



Elementos do fotorrealismo

Realismo em tempo real...

Paradigmas

do Graphic Process

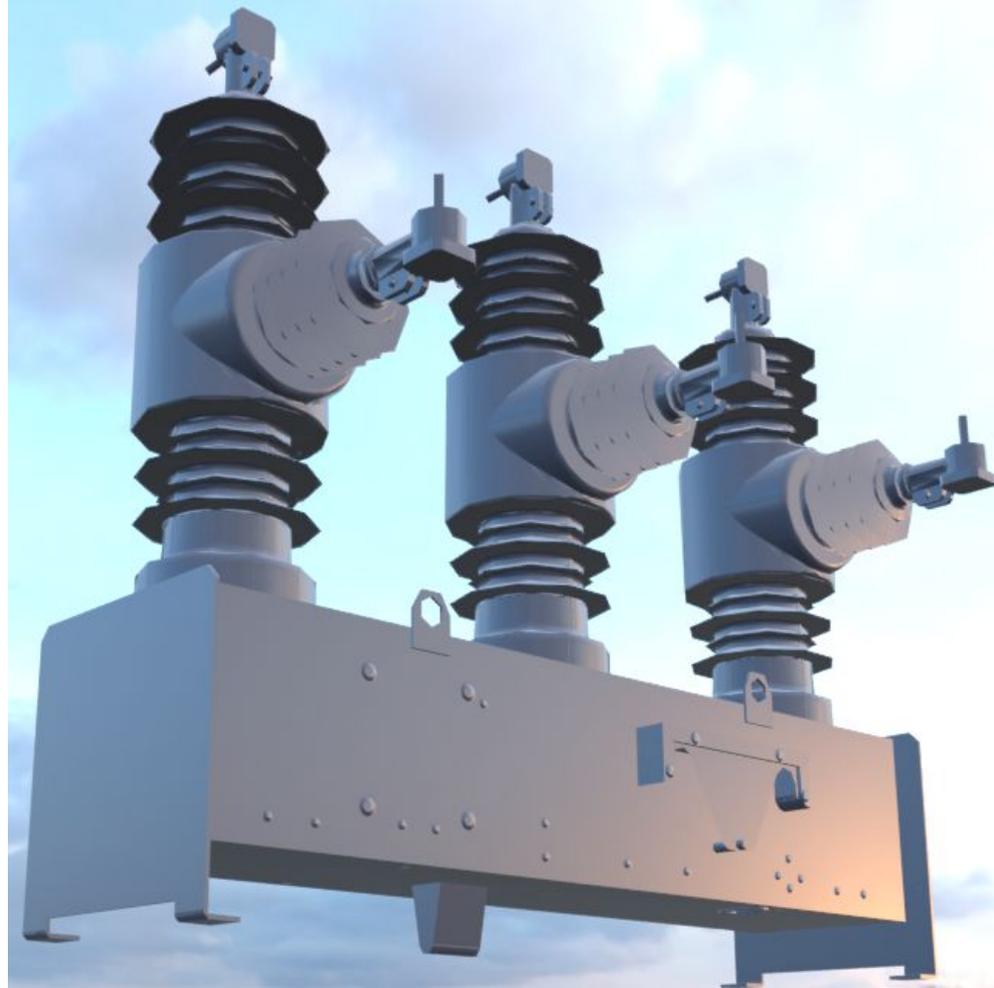
- Refino de geometria:
+ precisão visual, + polígonos
- Exemplo: 1M - 25 MB
- Simples no entanto a
capacidade ainda é limitada
para lidar com geometria.
Recomenda-se 3 M visíveis
(PCs) e 200 mil (mobile)



Paradigmas

do Graphic Process

- Mapeamento de texturas: qualidade similar com menos faces
- Neste exemplo: 6K tris (200 kB) + textura Normal Map (5 MB)
- Simular características com mapas 2D



