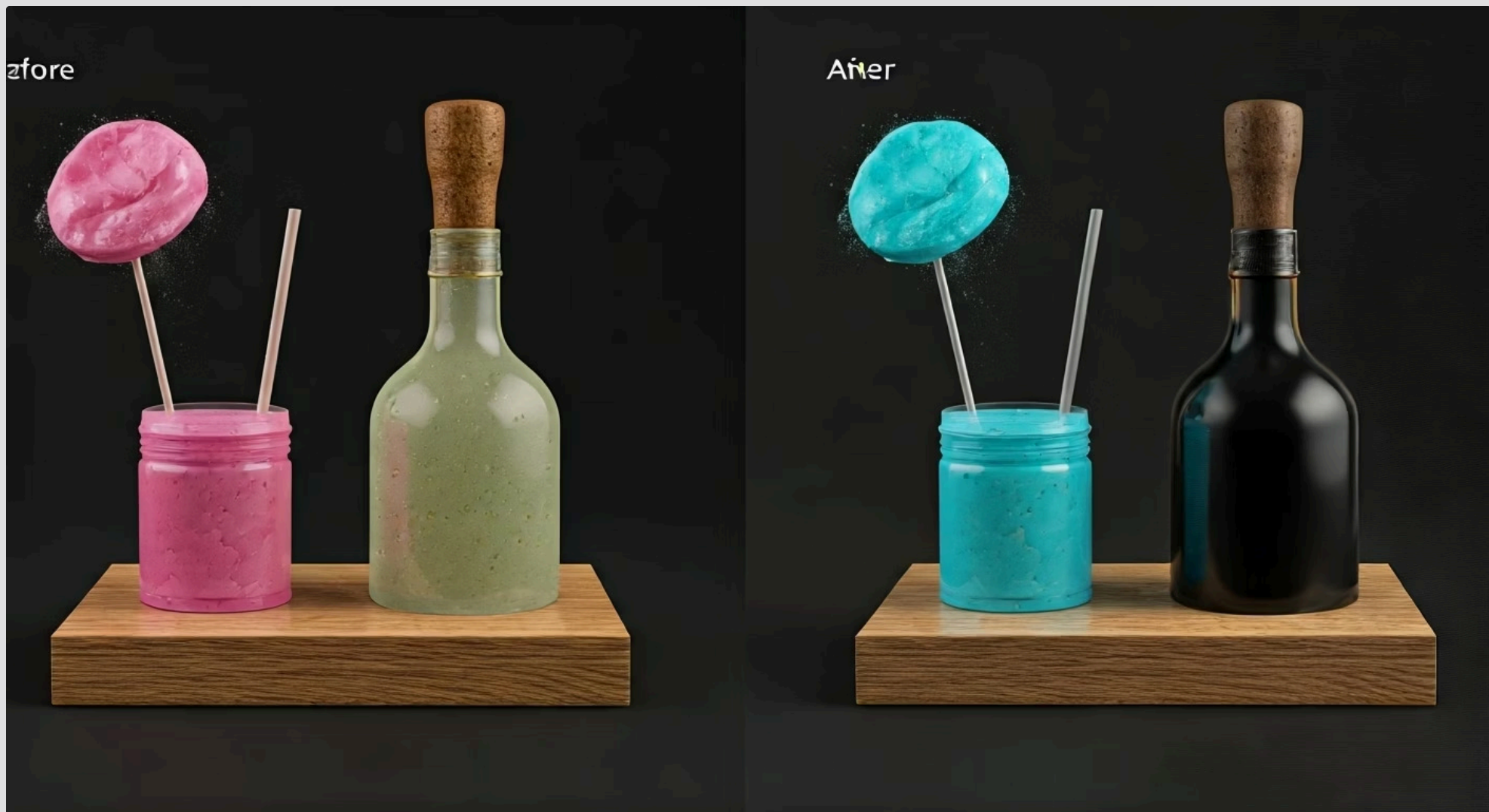


# Aula 8 – Criação e Aplicação de Materiais PBR



Imagine que você está construindo um mundo virtual. Você modelou personagens, cenários, objetos, mas algo parece "plástico", sem vida, como um brinquedo barato. A luz não interage de forma convincente, e a superfície dos objetos não reflete a realidade. Esse é um desafio comum para quem está começando no desenvolvimento de jogos 3D, e a solução para ele reside na arte e ciência dos materiais.

Nesta aula, vamos desvendar um dos pilares do realismo visual na indústria de jogos e animação: o Physically Based Rendering, ou PBR. Você aprenderá não apenas o que é o PBR, mas como ele se tornou o padrão ouro para criar superfícies que reagem à luz de forma crível, transformando modelos 3D em objetos tangíveis e imersivos. Ao final, você será capaz de compreender a função de cada mapa de textura, criar suas próprias texturas simples e aplicá-las em modelos 3D, preparando-os para brilhar em qualquer game engine moderna. Prepare-se para dar vida aos seus projetos e elevar o nível de realismo em suas criações.

