

Shading – Parte I

Depois das fases de modelação e iluminação, segue-se a fase de *shading*, que é o processo de atribuir ao objeto, um material (ou materiais) para que ele reaja à luz de forma realista. De facto, os materiais controlam a aparência dos objetos, definindo a substância de que estes são feitos, a sua cor e textura, bem como a forma como a luz interage com os mesmos.

As texturas são colocadas em cima dos materiais para lhes dar cores mais complexas e outros efeitos. Um objeto é coberto com um material, que pode conter várias texturas: uma textura de imagem de pedra, uma textura para fazer a pedra parecer com relevo, e uma textura para fazer a pedra deformar-se de diferentes maneiras.

Para aplicar texturas no Blender, é necessário que os objetos tenham um material atribuído. As texturas podem ser procedimentais (geradas pelo próprio Blender com base em fórmulas matemáticas), ou de imagem, sendo que estas últimas podem ser criadas do zero, ou a partir de imagens existentes.

O exemplo que se segue, pretende ilustrar o processo de *shading* num objeto que pretende representar uma laranja, o qual compreende três etapas:

- Criação do mapa UV (ou *UV map*) que consiste na representação plana da superfície do modelo 3D para permitir envolvê-lo numa textura 2D. Para tal, o objeto 3D é literalmente “desembrulhado” (*unwrapped*);
- Criação da textura através do modo *Texture Paint* que permite pintar diretamente a malha anteriormente mapeada;
- Criação dos materiais a atribuir ao modelo de forma a que a aparência deste se aproxime de uma laranja verdadeira.

1. UV Unwrap

Tal como referido, para aplicarmos uma textura a um modelo 3D, temos de fazer uma correspondência entre o elemento 2D (a imagem que constitui a textura) e o modelo tridimensional (à semelhança do invólucro de um Pai Natal de chocolate, ou da projeção do globo terrestre num planisfério). O processo de “desembrulhar” a malha poligonal para a projeção num plano bidimensional tem a designação de **UV Unwrap**, em que U e V são as designações dos eixos da textura bidimensional (da mesma forma que temos os eixos X, Y e Z para o espaço tridimensional).

- Abrir o ficheiro do FCG_05_Shading_A.blend;
 - Caso não esteja, ocultar a *Collection*, deixando visível apenas a *Collection 2*, onde podemos encontrar o modelo de uma laranja;

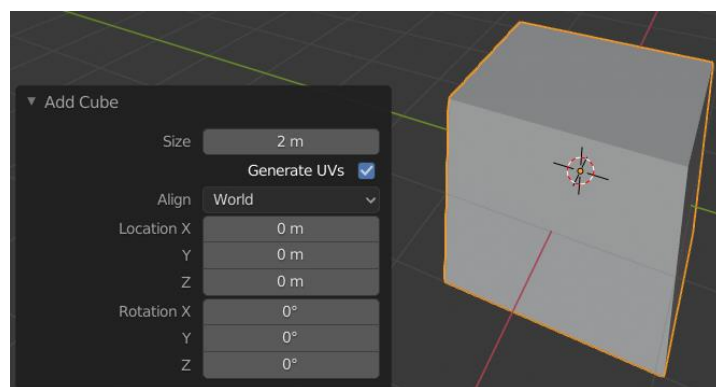
Para realizarmos a operação de *UV Unwrap* corretamente, **devemos garantir que a escala está definida como 1 nos três eixos**. Uma escala não-uniforme irá gerar erros no processo.

- Selecionar o objeto *Laranja* e observar (no ícone *Object Properties* do editor *Properties*) que a escala do modelo não é 1;
- No *Header*, selecionar **Object** → **Apply** (ou pressionar as teclas **CTRL+A**) e escolher a opção **Scale**.

Em seguida, é necessário criar o **UV map**:

- Mudar o *workspace* para **UV Editing** ficando o ecrã fica dividido em duas áreas: no lado direito o editor **3D Viewport** e no do lado esquerdo o **UV Editor**;
- Em modo de edição, selecionar todas as faces da *Laranja* (tecla **A**);
- Observar no **UV Editor** que a *Laranja* tem já um *UV map* atribuído.