Paradigmas: comparando



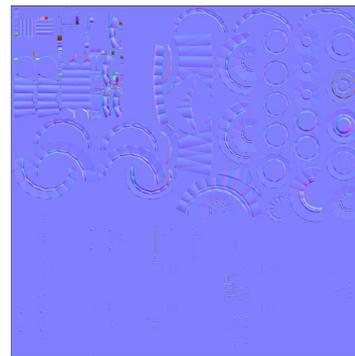
1 M polígonos, 25 MB



6k polígonos, 200 kB



+textura:
5 MB

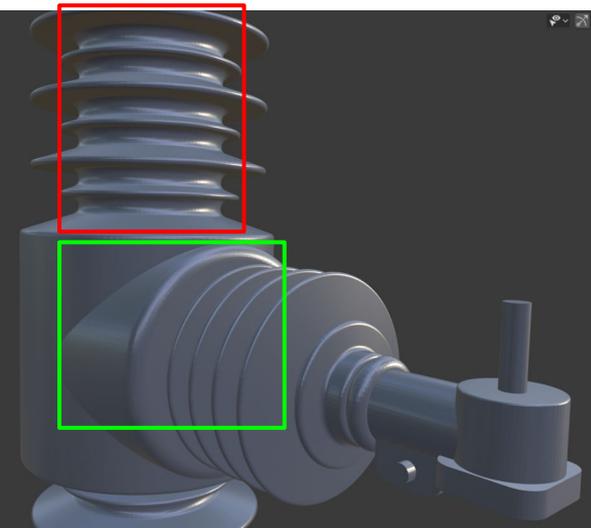


Paradigmas: comparando



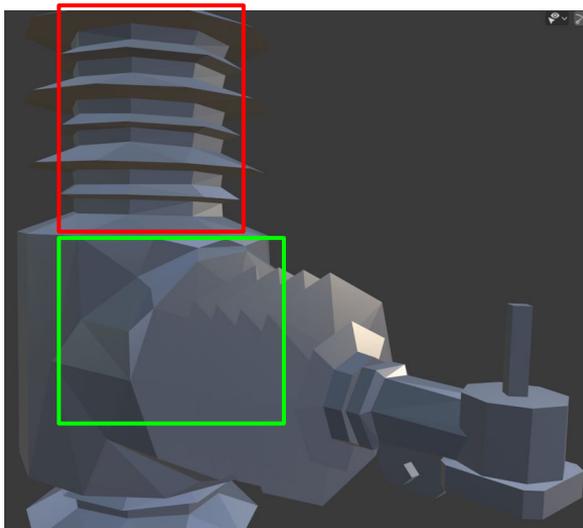
reflexões precisas

geometria suave



reflexões diferentes

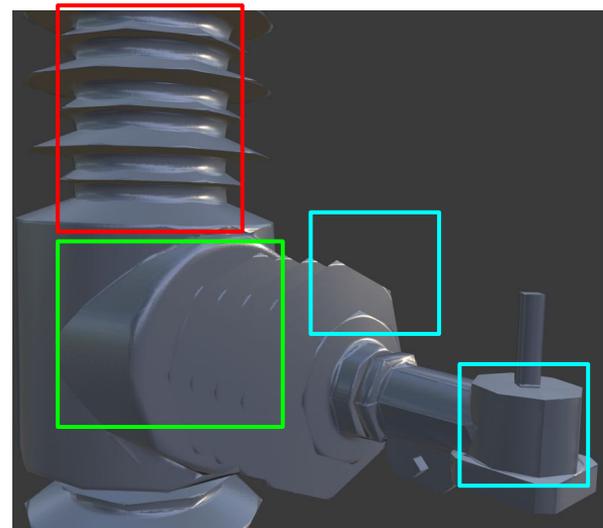
cantos vivos, geometria
facetada



reflexões muito similares

geometria muito suavizada

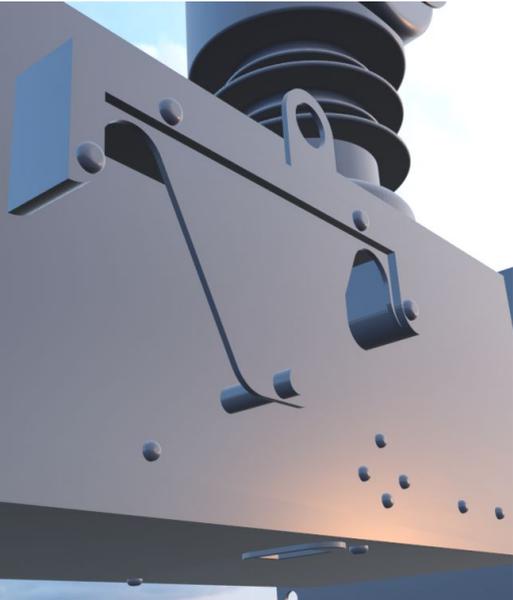
alguns cantos vivos visíveis



Paradigmas: comparando



detalhes precisos e suaves



ausência de detalhes



detalhes similares, mas menos protuberantes vistos de perto





Paradigmas

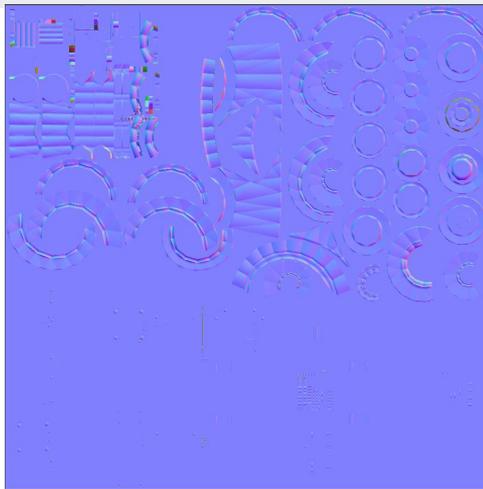
do Graphic Process

