

Tópicos Especiais em Engenharia de Software (Jogos II)

Aula 02 – Iluminação, Materiais e Shaders

Edirlei Soares de Lima
<edirlei@iprj.uerj.br>



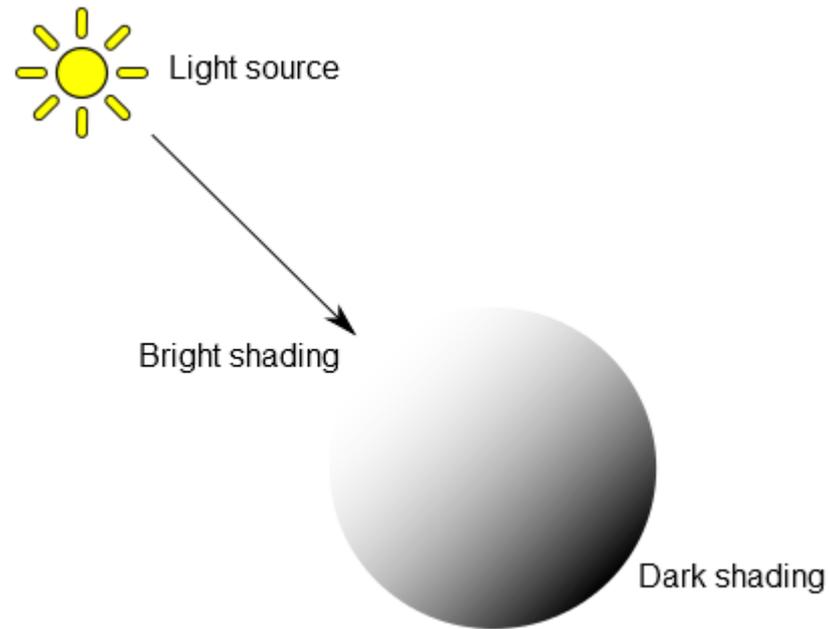
Unity 3D: Graphics

- A Unity oferece diversos elementos para a criação de ambientes virtuais realistas.
 - Iluminação em Tempo-Real
 - Sombras
 - Shaders
 - Sistema Partículas
 - Editor de Terrenos
 - Editor de Árvores
 - Água

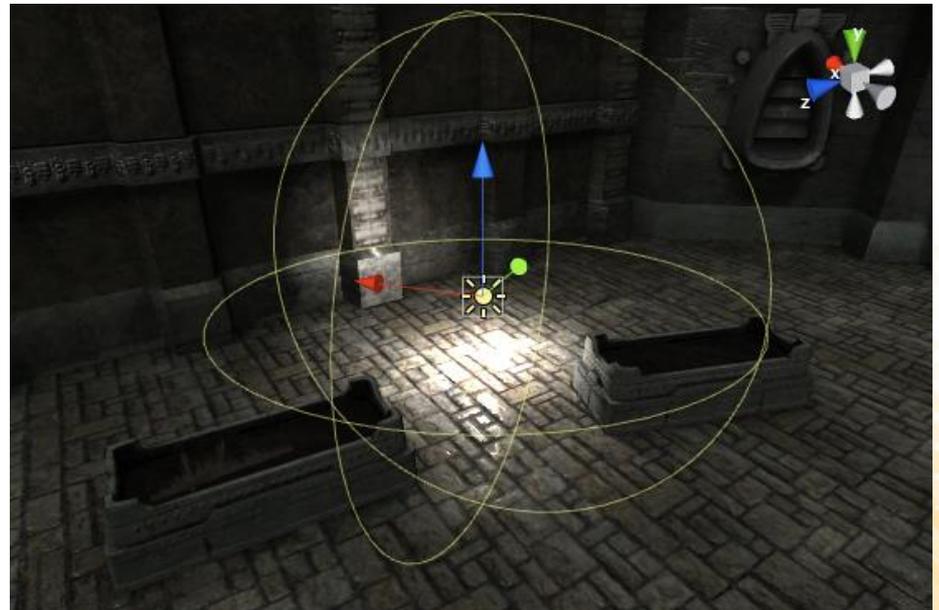
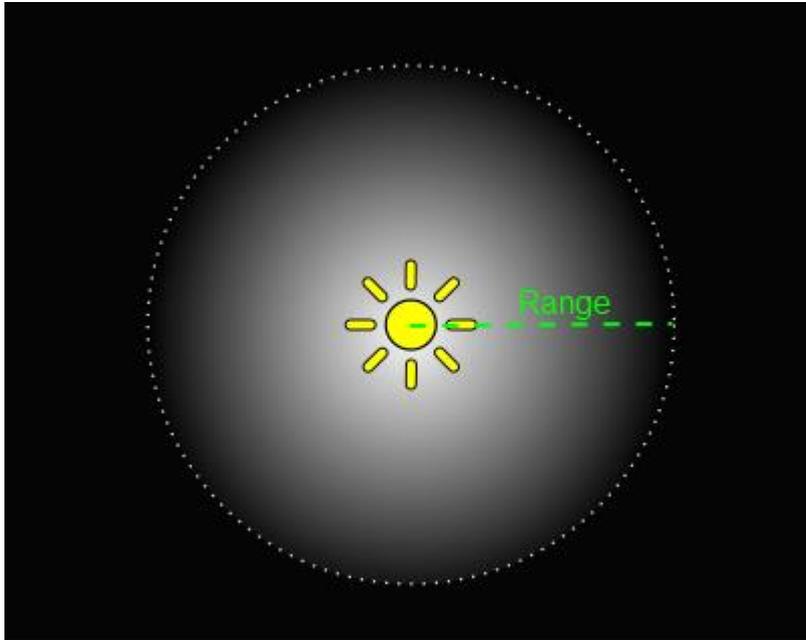


Iluminação

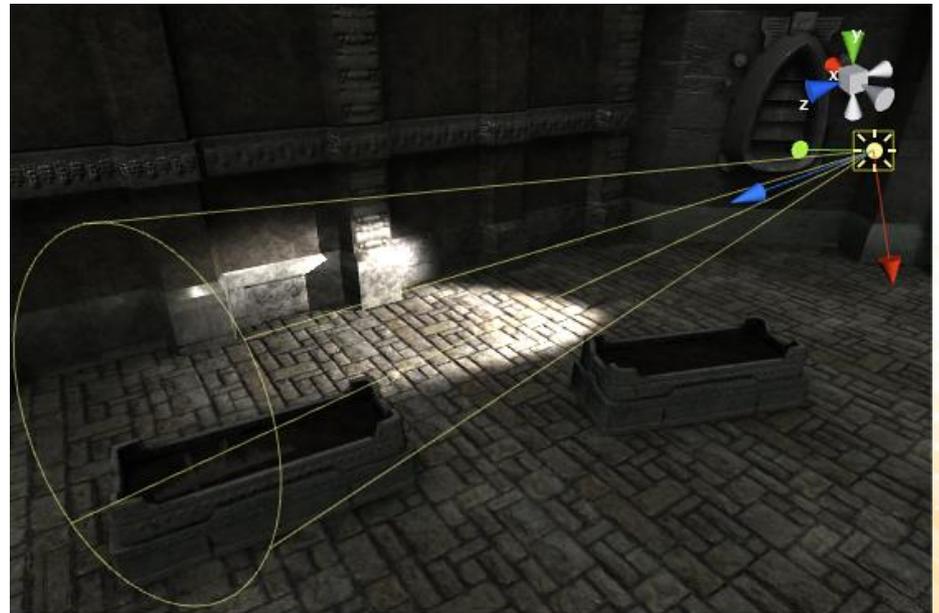
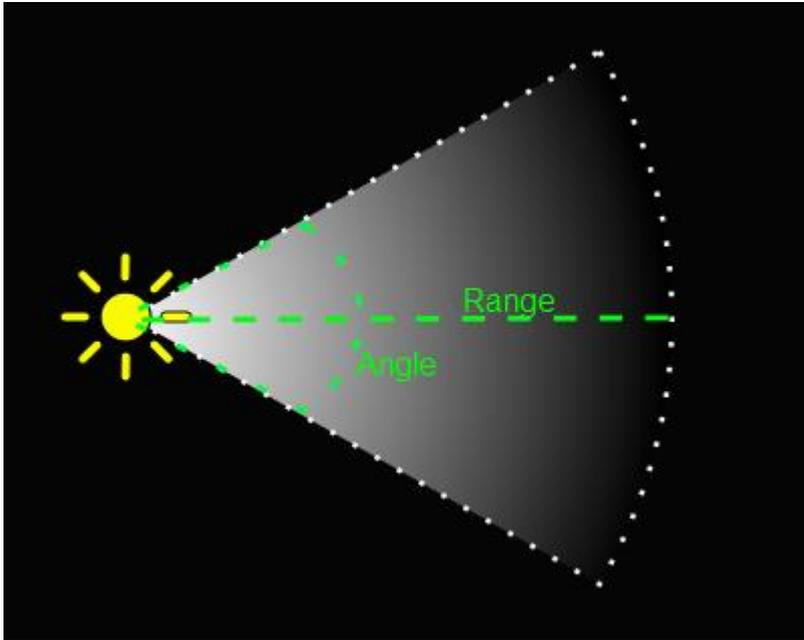
- Para calcular a incidência da iluminação em objetos, a Unity utiliza:
 - Intensidade da luz
 - Direção da luz
 - Cor da Luz
- Tipos de luz:
 - Point Lights
 - Spot Lights
 - Directional Lights
 - Area Lights



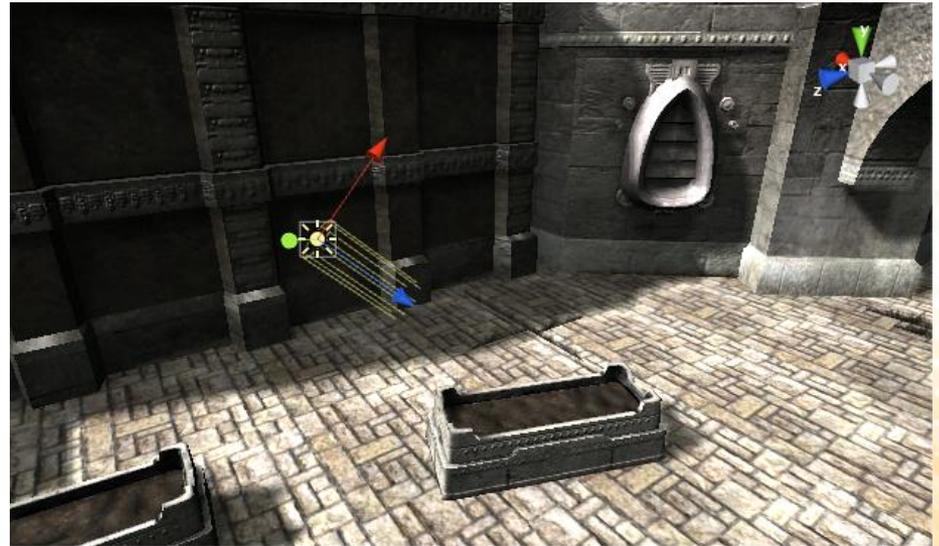
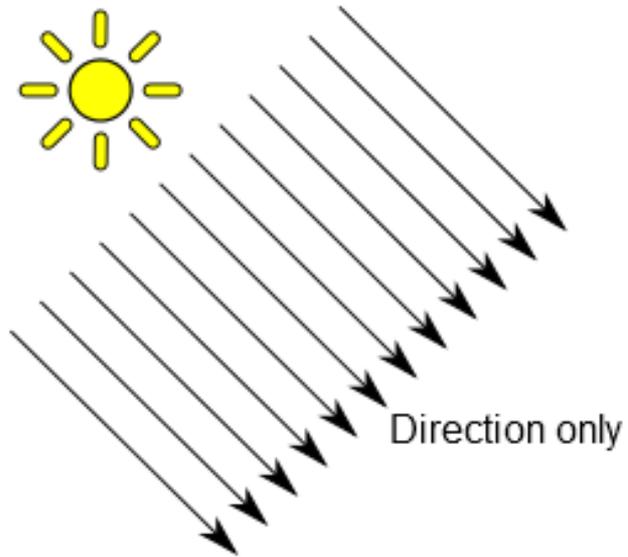
Iluminação – Point Lights



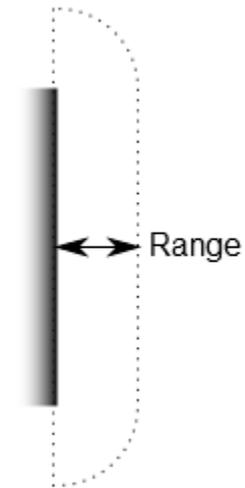
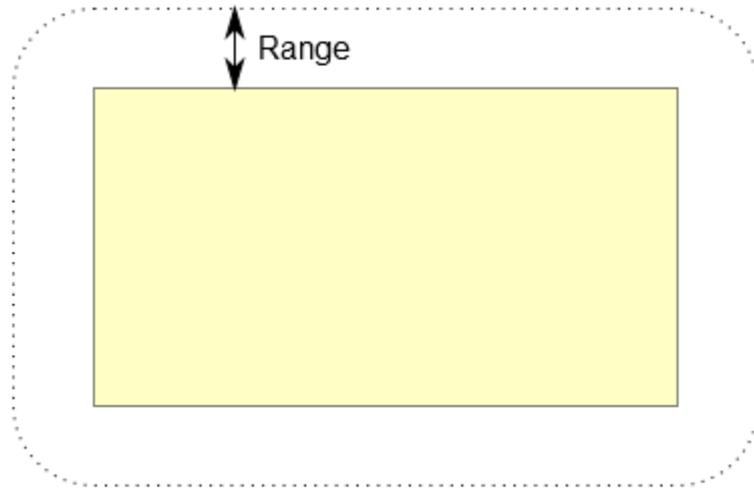
Iluminação – Spot Lights



Iluminação – Directional Lights



Iluminação – Area Lights

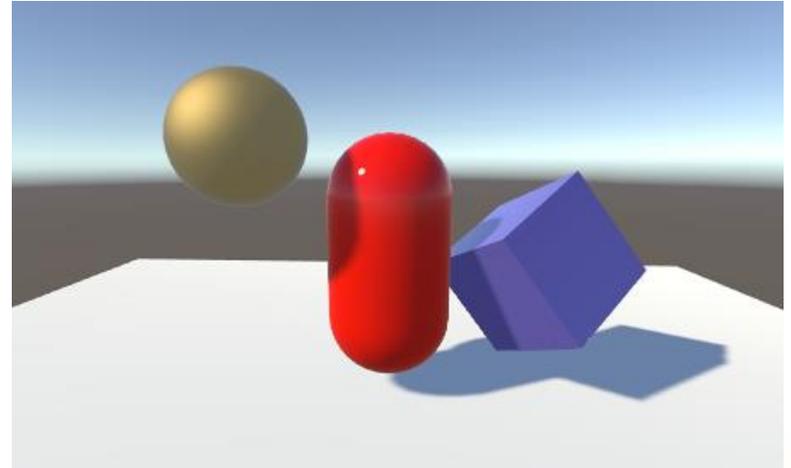


Iluminação

- **Directional Light:**
 - Geralmente utilizada para representar o sol.
 - Tem um efeito significativo sobre a aparência do ambiente.
- **Spot Light e Point Light:**
 - Geralmente utilizada para representar fontes de luz artificiais (lâmpadas).
 - O seu posicionamento normalmente é determinado pela posição dos objetos da cena.

