

# Computação Gráfica para Jogos Eletrônicos

## Métodos de renderização e shaders

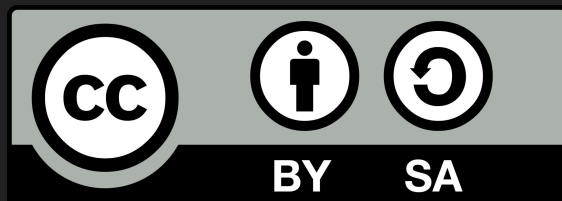
Slides por: Gustavo Ferreira Ceccon ([gustavo.ceccon@usp.br](mailto:gustavo.ceccon@usp.br))

Assistentes: Leonardo Tórtoro Pereira ([leonardop@usp.br](mailto:leonardop@usp.br)),

Gabriel Simmel ([gabriel.simmel.nascimento@usp.br](mailto:gabriel.simmel.nascimento@usp.br)) e Ítalo Tobler ([italo.tobler.silva@usp.br](mailto:italo.tobler.silva@usp.br))

Edição 2020: Renata Vinhaga ([renatavinhaga@usp.br](mailto:renatavinhaga@usp.br))





Este material é uma criação do  
Time de Ensino de Desenvolvimento de Jogos  
Eletrônicos (TEDJE)

Filiado ao grupo de cultura e extensão  
Fellowship of the Game (FoG), vinculado ao  
ICMC - USP

Este material possui licença CC By-SA. Mais informações em:  
<https://creativecommons.org/licenses/by-sa/4.0/legalcode>



# Objetivos

- Retomar e aprofundar o básico de GPU e do Pipeline de Renderização
- Unity templates : Built in, HDRP, URP
- Conceitos e algoritmos básicos por trás das principais técnicas utilizadas na área, além de exemplos de utilização
- Shader time
- Um pouco mais sobre modelos 3D



# Índice

1. Introdução
2. Shader
3. Iluminação
4. Animação + Blender



# 1. Introdução



# 1. Introdução

- Lembra das APIs?
  - ◆ OpenGL, Vulkan, DirectX
- GLSL vs. HLSL vs. CG
  - ◆ Linguagens de shader legíveis

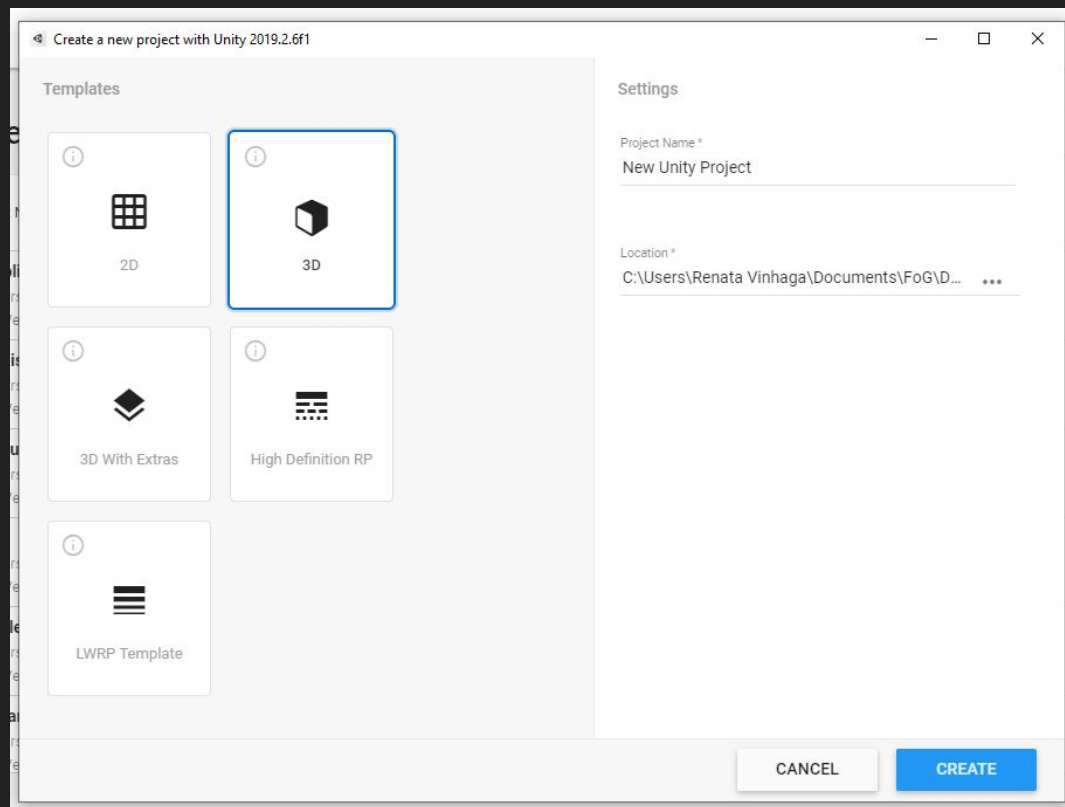


# 1. Introdução

- Quem quiser aprender mais:
- <https://learnopengl.com/> (curso de OpenGL)
- <https://www.youtube.com/watch?v=V5XFrIhLpGQ> (shader graph)
- <https://unity3d.com/pt/learn/tutorials/topics/graphics/gentle-introduction-shaders> (Unity shaders)
- <https://docs.unity3d.com/Manual/Shaders.html> (manual geral)
- <https://docs.unity3d.com/Manual/SL-Reference.html> (especificação CG)



