



Modelagem 2D

Aula 09 - Modelagem 3D – Parte 2



Material Didático do Instituto Metrópole Digital - IMD
Versão 5.0 - Todos os Direitos reservados



Apresentação

Olá, na aula de hoje daremos continuidade ao assunto de Modelagem 3D.



Objetivos

Entender as atuais funcionalidades da modelagem 3D;

Reconhecer quais as principais ferramentas de criação e edição de uma game em 3D.

1 - Modelagem 3D - Finalidades

Já parou para pensar como seriam o cinema e os games atuais sem a tecnologia da computação gráfica? Antes desse recurso existir e tomar a forma que conhecemos hoje, os efeitos práticos (termo referente a efeitos criados de forma totalmente artesanal) dominavam as produções cinematográficas mais ousadas e os pixels nos games apenas simulavam a experiência em duas dimensões. Mas a computação gráfica não apenas revolucionou esses dois ramos do entretenimento, como veremos mais adiante nesta aula.



Curiosidade

Pesquise anúncios publicitários antigos dos anos 1970 e também de anos anteriores e compare com anúncios atualizados, depois você pode compartilhar no fórum o que conseguiu visualizar de avanços tecnológicos nos recursos computacionais utilizados nos anúncios mais atuais em relação aos antigos.

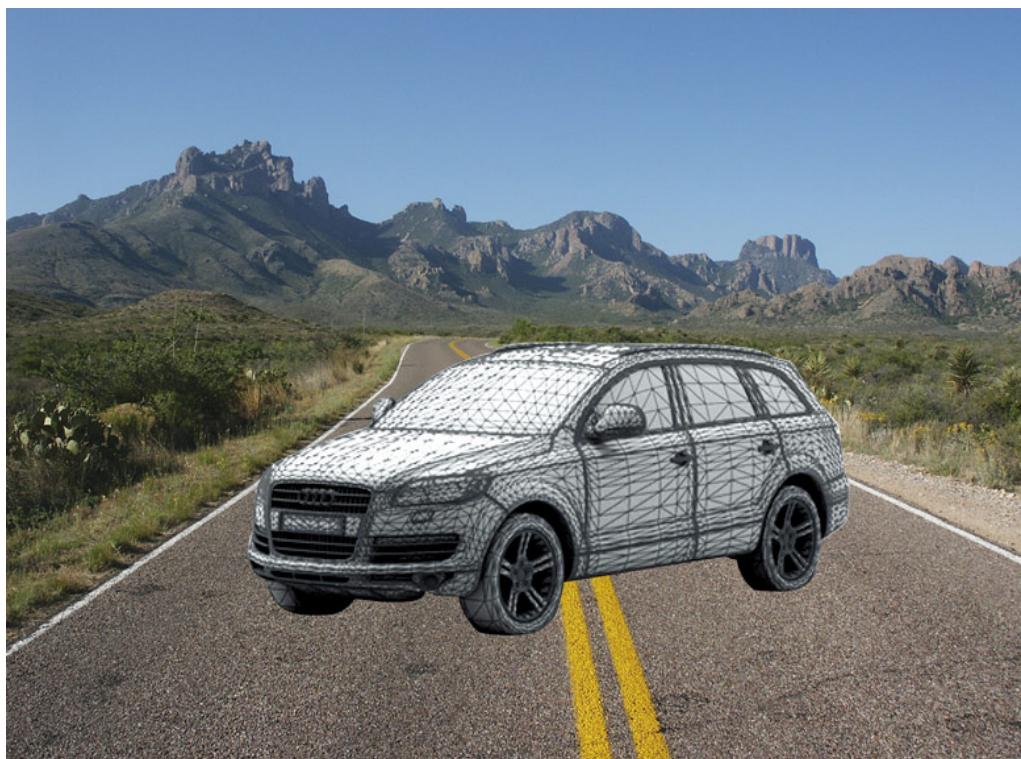
As ilustrações realistas, em que uma arte gráfica era feita com dúzias de canetas e pincéis diferentes, procuravam compensar os efeitos e detalhes não reproduzidos pela fotografia. A manipulação de imagens (entenda montagem de fotos e acréscimo de filtros) antigamente era realizada de maneira analógica, pouquíssimos profissionais no mundo conseguiam acertar os cortes nos negativos a fim de unir uma modelo *Pin Up* ao lado de uma garrafa de refrigerante sob um céu azul, com o mar de uma bela praia ao fundo. Assim, a ilustração sempre salvava a publicidade. Hoje em dia 99,99% das imagens publicitárias são fotos manipuladas com uso de programas de computador, muitos desses anúncios recorrem à modelagem 3D e você nem percebe!

Figura 01 - Imagine como seria esse anúncio...



Fonte: Anúncio Publicitário. Disponível em: <<http://laduarte.com/imaging.html>>. Acesso em: maio de 2017

Figura 02 - ... na década de 1960?



Fonte: Anúncio Publicitário. Disponível em: <<http://laduarte.com/imaging.html>>. Acesso em: maio de 2017

Analizando as duas figuras acima, temos uma noção do poder da arte gráfica atualmente. A primeira imagem engana muita gente, mas na verdade não há carro nenhum nessa estrada. O carro, feito em computação gráfica e renderizado em uma posição que respeite a perspectiva da estrada, passou por um processo de pós-produção em sua renderização e montagem com a foto da estrada. Na segunda imagem vemos o modelo 3D bruto, no início do processo.

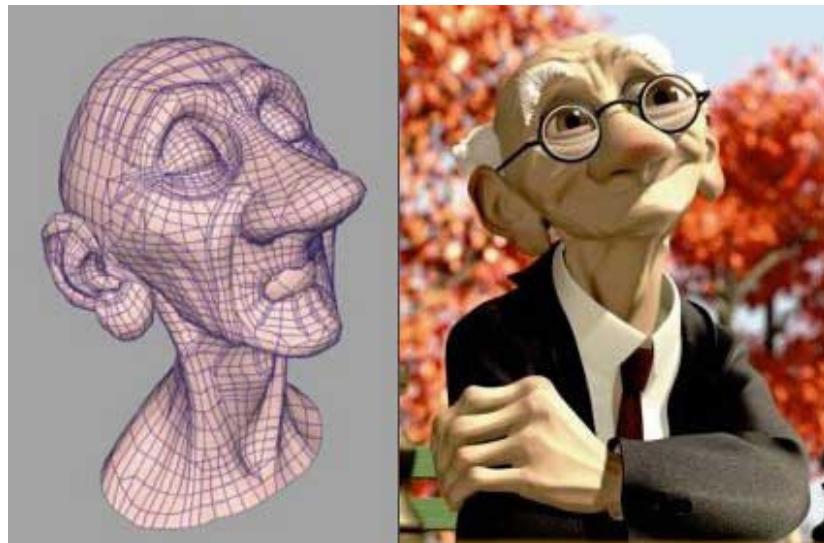
Além da modelagem 3D usada na composição de imagens para revistas e pôsteres, em que os processos de render são otimizados para imagens estáticas, existem as animações 3D, as quais exigem um acabamento mais rebuscado dos modelos.

Figura 03 - Algumas produções como Game Of Thrones utilizam a modelagem 3D de maneira dramática e convincente para ajudar a contar sua história.



Fonte: Heres-how-it-looked-after-the-effects. Disponível em: <<http://www.pixomondo.com/business-insider-what-the-game-of-thrones-dragons-look-like-before-visual-effects/>>. Acesso em: maio de 2017

Figura 04 - Outras produções em Modelagem 3D.



Fonte: Conhecendo a Computação Gráfica. Disponível em: <<https://leticiamagalhaesblog.wordpress.com/2016/02/21/conhecendo-a-computacao-grafica/>>. Acesso em: maio de 2017

Algumas produções cinematográficas que utilizam animação 3D chegam a renderizar um filme com mais de 5 mil computadores! Isso acontece, pois, a animação 3D para cinema não é em tempo real, assim é possível encher uma cena com centenas de luzes e dezenas de filtros de ajustes e pós-produção carregada, para converter tudo em arquivos de vídeo no final do processo.

Mas o que é a renderização em tempo real? Ficou curioso para saber como isso acontece?

Não saia daí! Agora chegou a hora de conhecermos um pouco mais sobre os processos de renderização e as suas aplicações!

2 - Modelagem 3D – Real Time Render

Renderização em tempo real é o mesmo processo de render usado nos games. Nele os modelos são otimizados para haver interação direta de diversas formas. Um personagem 3D de um game é projetado para receber diversas luzes em tempo real,

respeitar os relevos do mapa ao se mover, reagir com a interação da física... resumindo, um personagem de game deve responder de forma imediata a qualquer tipo de evento que seja submetido dentro das regras que seu ambiente fornece.

Já se perguntou por que em alguns games durante uma Cutscene os gráficos são semelhantes aos filmes de animação que vemos no cinema e, quando o jogo inicia, os gráficos têm uma certa “queda” de qualidade? Esse efeito conhecido como Downgrade ocorre quando as cutscenes de muitos games não são renderizadas em tempo real, diferente dos jogos no momento de Gameplay.

Figura 05 - Este é um Orc do Jogo World of Warcraft, na versão da cutscene.



Fonte: World of Warcraft: Warlords of Draenor Cinematic. Disponível em: <<https://www.youtube.com/watch?v=FkS2EVmdt9w>>. Acesso em: maio de 2017

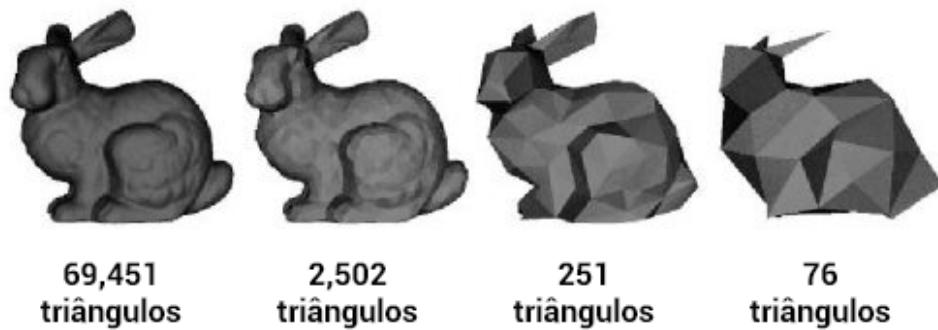
Figura 06 - Este é um Orc do Jogo World of Warcraft, na versão do Gameplay



Fonte: Site tom's HARDWARE. Disponível em: <<http://www.tomshardware.com/news/blizzard-world-warcraft-robin-williams,27466.html>>. Acesso em: maio de 2017

Dada essa característica, existem algumas preocupações quando se geram os modelos do jogo e, por isso, normalmente existem várias versões do mesmo modelo, com uma quantidade de polígonos diferenciada. Por exemplo, um dos aspectos que se trabalha bastante nos jogos é o nível de detalhe - **Level of Detail (LOD)**. O que é isso?