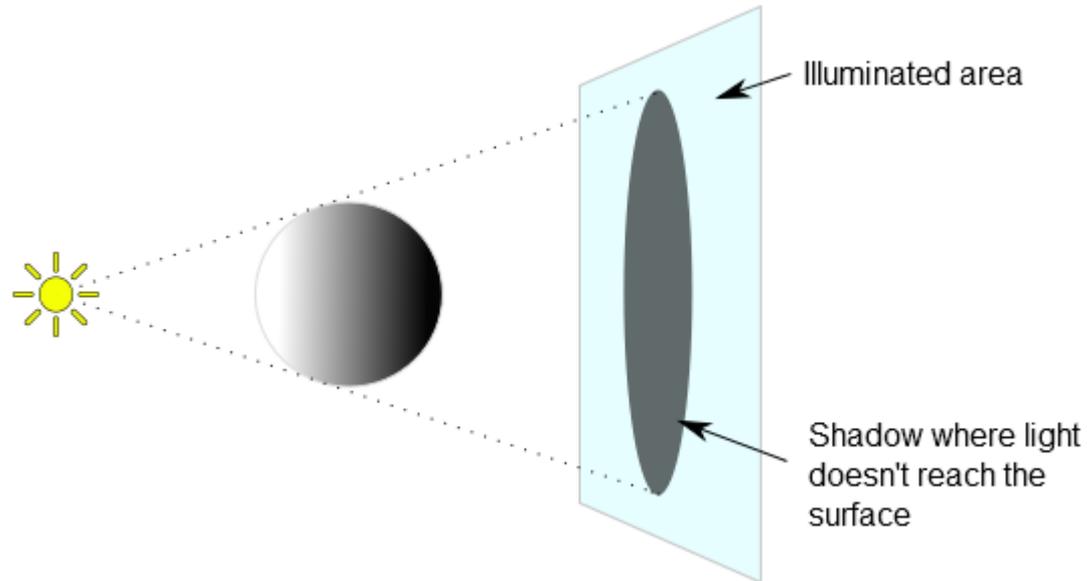
Sombras

- Assim como no mundo real, as luzes da Unity podem lançar sombras de um objeto em outras partes de si mesmo ou para outros objetos próximos.
- Sombras adicionam um grau de profundidade e realismo das cenas, uma vez que enfatizam a escala e a posição dos objetos.



Sombras

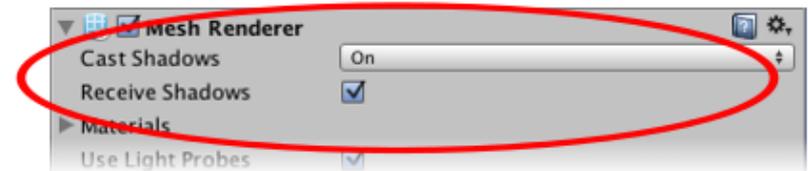
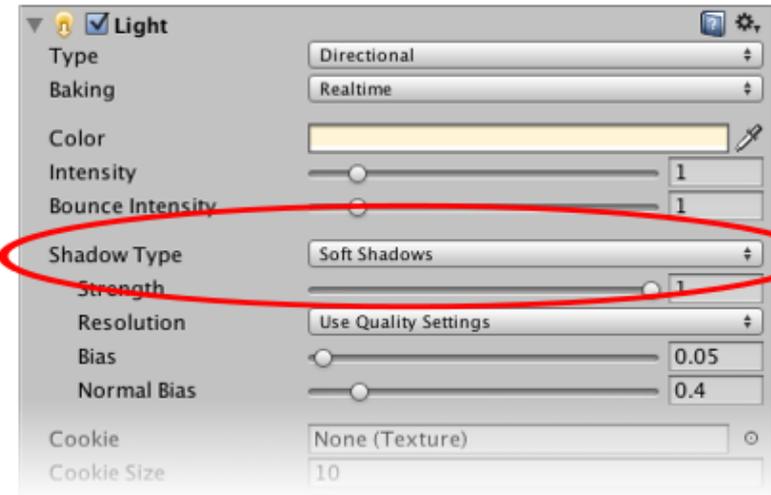
- Funcionamento:



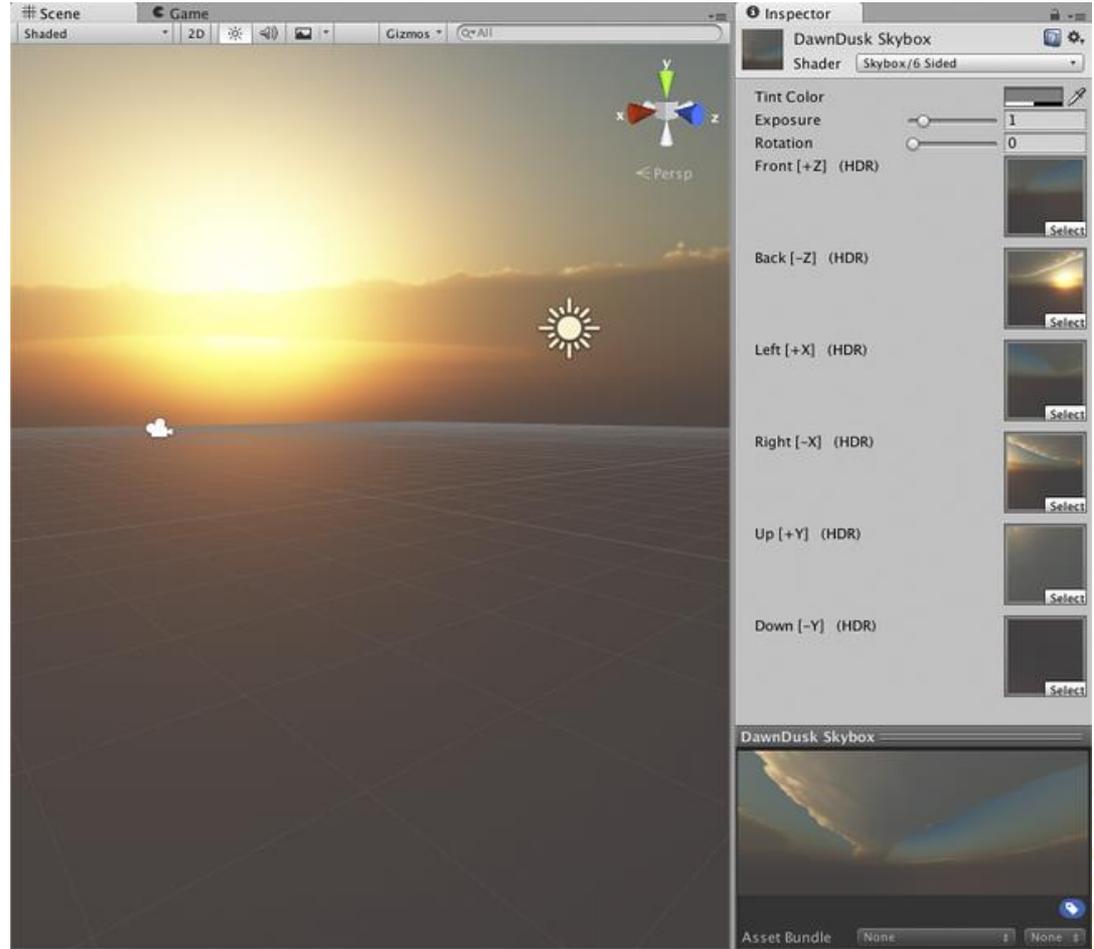
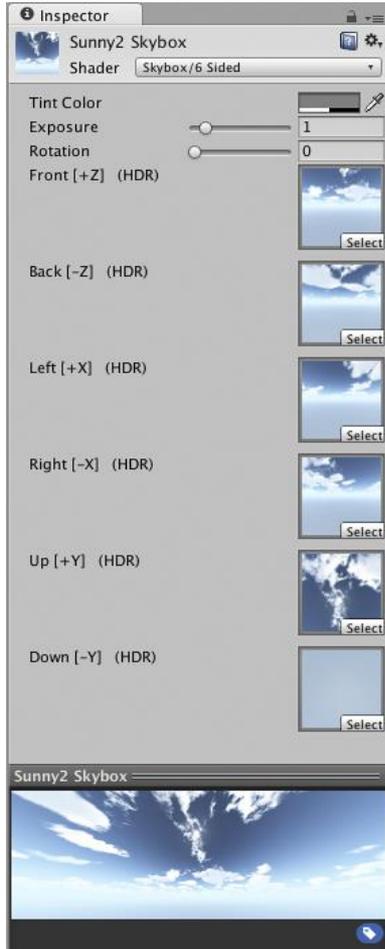
Sombras

- **Propriedades:**

- **Soft Shadows:** sombras com bordas suaves.
- **Hard Shadows:** sombras com bordas fortes (menos realistas, mas envolve menos tempo de processamento).
- **Strength:** determina o grau de escuridão das sombras.
- **Resolution:** resolução usada para renderização do mapa de sombras.



Skybox



Exercício 04

- Continue a implementação da fortaleza criada nos exercícios anteriores adicionando a ela os seguintes elementos:
 - a) Dentro da fortaleza, crie um ambiente interno (uma sala fechada).
 - b) Adicione e configure uma *spot light* dentro do ambiente interno.
 - c) Configure a *directional light* da cena para representar um “final de tarde” com pouca luminosidade.
 - d) Crie duas tochas na entrada na fortaleza utilizando *point lights*.
 - e) Configure as sombras da cena.
 - f) Crie e adicione a cena um *skybox* utilizando as seguintes texturas:



http://www.inf.puc-rio.br/~elima/jogos3d/exercicio4_skybox.html

Materiais, Shaders e Texturas

- **Material:** define como a superfície deve ser renderizada, incluindo referências a texturas, informações de tiling, cores, etc.
 - As opções disponíveis para um material dependem do Shader utilizado pelo material.
- **Shaders:** são pequenos scripts que contêm os cálculos matemáticos e algoritmos para determinar a cor de cada pixel do objeto representado de acordo com a luz e configuração do material.
- **Texturas:** são imagens.
 - Um material pode conter referências a texturas utilizadas pelo Shader para determinar a cor da superfície do objeto.
 - Texturas também podem representar outros aspectos da superfície de um material, tal como a sua refletividade ou deformações.