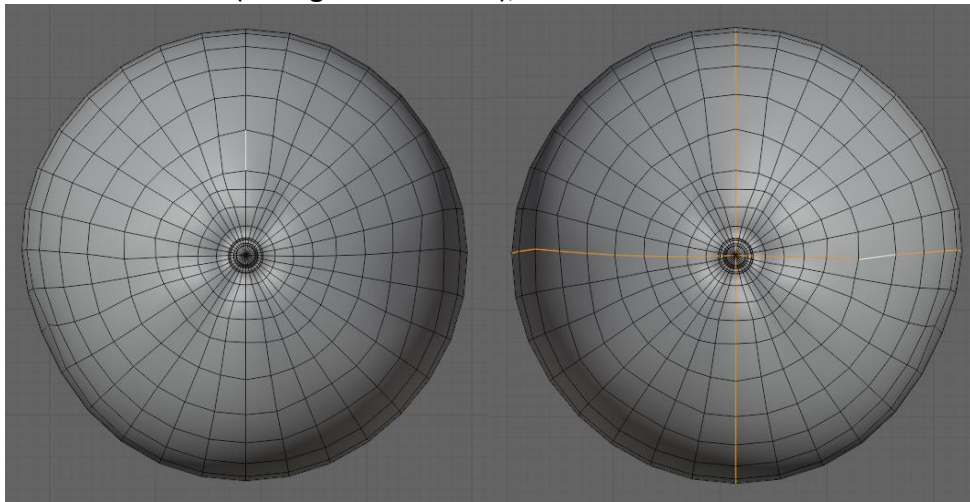
Isto acontece porque a *Laranja* foi criada a partir de uma *UV Sphere* e qualquer malha poligonal pré-definida no Blender tem já um *UV map* atribuído. A título de exemplo, pode criar-se um cubo, entrar em *Edit Mode* e selecionar todas as faces (observar o **UV Editor**). No final, eliminar o cubo.



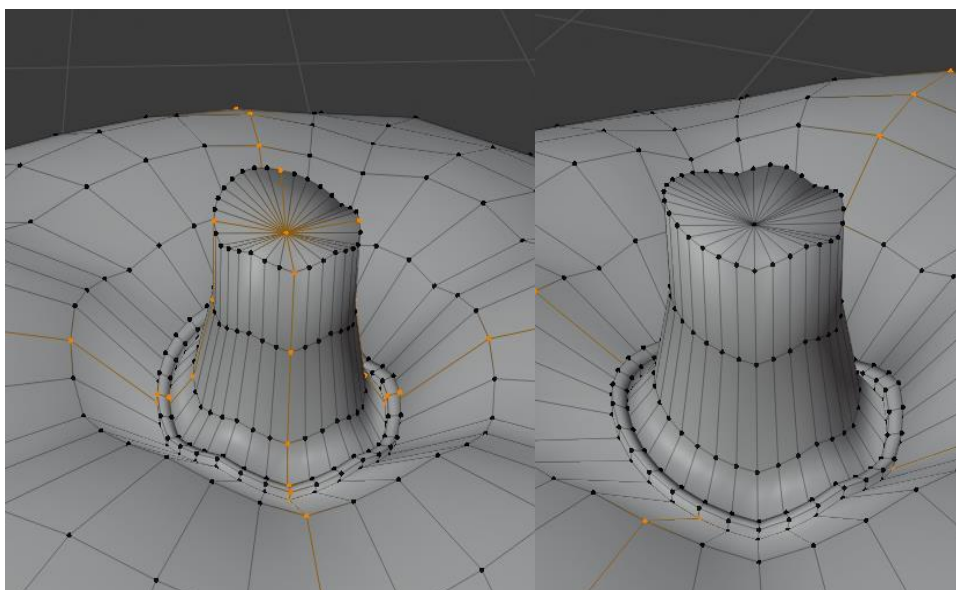
- No entanto, vamos criar manualmente um *UV map* para a *Laranja*, que seja mais adequado para trabalharmos a sua textura;
- Podemos observar que, ao selecionarmos um conjunto de vértices do modelo, apenas os vértices correspondentes aparecem no **UV Editor**. Se selecionarmos todos os vértices (tecla **A**), surge o *UV map* completo. Para simplificar o trabalho, no canto superior esquerdo do **UV Editor** podemos ativar a opção o **UV Sync Selection** (ícone de dois setas), de forma a sincronizar o que é selecionado e apresentado no **UV Editor** e no **3D Viewport**.

Para indicar ao *software* como deve “desembrulhar” a superfície do modelo (*unwrap*), temos que definir quais são as arestas que irão funcionar como “costuras” (*seams*).