# O Padrão da Indústria: Entendendo o PBR

No universo do desenvolvimento de jogos 3D, a busca pelo realismo é uma constante. Antigamente, os artistas passavam horas tentando "enganar" o olho humano, ajustando cores e reflexos de forma manual para simular como a luz interagiria com diferentes superfícies. Era um processo subjetivo, demorado e, muitas vezes, inconsistente, resultando em objetos que pareciam diferentes dependendo da iluminação da cena. Essa abordagem, embora funcional para a época, limitava a capacidade de criar mundos verdadeiramente imersivos.

Foi nesse cenário que o **Physically Based Rendering (PBR)** surgiu como uma revolução. O PBR não é apenas uma técnica; é uma filosofia de trabalho que se baseia em princípios físicos reais de como a luz interage com a matéria. Em vez de adivinhar, o PBR simula essa interação de forma matematicamente precisa, garantindo que um material se comporte de maneira consistente sob qualquer condição de iluminação.

A adoção do PBR democratizou o realismo. Com ele, artistas podem criar materiais que se comportam de forma previsível, economizando tempo e garantindo uma qualidade visual superior e uniforme em todo o projeto. Hoje, é o padrão inquestionável em game engines como Unity e Unreal Engine, e dominá-lo é essencial para qualquer desenvolvedor que busca criar experiências visuais de ponta.

## 📄 Por que PBR?

Isso significa que um pedaço de madeira parecerá madeira, e um metal parecerá metal, não importa se está sob a luz do sol, a luz de uma lâmpada ou na penumbra de uma caverna.



# A Sinfonia dos Mapas de Textura: Criando Realismo



## Múltiplas Camadas

Cada mapa é uma camada de informação específica



## Controle Granular

Defina cada aspecto visual do material



## Interação com Luz

Simulação física precisa da luz

Se o PBR é a orquestra que rege o realismo, os mapas de textura são os músicos, cada um com seu instrumento e partitura, trabalhando em perfeita harmonia para criar uma melodia visual complexa. Não se trata apenas de uma única imagem colorida, mas de uma coleção de imagens em tons de cinza ou coloridas, cada uma com uma função específica, que juntas definem as propriedades de uma superfície. É a combinação inteligente desses mapas que permite que um objeto 3D pareça áspero, liso, brilhante, opaco, enferrujado ou polido.

**Pense nos mapas de textura como diferentes camadas de informação que você "pinta" sobre seu modelo 3D.** Uma camada define a cor base, outra define o quão áspera a superfície é, outra indica se é um metal ou não, e assim por diante.

Quando a game engine processa essas informações, ela as interpreta de acordo com os princípios do PBR, resultando em uma interação de luz que é incrivelmente próxima da realidade. Sem essa colaboração, o material seria apenas uma cor plana e sem vida, incapaz de transmitir a riqueza de detalhes que vemos no mundo real.

A beleza dessa abordagem é que ela permite um controle granular sobre cada aspecto visual do material. Você pode ter um metal que é brilhante em uma área e enferrujado em outra, ou uma madeira que é lisa onde foi polida e áspera onde está desgastada. Essa flexibilidade é o que torna o PBR tão poderoso e a razão pela qual ele se tornou a espinha dorsal da criação de materiais na indústria de jogos.

# Desvendando os Mapas Essenciais do PBR

Para entender como essa sinfonia funciona, precisamos conhecer os principais instrumentos. Cada mapa de textura desempenha um papel crucial na definição do material, e a ausência ou o uso incorreto de um deles pode comprometer todo o realismo. Vamos explorar os mais importantes, começando pelos que definem a cor e a geometria aparente.