Standard Shader

- **Standard Shader:** shader padrão para a criação de materiais.
 - Suporta reflexão, bump mapping, mapa de oclusão, emissão, transparência, etc.
 - Physically Based Shading: simula a interação da luz com os materiais de forma que imita a realidade.
 - Exemplo de cena renderizada usando somente o Standard Shader:



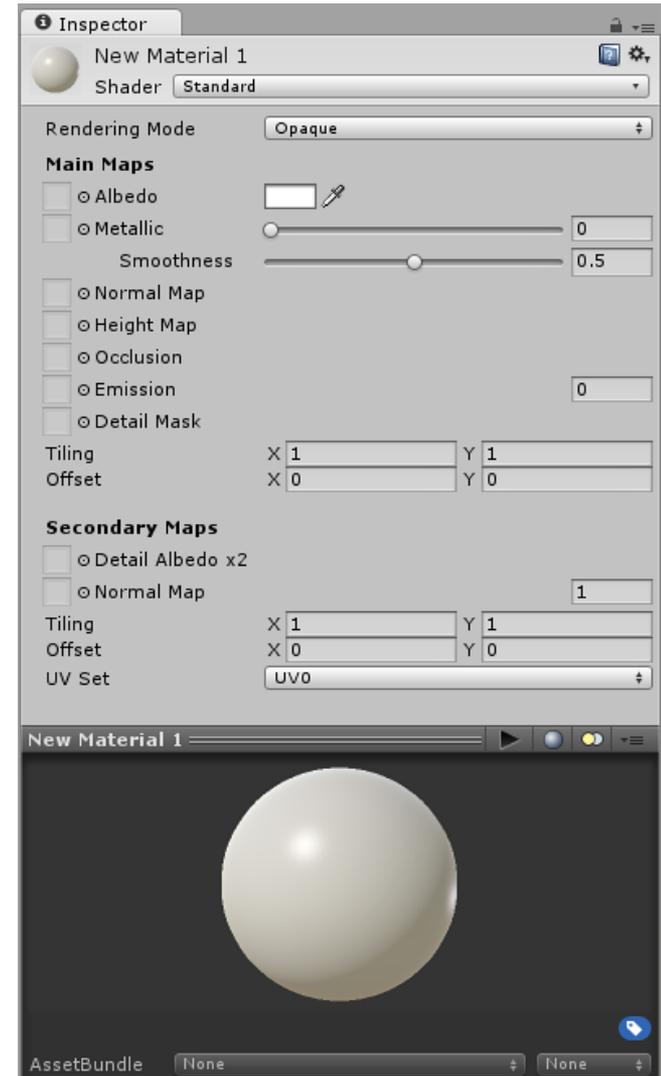
Standard Shader: Conteúdo e Contexto

- A aparência dos materiais baseados no Standard Shader é influenciada pelo contexto e conteúdo das cenas.
 - **Contexto:** fontes de luz, skybox, luz refletida de outros objetos, etc.
 - **Conteúdo:** objetos da cena, texturas, etc.
- Exemplo de cena com variações no contexto:



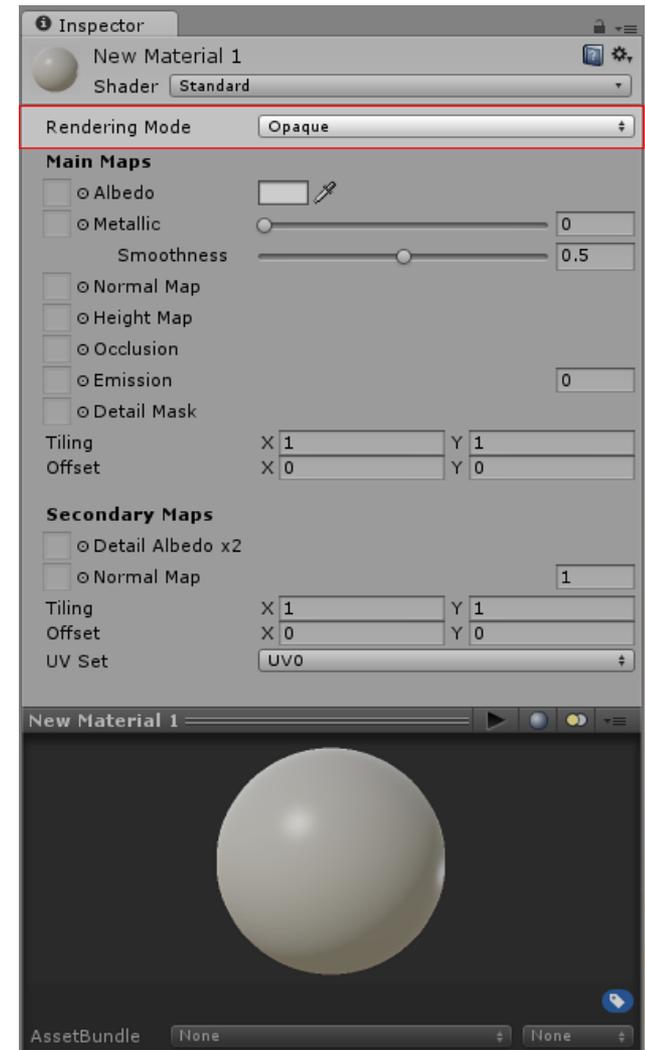
Standard Shader

- **Parâmetros do Material:**
 - Rendering Mode
 - Albedo Color & Transparency
 - Specular Mode: Specular
 - Metallic Mode: Metallic
 - Smoothness
 - Normal Map (Bump Mapping)
 - Height Map (Parallax Mapping)
 - Occlusion Map
 - Emission



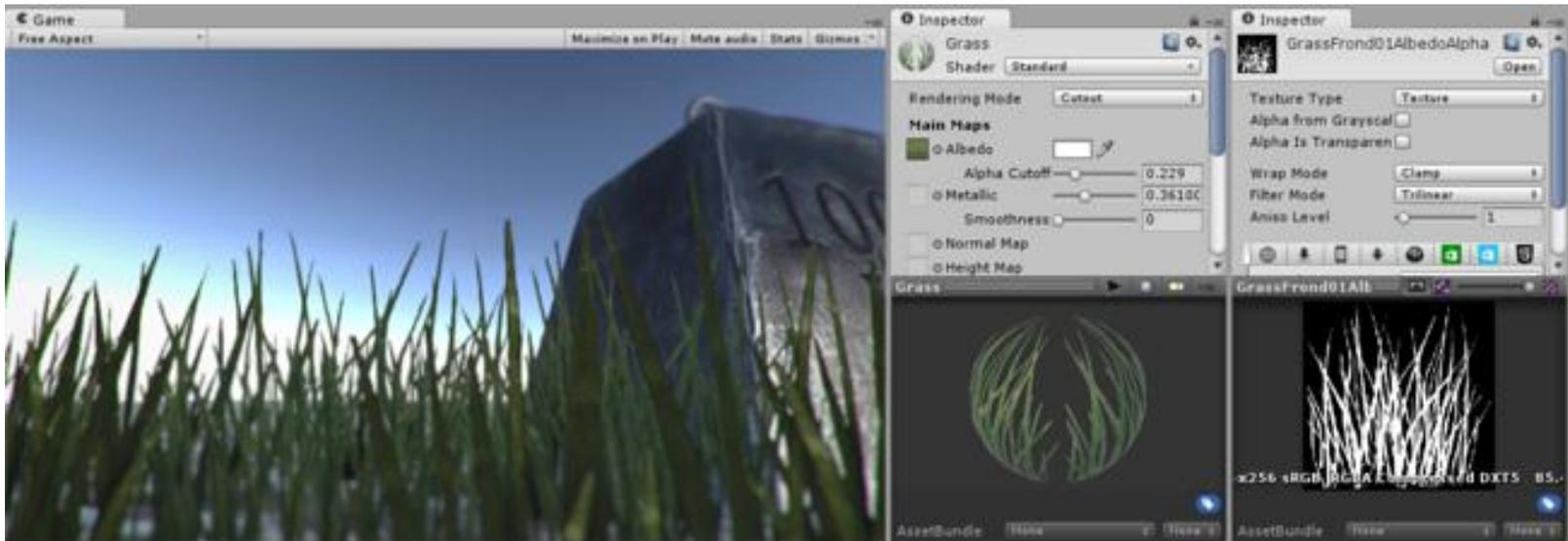
Standard Shader – Rendering Mode

- **Define a forma como o material será renderizado.**
 - **Opaque:** usado para materiais sólidos sem transparência.
 - **Cutout:** usado para criar materiais com áreas transparentes. Esse modo não permite áreas semitransparentes. Geralmente usado para representar folhas ou roupas com buracos.
 - **Transparent:** usado para criar materiais transparentes. Esse modo permite áreas semitransparentes. Geralmente usado para representar plástico ou vidro transparente.
 - **Fade:** usado para criar materiais transparentes sem reflexão.



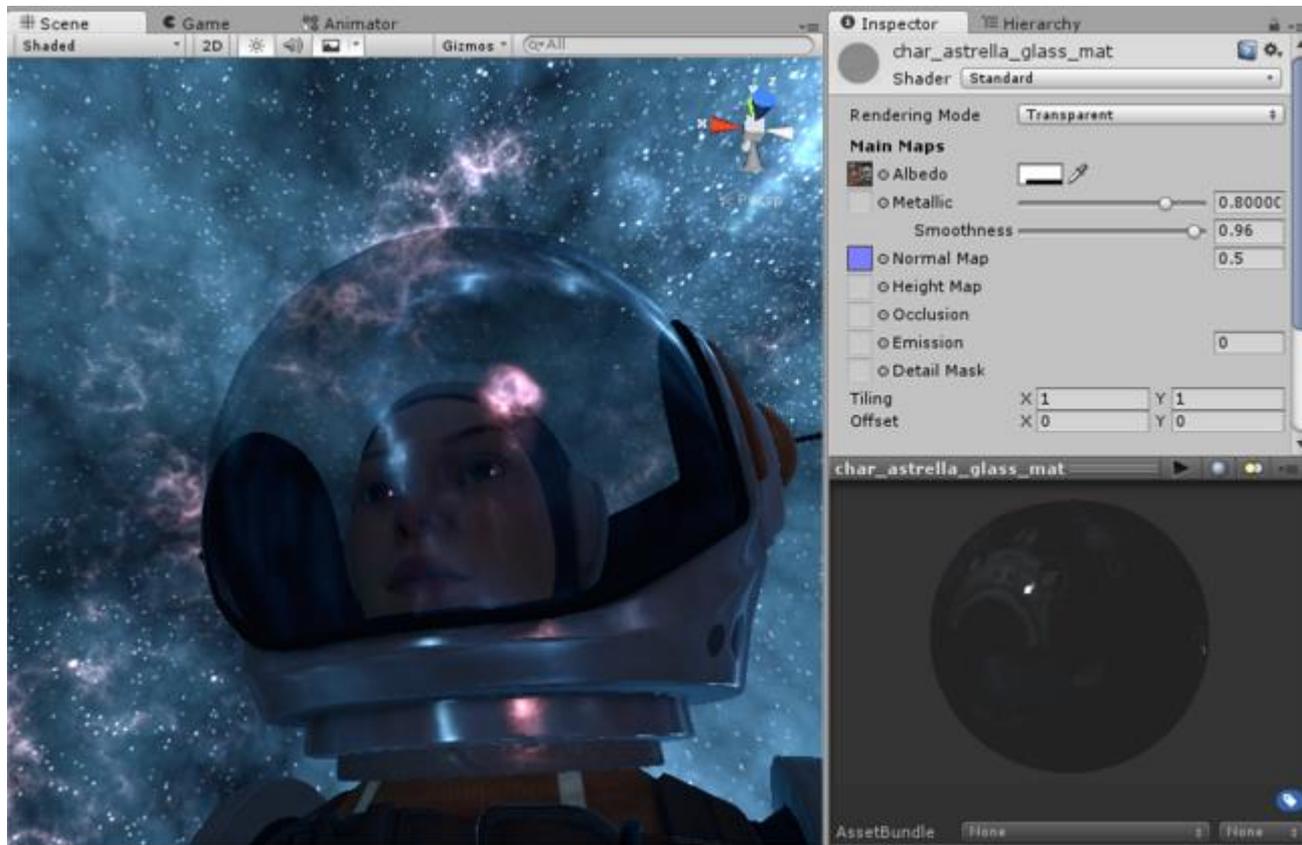
Standard Shader – Rendering Mode

- Exemplo Cutout:



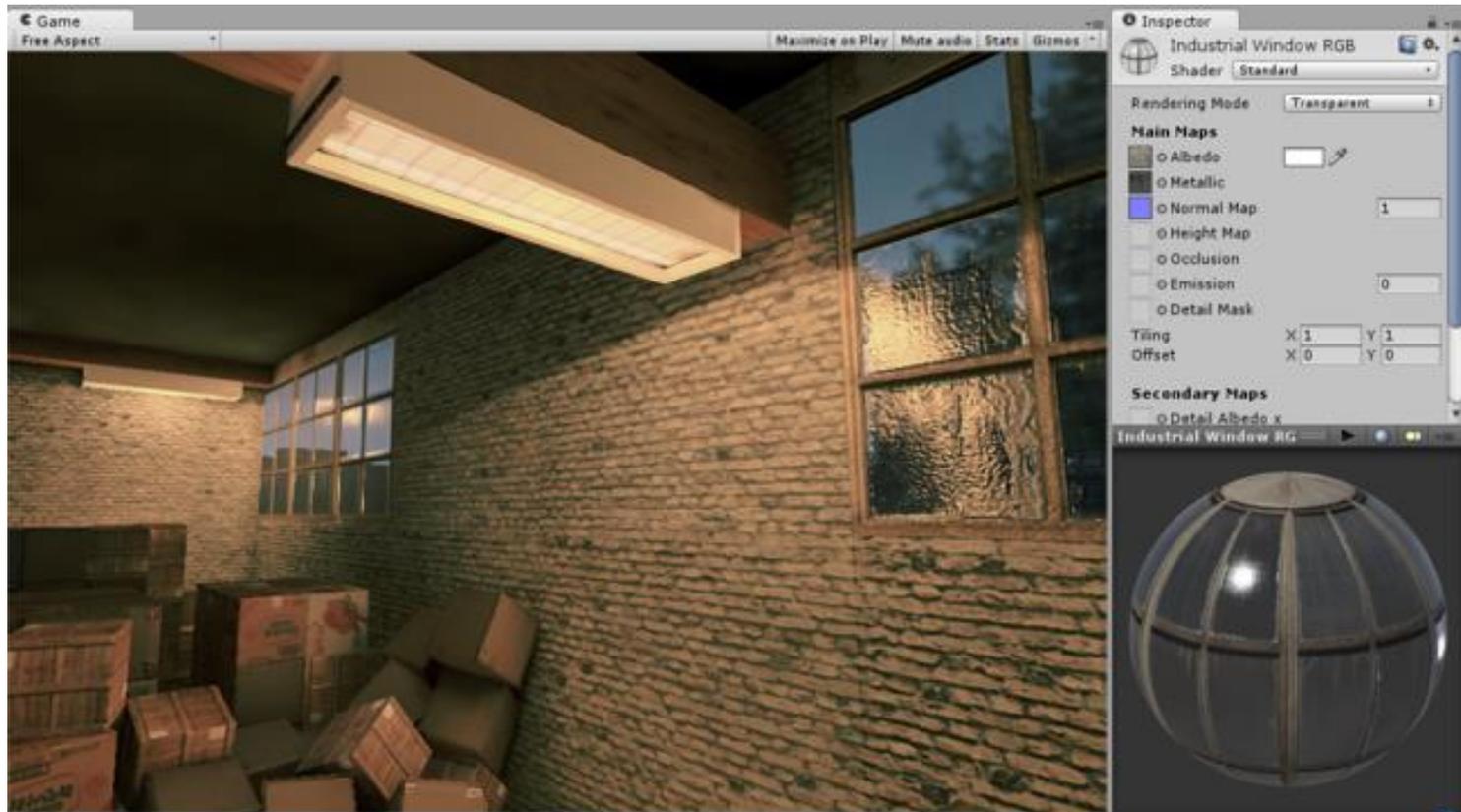
Standard Shader – Rendering Mode

- Exemplo Transparent:



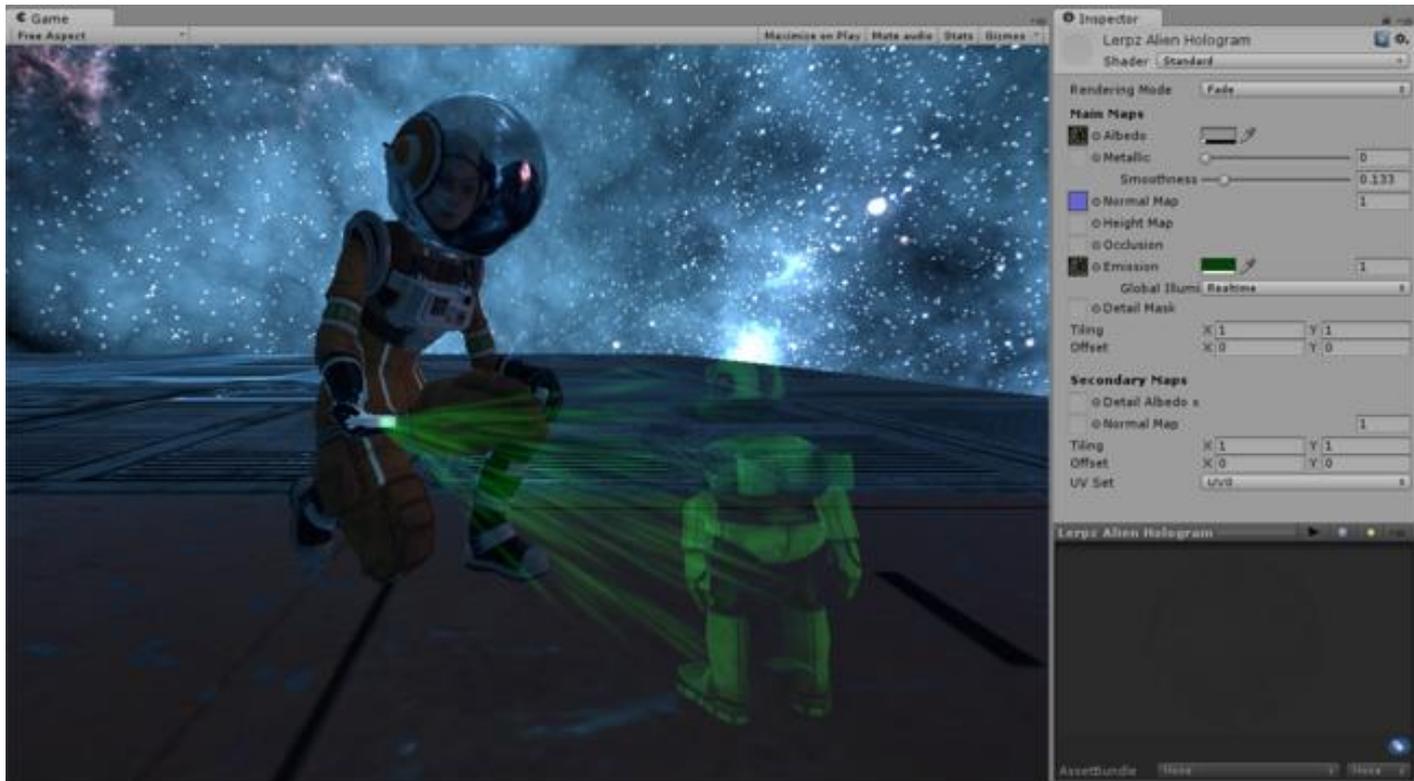
Standard Shader – Rendering Mode

- Exemplo Transparent (com áreas opacas):



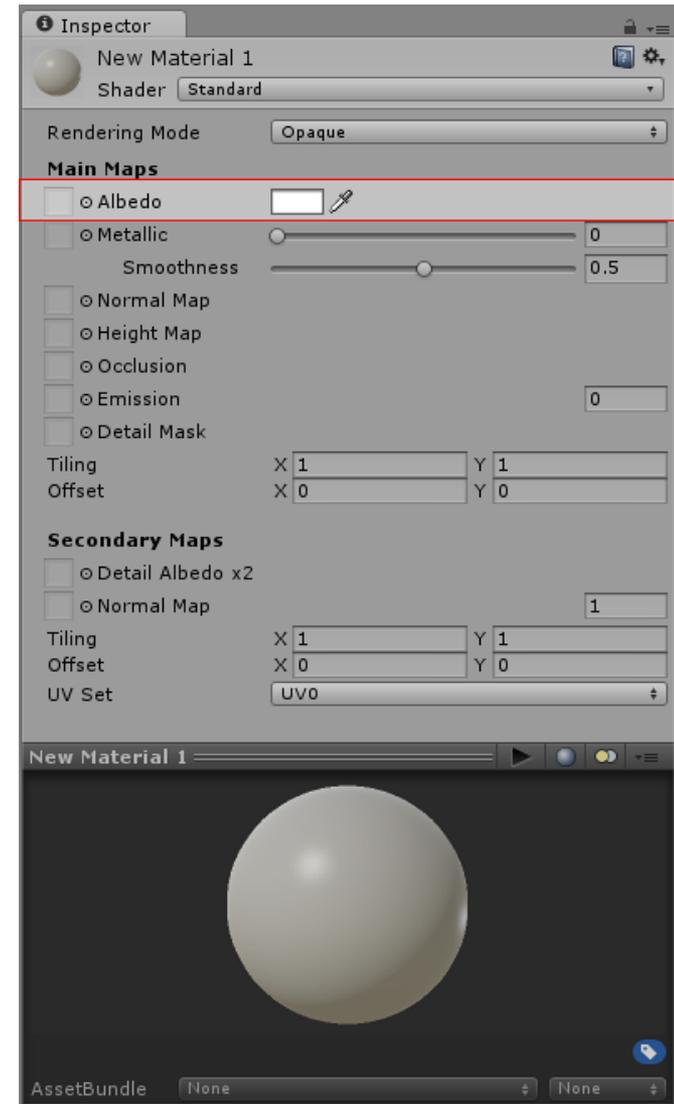
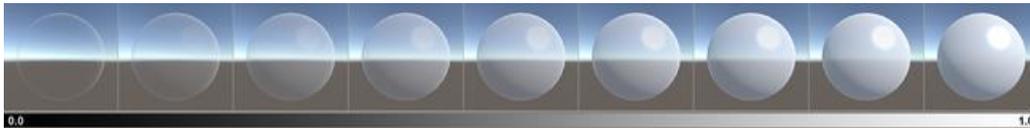
Standard Shader – Rendering Mode

- Exemplo Fade:



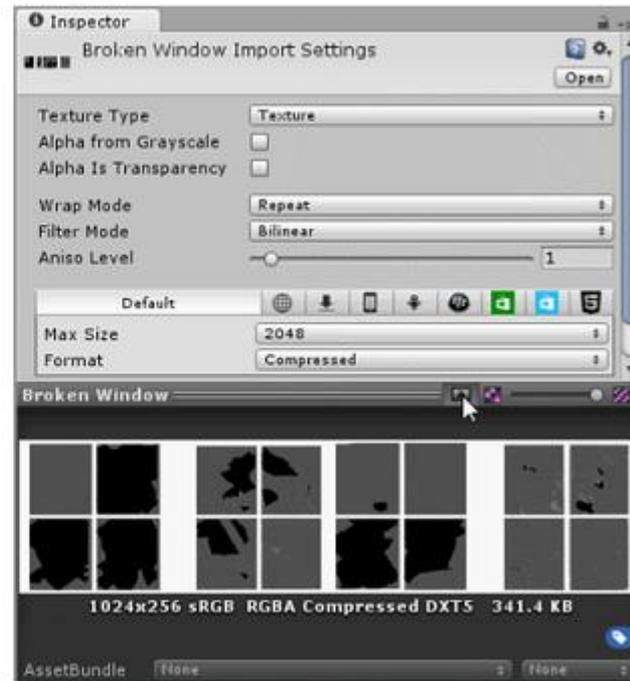
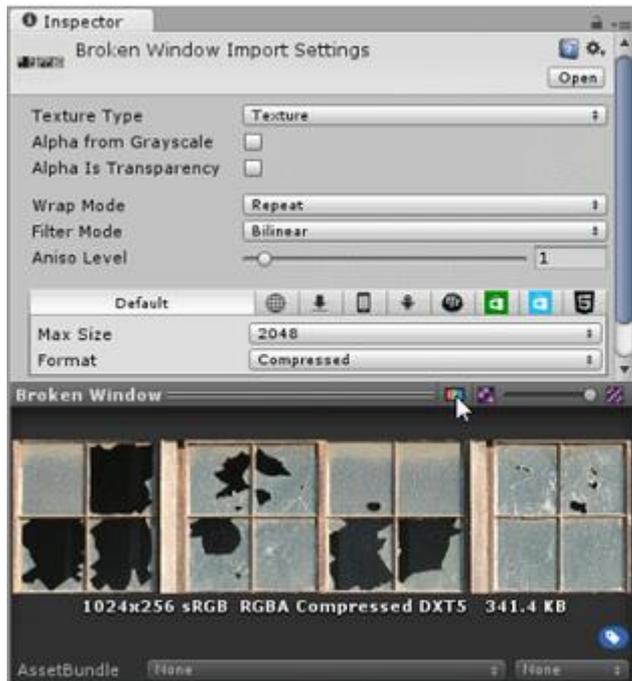
Standard Shader – Albedo

- Determina a cor ou textura base do objeto.
- A transparência total do material pode ser determinada pelo canal alpha da cor.



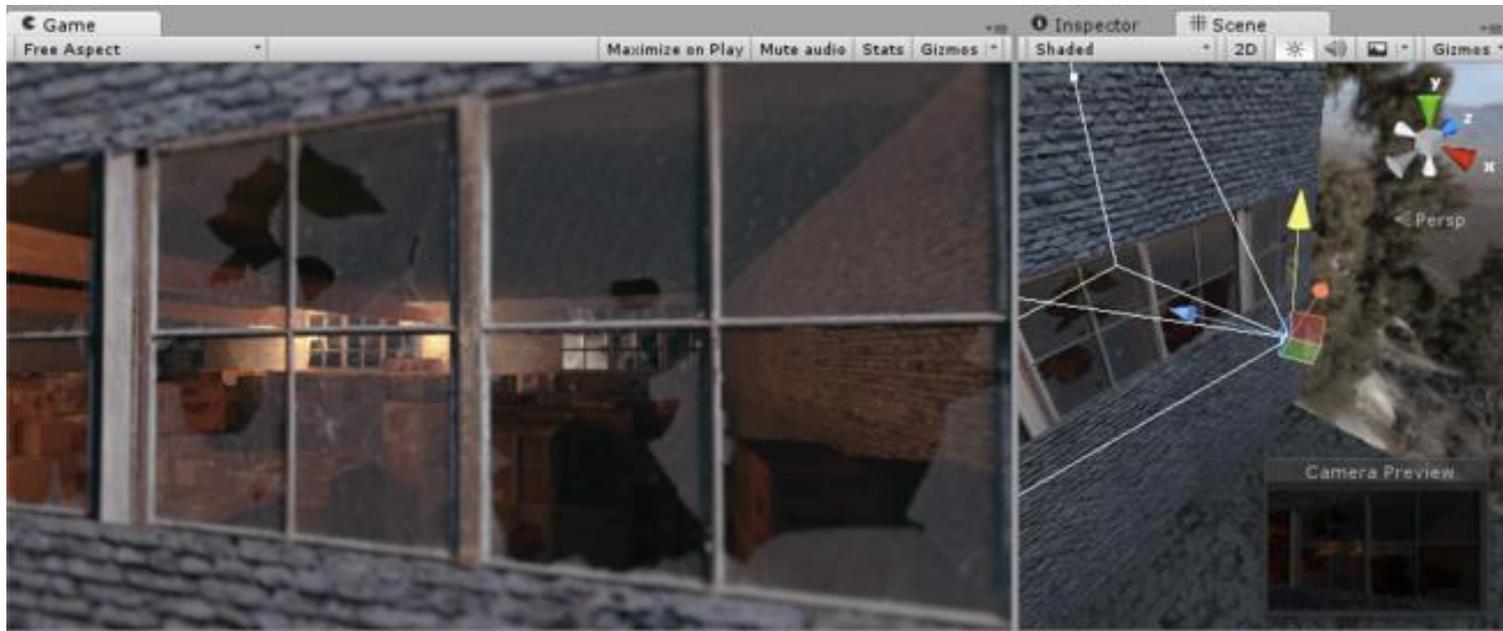
Standard Shader – Albedo

- Transparências em áreas do material devem ser definidas no canal alpha da textura:



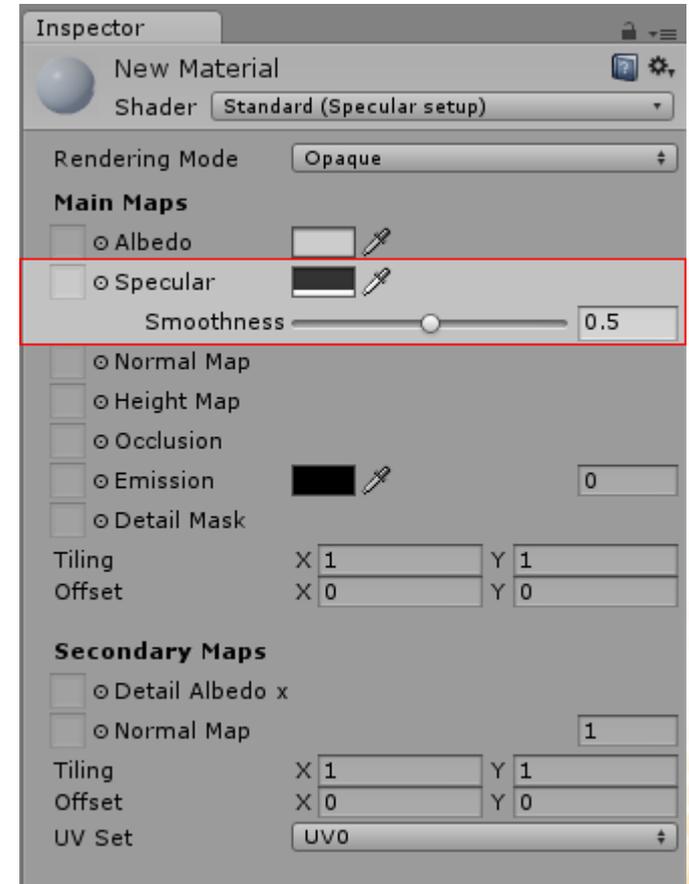
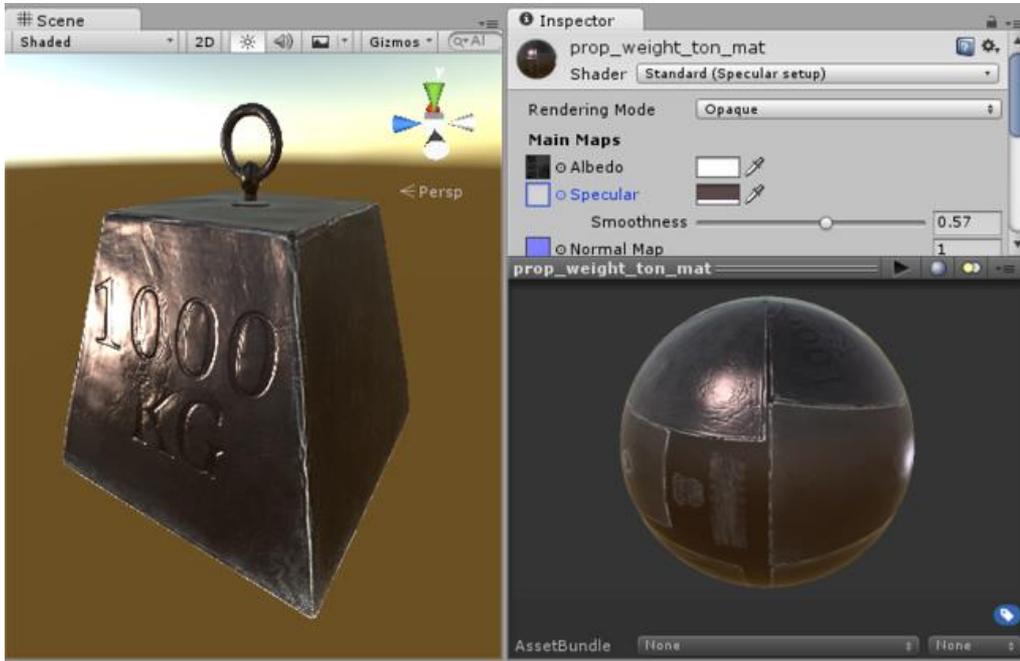
Standard Shader – Albedo

- Transparências em áreas do material devem ser definidas no canal alpha da textura:



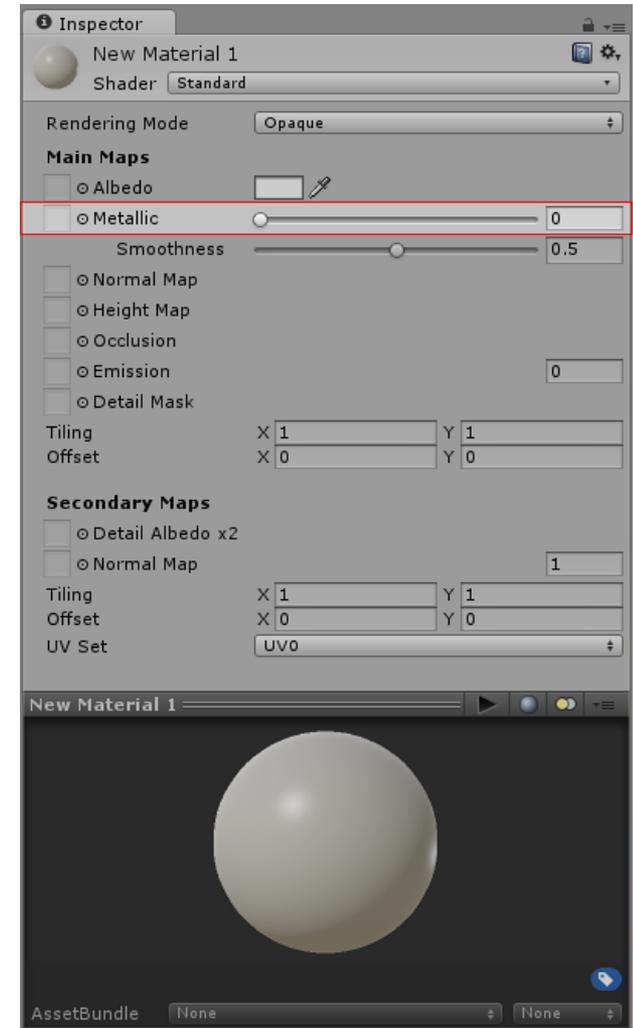
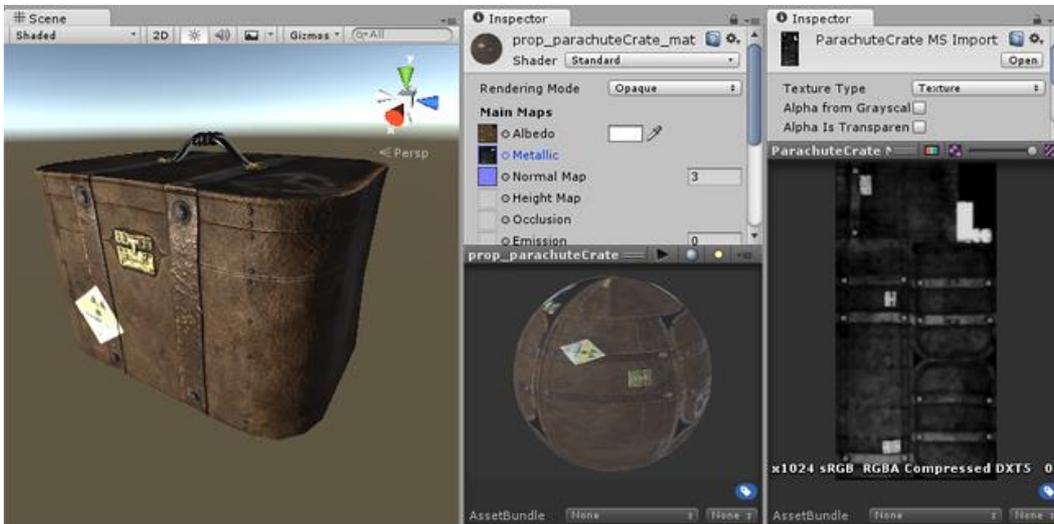
Standard Shader – Specular (Specular Mode)

- Determina a reflexão do objeto.



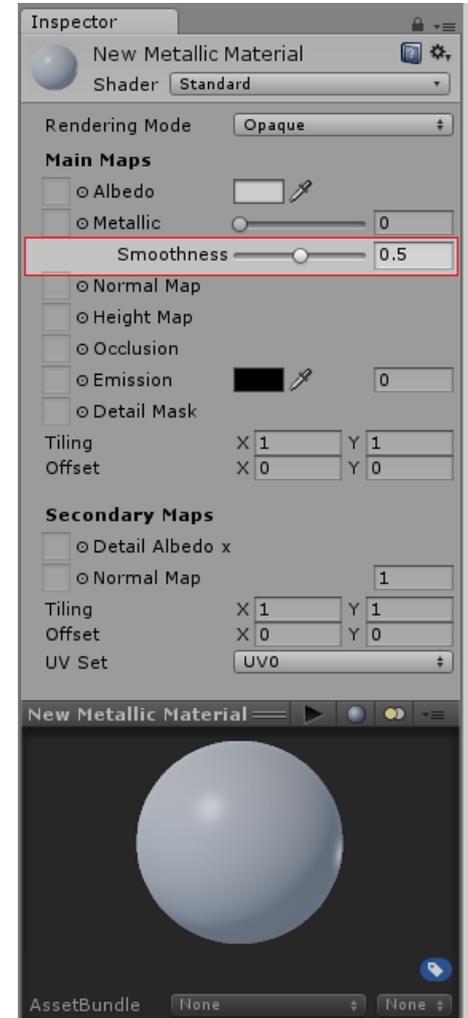
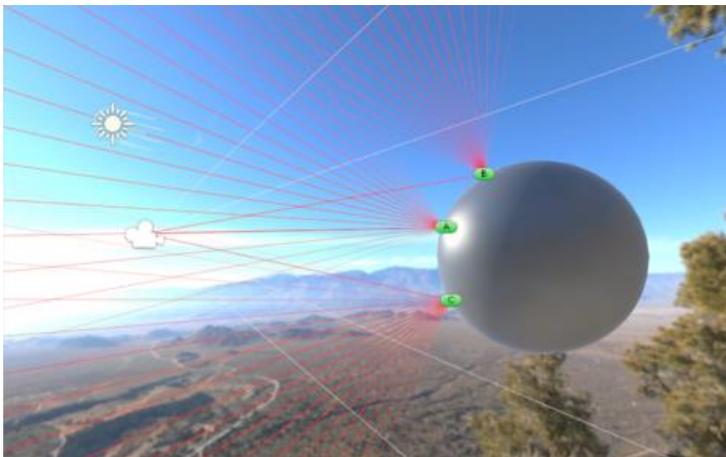
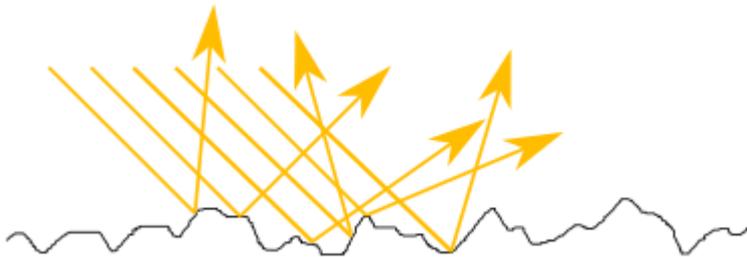
Standard Shader – Metallic

- Determina o grau do aspecto metálico da superfície do objeto.