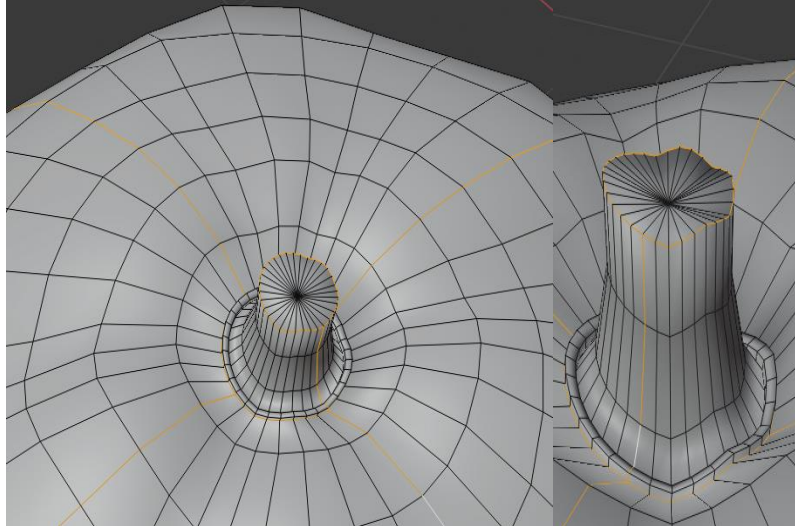
- No editor **3D Viewport**, mudar para a vista de topo ortográfica;
- Remover todas as seleções (**ALT+A**);
- Selecionar uma aresta (ver figura da esquerda) e pressionar simultaneamente a tecla **ALT** e o botão esquerdo do rato (**LMB**) criando um *Edge Loop* (equivalente a selecionar uma aresta e as opções do *Header*: **Select** → **Select Loops** → **Edge Loops**);
- Repetir a operação anterior (mantendo a tecla **SHIFT** pressionada) para criar mais três *Edge Loops*, dividindo o modelo em quadrantes mais ou menos com a mesma dimensão (ver figura da direita);




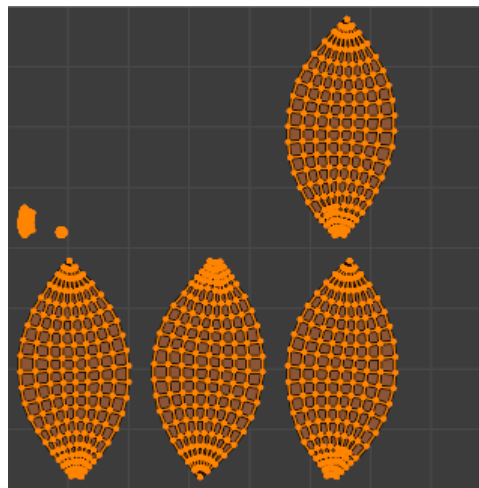
- Na vista de perspectiva, fazer *zoom* na zona do caule (topo da *Laranja*) e eliminar os vértices selecionados (ver figura da esquerda). O resultado deverá ser o que se apresenta na figura da direita. Neste processo, não esquecer de manter a tecla **SHIFT** pressionada;





- Mantendo ainda a tecla **SHIFT** pressionada, criar um *Edge Loop* (tecla **ALT+LMB**) em torno da base do caule, um *Edge Loop* no topo do caule e um *Edge Loop* dividindo o caule na vertical, de forma a obter um resultado semelhante ao das figuras seguintes.



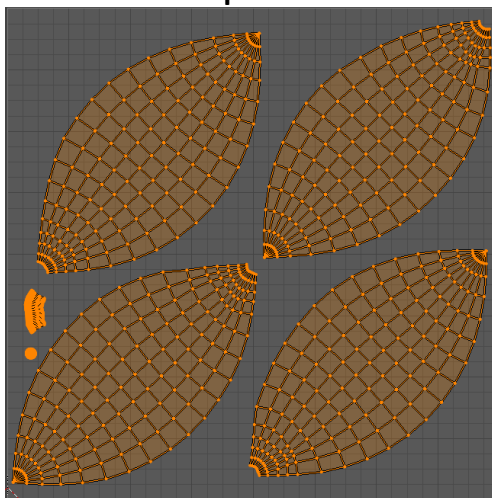
- Ainda em modo de edição de arestas (), pressionar o botão direito do rato (**RMB**) e selecionar **Mark Seam**. As *seams* deverão aparecer marcadas num tom avermelhado. Se for necessário corrigir alguma *seam* marcada por engano, podemos selecionar a *seam* em questão, pressionar o botão direito do rato e escolher **Clear Seam**;
- Selecionar toda a malha (tecla **A**);
- No *Header* do editor **3D Viewport**, selecionar **UV → Unwrap** (em alternativa, usar o atalho **U** para abrir o menu *UV Mapping* e selecionar **Unwrap**);
- Podemos observar o novo *UV map* criado no lado esquerdo, no **UV Editor**;
- Se alterarmos o valor de **Margin** no painel *Unwrap* que surge no canto inferior esquerdo do **3D Viewport**, podemos ver que mudamos a distância entre as “ilhas” (*islands*) do *UV map*;
- Definir o valor **Margin** como **0.2**, obtendo um *UV map* semelhante ao da figura seguinte:



- Desativando o **UV Sync Selection** (), observamos que temos diversos modos de seleção no **UV Editor**, incluindo a seleção por ilhas ();
- Se selecionarmos uma ilha, verificamos que, à semelhança do que acontece com uma malha, podemos editar a sua posição, rotação ou escala utilizando as teclas **G**, **R** e **S**, respetivamente.

Contudo, **alterar a escala de uma ilha isoladamente não é recomendável**. Isto porque, neste momento, todas as ilhas do *UV map* têm dimensões proporcionais à área que ocupam no modelo. Alterando a escala de uma ilha, vamos criar discrepâncias a nível da resolução da textura nas diversas partes do modelo. Para alterar a escala das ilhas, devemos sempre ter **todas as ilhas selecionadas**. Convém ainda que as ilhas ocupem a maior área possível do *UV map* (reduzindo ao mínimo os espaços vazios), de forma a otimizar a resolução da textura.

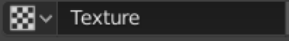
- Selecionar todas as ilhas (tecla **A**) e aumentar ligeiramente a escala (tecla **S**);
- Arrumar individualmente as ilhas de forma a obter um aproveitamento eficiente do *UV map*, utilizando as teclas **G** e **R**, chegando a um resultado semelhante ao da figura seguinte (é importante garantir que **não existe sobreposição entre as ilhas** e que estas se encontram **completamente dentro do UV map**):

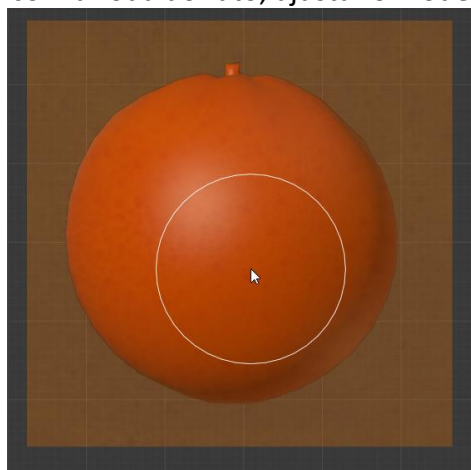


2. Texture Painting

Após a realização do *UV Unwrap*, podemos começar a pintar a textura no nosso modelo recorrendo ao modo *Texture Paint*.

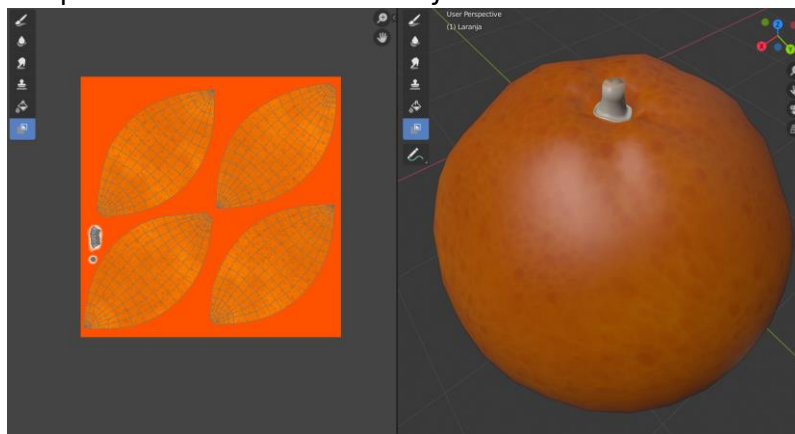
- Selecionar o *workspace Texture Paint*, ficando o espaço de trabalho dividido entre o editor **3D Viewport** (no lado direito) e o **Image Editor** (no lado esquerdo);
- No **3D Viewport** podemos observar que estamos a trabalhar no modo *Texture Paint* e que o modelo apresenta uma cor roxa, o que indica que não se encontra atribuída nenhuma textura;
- Para podermos pintar a textura de um objeto, temos de partir de uma imagem de base. No **Image Editor** clicar no botão **+New**;