# Templates





## Built in Renderer

- Rendering de propósito geral
- Tenta suprir todas as necessidades
- Contra: acaba não sendo útil para coisas específicas



# Scriptable Render Pipeline

- Unity recriou o sistema de Renderização
- Um engenheiro gráfico pode customizar a Renderização por Scripts em C#
- Com palavras mais simples: Agora a Unity tem mais capacidade gráfica
- 2 pipelines pré definidos para nós mortais

## HDRP e URP

- High Definition Pipeline Render
  - ◆ Foca em hardwares mais avançados como PCs, Xbox e Playstation
  - ◆ Para finalidades de alto nível gráfico
  - ◆ Cada material vai exigir inúmeros mapeamentos
  - ◆ Para projetos pequenos ou projetos em 2D é DESNECESSÁRIO

## HDRP e URP

- Universal Render Pipeline (ou LWRP)
  - ◆ Serve para todas as plataformas
  - ◆ Não quer dizer que é inferior!
  - ◆ Tem as coisas do HDRP mas foca em performance para todas as plataformas
  - ◆ Iluminação e sombras 2D (não tem no HDRP)
  - ◆ Se seu projeto é 2D essa é a escolha

# HDRP e URP

→ O que ambos tem e o Built-in não:

- ◆ VFX Graph

- ◆ Shader Graph

→ Principal diferença:

- ◆ Iluminação

Explicação do Brackeys: [https://www.youtube.com/watch?v=5MuA92xUJCA&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=5MuA92xUJCA&ab_channel=Brackeys)



# PBR

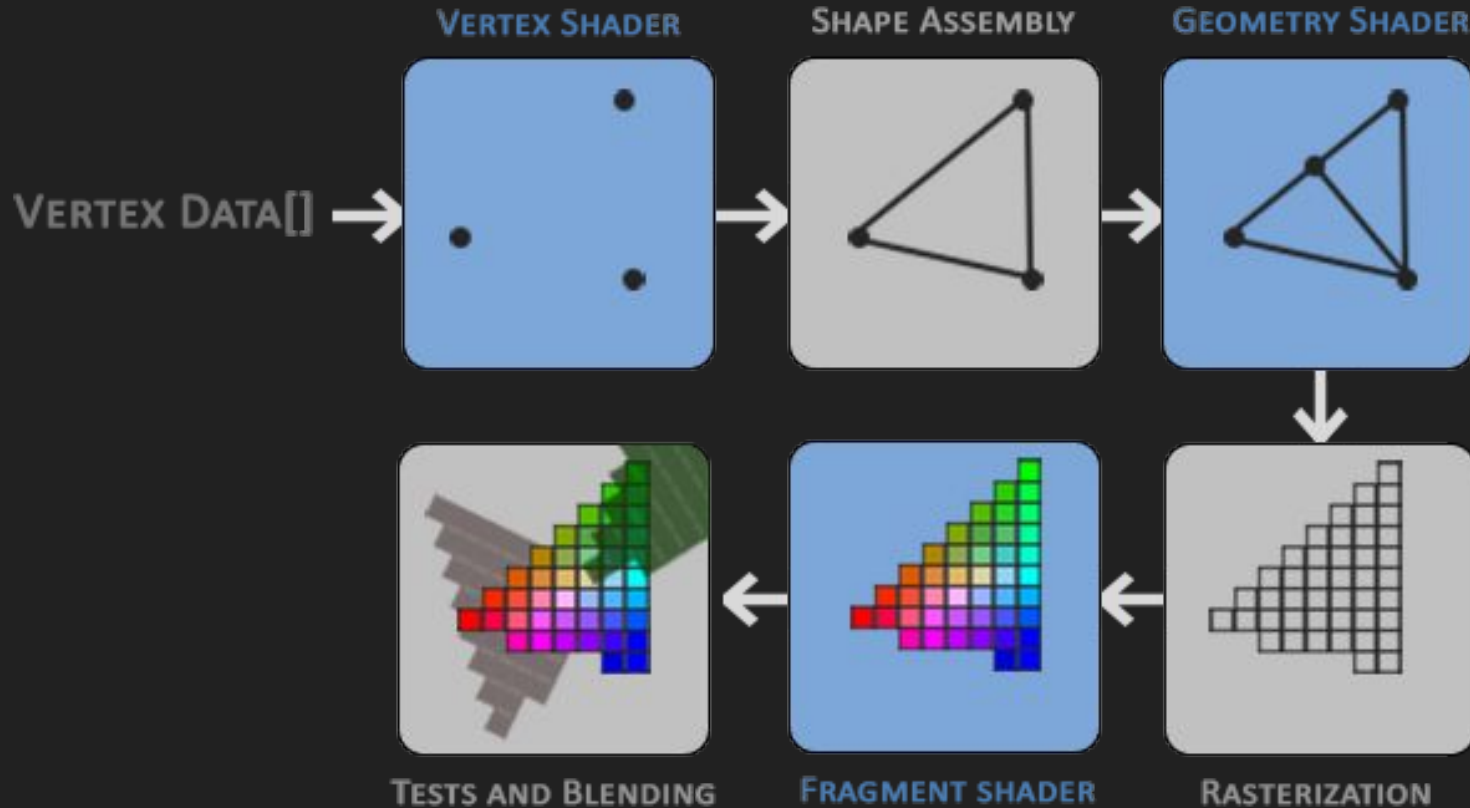
## Physically Based Rendering

- Usa o comportamento físico da luz para criar materiais com aspecto natural
- Modelo de iluminação que segue algumas das regras da física, como conservação de energia, Fresnel e oclusão
- Chamado Standard Shader na Unity, é o modelo de iluminação padrão
- Dois principais parâmetros: metallic e smoothness