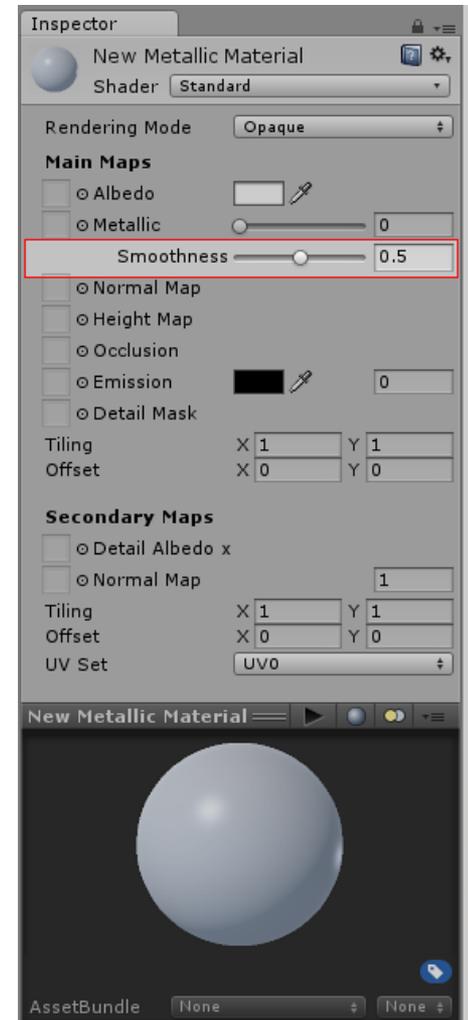
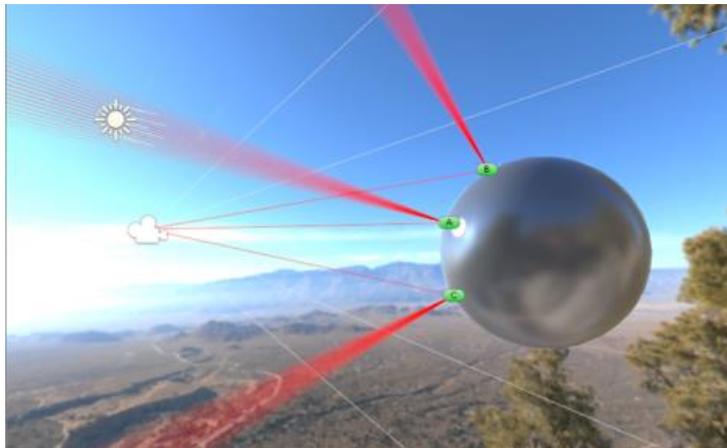
Standard Shader – Smoothness

- Determina a forma como a luz é refletida quando ela atinge a superfície do objeto.
 - Smoothness baixo:



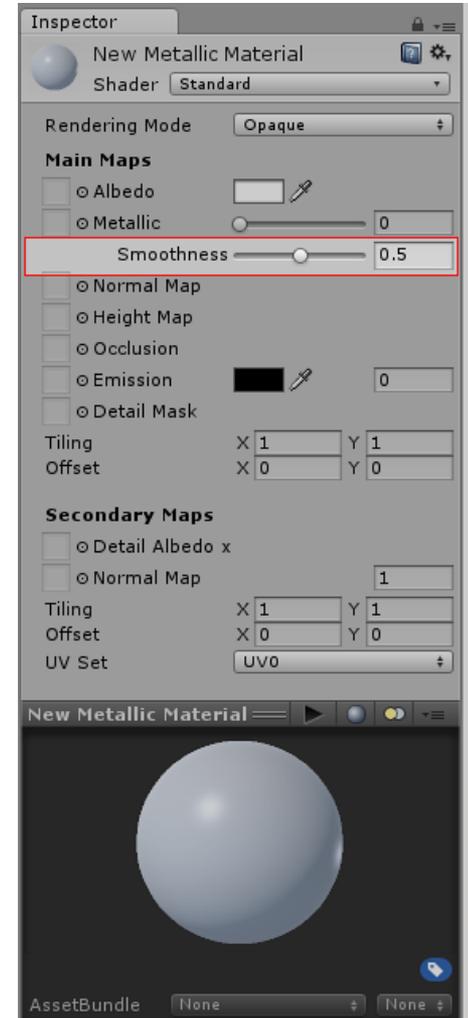
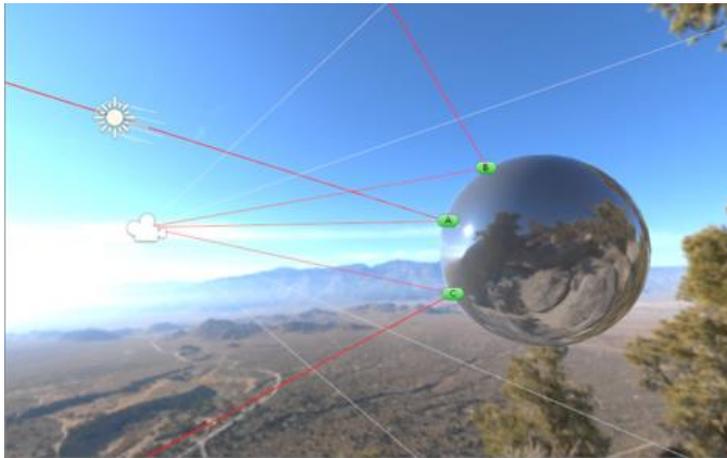
Standard Shader – Smoothness

- Determina a forma como a luz é refletida quando ela atinge a superfície do objeto.
 - Smoothness médio:



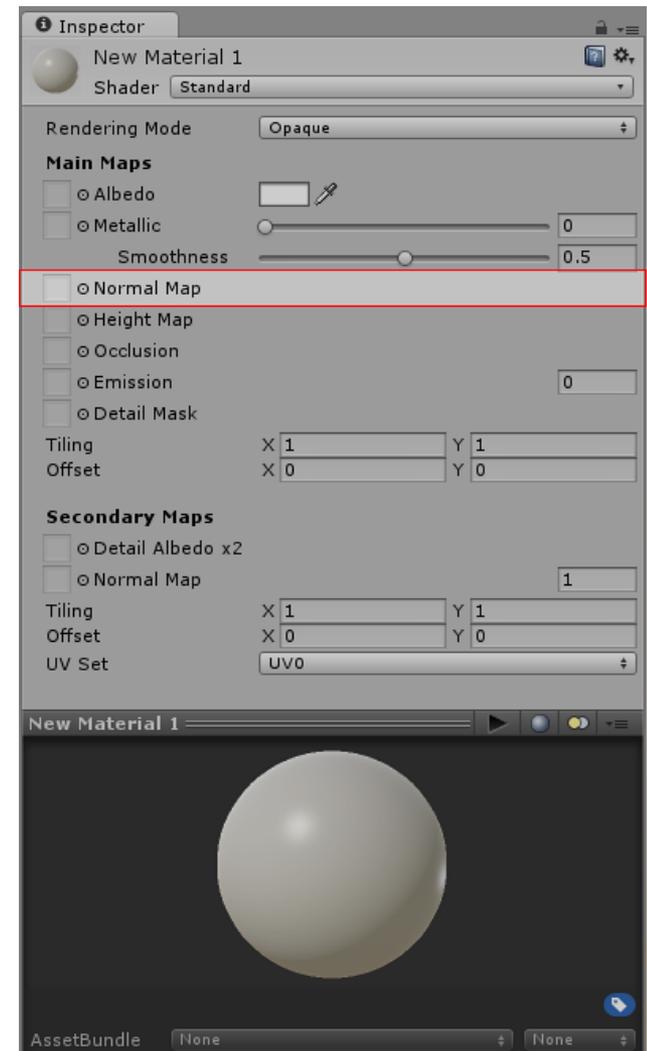
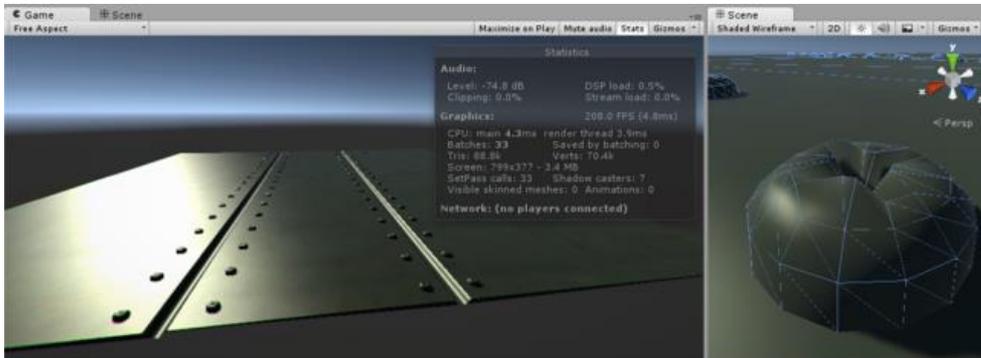
Standard Shader – Smoothness

- Determina a forma como a luz é refletida quando ela atinge a superfície do objeto.
 - Smoothness alto:



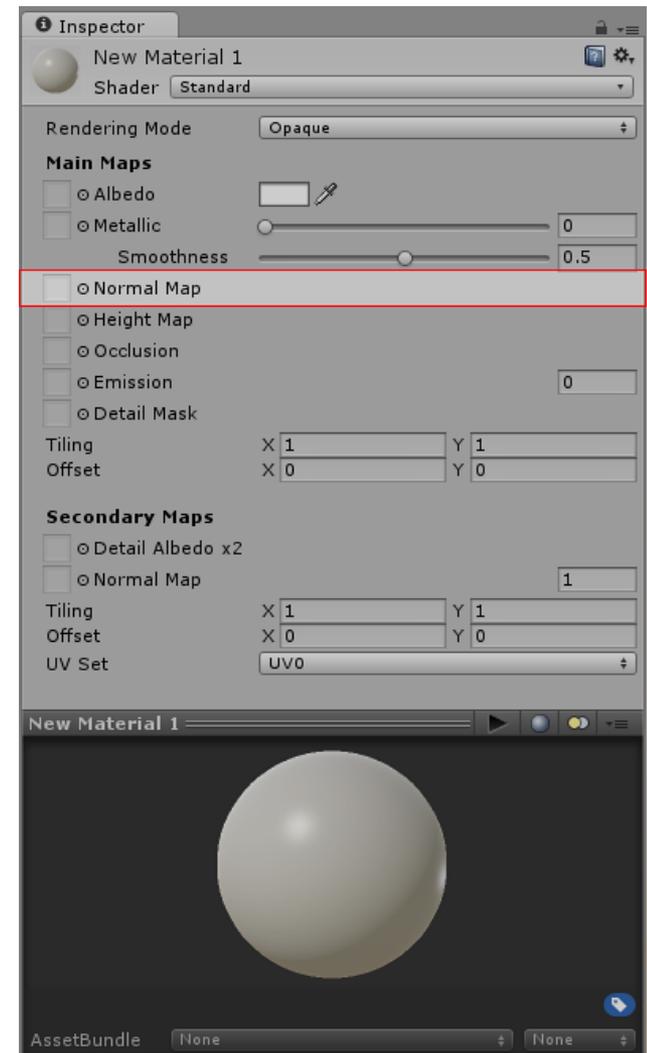
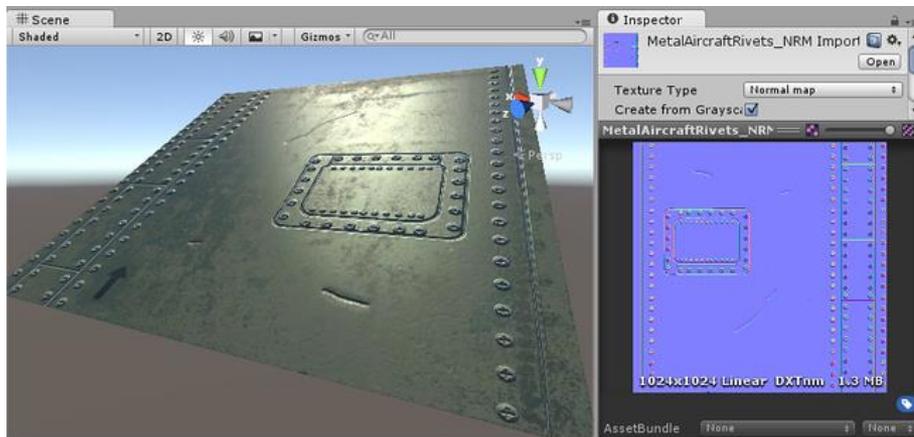
Standard Shader – Normal Map

- Permite representar detalhes em superfícies sem acrescentar vértices a geometria dos objetos.
 - Bump Mapping
- Exemplo: parafusos em uma placa de metal (sem bump mapping)



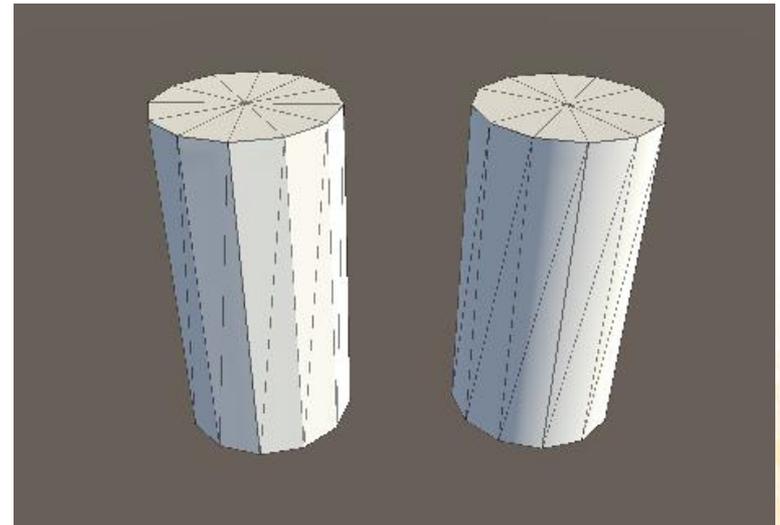
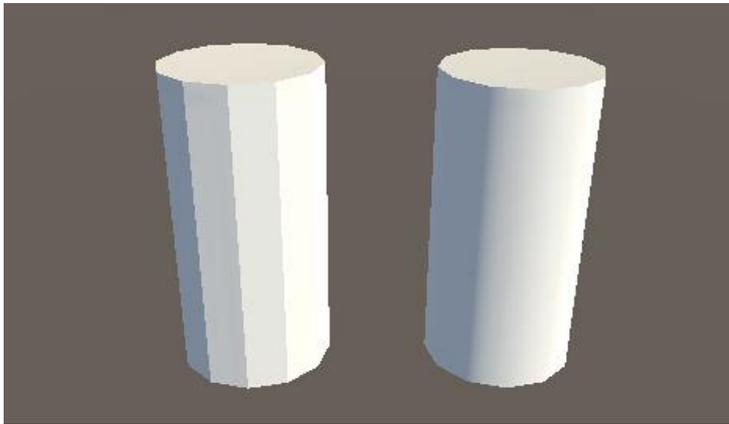
Standard Shader – Normal Map

- Permite representar detalhes em superfícies sem acrescentar vértices a geometria dos objetos.
 - Bump Mapping
- Exemplo: parafusos em uma placa de metal (com bump mapping)



Standard Shader – Normal Map

- O Bump Mapping é uma técnica que utiliza um mapa de normais para aplicar perturbações no vetor de normais da superfície dos objetos.
- O que são normais?