- Texturas - padrão da indústria
- Existem diversos outros mapas para reproduzir detalhes
- Mapas de normais: economia de memória e de geometria
- Em jogos existem muitos elementos; nosso exemplo já possuía 33% do máx. de triângulos recomendado para PC e 500% para mobile
- LOD: técnica para lidar com o detalhamento conforme distância
- Texturas são usadas a muito tempo; apenas recentemente foi determinado um padrão voltado ao fotorrealismo: o PBR (Physically Based Rendering)

PBR: exemplo



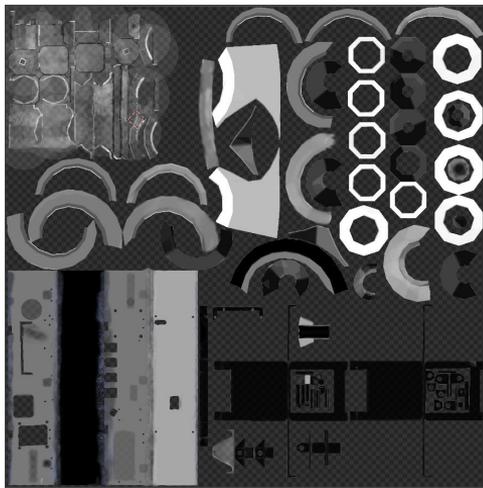
+



=



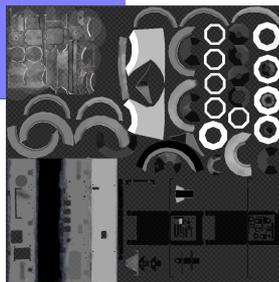
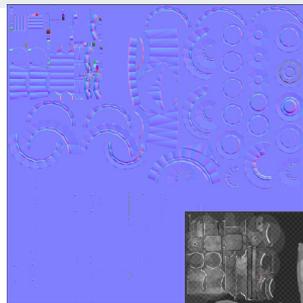
+ mapa 'Metálico': regiões 'metalizadas' (canal R) e rugosidade (canal Alpha)