# 3. Shader

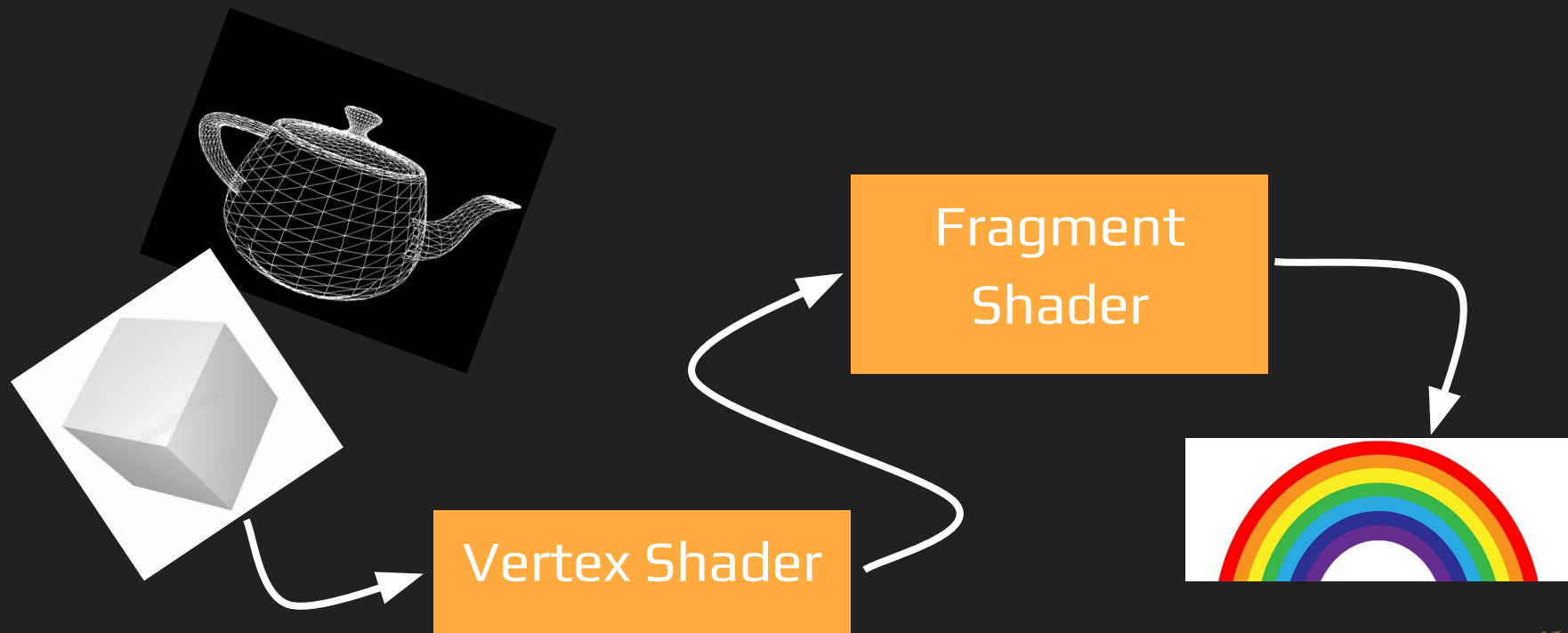
# Pipeline de renderização



<http://learnopengl.com/#!/Getting-started/Hello-Triangle>



# Vamos Simplificar!



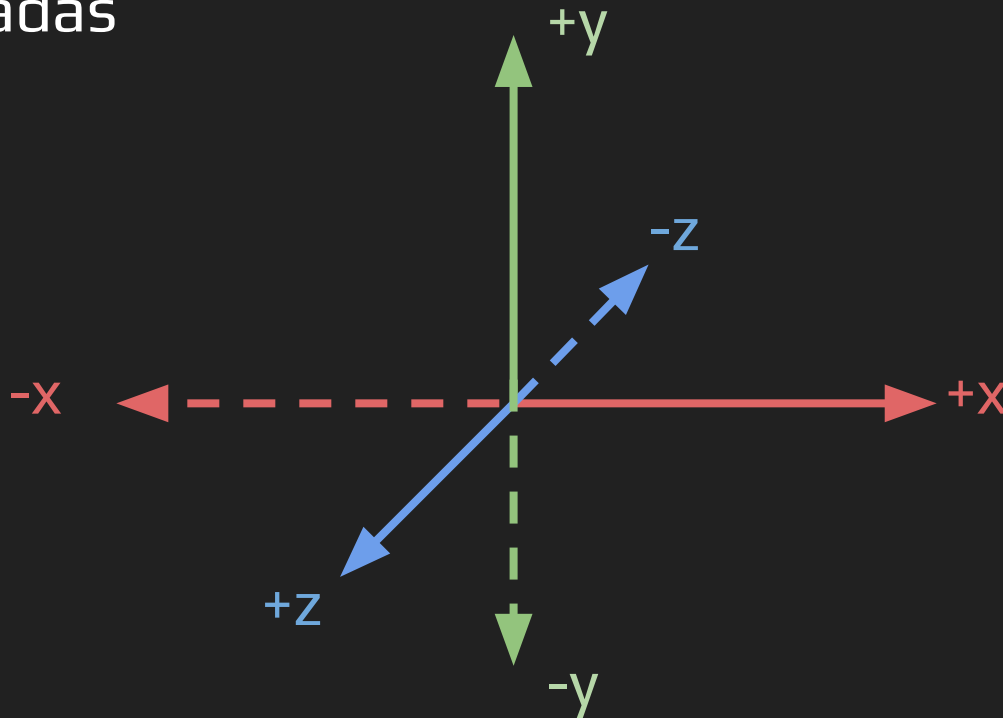
# 1. Relembrando

## → Shaders

- ◆ São programas que vão rodar na GPU
- ◆ Tudo paralelo, então não existem informações de vizinhos
- ◆ O shader vai rodar por elemento!
- ◆ A linguagem de shader é baseada em C !