## Albedo (Base Color) e Normal Map: A Cor e o Detalhe Falso

### Albedo (Base Color)

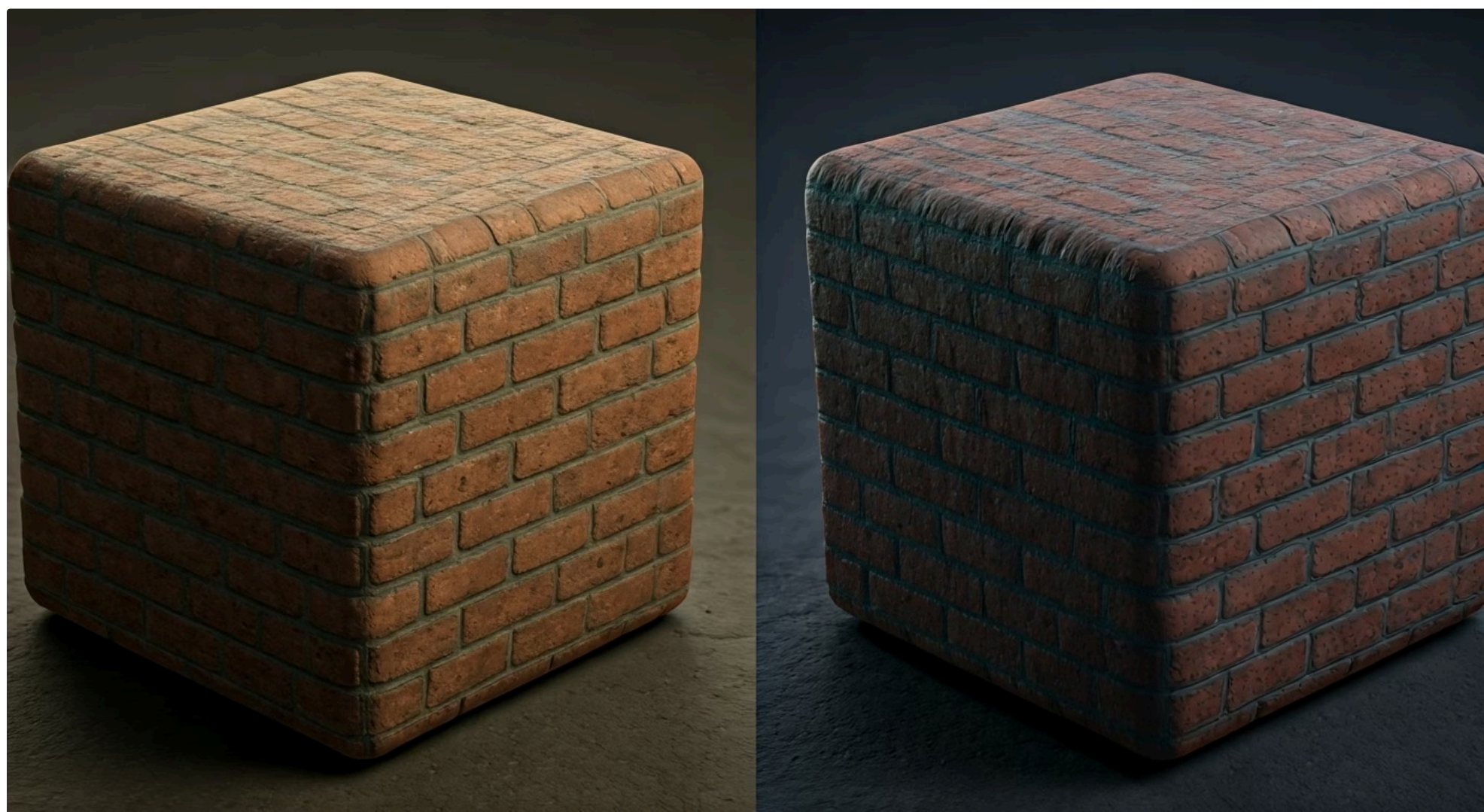
O mapa de **Albedo**, também conhecido como **Base Color**, é o ponto de partida. Ele define a cor intrínseca do material, ou seja, a cor que vemos quando a superfície é iluminada de forma difusa, sem reflexos ou sombras. Imagine-o como a tinta que você usaria para pintar um objeto.

- ❏ **Importante:** Este mapa não deve conter informações de iluminação ou sombra pré-assadas, pois o PBR se encarrega de calcular isso em tempo real. Ele deve ser o mais "plano" possível em termos de luz.

### Normal Map

Já o **Normal Map** é um truque de magia. Em vez de adicionar geometria real ao seu modelo (o que aumentaria drasticamente o número de polígonos e o custo de processamento), ele simula detalhes de superfície, como arranhões, relevos e texturas, usando informações de direção da superfície.

Pense nele como um mapa que "engana" a luz, fazendo com que ela reaja como se houvesse pequenos altos e baixos, mesmo que a superfície seja perfeitamente plana. Um normal map é geralmente azulado e roxo, e cada pixel indica a direção que a superfície "aparenta" ter naquele ponto.



# Mais Profundidade: Roughness e Metallic Maps

Continuando nossa jornada pelos mapas de textura, chegamos a dois que são fundamentais para definir como a luz se comporta na superfície: o Roughness e o Metallic. Eles são a chave para diferenciar um metal polido de uma madeira áspera, ou um plástico brilhante de um tecido fosco.



## Roughness Map

### O Quão Áspera é a Superfície?

O mapa de **Roughness** (Rugosidade) é um mapa em tons de cinza que controla a dispersão da luz na superfície. Em termos simples, ele define o quão "áspera" ou "lisa" microscopicamente uma superfície é.