Figura 07 - Exemplo de modelo 3D com diferentes níveis de detalhe



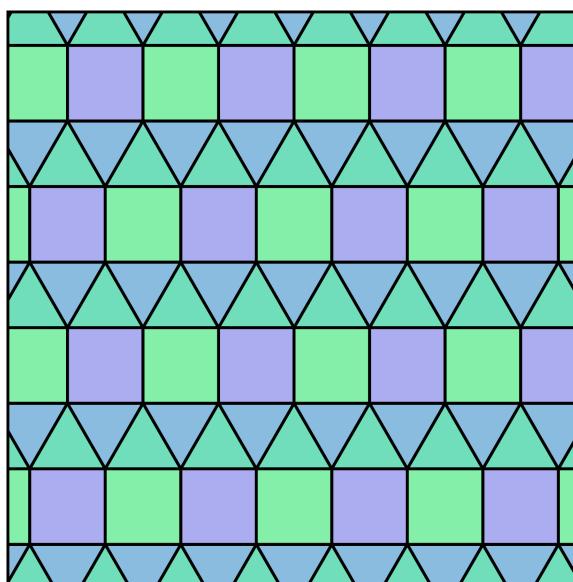
Fonte: Disponível em: <<http://polygon-reducer.pc-guru.cz/blog/polygon-reducer-lod-1.jpg>>. Acesso em: 23 maio 2018

O nível de detalhe é uma técnica para agilizar o desempenho de renderização do jogo. Objetos próximos da câmera e da visão do personagem (e jogador) são renderizados com modelos mais detalhados, contendo mais polígonos (logo, mais custosos). A medida que um objeto fica mais distante da câmera, substitui-se o modelo por uma versão com menos polígonos, com o objetivo de economizar processamento. Afinal, estando longe, o jogador não vai ver direito mesmo! Essa técnica deve ser bem utilizada, para evitar aquele velho problema de os objetos aparecerem do nada dentro da visão do jogador. Quem nunca bateu em uma árvore ninja, que aparece instantaneamente na sua frente, não é mesmo?

A partir do [Directx11](#) (Conjunto de bibliotecas do Windows que contém operações para a parte gráfica.) foi possível fazer alguns desses recursos de forma dinâmica. Imagina que, dependendo de como você for tratar o LOD, você tenha de criar uns 4 modelos do mesmo objeto, apenas com uma quantidade de polígonos diferentes. O DirectX11 permite que você faça isso dinamicamente a partir de um único modelo, graças a técnica de **Tesselação**.

A tesselação nada mais é do que a divisão de uma forma geométrica mais complexa em agrupamentos de formas geométricas mais simples. Por exemplo, eu posso ter um quadrado que é formado por um quadrado ou por dois triângulos. Dessa forma, podemos construir um modelo com um número menor de polígonos e, por meio da tesselação, gerar de forma automática uma malha mais detalhada e suave. Isso economiza em muito o trabalho do pessoal da modelagem!

Figura 08 - Exemplo de quadrado formado com várias formas menores

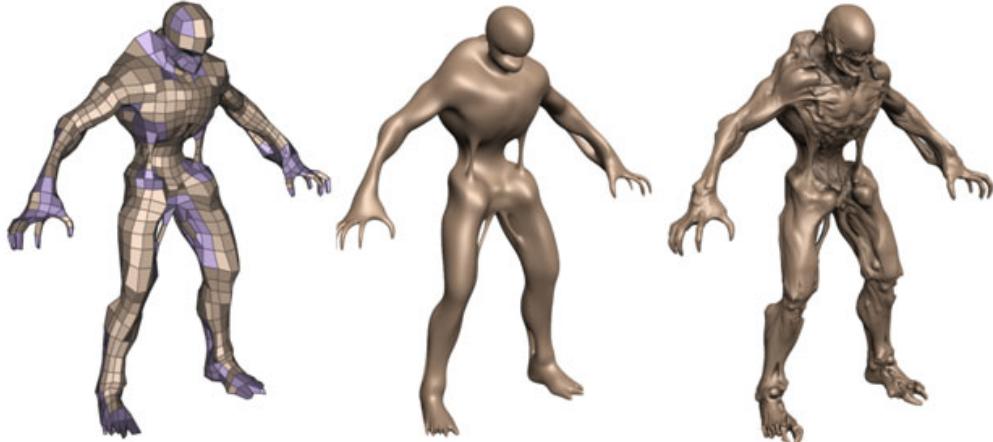


Fonte: ACADEMIC. **Mosaico regular.** Disponível em:

<<http://www.esacademic.com/dic.nsf/eswiki/1448871>>. Acesso em: 29 maio 2018

Com a tesselação conseguimos um novo nível de flexibilidade no trabalho de modelos 3D em jogos, e esse é talvez o grande diferencial dos gráficos dessa nova geração!

Figura 09 - Exemplo de tesselação em modelo 3D



Fonte: Disponível em: <http://www.nvidia.com.br/docs/IO/92640/coarse_model.jpg>. Acesso em: 23 maio 2018

3 - Operações sobre Modelos 3D

Existe um conjunto de operações que podem ser feitas sobre as malhas e objetos criados. A primeira classe de operações são as transformações, que caracterizam as alterações mais simples sobre o objeto. Nelas ocorrem mudanças com relação ao posicionamento dele no espaço, mas sem deformar o seu formato básico padrão:

- **Translação:** mover um objeto, mudá-lo de posição. Apenas altera os valores das coordenadas x,y e z;
- **Rotação:** faz com que o objeto rotacione sobre um ou mais eixos, alterando as coordenadas dos seus vértices;
- **Mudança de escala:** o objeto aumenta ou diminui de tamanho, mantendo ainda a relação estrutural com o formato original;
- **Espelhamento:** vira o objeto verticalmente/horizontalmente (ou cria uma cópia espelhada);
- **Alinhamento:** posiciona os elementos da cena através de um referencial, que podem ser coordenadas específicas do mundo virtual da modelagem, ou a posição relativa à outros objetos.

Figura 10 - Exemplos de operações sobre o modelo 3D: translação, rotação, escala, espelhamento e alinhamento.