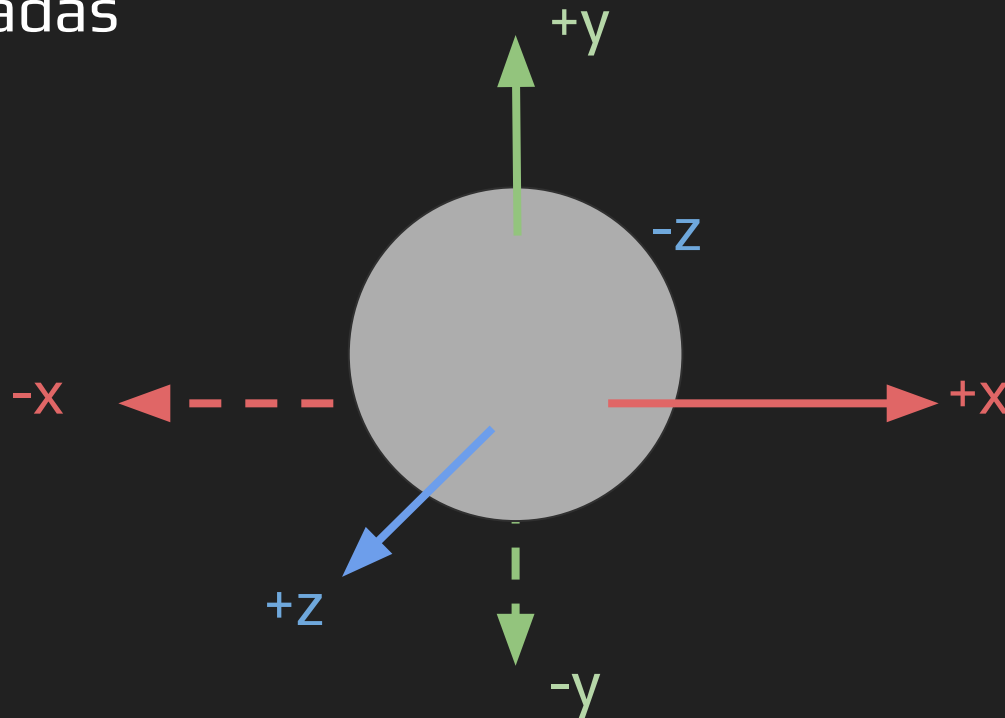
# 1. Relembrando

→ Coordenadas



# 1. Relembrando

→ Coordenadas

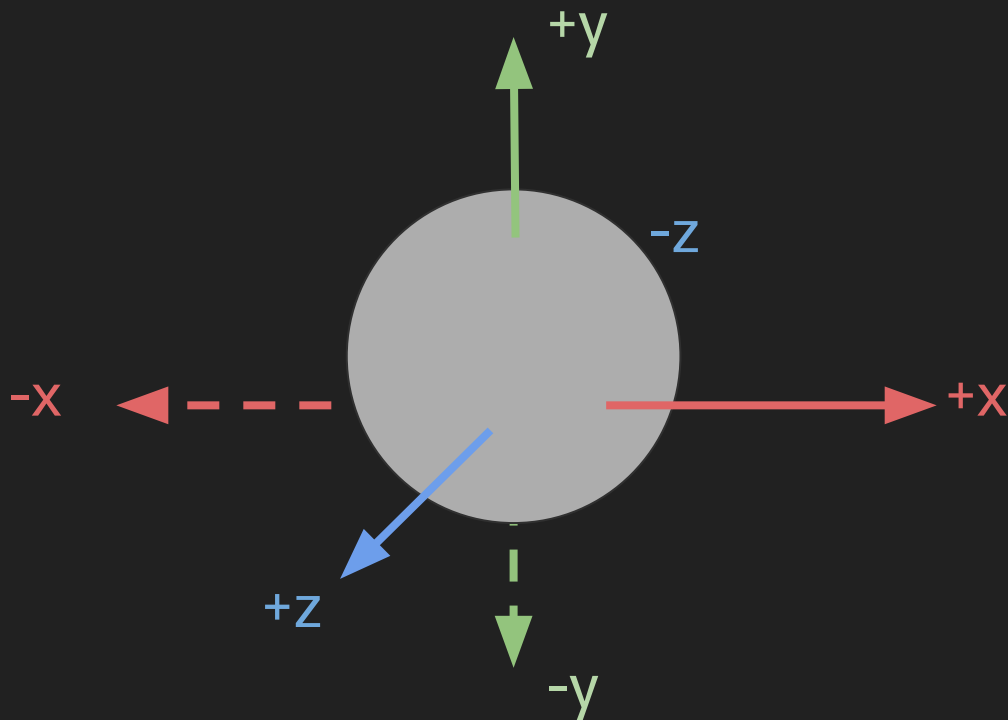


# 1. Relembrando

→ Cores

x y z w

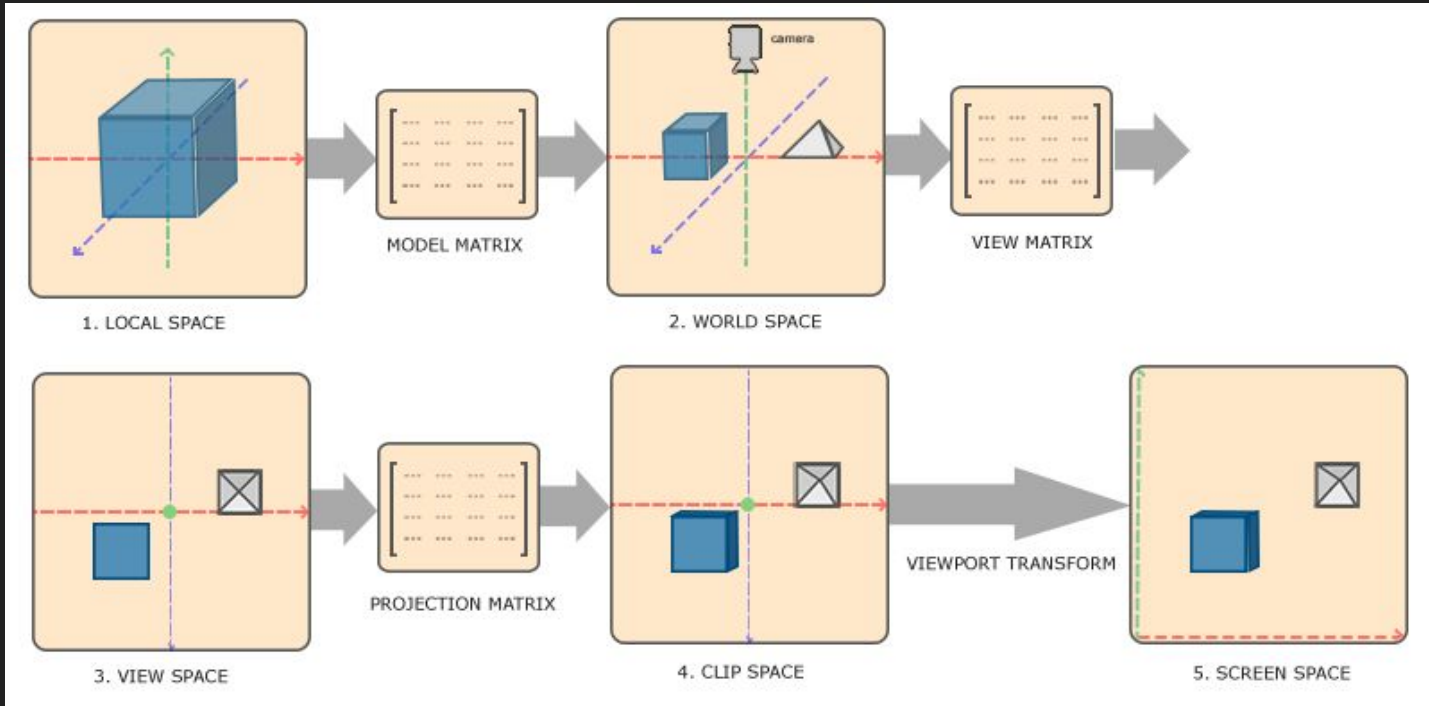
r g b a



# Shader time na Unity !



# 1.Transformações de Espaço



### 3. Renderização

- Vertex Shader
  - ◆ Transformações de mundo
  - ◆ Matriz MVP (Model View Projection)
  - ◆ Mapeamento de coordenadas
- Fragment Shader
  - ◆ Nível de pixel/fragmento
  - ◆ Definir cor de saída



### 3. Renderização

#### → Mapeamentos (Mapping)

- ◆ Texturas
- ◆ Bump
  - Displacement
  - Normal
  - Parallax
  - Height
- ◆ Cube
- ◆ Shadow