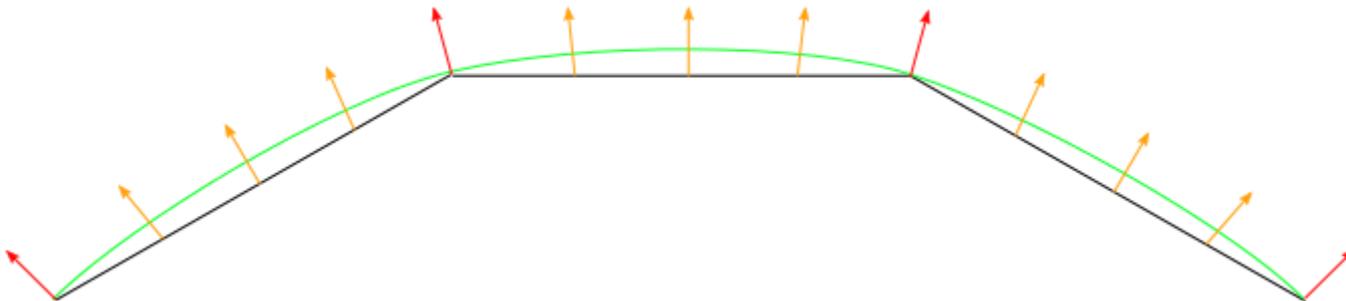
Standard Shader – Normal Map

- O que são normais?

Superfície achatada:

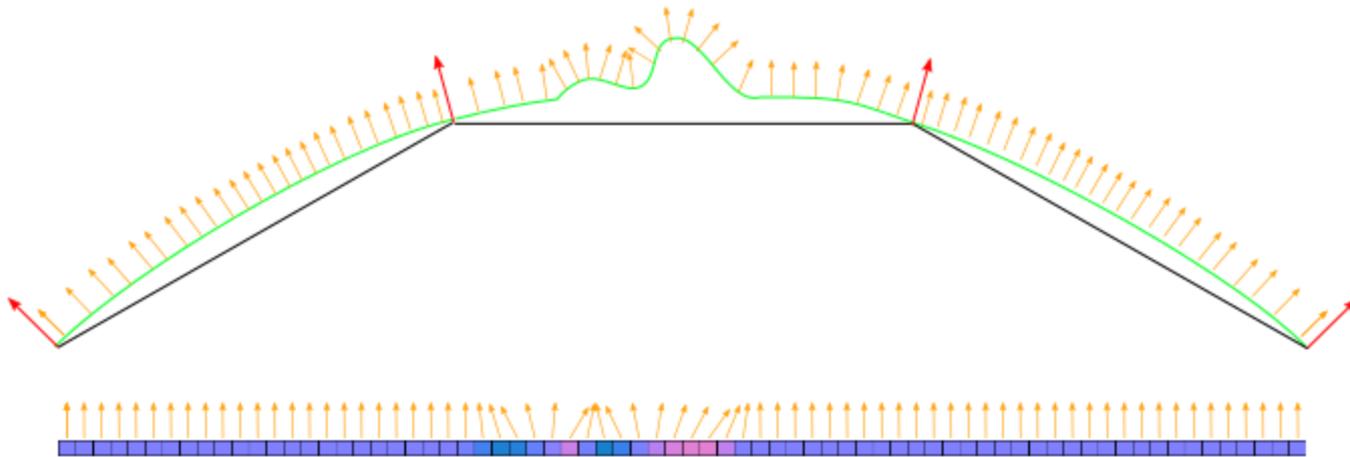


Superfície suave:



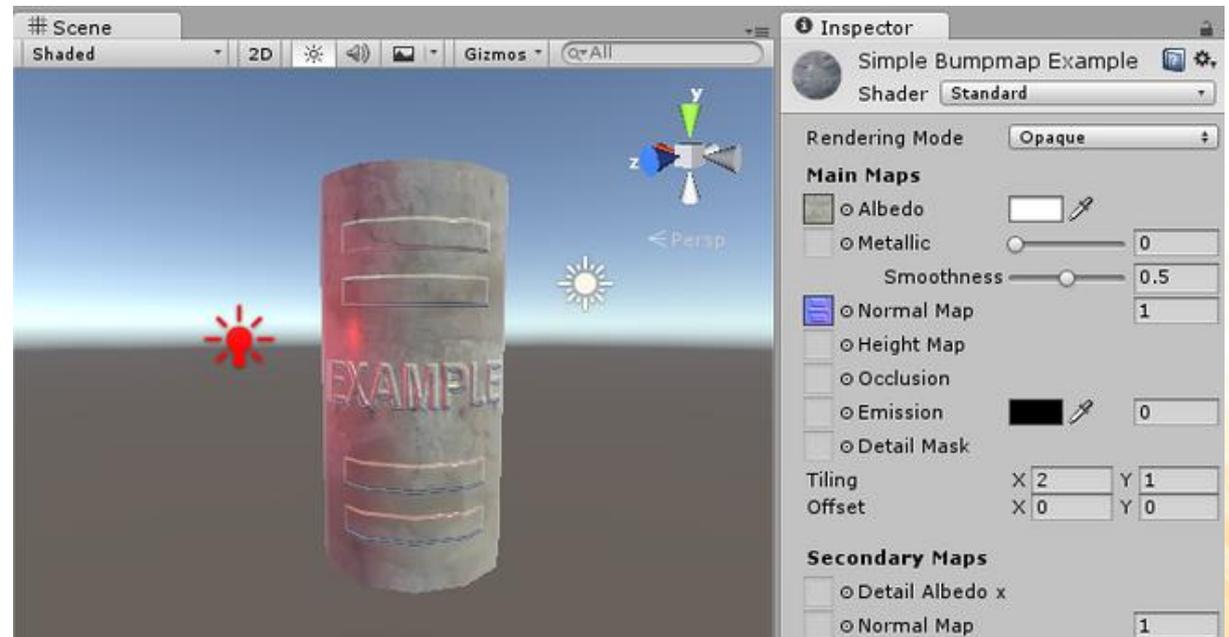
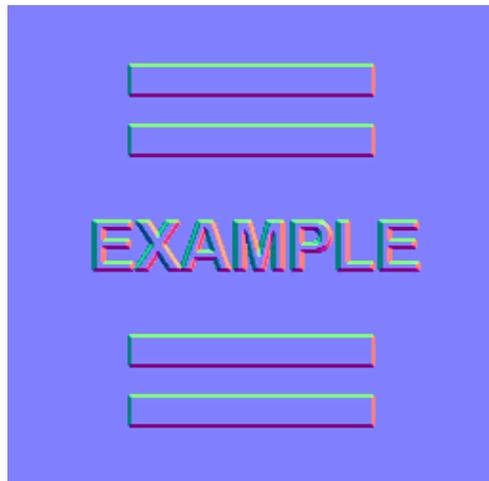
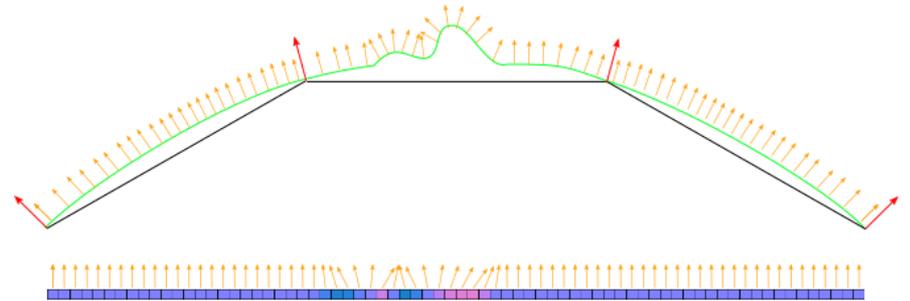
Standard Shader – Normal Map

- O que é normal mapping?



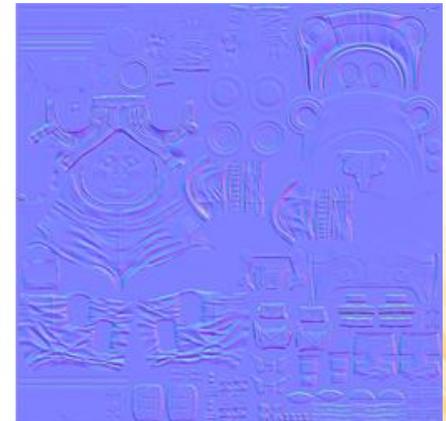
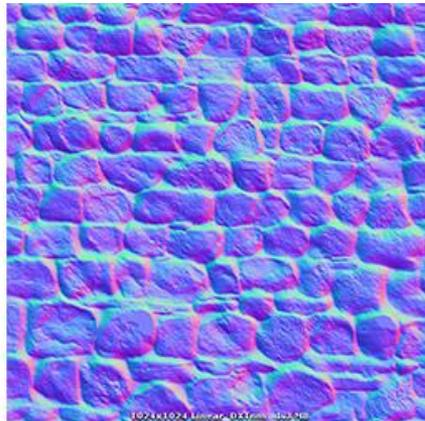
Standard Shader – Normal Map

- O que é normal mapping?



Standard Shader – Normal Map

- **Como criar um normal map?**
 - Durante a modelagem dos objetos 3D
 - Versão detalhada + versão menos detalhada do modelo 3D
 - A partir da textura
 - ShaderMap - <http://shadermap.com/>
 - Manualmente



Standard Shader – Normal Map

- **Resultado:**



Sem Normal Mapping

Standard Shader – Normal Map

- **Resultado:**



Com Normal Mapping (directional light)

Standard Shader – Normal Map

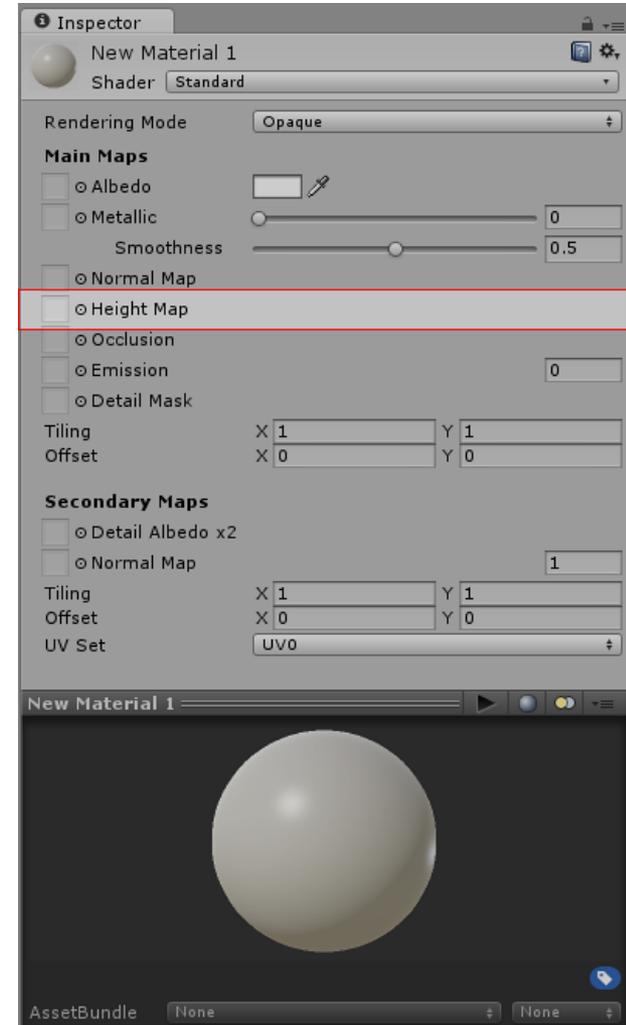
- **Resultado:**



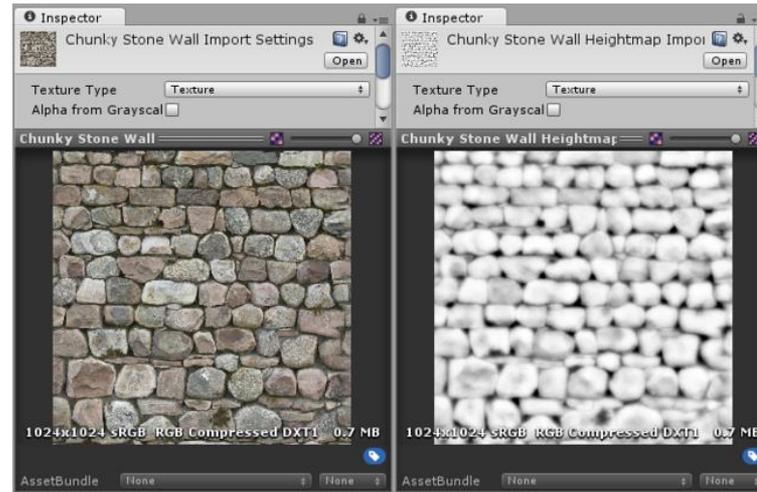
Com Normal Mapping (point light)

Standard Shader – Heightmap

- Height mapping (parallax mapping) é uma técnica semelhante ao normal mapping, porém é mais complexa (e demanda mais poder de processamento).
- Enquanto o normal mapping modifica a iluminação na superfície do objeto, o height mapping vai além e desloca a textura da superfície visível.