Figura 10a - Translação

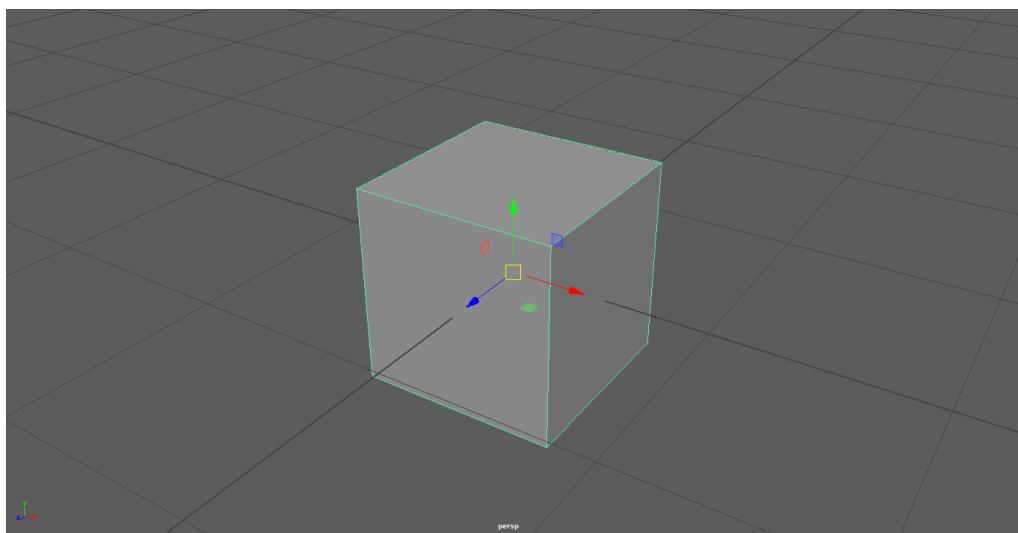


Figura 10b - Rotação

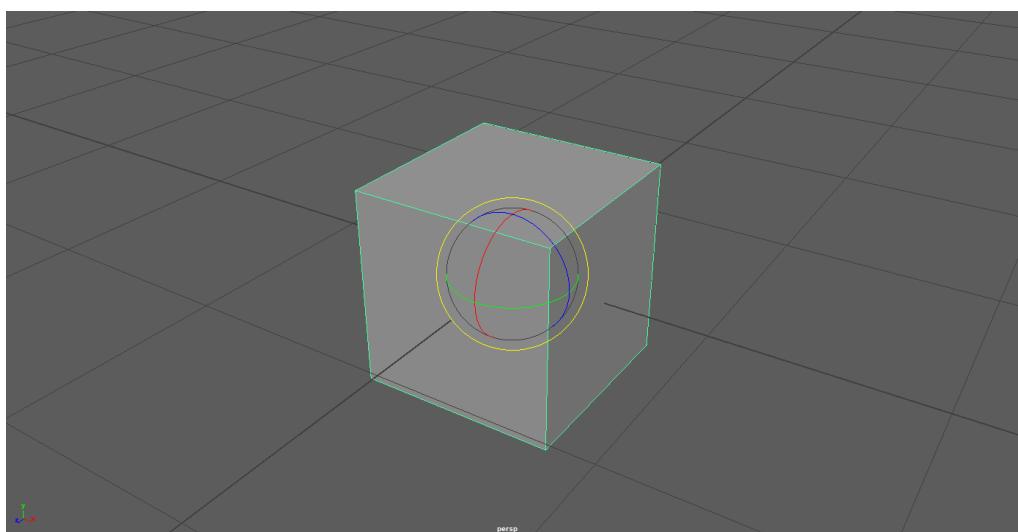


Figura 10c - Mudança de escala

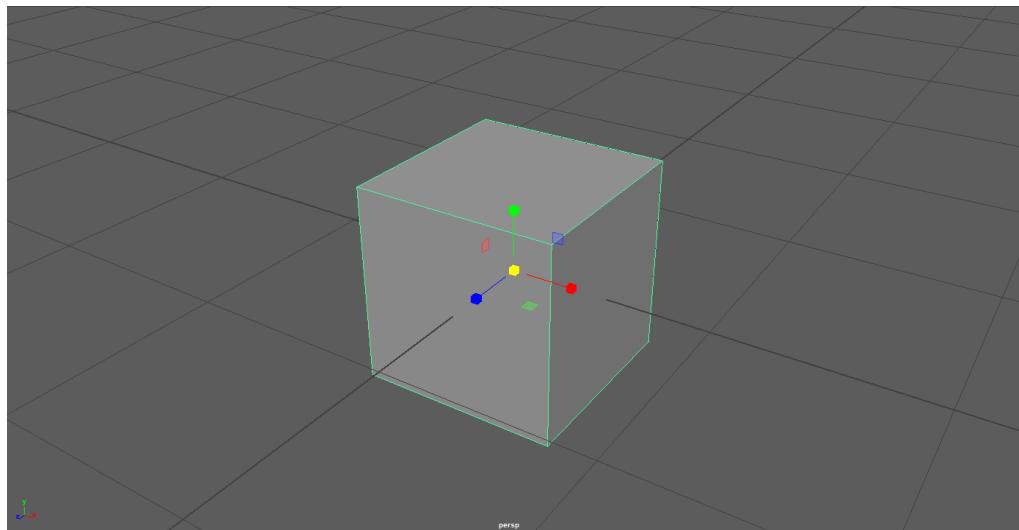


Figura 10d - Espelhamento

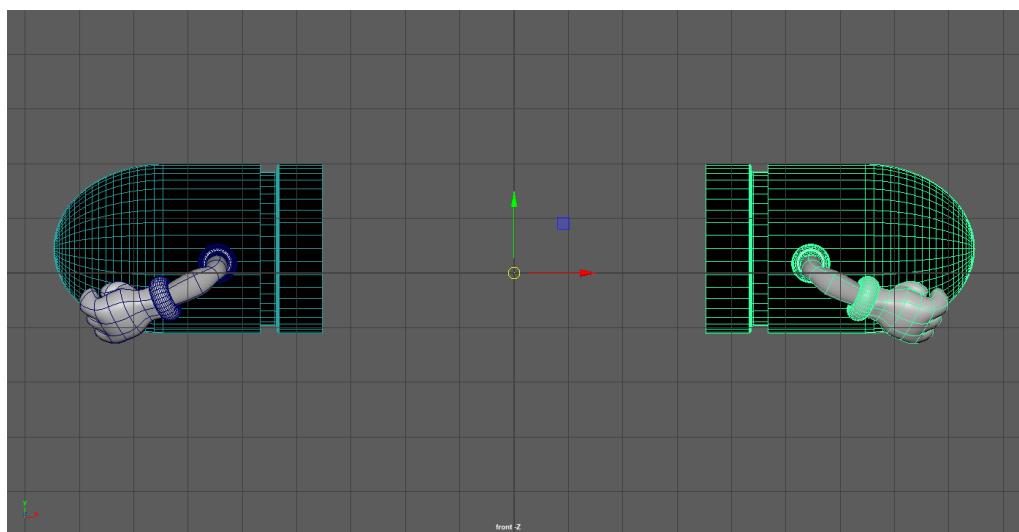
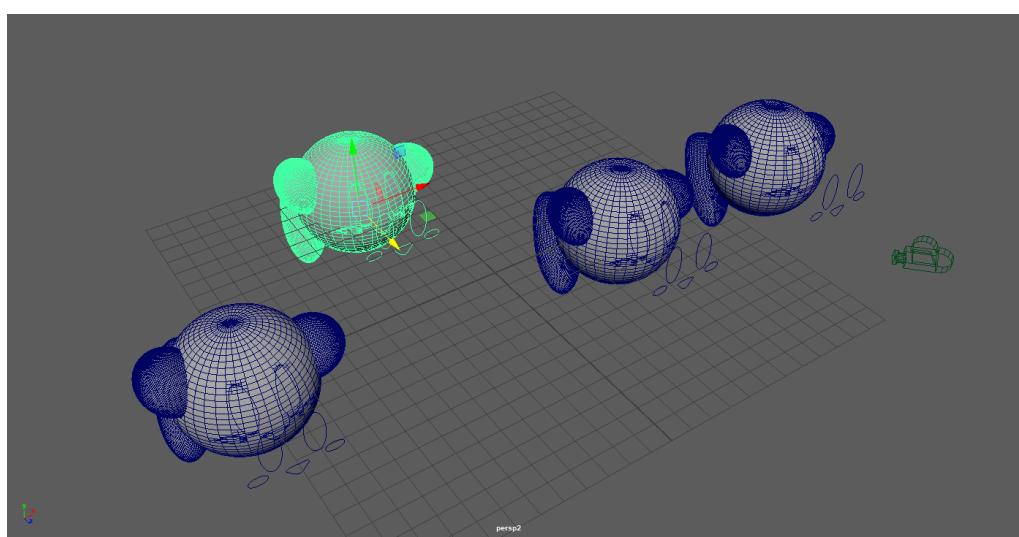


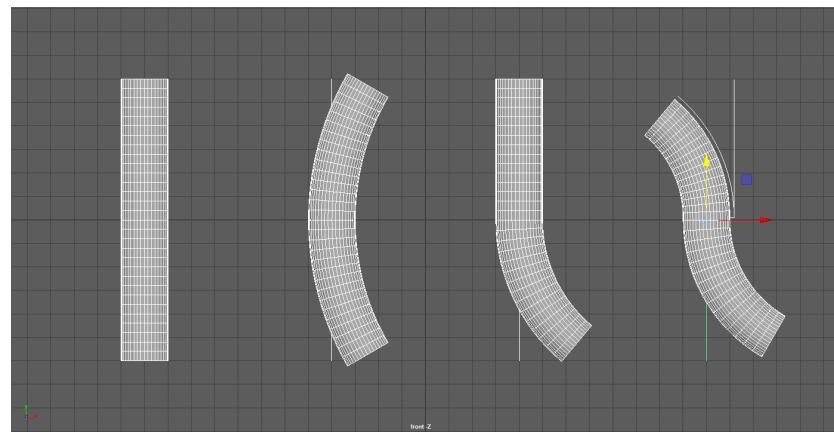
Figura 10e - Alinhamento



Outro conjunto comum de operações são as deformações. Essas operações causam uma alteração na estrutura da [mesh](#) (O mesh é a malha de polígonos que forma o modelo 3D, ou seja, cada pontinho que faz parte do modelo.), e por isso devem ser bem planejadas.

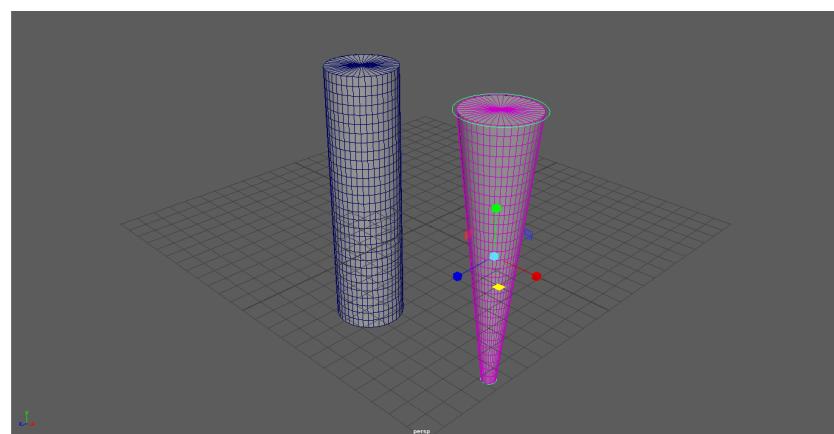
- **Bend (Dobra):** distorce o objeto em função de um dos eixos. Normalmente se segmenta o objeto em várias partes, para que a dobra fique com aspecto mais natural.

Figura 11 - Exemplo da operação Bend



- **Taper:** esse eu não encontrei uma boa tradução. Consiste em expandir ou comprimir um objeto em relação a um ponto de pivô ou controle (normalmente o que direciona a rotação). Essa operação deforma apenas a extremidade desejada.

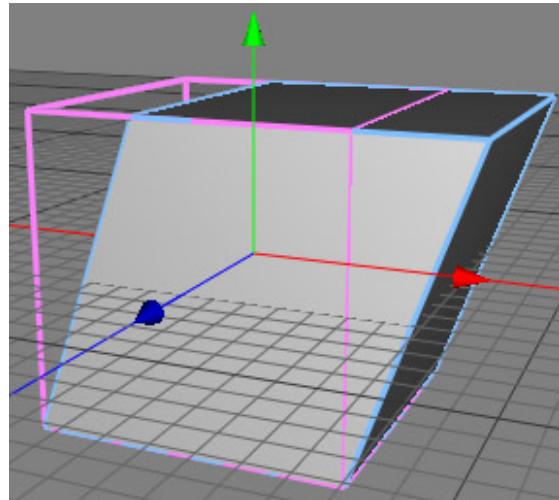
Figura 12 - Exemplo da operação Taper



- **Skew:** a partir de um eixo, é aplicada uma alteração em um sentido específico em uma das extremidades, sendo aplicada a alteração

oposta na outra extremidade. É como se você pegasse uma caixa e puxasse a parte de cima para a esquerda e a de baixo para a direita.