# Normal Map

## → Normal Map

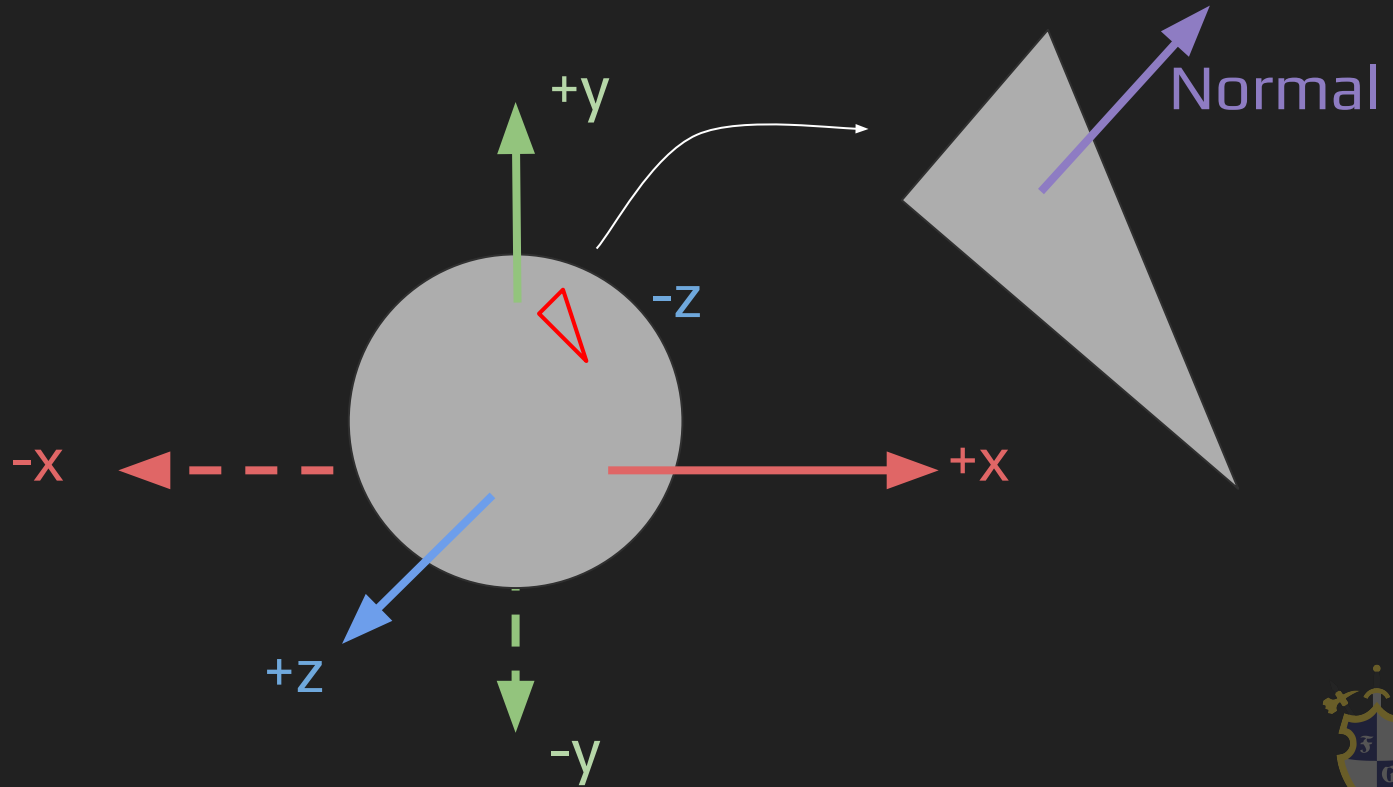
- ◆ Modifica a luz através da superfície da textura
- ◆ Baseia-se no vetor normal à superfície



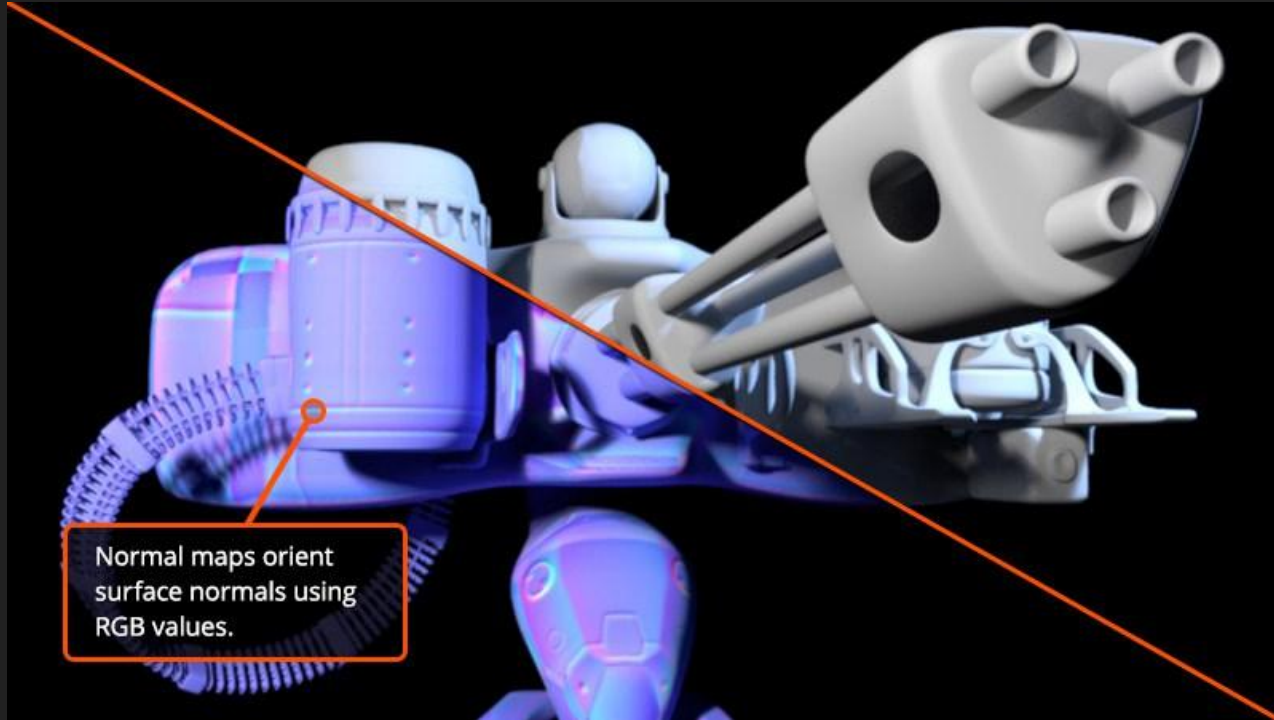
<https://learnopengl.com/Lighting/Basic-Lighting>

# Normal

→ Cores  
x y z w  
r g b a



# Normal Map



<http://blog.digitaltutors.com/bump-normal-and-displacement-maps/>

## 2. Iluminação

## 2. Modelos de Iluminação

### → Materiais

- ◆ Instâncias de um programa (um ou mais shaders)
- ◆ Descrevem como o objeto deve se comportar visualmente
- ◆ Podem conter informações como reflexão, transparência, quão metálico, quão liso, etc.

