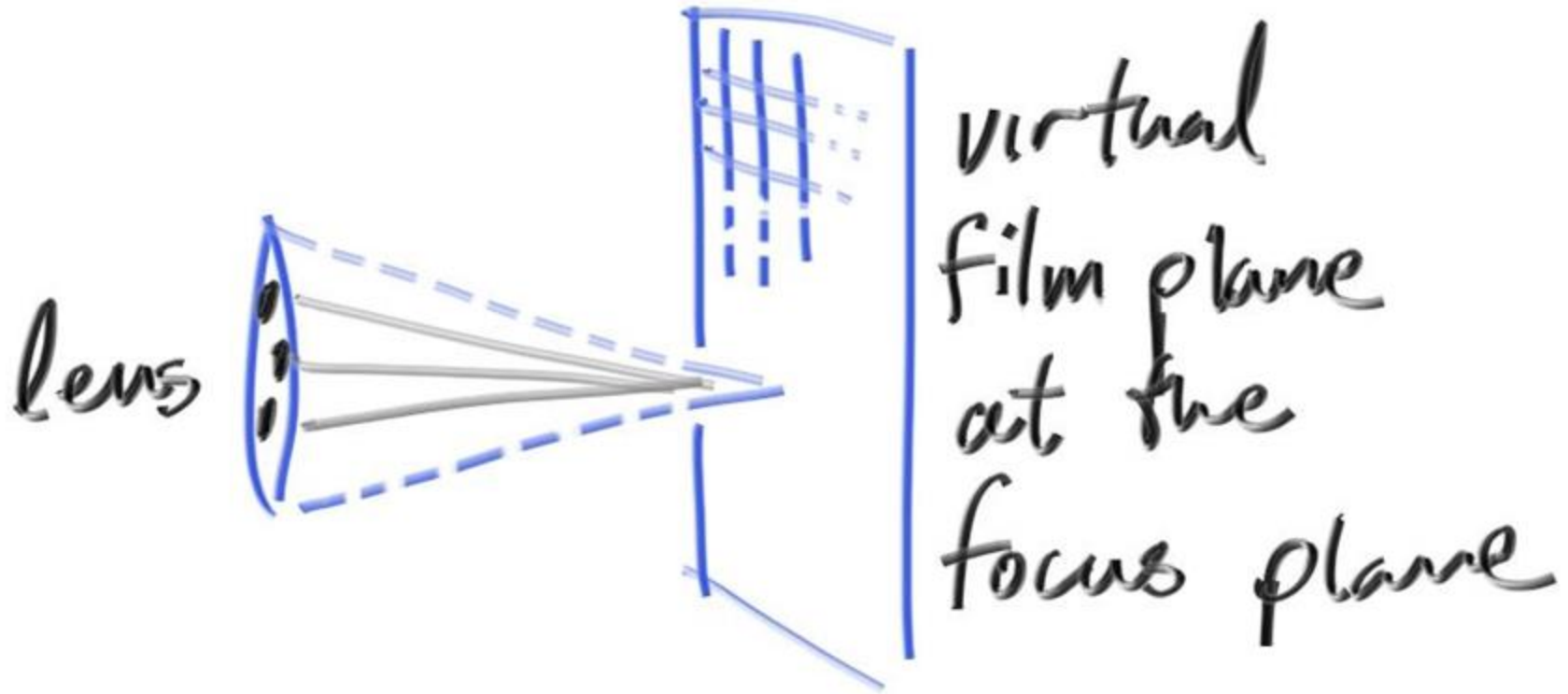
# Desfocando - Blur

Simula um efeito de profundidade de campo (depth of field)  
Podemos criar o efeito por um conjunto sensor, lente e abertura.



# Desfocando - Blur

O truque a ser usado é lançar os raios como se fosse da lente.



# Projeto Raytracer

▼ Sphere 2

▼ transform

|   |                                   |
|---|-----------------------------------|
| X | <input type="text" value="0"/>    |
| Y | <input type="text" value="-0,5"/> |
| Z | <input type="text" value="-2"/>   |
| W | <input type="text" value="0,5"/>  |

▼ color

|     |                                       |                                     |
|-----|---------------------------------------|-------------------------------------|
| Hex | <input type="color" value="#ff0000"/> | <input type="text" value="ff0000"/> |
|-----|---------------------------------------|-------------------------------------|

▼ material

|            |                                  |
|------------|----------------------------------|
| Smoothness | <input type="text" value="1"/>   |
| Absorption | <input type="text" value="0,2"/> |
| Specular   | <input type="text" value="0,8"/> |
| Light      | <input type="text" value="0"/>   |

## Smoothness:

- Se  $> 0$ , componente metallic
- Se  $< 0$ , o material é Dielétrico

## Absorption = Fuzz

# Computação Gráfica

Luciano Soares  
<lpsoares@insper.edu.br>

Fabio Orfali  
<fabioo1@insper.edu.br>

Gustavo Braga  
<gustavobb1@insper.edu.br>