PBR: exemplo



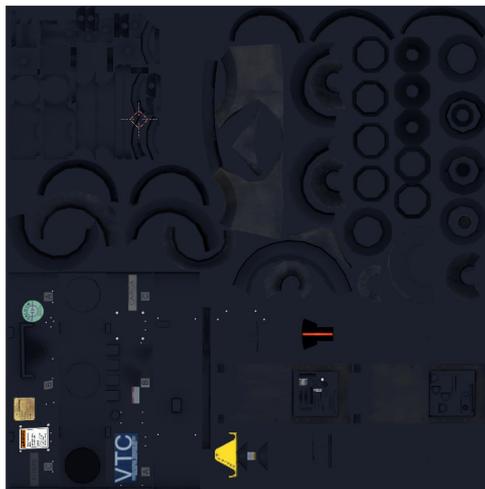
+



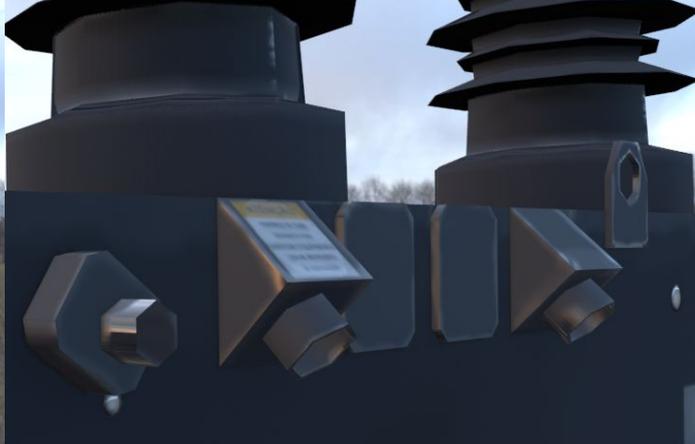
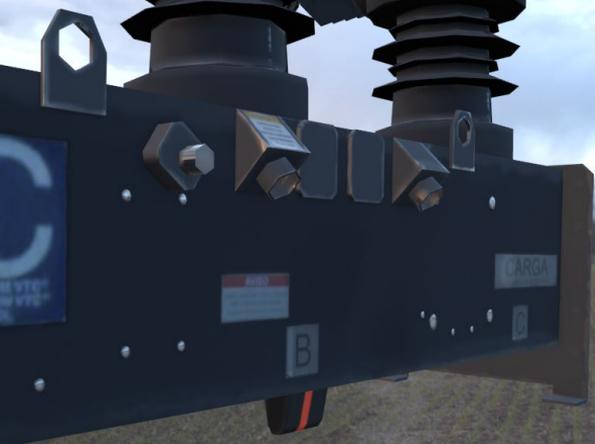
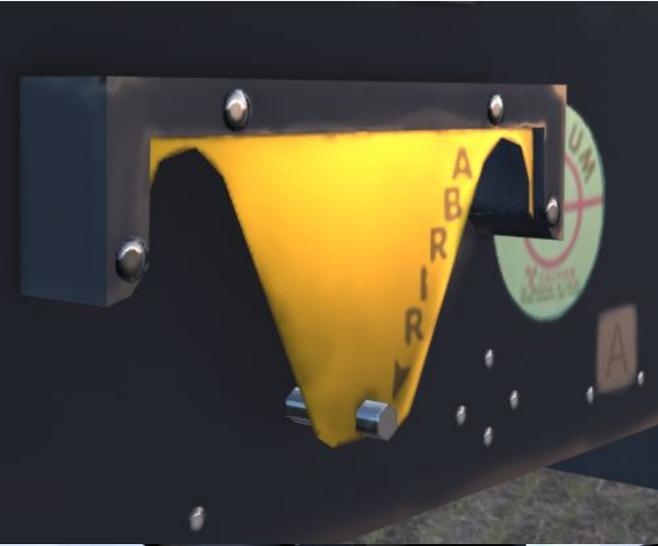
=



+ mapa 'Albedo': cores difusas do objeto (RGB)



PBR: detalhes



Texturas: reproduz geometria + cor + brilho

Exemplo: 10 MB (texturas) +
200 kB (6 K tris) X
25 MB (1 M tris)