- **Valor baixo (preto/escuro):** Superfície lisa e polida, reflexos nítidos como um espelho
- **Valor alto (branco/claro):** Superfície áspera, reflexos difusos e aspecto fosco

Imagine uma pista de boliche (lisa, reflexiva) versus uma estrada de terra (áspera, fosca). O mapa de Roughness permite que você pinte essas variações diretamente no seu modelo, fazendo com que a luz se comporte de maneira diferente em cada parte do objeto.



## Metallic Map

### É Metal ou Não É?

O mapa de **Metallic** (Metalicidade) é outro mapa em tons de cinza, mas com uma função binária: ele informa à game engine se uma área do material é metálica (branco) ou dielétrica (não metálica, preto).

- **Materiais metálicos:** Absorvem luz difusa e refletem luz especular com a cor do próprio metal
- **Materiais dielétricos:** Refletem luz especular sem coloração e absorvem ou dispersam luz difusa

Este mapa é crucial para o sistema PBR, pois ele dita qual modelo de reflexão física deve ser aplicado. Um valor intermediário de cinza geralmente não é recomendado, pois na natureza, um material é ou não é metal. No entanto, pode ser usado para simular superfícies com corrosão ou oxidação.

# Mapas Auxiliares: Detalhes que Fazem a Diferença

Além dos mapas essenciais que já vimos, existem outros que, embora não sejam estritamente obrigatórios para todos os materiais, adicionam camadas extras de realismo e complexidade visual. Eles são como os temperos finos que elevam o sabor de um prato, transformando um material bom em um material excepcional.



## Ambient Occlusion (AO) Sombras Sutis e Profundidade

O mapa de **Ambient Occlusion (AO)** é um mapa em tons de cinza que simula a oclusão da luz ambiente em áreas onde a luz tem dificuldade de alcançar, como fendas, cantos e reentrâncias. Ele não é uma sombra dinâmica, mas sim uma sombra pré-calculada que adiciona profundidade e realismo ao material.

Pense em como a sujeira se acumula nos cantos de um cômodo ou como a luz é mais fraca nas dobras de um tecido. O AO simula esse efeito, escurecendo sutilmente essas áreas e dando a impressão de que o objeto tem mais volume e está mais integrado ao ambiente.



## Height/Displacement Map Geometria Realista (e Pesada)

O mapa de **Height** (Altura) ou **Displacement** (Deslocamento) é um mapa em tons de cinza que, ao contrário do Normal Map, realmente modifica a geometria do modelo. Ele usa os valores de cinza para "empurrar" ou "puxar" os vértices da malha, criando relevos e depressões reais na superfície.

Embora ofereça o maior nível de realismo em termos de detalhes de superfície, o uso de Height/Displacement Maps é computacionalmente muito mais caro do que o Normal Map, pois exige um número significativamente maior de polígonos. Por isso, é geralmente reservado para detalhes muito próximos à câmera ou para superfícies que exigem uma silhueta complexa.

# Criando Texturas Simples: Seus Primeiros Passos no GIMP/Photoshop

Agora que entendemos a teoria por trás dos mapas de textura, é hora de colocar a mão na massa. Você não precisa ser um artista digital experiente para começar a criar suas próprias texturas PBR. Com editores de imagem como GIMP (gratuito e de código aberto) ou Photoshop (padrão da indústria), você pode criar texturas simples que já farão uma grande diferença em seus projetos. O segredo está em entender o propósito de cada mapa e como as cores e tons de cinza se traduzem em propriedades físicas.

## Exemplo Prático: Parede de Metal Enferrujado

Vamos imaginar que queremos criar uma textura PBR para uma parede de metal enferrujado.

01

### Albedo (Base Color)

Comece com uma cor base para o metal. Adicione variações de cor para simular a ferrugem, usando tons de laranja, marrom e vermelho. Lembre-se de que este mapa deve ser o mais "plano" possível em termos de luz.

03

### Metallic Map

As áreas enferrujadas não são metálicas, então devem ser pretas. As áreas de metal exposto devem ser brancas. Você pode usar uma máscara para pintar essas áreas.