Figura 13 - Exemplo da operação Skew

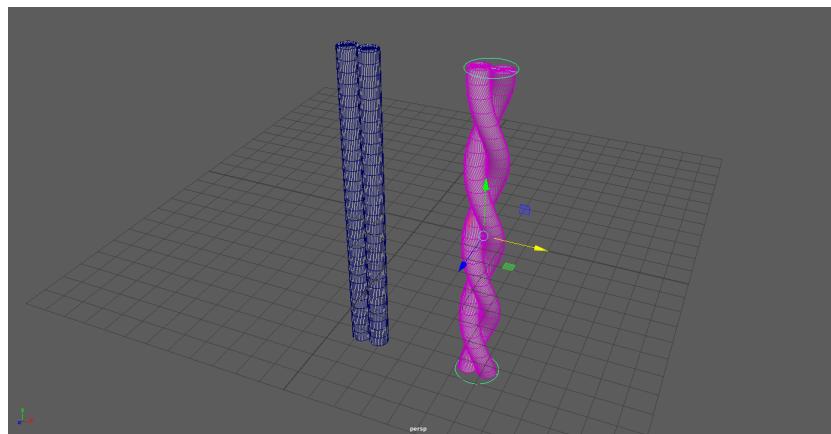


Fonte: Disponível em:

<https://tgjp.github.io/Cheetah3D_jp/Cheetah3Dmanual/Objects/ModifierObjects/Shear/Shear2.jpg> Acesso em: 23 maio 2018

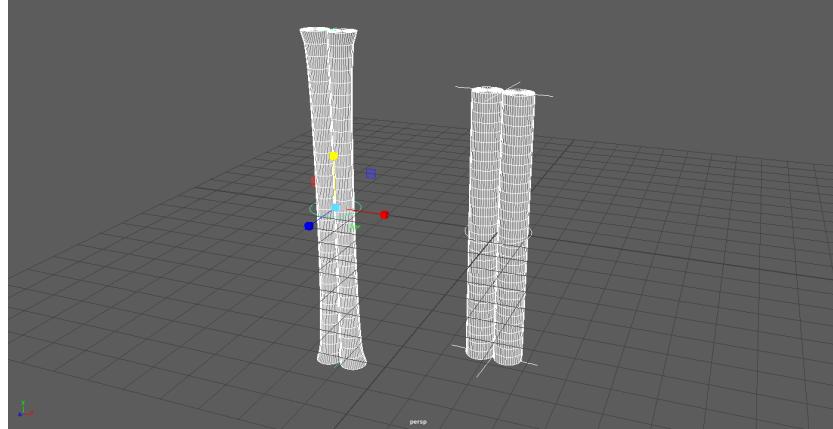
- **Twist (Torção):** contorce um objeto ao longo de um eixo. Como se fosse fazer um espiral!

Figura 14 - Exemplo da operação Twist



- **Squash e Stretch:** similar ao que vimos em animação, trata a deformação do objeto em dois sentidos de forma inversa, aumentando na horizontal quando diminui na vertical e vice-versa. Imagine como se fosse uma bola mole cheia de água, ou uma massinha de puxa puxa.

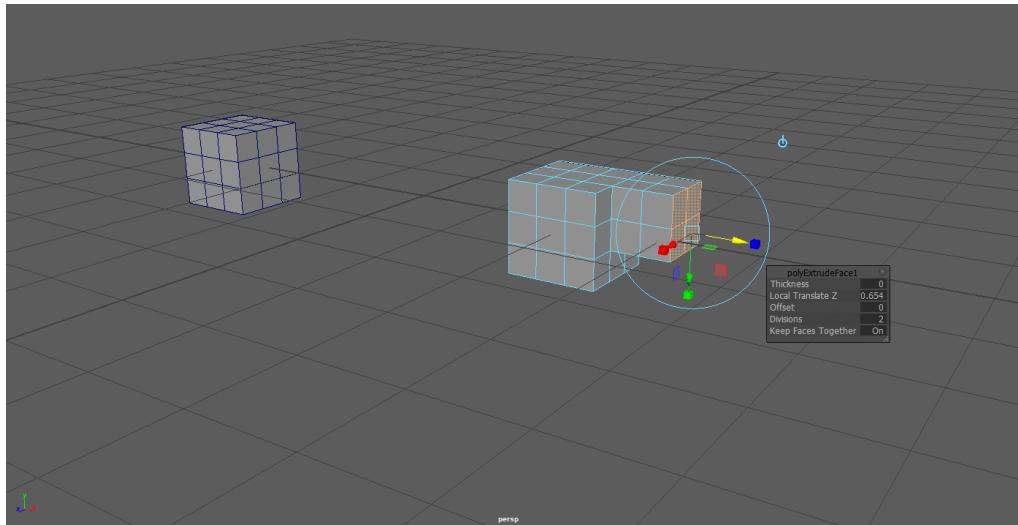
Figura 15 - Exemplo da operação Squash and Stretch



Existem outras operações que permitem gerar os modelos a partir de imagens 2D ou de *splines*. E elas tem nomes muito legais, então você pode encher a boca e sair falando na rua para se amostrar!

A primeira delas é a **extrusão**. Colocando em miúdos, essa operação consiste em pegar uma forma 2D e acrescentar volume a ela. Imagine que você tem um triângulo 2D. Extruir o triângulo seria adicionar uma altura a ele, de forma que ele fique 3D! Um pouco mais bonito: extrusão refere-se à alteração de volume em uma das faces do modelo, podendo ser a adição ou redução do volume.

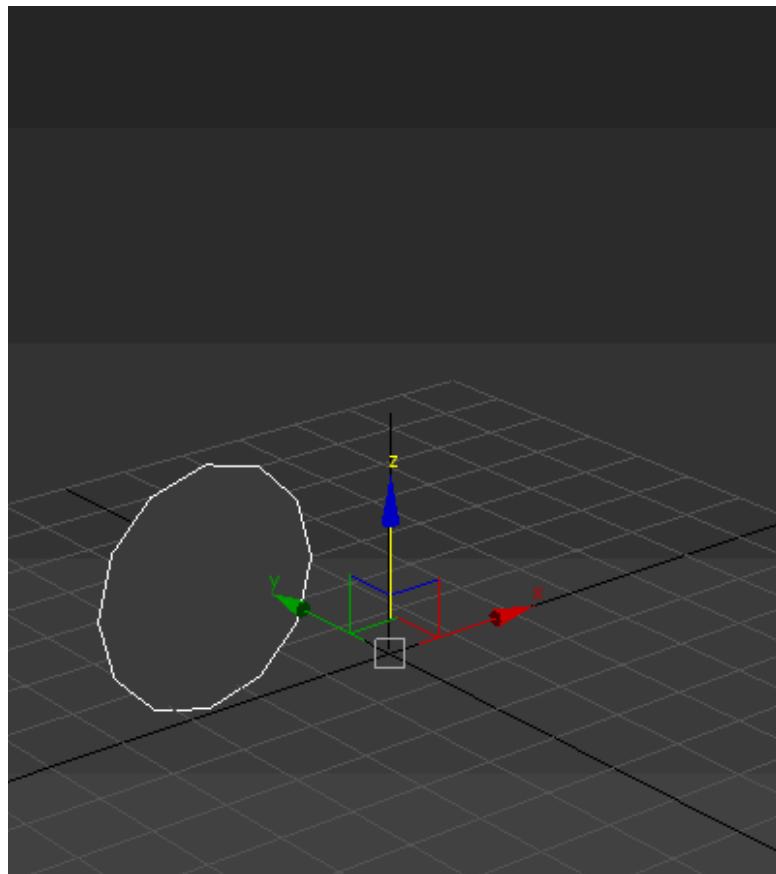
Figura 16 - Exemplo de extrusão de faces



A extrusão é uma operação básica, e que é utilizada por outras operações mais complexas. Por exemplo, existe um conjunto de tipos de operações em que o usuário determina apenas um caminho ou sentido de “movimento” para uma imagem 2D, e a ferramenta gera, através de várias extrusões, um objeto bastante complexo!

Uma dela é a operação de **Lathing**, ou torneamento. Você pega uma imagem 2D e diz que ela vai girar 360° em um eixo específico. A ferramenta então vai realizando extrusões de face aos poucos, de forma que gera um modelo preenchido no sentido da rotação realizada. Esse giro pode ser completo e gerar um modelo fechadinho, ou pode ser feito parcialmente e gerar um modelo aberto. O que você precisar!

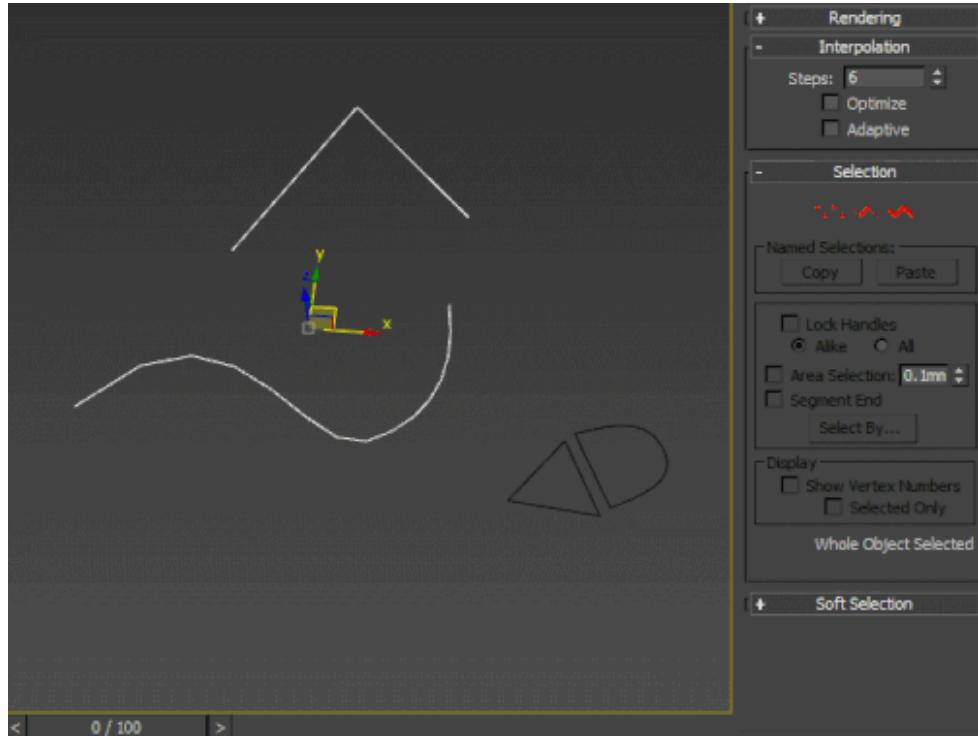
Figura 17 - Exemplo da operação Lathing



Fonte: Disponível em: <<https://helpx.adobe.com/content/dam/help/en/fuse/visual-dictionary/Lathe.gif>> Acesso em: 23 maio 2018

Outra operação bem legal para gerar um modelo 3D a partir de uma imagem 2D é a de **Sweeping**, ou varredura. Nela você desenha uma forma 2D e traça um caminho perpendicular a partir da forma até um ponto que você desejar. Esse caminho pode ser reto ou curvo. Quando a operação for executada, as extrusões vão formar um modelo preenchendo esse caminho com a mesma forma geométrica da imagem inicial, seguindo a orientação determinada pelo modelador. É como se você fizesse uma extrusão supercomplexa!