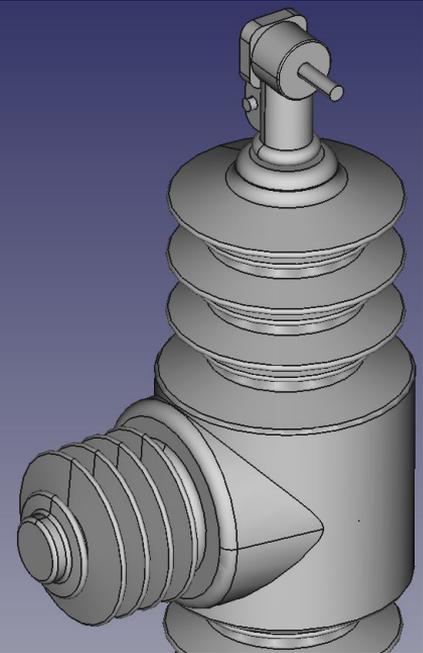


PBR: produção

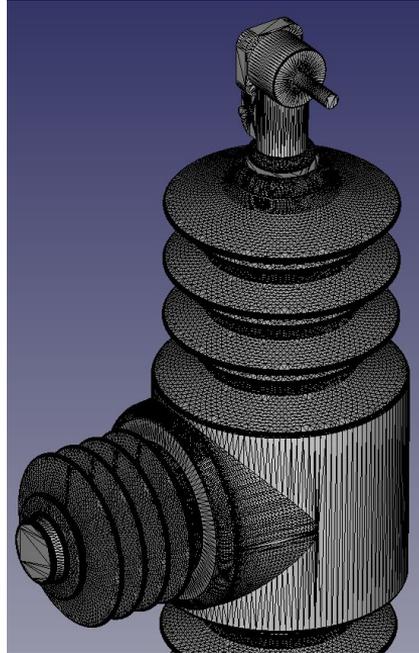
- Modelagem não muda: alta e baixa resoluções necessárias
- Conversão alta-baixa resolução exige intervenção manual
- Alta resolução: para produzir o mapa de normal por [‘baking’](#)
- Baking também usado para [embutir iluminação estática no modelo](#) - sombreamento detalhado. Pode ser embutido no Albedo, mas existem mapas específicos
- Albedo e Metálico: processos artísticos / semi automático, determina-se cores, metalização e rugosidade de cada região
- Pintura de mapas: por mapa / canal; alguns programas tem etapas automáticas e permitem pintar padrões em múltiplos mapas
- Blender: permite produção de conteúdo 3D PBR, mas criação de mapas é manual

PBR: exemplo de produção

CAD: produção do modelo



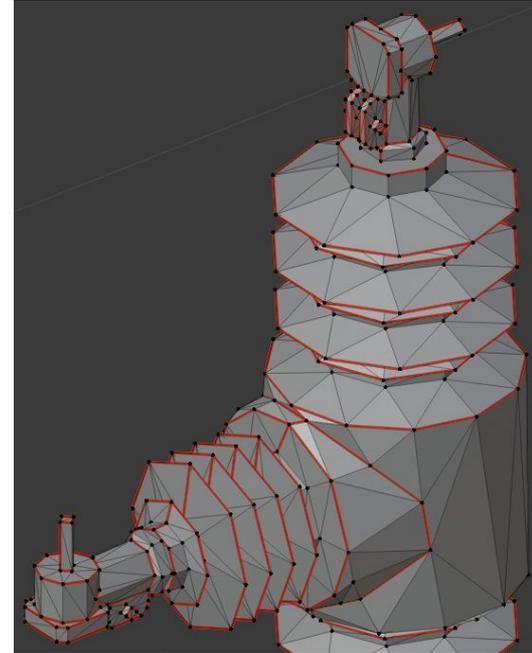
CAD: criação das malhas (automática)



Blender: simplificação da malha (automática)