02

### Roughness Map

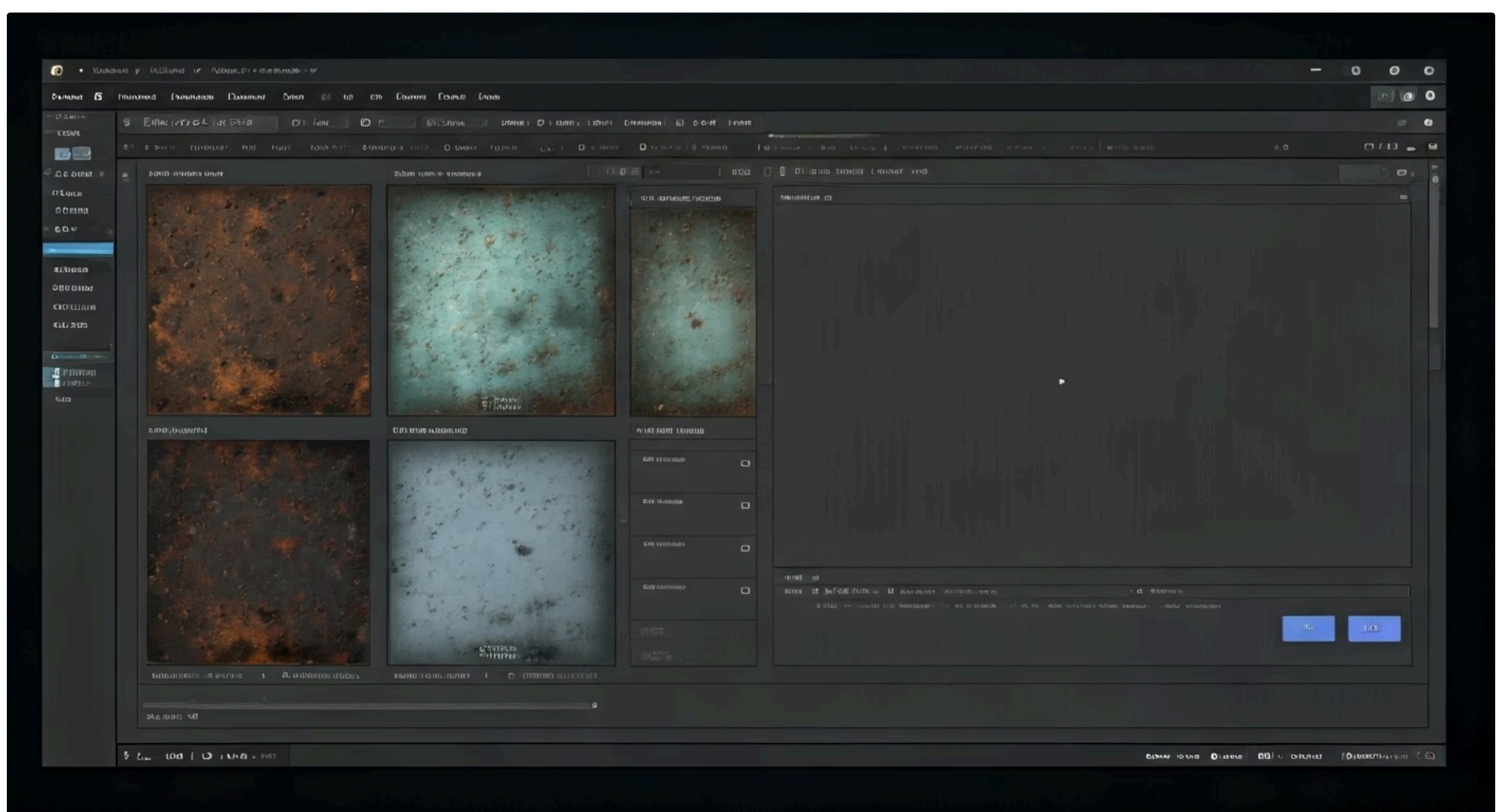
Para a ferrugem, a superfície será áspera, então as áreas enferrujadas devem ser mais claras (tons de cinza médio a branco). O metal exposto, se for mais liso, terá tons mais escuros (cinza escuro a preto). Use pincéis com texturas para criar variações orgânicas.

04

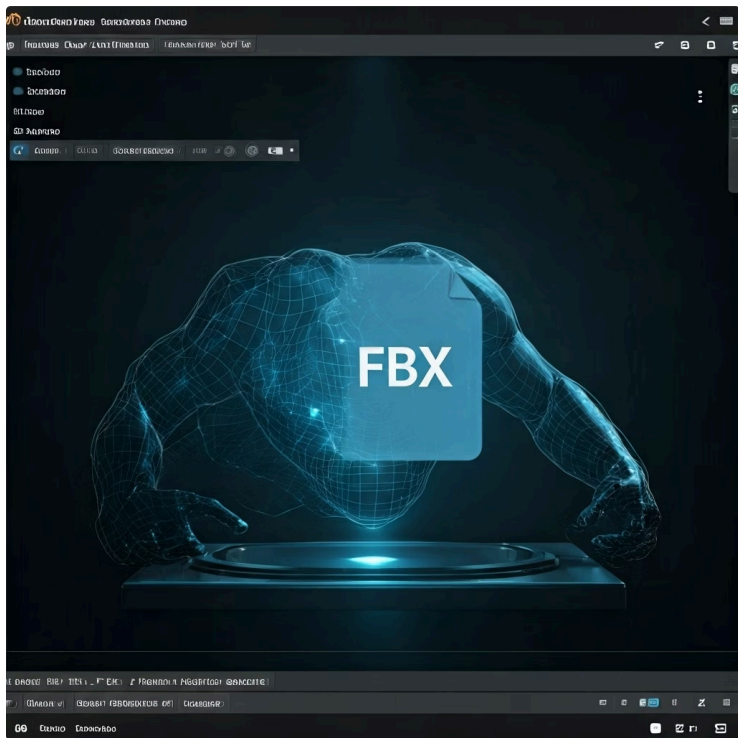
### Normal Map

Para adicionar detalhes de relevo, como a textura da ferrugem ou pequenos amassados no metal, você pode usar filtros de "Normal Map" disponíveis no GIMP ou Photoshop, ou até mesmo gerar um a partir do seu mapa de Height (se você criar um). Para isso, você pode pintar áreas de alto contraste no seu mapa de Height e depois convertê-lo.