Figura 18 - Exemplo de Sweeping



Fonte: Disponível em:

<<https://i.pinimg.com/originals/09/2f/17/092f17747343ffc4a5a52dca96c97654.gif>> Acesso em: 23 maio 2018

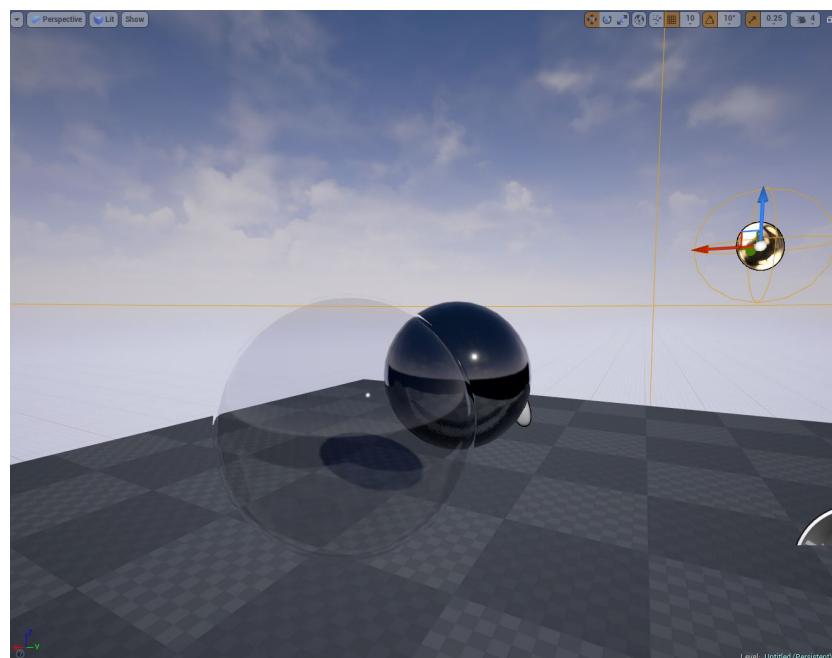
4 - Texturas

Quando nós fazemos um modelo 3D, estamos definindo basicamente a forma do objeto. Mas como sabemos, as coisas no mundo possuem vários detalhes, que dão aspectos únicos e interessantes aos objetos, e nos ajudam a distingui-los na natureza. A textura serve justamente para nos ajudar a definir detalhes do aspecto visual do objeto, como:

- **Cor:** isso aqui você já tá até abusado de ouvir falar!
- **Aspereza ou Rugosidade:** a superfície de um objeto dificilmente é tão lisinha quanto a minha careca. Elas parecem mais com o meu rosto: cheio de pelo encravado, espinhas, cicatriz de briga de faca... Quando falamos sobre rugosidade de uma superfície, estamos falando sobre o relevo dela, as pequenas variações de altura existentes, que fazem uma diferença enorme quando falamos de iluminação! Já pensou se você tivesse de desenhar uma malha de polígonos com esse nível de detalhe? Estaríamos nos jogos 2D ainda...

- **Translucidez:** o detalhe aqui é o quanto a superfície é um fantasma... eh... o pessoal dos materiais ligou e pediu para eu mudar essa explicação para uma correta. A translucidez de um objeto tem a ver com o quanto ele é transparente ou não, o quanto você consegue ver através dele. Não sei qual foi o problema com a explicação do fantasma... Um objeto translúcido implica que parte da luz que o ilumina atravessa, com um comprimento de onda filtrado e influencia também na iluminação dos objetos "por trás" dele.

Figura 19 - Uma bola translúcida e uma bola sólida.

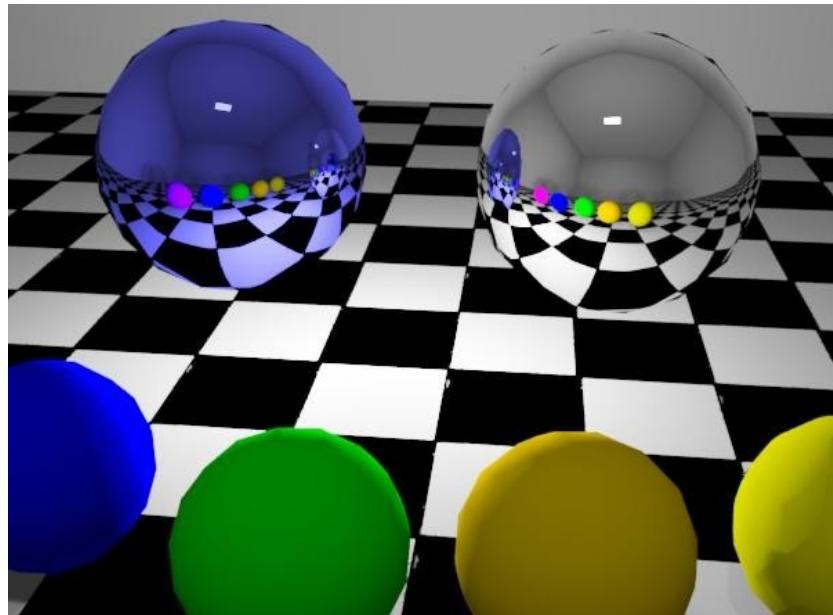


Fonte: Disponível em:

<https://answers.unrealengine.com/storage/temp/68786-reflection_opacity_translucent_test.jpg>. Acesso em: 23 maio 2018

- **Refletividade:** aqui minha careca é quase um exemplo perfeito... estamos falando de o quanto um objeto reflete a luz, e consequentemente, do reflexo que ele consegue gerar. Imagine aquela superfície tão polida, mas tão polida, que você vê seu próprio reflexo nela. Só com produtos de limpeza Ajax! Clique no link... Opa, não pode ter propaganda aqui, esquece!

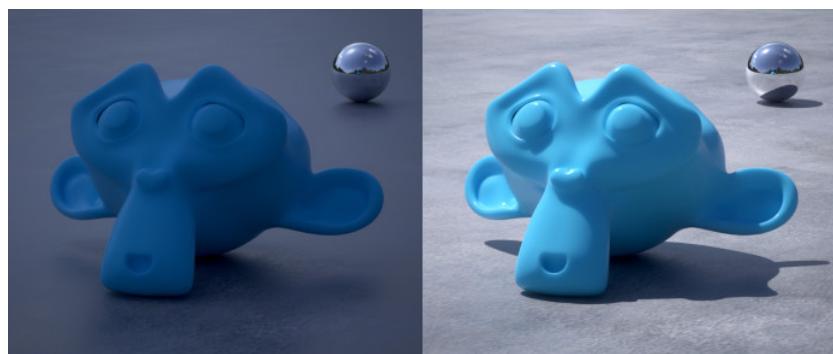
Figura 20 - Duas bolas com bastante Refletividade!



Fonte: COMPUTER GRAPHIC. **Examples of computer graphics.** Disponível em: <<https://artxinc.com/examples-of-computer-graphics.html>>. Acesso em: 29 maio 2018

- **Luminância:** esse aqui é mais transcendental... luminância se refere ao quanto de claridade na tonalidade da cor a iluminação influencia no objeto. Normalmente essa propriedade é utilizada para se gerar mecanismos alternativos de iluminação e sombreamento no modelo, sem precisar calcular o resultado de uma luz em tempo real. Tipo: aqui o azul fica mais claro, aqui mais escuro...

Figura 21 - Dois cenários com diferentes características de luminância



Fonte: Disponível em: <http://adaptivesamples.com/wp-content/uploads/2016/03/scene_sun.jpg>. Acesso em: 23 maio 2018

Ok, mas o que é exatamente uma textura? É uma propriedade? É uma funcionalidade da ferramenta 3D?

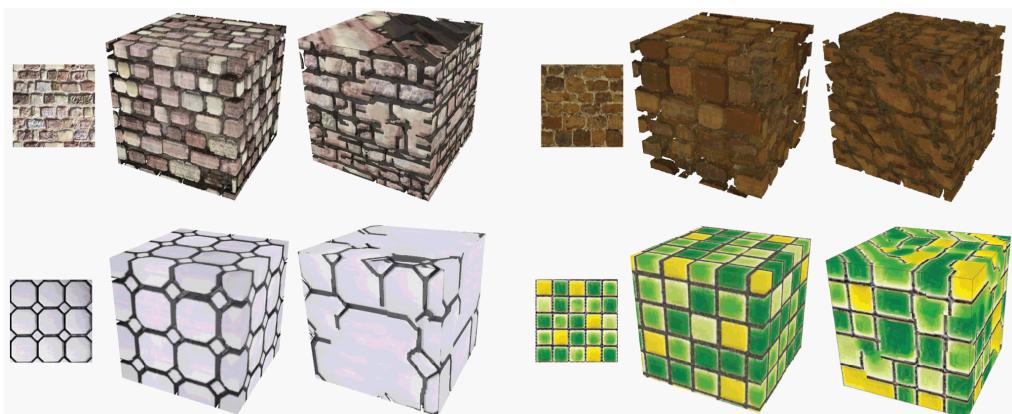
Uma textura é... uma imagem 2D.

Sério.

Juro.

Sabe quando você tá em casa olhando para aquela parede branca e pensa que ela ficaria mais bonita com um papel de parede? E, chique como vocês são, ainda compram aquele papel de parede com relevo, que parece feito de carvalho envelhecido dos Alpes suíços? Pois a ideia é parecida: nós pegamos uma imagem 2D e cobrimos o modelo 3D com ela.

Figura 22 - Texturas para todos os gostos!!