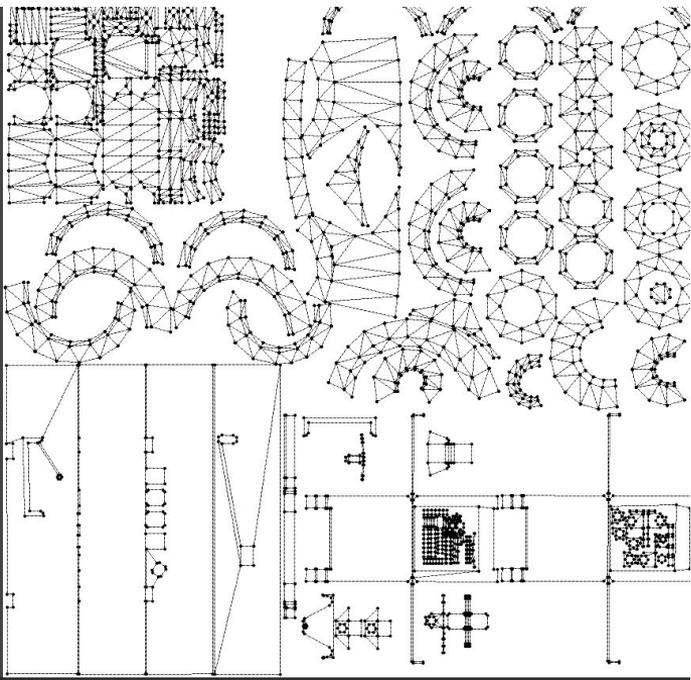


Blender: recorte da malha

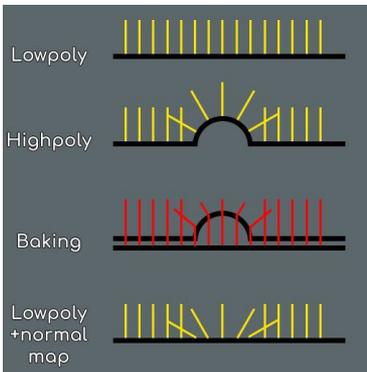
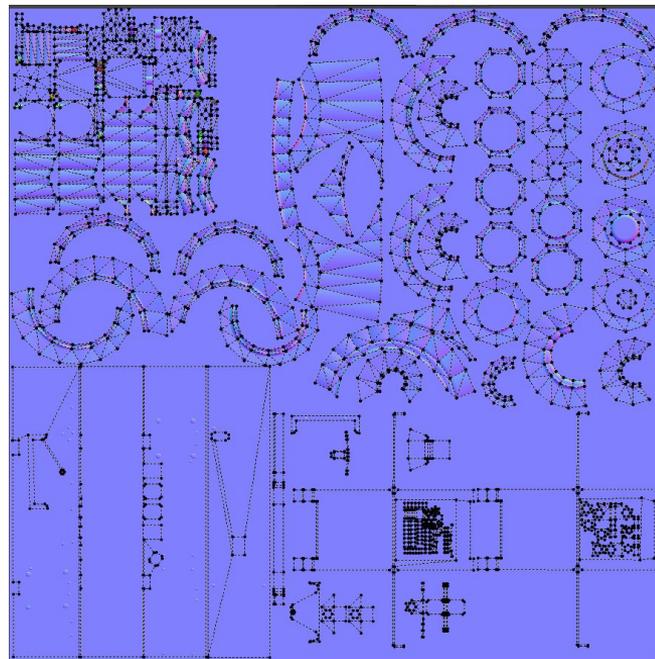


PBR: exemplo de produção

Blender: [desdobramento](#) de malha
(semi-automática)



Blender: baking de normal (automática)