**A chave é pensar em cada mapa como uma camada de informação separada.** Não tente colocar tudo no Albedo. A prática leva à perfeição, então experimente com diferentes materiais e observe como a luz interage com eles no mundo real para guiar suas criações.



# Exportando do Blender: Preparando Seu Modelo para a Game Engine



Depois de modelar seu objeto 3D no Blender e criar suas texturas PBR, o próximo passo crucial é exportar tudo de forma que a game engine possa entender e utilizar. Este processo, embora pareça técnico, é bastante padronizado e essencial para garantir que seu trabalho seja transferido corretamente.

O formato de arquivo mais comum e recomendado para exportar modelos 3D com suas informações de malha, UVs e até mesmo animações é o **FBX (.fbx)**. Ele é amplamente suportado por praticamente todas as game engines e softwares 3D. Outra opção é o **OBJ (.obj)**, que é mais simples e geralmente usado apenas para geometria, sem informações de animação ou hierarquia complexa.

## Passos para exportar do Blender:

### 1 Verifique a escala

Certifique-se de que seu modelo esteja na escala correta. Game engines como Unity e Unreal Engine geralmente usam metros como unidade padrão. No Blender, você pode ajustar as unidades de cena e aplicar a escala (Ctrl+A > Scale) antes de exportar.

### 2 Aplique transformações

Certifique-se de que a rotação e a escala do seu objeto estejam "aplicadas" (resetadas para 0 e 1, respectivamente). Isso evita problemas de orientação e tamanho na game engine.

### 3 Verifique os UVs

Seus UVs (coordenadas de mapeamento de textura) devem estar corretamente desdobrados e sem sobreposições significativas, para que as texturas sejam aplicadas de forma limpa.

### 4 Selecione o objeto

No Blender, selecione o(s) objeto(s) que deseja exportar.

### 5 Exportar

Vá em File > Export > FBX (.fbx). Na janela de exportação, certifique-se de marcar "Selected Objects" (Objetos Seleccionados) e, dependendo da engine, ajuste as opções de "Scale" (Escala) e "Forward/Up Axis" (Eixos de Frente/Cima) para corresponder às convenções da sua game engine (geralmente Y-Up ou Z-Up).

- Importante:** Suas texturas PBR (Albedo, Normal, Roughness, Metallic, etc.) devem ser exportadas como arquivos de imagem separados (PNG, JPG, TGA) e salvas em uma pasta organizada. Elas serão importadas separadamente na game engine e atribuídas ao material do seu modelo.

# Importando e Aplicando Materiais PBR em Game Engines

Com seu modelo 3D exportado e suas texturas prontas, o último passo é trazê-los para a game engine e dar vida ao seu material PBR. Tanto Unity quanto Unreal Engine, os motores de jogo dominantes no mercado, oferecem ferramentas visuais poderosas para gerenciar e aplicar materiais, tornando o processo intuitivo.

## No Unity

1. **Importar Modelo:** Arraste seu arquivo .fbx para a pasta Assets do seu projeto Unity. O Unity o importará automaticamente.
2. **Importar Texturas:** Arraste seus arquivos de imagem de textura (Albedo, Normal, Roughness, Metallic, etc.) para a mesma pasta Assets.
3. **Criar Material:** Clique com o botão direito na pasta Assets, vá em Create > Material. Nomeie-o adequadamente (ex: M\_MetalEnferrujado).
4. **Configurar Material:** Selecione o novo material. No painel Inspector, mude o Shader para Standard (que é o shader PBR padrão do Unity).
5. **Atribuir Texturas:** Arraste cada textura para o slot correspondente no Inspector:
  - o Albedo para Albedo
  - o Normal para Normal Map (certifique-se de que o tipo de textura no Inspector da imagem esteja como Normal Map)
  - o Roughness para Metallic (no Unity, o mapa de Roughness é geralmente invertido e conectado ao slot Smoothness ou ao canal Alpha do Metallic, dependendo do workflow)
  - o Metallic para Metallic (se for um mapa separado)
  - o Ambient Occlusion para Occlusion
6. **Aplicar ao Modelo:** Arraste o material criado para o seu modelo na cena ou para o slot de material no Renderer do modelo.