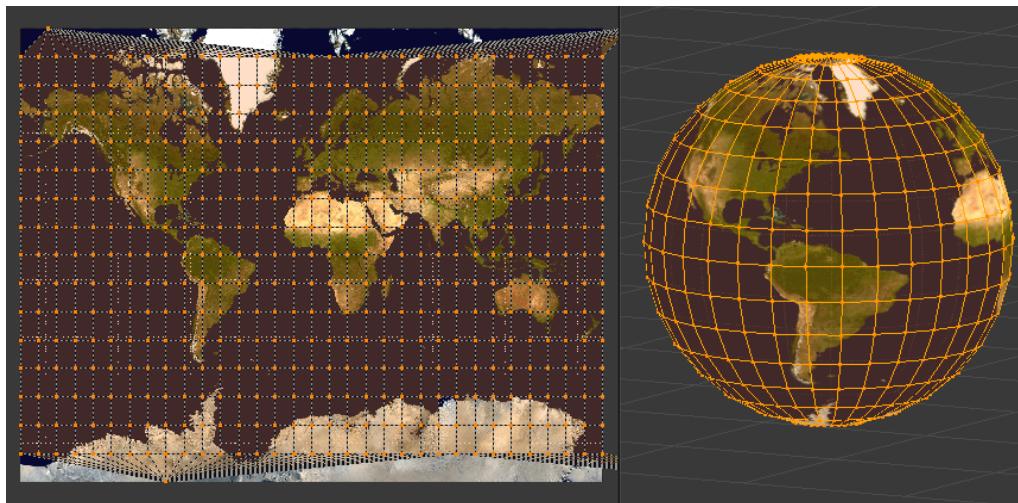
Fonte: Disponível em

<<http://doi.ieeecomputersociety.org/cms/Computer.org/dl/trans/tg/2013/03/figures/ttg20130304609.gif>>. Acesso em: 23 maio 2018

OK, esse da figura não ficou tão legal... mas é porque também não é só enrolar o objeto de qualquer forma, né? Temos que fazer um passo importante para que a textura fique bonita: temos de mapear quais posições da imagem queremos mapear em que lugar do objeto 3D!

Sabe quando pegamos o globo e o mostramos como um mapa 2D?

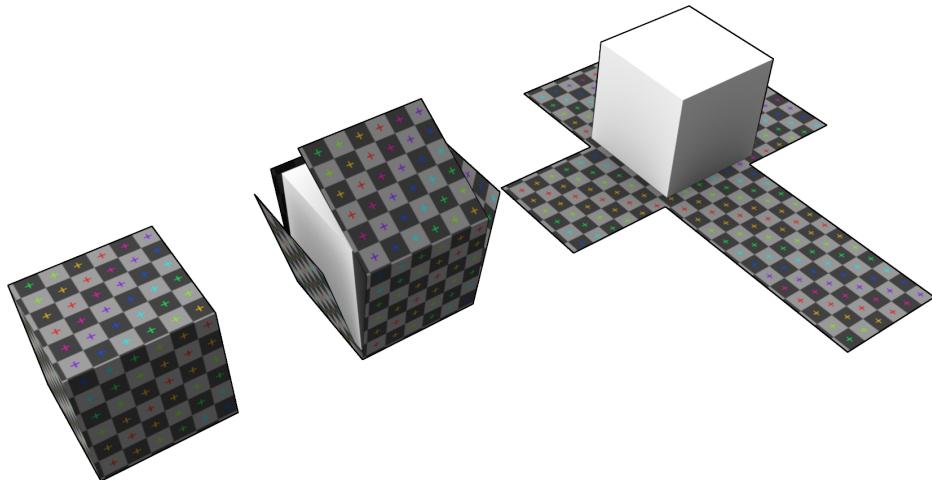
Figura 23 - Mapa-múndi



Fonte: Disponível em: <http://builder.openhmd.net/blender-hmd-viewport-temp/_images/editors_uv_image_uv_editing_unwrapping_sphere-projection.png>. Acesso em: 24 maio 2018

A ideia é a mesma: vamos pegar o nosso modelo e dividi-lo em partes, de forma que cada parte possa ser representada por uma forma 2D. Veja esse exemplo abaixo para um cubo:

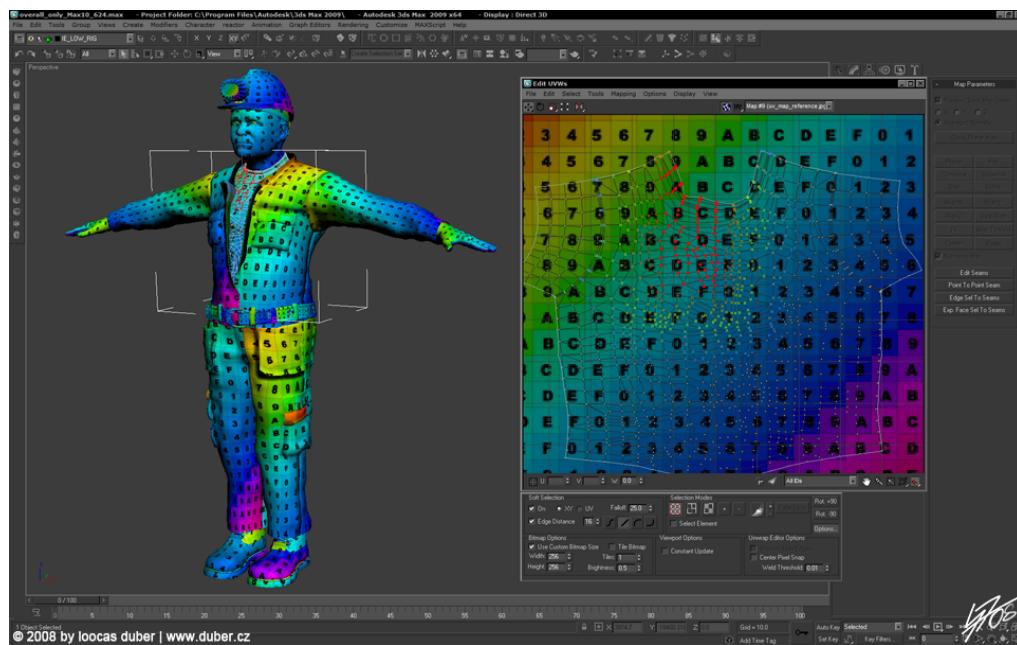
Figura 24 - Formas de 2D



Fonte: Disponível em: <https://andmoor1.files.wordpress.com/2015/03/cube_representative_uv_unwrapping.png>. Acesso em: 24 maio 2018

Esse mesmo procedimento pode ser aplicado para uma forma mais complexa:

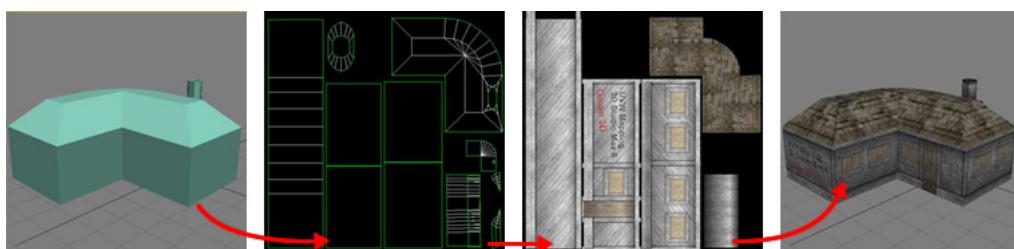
Figura 25 - Um mapeamento para a textura de um personagem



Fonte: Disponível em: <http://blog.duber.cz/wp-content/gallery/screenshots/custom_uv_map_example.png> Acesso em: 23 maio 2018

Viu os pedacinhos do personagem no mapinha ao lado? Com esse mapa, chamado de **mapa UV**, você tem uma relação de como mapear cada coordenada do modelo 3D com a imagem 2D. Agora, basta aplicar as imagens ou texturas criadas em cima desse mapa, e elas serão desenhadas de acordo sobre o modelo 3D. Assim fica beeeeeeeeeeeeem mais tranquilo de fazer!

Figura 26 - Do modelo até a versão com textura!



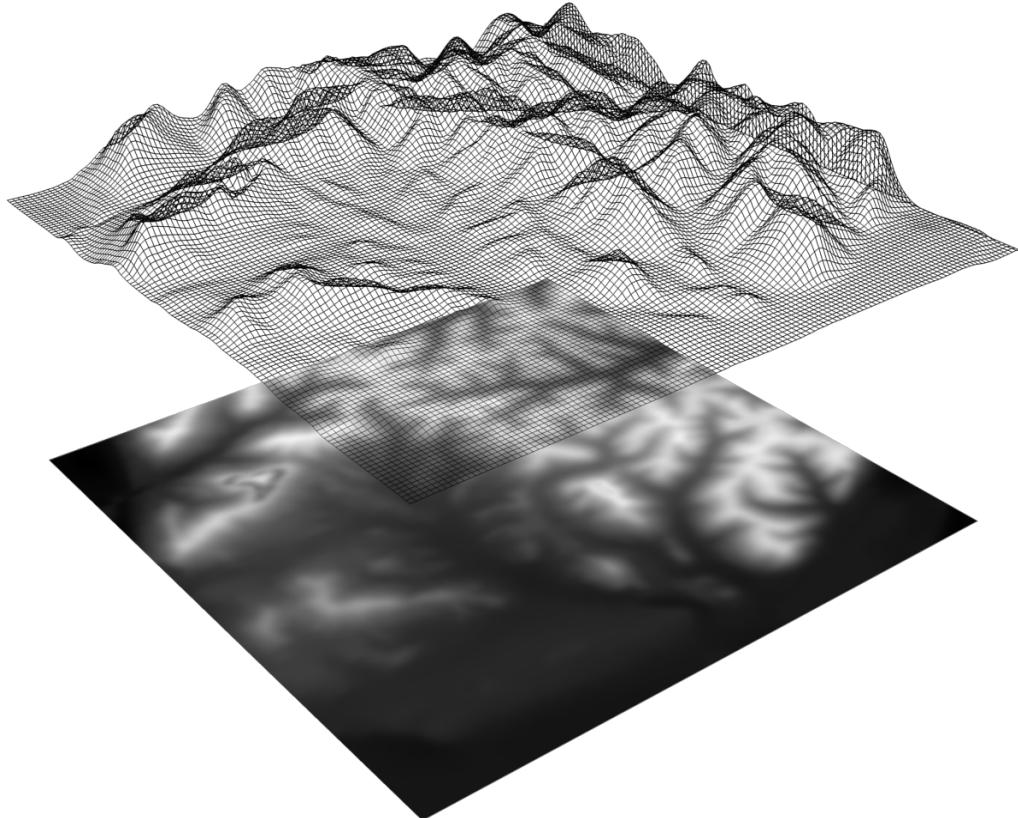
Fonte: Disponível em:
<http://www.republicofcode.com/tutorials/3ds/unwrap_uv_mapping/intro.jpg>. Acesso em: 23 maio 2018

Então, como o mapa UV funciona? Ele pega as coordenadas de cada vértice do objeto (xyz) e mapeia em um par de coordenadas xy em uma imagem 2D. Para não confundir o xy original da coordenada tridimensional com o do mapa, o pessoal por convenção adotou as letras uv para representar a posição xy na imagem 2D. Por isso mapa UV. :P

Uma grande vantagem nessa abordagem é a possibilidade de criar diferentes mapas com camadas de informações que permitem alterar vários atributos do objeto criado. Muitas vezes esses mapas são utilizados para otimizar operações de iluminação ou adicionar detalhes às imagens sem a necessidade de esculpi-los na malha.