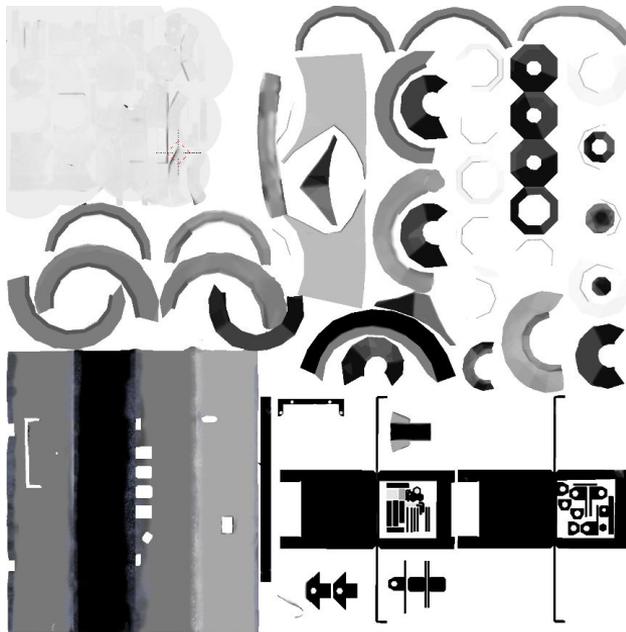


PBR: exemplo de produção

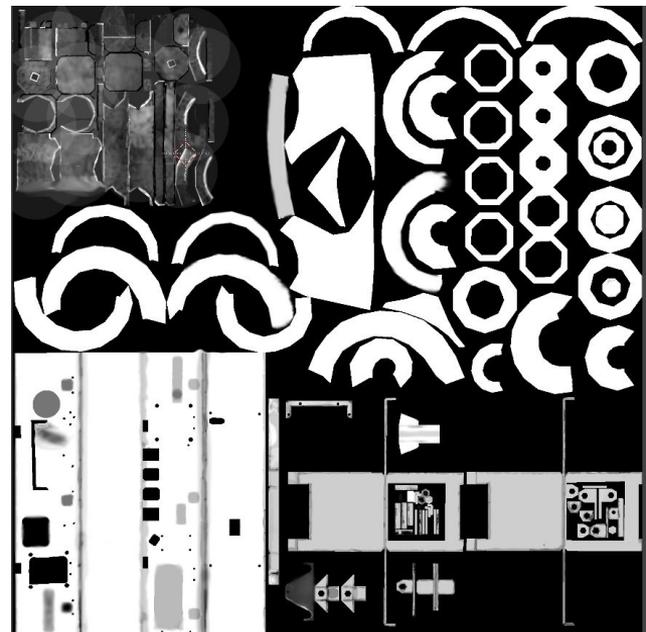
Blender: Albedo por pintura de texturas RGB



Blender: Metallic, pintura em dois canais



Canal R -
branco: metal / preto: não metal



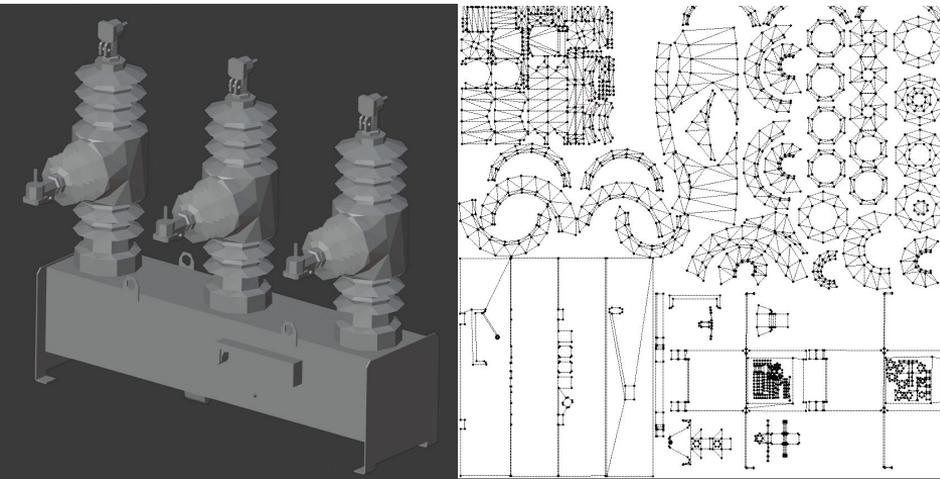
Canal Alpha -
branco: máximo rugoso / preto: liso