## No Unreal Engine

1. **Importar Modelo:** No Content Browser, clique em Import e selecione seu arquivo .fbx.
2. **Importar Texturas:** Arraste seus arquivos de imagem de textura para o Content Browser.
3. **Criar Material:** Clique com o botão direito no Content Browser, vá em Material. Nomeie-o (ex: M\_MetalEnferrujado).
4. **Configurar Material:** Dê um clique duplo no material para abrir o Material Editor.
5. **Atribuir Texturas:** Arraste suas texturas do Content Browser para o Material Editor. Conecte os nós de textura aos pinos correspondentes do nó Main Material:
  - o Albedo (RGB) para Base Color
  - o Normal (RGB) para Normal
  - o Roughness (Red ou RGB) para Roughness
  - o Metallic (Red ou RGB) para Metallic
  - o Ambient Occlusion (Red ou RGB) para Ambient Occlusion
6. **Salvar e Aplicar:** Salve o material e arraste-o para o seu modelo na cena ou para o slot de material no Details do modelo.



# Consolidação: Dando Vida aos Seus Mundos Virtuais

## Fundamentos do PBR

Metodologia baseada em princípios físicos que transformou a indústria de jogos

## Mapas de Textura

Albedo, Normal, Roughness, Metallic, AO e Height trabalham em conjunto

## Ferramentas Práticas

GIMP e Photoshop para criar texturas simples e eficazes

## Pipeline Completo

Do Blender para Unity e Unreal Engine com confiança

Chegamos ao fim de nossa jornada pela criação e aplicação de materiais PBR. Vimos como essa metodologia, baseada em princípios físicos, transformou a indústria de jogos, permitindo que artistas e desenvolvedores criem superfícies incrivelmente realistas que interagem com a luz de forma consistente e convincente. Desde a compreensão do que é o PBR e por que ele é o padrão, até a exploração detalhada de cada mapa de textura – Albedo, Normal, Roughness, Metallic, AO e Height – você agora tem o conhecimento fundamental para dar vida aos seus modelos 3D.

Aprendemos que cada mapa é uma peça vital de um quebra-cabeça visual, e que a combinação correta dessas informações é o que gera o realismo. Exploramos como ferramentas como GIMP e Photoshop podem ser usadas para criar texturas simples, e como o pipeline de exportação do Blender para game engines como Unity e Unreal Engine é essencial para levar suas criações do software de modelagem para o ambiente interativo. Dominar o PBR não é apenas uma habilidade técnica; é uma forma de arte que permite contar histórias visuais mais ricas e imersivas.

### Em prática:

Comece com materiais simples: uma parede de tijolos, um pedaço de madeira, um metal polido. Crie os mapas essenciais (Albedo, Normal, Roughness, Metallic) e observe como cada um afeta o resultado final. Experimente com diferentes valores e veja como a luz reage. A prática constante e a observação do mundo real são seus melhores guias para dominar o PBR.

# Autoavaliação

## Questão 1

Qual é a principal vantagem do Physically Based Rendering (PBR) em comparação com métodos de renderização mais antigos?

1. Permite criar modelos 3D com mais polígonos.
2. Garante que os materiais se comportem de forma consistente sob qualquer condição de iluminação.
3. Reduz a necessidade de mapas de textura.
4. É exclusivo para game engines de alta performance.

## Questão 2

Qual mapa de textura é responsável por simular detalhes de superfície, como arranhões e relevos, sem adicionar geometria real ao modelo?

1. Albedo Map
2. Metallic Map
3. Normal Map
4. Roughness Map

## Questão 3

No contexto PBR, um valor alto (próximo ao branco) no mapa de Roughness indica uma superfície:

1. Lisa e polida, com reflexos nítidos.
2. Áspera e fosca, com reflexos difusos.
3. Metálica e brilhante.
4. Transparente.

## Questão 4

Ao exportar um modelo do Blender para uma game engine como Unity ou Unreal Engine, qual formato de arquivo é mais comumente recomendado por sua ampla compatibilidade e suporte a diversas informações (malha, UVs, animações)?