Por exemplo: um tipo de mapa muito utilizado é o **Height Map**, ou mapa de altura. Esse mapa indica o relevo de uma superfície, especificando a relação de altura entre os vértices da malha.

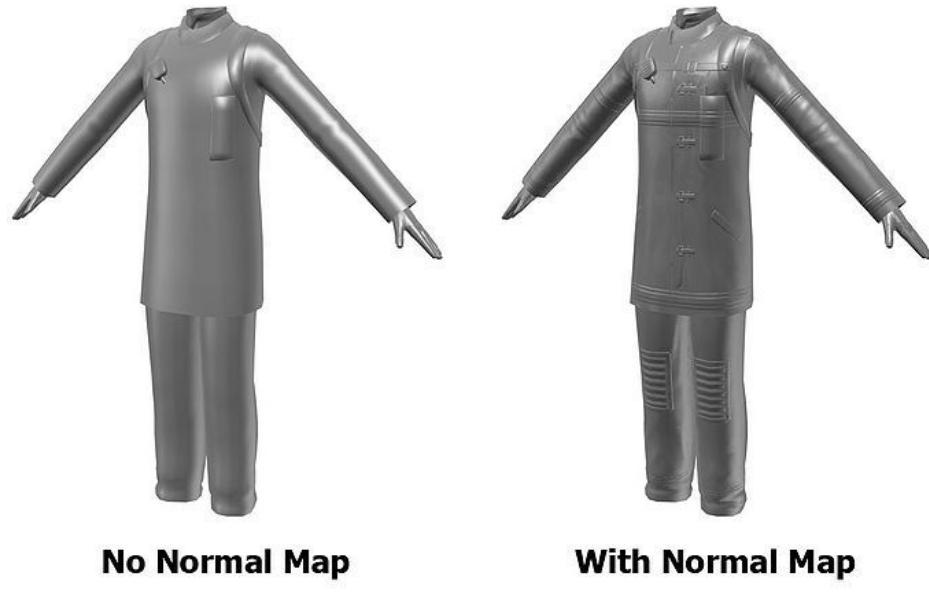
Figura 27 - Mapa de Altura (height map)



Fonte: Disponível em: <<https://www.ssucet.org/~jhudson/15/2802/heightmap/hm1.png>>. Acesso em: 23 maio 2018

Um **mapa normal** é um mapa que simula a iluminação e sombreamento em uma superfície com detalhes, uma espécie de implementação de mapa de relevo. Uma vantagem do uso desse mapa é que podemos criá-lo para simular a aparência visual de um modelo *high poly* e aplicá-lo sobre um modelo *low poly*. Acredite, o desempenho disso é muito melhor do que se usássemos um modelo *high poly* de verdade! Com o mapa aplicado como textura, teríamos um modelo com poucos polígonos e com um bom nível de detalhe!

Figura 28 - O mapa normal acrescenta detalhes interessantes ao modelo.



Fonte: Disponível em:

<<https://i2.wp.com/wiki.thesimsresource.com/images/thumb/1/1f/NormalMapExample.jpg/740px-NormalMapExample.jpg>>. Acesso em: 23 maio 2018

Um **mapa de deslocamento**, por sua vez, opera transformando a malha poligonal sobre a qual é aplicada. Através dele, cada vértice é repositionado de acordo com o valor informado para a posição, e um novo modelo pode ser gerado a partir desse mapa. Uma vantagem é que, a partir de um único modelo, pode-se criar vários outros, apenas trocando os mapas aplicados sobre eles.

Figura 29 - Mapa de deslocamento



Fonte: Disponível em:

<<https://i.pinimg.com/originals/27/7e/8d/277e8d7f854261bfc9966593c1e288f0.jpg>>. Acesso em: 23

Outras formas que as texturas ajudam a renderização do jogo a ganhar desempenho é por meio da simulação de resultados que seriam obtidos por processamento em tempo real, como iluminação, reflexo dos objetos e mapas de especularidade.

Agora vamos falar um pouco de como as texturas são criadas (caso você não queira pegar uma pronta :P)!

O primeiro ponto a discutir seria a abordagem para se criar uma textura. Aqui estamos falando de criação de imagens 2D! Classicamente podemos fazer de três formas:

- **Manualmente:** aqui é a mesma coisa de se trabalhar a criação de *sprites* para um jogo 2D. Vai depender um pouco mais do talento artístico do texturizador, que vai criar a imagem do zero, seja a partir de uma referência, ou por pura inspiração! Aqui a liberdade é muito maior, sendo as únicas limitações a criatividade e a habilidade do artista.

Figura 30 - Textura criada manualmente



Fonte: Disponível em: <[>](#). Acesso em: 23 maio 2018

- **Imagen de referencia:** outra forma comum de se criar uma textura é partir de uma imagem de referência. Seleciona-se uma região da imagem (ou a imagem toda) e faz-se um tratamento para que ela fique no tamanho adequado para uso. Embora seja mais restritiva, já que se fica preso à imagem existente, esse método permite a criação muito

rápida da textura, sem exigir nenhuma habilidade elevada no campo artístico.

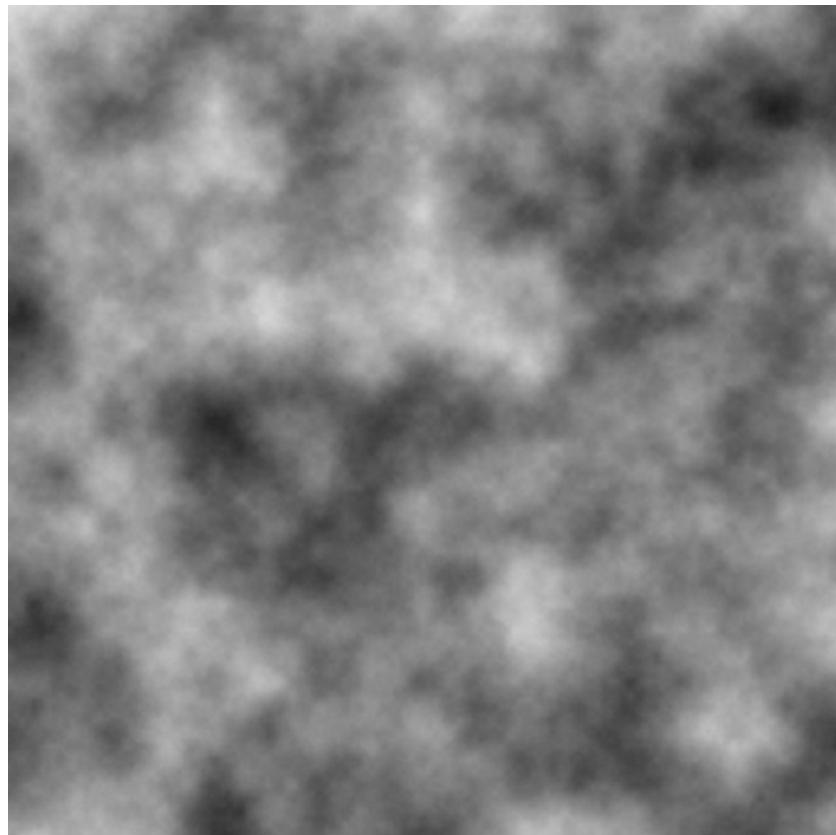
Figura 31 - Textura usando uma imagem de referência



Fonte: Disponível em:
<<https://i.pinimg.com/originals/a3/04/c7/a304c70f27c87ceddba385bd5de3a8af.jpg>>. Acesso em: 23 maio 2018

- **Geração automática:** através do uso de formas matemáticas para gerar ruído, pode-se criar padrões de imagens e utilizá-los como texturas. Uma aplicação comum é na geração de mapas de altura através de ruídos, que permitem uma geração aleatória de terrenos com relevo variado.

Figura 32 - Textura gerada por uma função matemática de ruído



Fonte: Disponível em: <<http://pleek.net/images/projects/16/image01.jpg>>.

Acesso em: 23 maio 2018

Uma coisa muito importante na criação da textura é o momento em que vamos definir o tamanho da imagem. Não queremos gerar uma textura muito grande, porque ela terá um custo maior de armazenamento. Então o que normalmente fazemos é gerar imagens menores (potências de 2, de preferência) e a partir da combinação delas, gerar o padrão que se deseja na imagem ou objeto maior. O que fazemos daqui é criar figuras tendo cuidado com as bordas da imagem, para garantir que elas não tenham uma quebra de continuidade, e então fazemos um processo de ***tiling***, em que as colocamos em posições adjacentes para compor a textura final do objeto. Quando existe uma quebra da continuidade, dizemos que existe um ***seam*** entre as texturas, e então devemos efetuar as correções necessárias para que essas diferenças não fiquem visíveis. Texturas que possuem uma boa continuidade são chamadas de ***seamless***, literalmente sem *seam*.

Agora vamos ver um processo de produção de modelos 3D envolvendo esses conceitos estudados até aqui!

3 - Processo de Confecção para Games

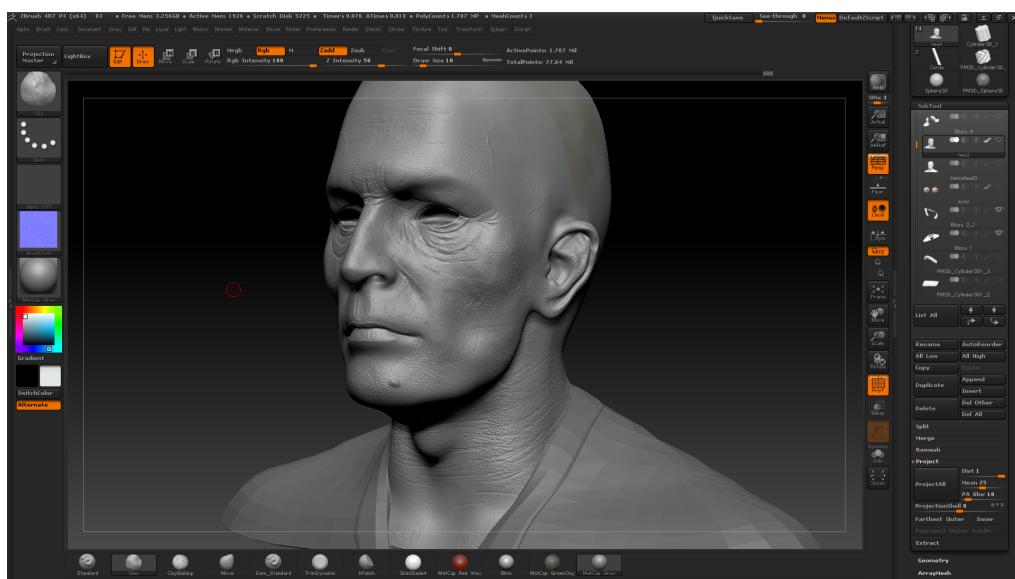
Um modelo 3D para games deve passar por diversos processos antes de sua aprovação na Engine. É preciso levar em conta vários detalhes, por exemplo, se o modelo for uma estátua que aparece em apenas um ângulo da câmera, não é preciso modelar as partes que não aparecem. São esses pequenos detalhes que fazem a diferença no desempenho do seu game.