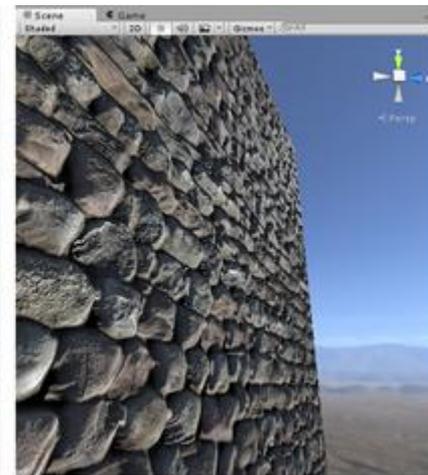
Standard Shader – Heightmap



Sem Bump Mapping



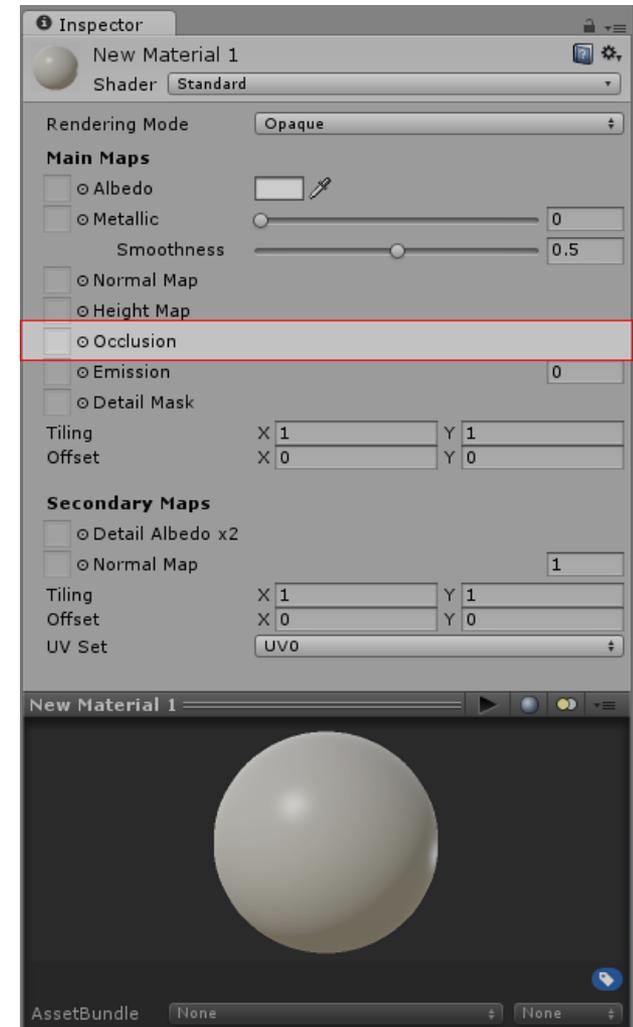
Normal Mapping



Height Mapping

Standard Shader – Occlusion Map

- Utilizado para determinar quais partes do objeto devem receber mais/menos luz indireta.



Standard Shader – Occlusion Map



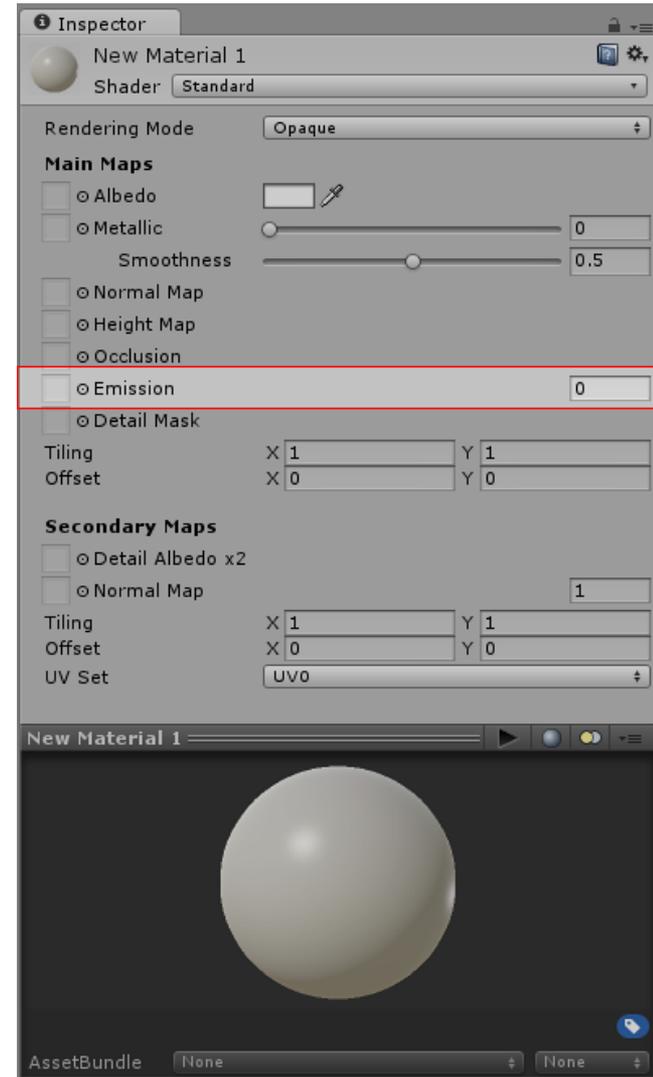
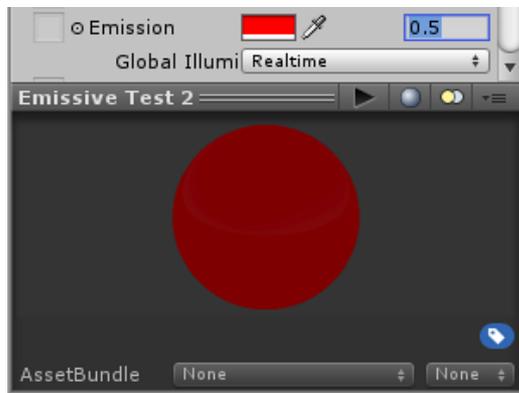
Sem Occlusion Mapping



Com Occlusion Mapping

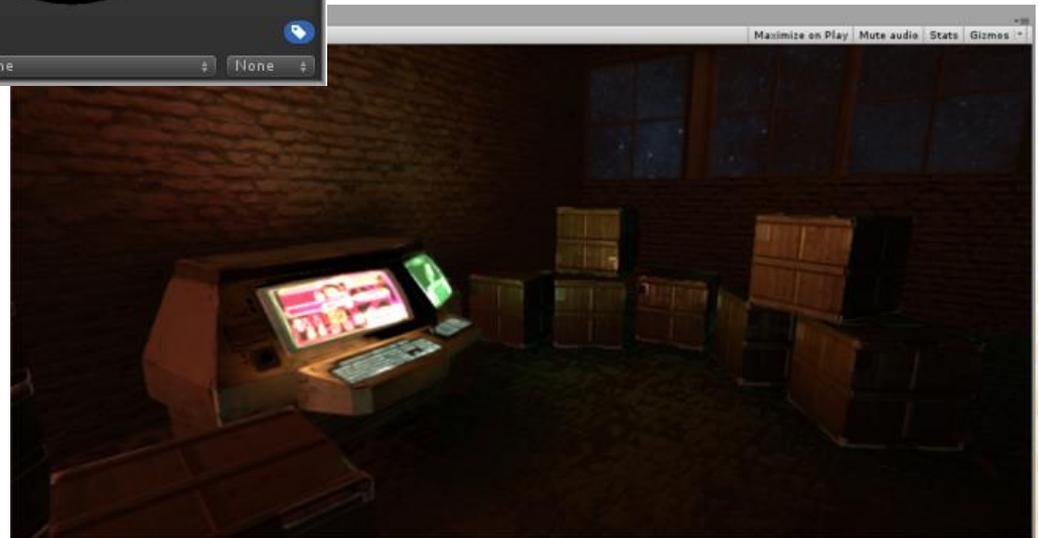
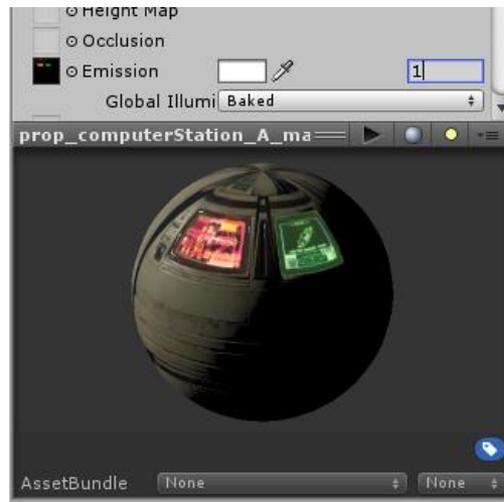
Standard Shader – Emission

- Controla a cor e intensidade da luz emitida pela superfície.



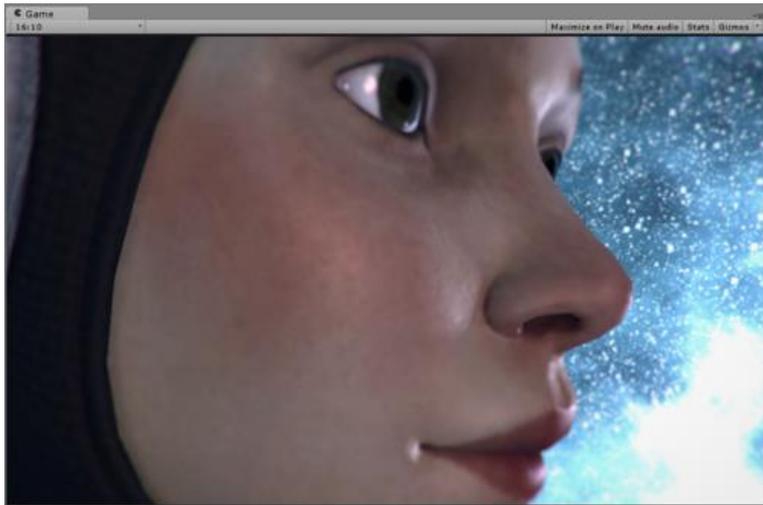
Standard Shader – Emission

- Controla a cor e intensidade da luz emitida pela superfície.



Standard Shader – Secondary Maps & Detail Mask

- Secondary maps permitem que um segundo conjunto de texturas seja aplicada sobre o primeiro conjunto.



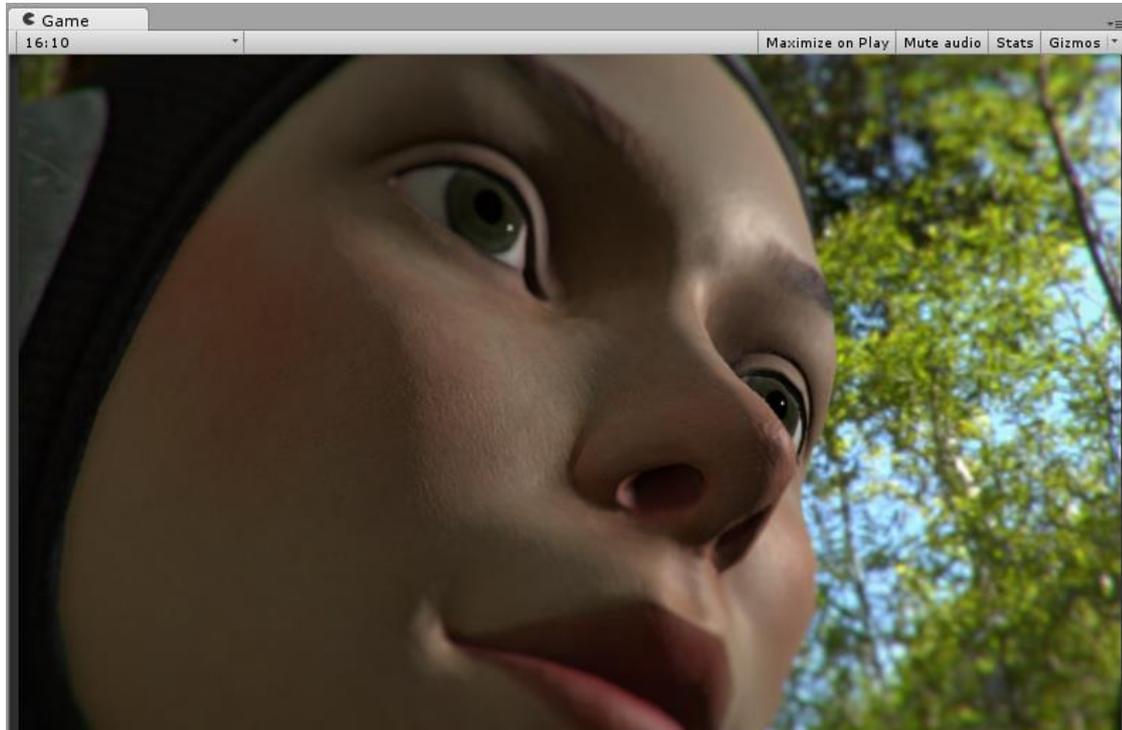
Sem Secondary Maps



Com Secondary Maps

Standard Shader – Secondary Maps & Detail Mask

- Detail Mask permite aplicar uma mascara sobre o Secondary Maps que determina em quais regiões o Secondary Maps será aplicado.



Exercício 05

- Continue a implementação da fortaleza criada nos exercícios anteriores adicionando a ela os seguintes elementos:
 - a) Altere as texturas utilizadas na fortaleza de acordo com as imagens ao lado.
 - b) Altere o Standard Shader de todos os materiais para o modo Specular.
 - c) Adicione as imagens Specular, Normal Map e Occlusion das texturas aos materiais.
 - d) Configure os demais parâmetros dos materiais para criar um ambiente semelhante ao mostrado nas imagens.
 - e) Altere a cor Emission das tochas.