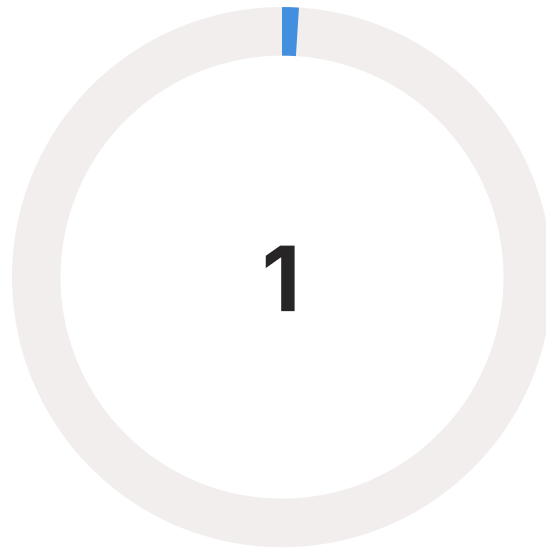
1. .blend
2. .obj
3. .fbx
4. .dae

---

## Questão Discursiva:

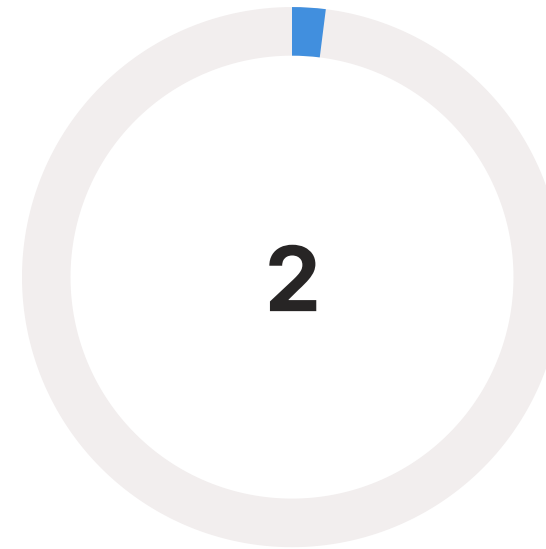
Explique a diferença fundamental entre o Normal Map e o Height/Displacement Map em termos de como eles afetam a geometria e o realismo visual de um modelo 3D, e quando cada um seria mais apropriado para uso em um projeto de game development.

# Gabarito



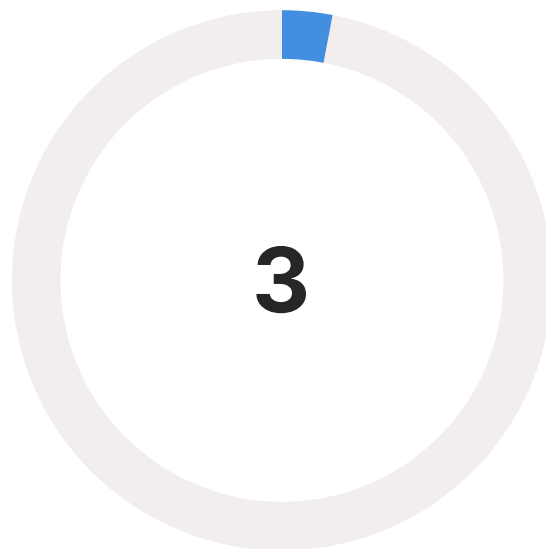
**Resposta**

**b)** Garante que os materiais se comportem de forma consistente sob qualquer condição de iluminação.



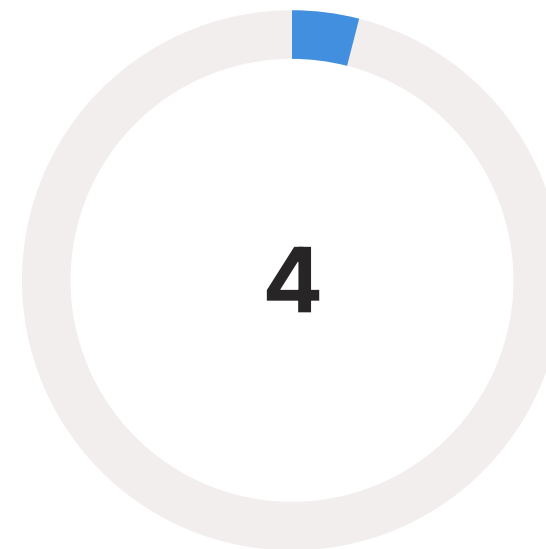
**Resposta**

**c)** Normal Map



**Resposta**

**b)** Áspera e fosca, com reflexos difusos.



**Resposta**

**c)** .fbx

# Próximos Passos

## Próxima Aula:

# Aula 9 – Animação 3D: Conceitos Fundamentais

Você dará o próximo passo para dar vida aos seus modelos, explorando os princípios e técnicas por trás da animação de personagens e objetos em 3D.

## Recursos Adicionais:

- **Documentação Oficial do Unity sobre PBR:** Para aprofundar nos detalhes de implementação no Unity.
- **Documentação Oficial da Unreal Engine sobre Materiais:** Para entender o sistema de materiais e shaders da Unreal.
- **Artigos da Allegorithmic (Substance 3D) sobre PBR:** Uma fonte rica de informações teóricas e práticas sobre PBR.

**NOTA IMPORTANTE:** As informações técnicas desta aula estão atualizadas até 2025. Consulte sempre as documentações oficiais das game engines e softwares para verificar alterações e novas funcionalidades.