Vamos imaginar que você pretende lançar um game para a plataforma PC Desktop, quanto mais jogadores puderem jogar o seu jogo melhor, não é mesmo?

Isso significa que mais ele será comprado, pois apenas uma pequena parte do mercado de consumidor de games para pc possui máquinas acima da configuração mediana de mercado.

Depois da criação do Mesh Base (Forma 3D Base), este deve passar por alguns processos:

Figura 33 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:

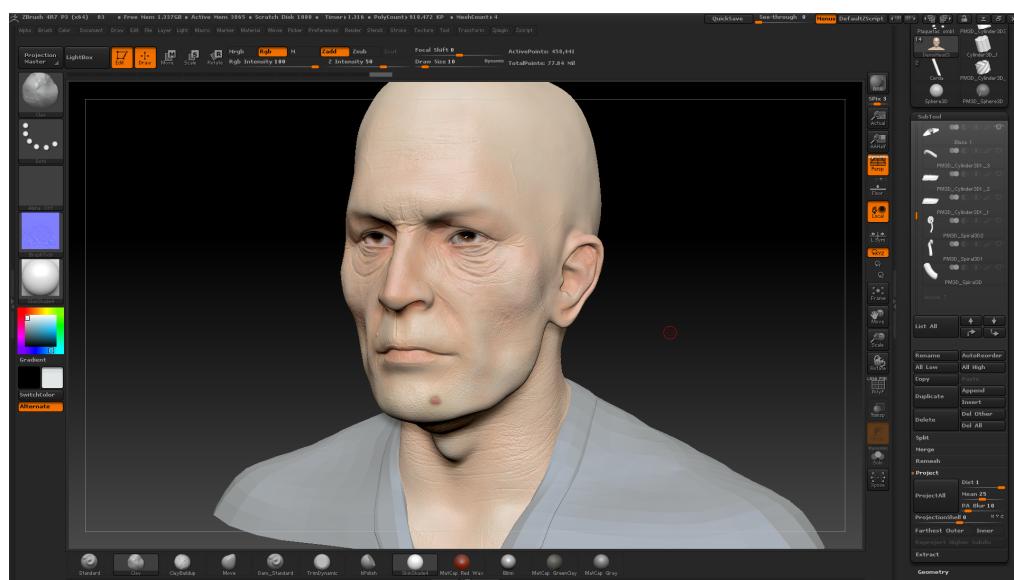
<<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>>. Acesso em: maio de 2017

A pintura digital pode ser feita antes ou depois da retopologia, depende do fluxo de trabalho do artista.

Mas o que é Retopologia?

Como explicado na aula passada, a composição básica de um modelo 3D é o polígono. Entendemos como Topologia uma malha derivada do agrupamento desses polígonos. Quando estamos esculpindo um modelo 3D no software de origem em modo de edição, esse modelo normalmente tem uma alta taxa de polígonos em sua composição (dizemos que esse modelo está em formato Highpoly), a retopologia de um modelo 3D nada mais é do que o processo de redesenhar a malha do personagem com uma baixa taxa de polígonos, tornando-o, assim, um modelo Lowpoly.

Figura 34 - Modelo feito pelo artista digital Alexandre Ferreira.

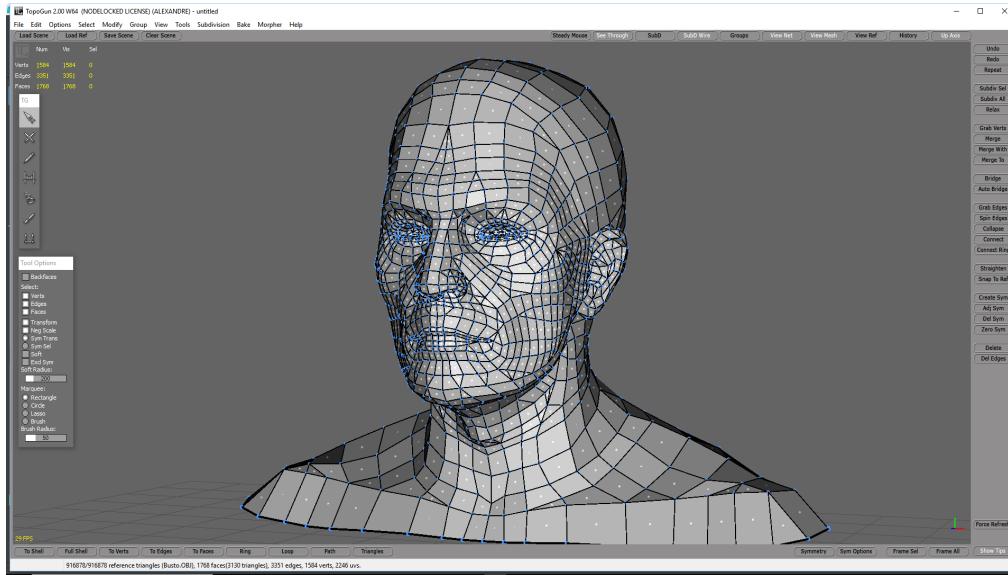


Fonte: Elaborada pelo Autor. Disponível em:

<<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>>. Acesso em: maio de 2017

O mesh base de um personagem como esse, com detalhes sutis na pele, não funcionaria muito bem em uma Engine de Games, devido à alta taxa de polígonos, com isso, depois de feita a versão Highpoly do modelo, o artista deve reduzir a sua quantidade de polígonos criando outro mesh com base nesse.

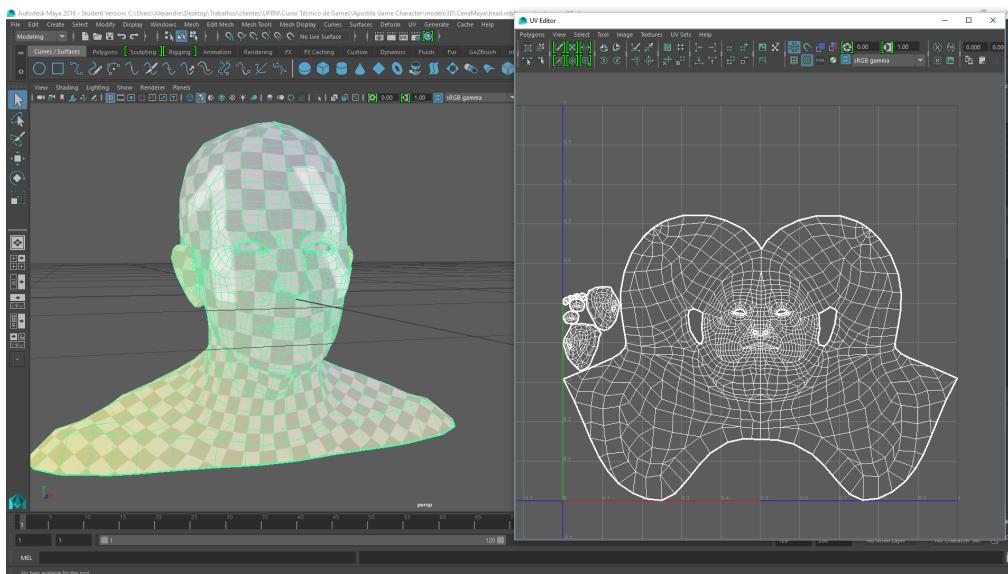
Figura 35 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:
<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>. Acesso em: maio de 2017

Agora, depois do processo de retopologia, a malha UV deve ser editada e seus mapas de textura devem ser criados para que o modelo possa entrar na Engine.

Figura 36 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:
<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>. Acesso em: maio de 2017

Figura 37 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:

<<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>>. Acesso em: maio de 2017

Os PBR Maps (Physically-Based Rendering) trabalharão junto com a física do motor gráfico para dar mais realismo à superfície do mesh, podendo simular rebatimento de luz, relevo e todo tipo de influência de luz e sombra causada pela interação com o cenário.

Figura 38 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:

<<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>>. Acesso em: maio de 2017

Figura 39 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:
<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>. Acesso em: maio de 2017

Figura 40 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:
<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>. Acesso em: maio de 2017

Agora, após configurado, o busto está sendo renderizado em tempo real.

Figura 41 - Modelo feito pelo artista digital Alexandre Ferreira.



Fonte: Elaborada pelo Autor. Disponível em:
<http://alexandreilustrador.deviantart.com/art/Workflow-659559317>. Acesso em: maio de 2017

E aí, curtiu a aula de hoje? Na próxima aula aprenderemos como configurar esse modelo 3D dentro da Engine Unity 3D!

Até lá!



Autoavaliação

1. De que forma a modelagem 3D auxiliou a produção de propaganda na mídia impressa?
2. Faça um resumo explicando como se dá o processo de confecção e otimização de um modelo 3D para ser renderizado em tempo real.



Leitura Complementar

- Game Impact Theory: The Five Forces That Are Driving the Adoption of Game Technologies within Multiple Established Industries
- Modelagem 3D e suas aplicações na pesquisa paleontológica – <http://revistas.unisinos.br/index.php/gaea/article/view/4697>
- Video - The New World of 3D Printing – <https://www.youtube.com/watch?v=s1fOdPrnPog>



Referências

Basemesh - <http://wiki.polycount.com/wiki/BaseMesh>

Retopologia: Ferramenta gratuita para refazer seu 3D -
<https://www.allanbrito.com/2016/08/31/retopologia-ferramenta-gratuita-para-3d/>

Introdução a Retopologia - <http://blenderpower.com.br/tutorial/introducao-a-retopologia/>