PBR: exemplo de produção

Blender: exportar modelo em FBX



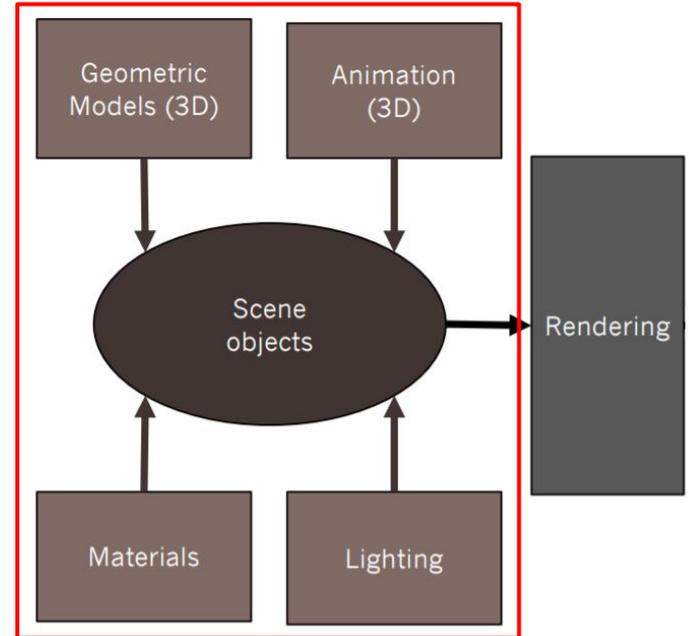
Unity: importar FBX, criar material e associar texturas



PBR: repercussões no 'baixo nível'

- CPU - redução de draw calls - depende de número de objetos e materiais

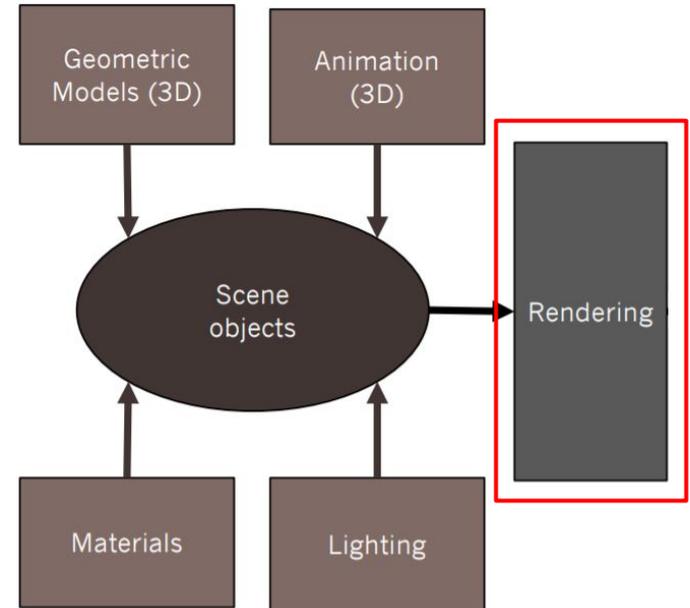
Detalhamento por texturas permite agrupar partes/objetos similares -> menos materiais



PBR: repercussões no 'baixo nível'

GPU - redução de operações em geométricas e otimização de memória

- Texturas reduzem polígonos (p/ mesma topologia visual) e incorporam iluminação estática -> menos operações no shader.
- Agrupar / comprimir texturas -> menor uso de memória de vídeo
- Culling / Level of Detail / mipmaps -> administram qualidade conforme distância dos objetos





PBR: ferramentas

- Modelagem CAD: FreeCad (livre), Fusion e SolidWorks (pagos)
- Modelagem artística: Blender (livre), 3DS Max e Maya (pagos)
- Texturizadores e bakers: ArmorPaint (free, Beta), Quixer Mixer (free p/ uso na Unreal); Substance, 3D Coat e ZBrush (pagos)
- Game Engines: Godot (livre); Unity, Unreal, CryEngine (livres para uso pessoal)



PBR: mercado e inovação

- Artistas: conhecer o workflow; processos de modelagem / texturização, tendo em vista otimizações na geometria e texturas; processos e detalhes de baking
- Programadores de engines: conhecer boas práticas; importar modelos, configurar materiais, alocar texturas e testar o desempenho gráfico. Conhecer mecanismos de otimização gráfica da engine
- Programadores de shaders: implementações e algoritmos de PBR, como lidar com texturas em shading. Estudar papers recentes
- Hardware: estado da arte, principais desafios, repercussões do PBR em arquiteturas. Tecnologias emergentes (raytracing em tempo real), estudar artigos recentes



OBRIGADO !