## 2. Modelos de Iluminação

- Modelos de iluminação são os modelos matemáticos e físicos de como os objetos interagem com a luz
  - ◆ Lambert Blinn-Phong
  - ◆ Physically Based Rendering (PBR)

## 2. Modelos de Iluminação

### → Difusa

- ◆ Cor que objeto recebe sob luz direta
- ◆ Mais forte na direção da luz e esmaece conforme o ângulo da superfície aumenta

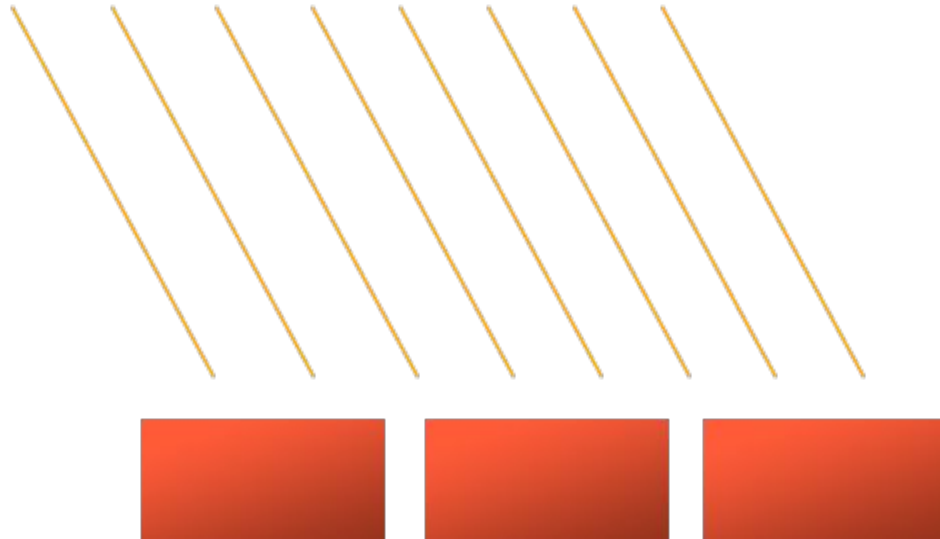
### → Especular

- ◆ Cor de destaque de um objeto.
- ◆ Aparece como reflexão da luz na superfície

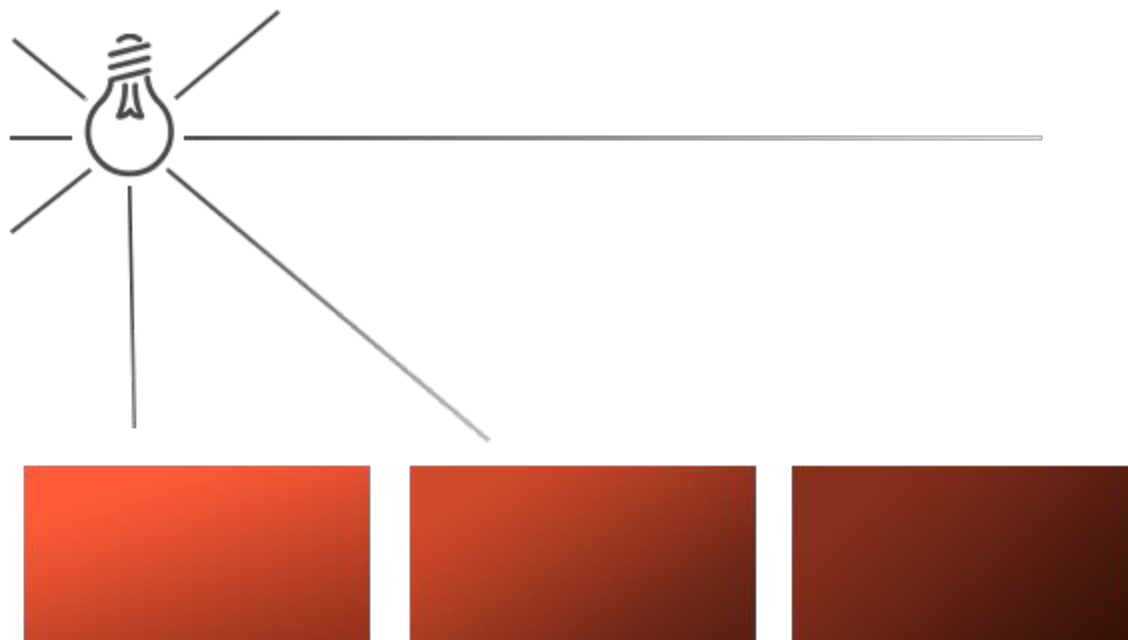
### → Exemplo



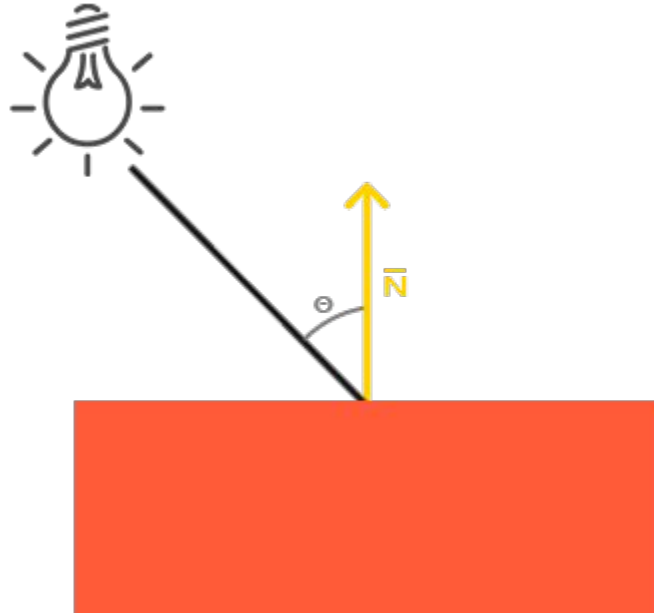
# Luz Direccional



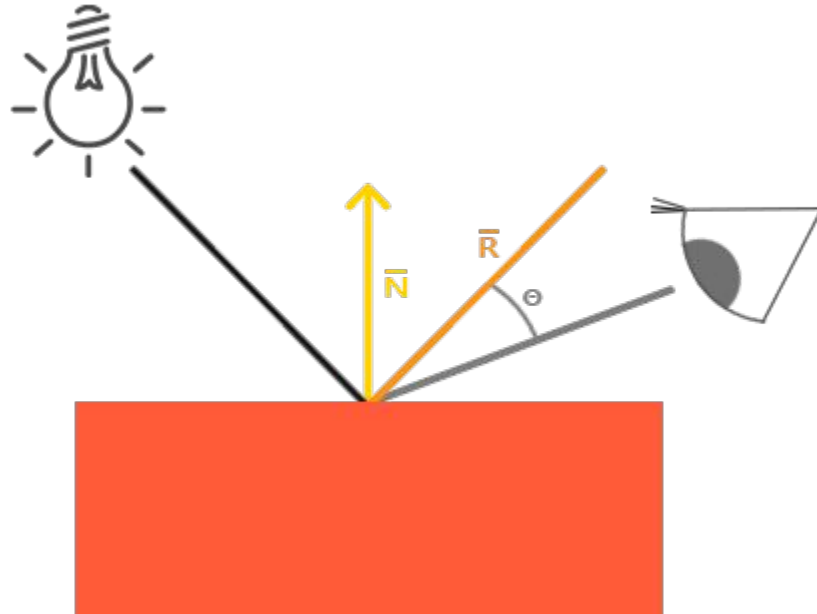
# Ponto de Luz



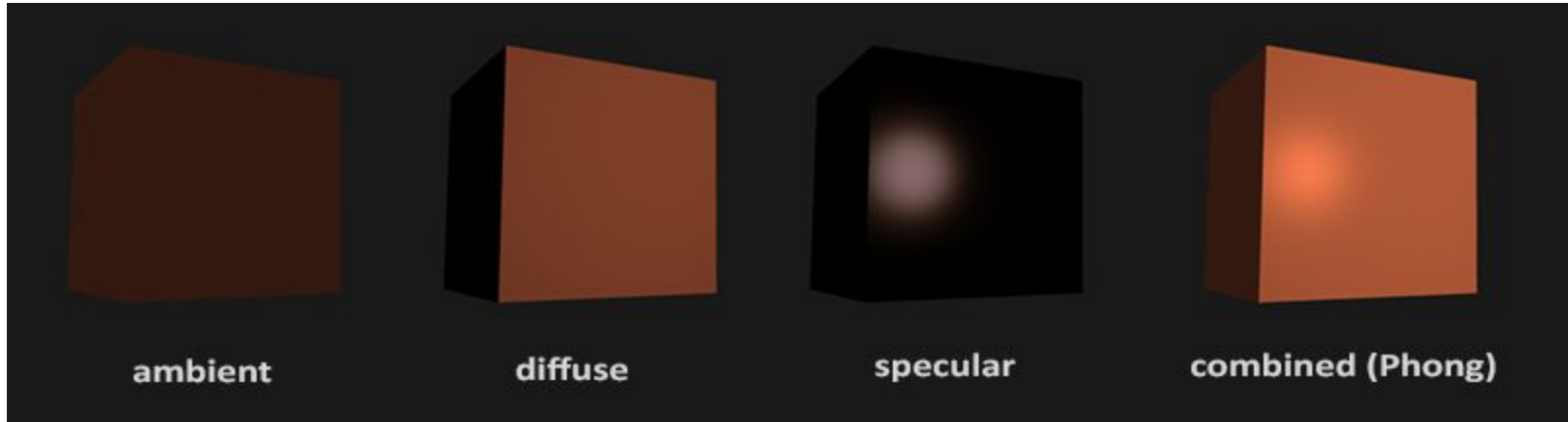
# Difusa



# Especular



# Modelos de iluminação



## 2. Modelos de Iluminação

- Physically Based Rendering (PBR)
  - ◆ Modelo de iluminação que segue algumas das regras da física, como conservação de energia, [Fresnel](#) e oclusão
  - ◆ Chamado Standard Shader na Unity, é o modelo de iluminação padrão
  - ◆ Dois principais parâmetros: metallic e smoothness

# Import e Export

# Animação + Mixamo





# Animação

→ Animação por bones



# Shader Graph

- Programação visual de shaders!
- Te poupa muito trabalho
- Menos controle mas pra um iniciante é show!
- Não tem no template built-in da Unity
- [https://www.youtube.com/watch?v=Ar9eIn4z6XE&app=desktop&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=Ar9eIn4z6XE&app=desktop&ab_channel=Brackeys)

# Dúvidas?