- No menu que aparece, inserir no campo *Name* o nome para a textura (*Laranja_textura*) e alterar a resolução da imagem para 2048 x 2048 px;
- No campo *Color* podemos definir uma cor-base para o modelo. Selecionar um tom alaranjado e clicar **OK**;
- Verificar que o *UV map* aparece pintado a cor-de-laranja;
- Para fazer a ligação entre a textura e o modelo, temos que seleccionar o ícone **Active Tool and Workspace Settings** (🔧) do editor **Properties**. No painel **Texture Slots** alterar o **Mode** para **Single Image**;
- De seguida, clicar no ícone da imagem (🖼️), ao lado do botão **+New**) e seleccionar *Laranja_textura*, ficando agora o modelo pintado de cor-de-laranja;
- Para controlarmos o tamanho do pincel com que vamos pintar o modelo, podemos definir o raio na secção **Brush Settings** → **Radius**. Em alternativa, no editor **3D Viewport** podemos pressionar a tecla **F**, ajustar o tamanho movendo o rato e confirmar com o **LMB**;
- No painel **Brush Settings**, colocar o valor de **Strength** no máximo (1.000), caso não esteja;
- No ícone **Texture Properties** (🎨), do editor **Properties**, clicar em **New** para adicionar uma nova imagem. No campo  alterar o nome para *Laranja*;
- No sub-painel **Settings**, do painel **Image**, carregar em **Open** e seleccionar o ficheiro *Laranja_text.jpg*;
- Voltar ao separador **Active Tool and Workspace Settings** (🔧) do editor **Properties** e no sub-painel **Texture**, do painel **Brush Settings**, no campo **Mapping** seleccionar a opção **Stencil**;
- Podemos observar que a imagem da textura aparece no **3D Viewport**. Mudar para a vista frontal ortográfica. Arrastar a imagem para cima da *Laranja* com o **RMB** e, juntamente com a roda do rato, ajustar o modelo à imagem:






- Utilizar o **LMB** para pintar a textura no modelo. Alterar depois para a vista oposta e repetir o processo;
- Repetir este procedimento para todas as vistas;
- Na vista de perspectiva, garantir que a totalidade da superfície do modelo se encontra pintada corretamente e corrigir caso se encontre algum defeito. Observar que esta textura também aparece aplicada ao *UV map* do **Image Editor**;

- No **Image Editor** selecionar a opção **Mask** () e eliminar a textura nas duas ilhas correspondentes ao caule da **Laranja**:





3. Materials

Após termos criado a textura da casca da laranja, podemos aplicá-la ao modelo mediante a criação de um material.




- Mudar o *workspace* para **Shading** que, por omissão, tem o modo **Material Preview** selecionado no **Viewport Shading** ();
- Podemos observar que, para além dos editores **3D Viewport**, **Outliner** e **Properties**, o *workspace* está dividido no **Shader Editor** (abaixo do **3D Viewport**), **Image Editor** (canto inferior esquerdo) e **File Browser** (canto superior esquerdo);
- Com a **Laranja** selecionada, no editor **Properties** selecionar o separador **Material Properties** () e escolher **+ New**;
- Observamos que foi atribuído um material ao modelo, aparecendo ainda um sistema de *nodes* (**Principled BSDF**) no **Shader Editor**. Este *node* é equivalente às propriedades que aparecem no painel **Surface** do ícone **Material Properties** do editor **Properties**;
- Nesse mesmo ícone, alterar o nome do material para “**Laranja**” ().

BSDF (*Bidirectional scattering distribution function*), é a função matemática utilizada para descrever a forma como a luz interage com uma superfície. O **Principled BSDF** apresenta múltiplos parâmetros que podem ser utilizados para configurar o material, dos quais podemos salientar os seguintes:

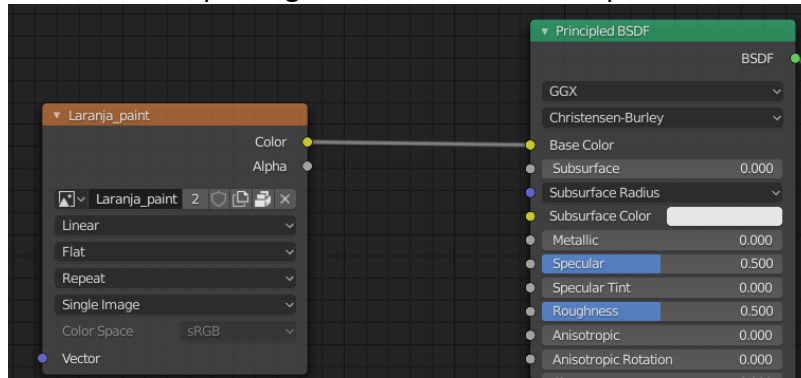
- O campo **Base Color** controla a cor da superfície do material;
- O campo **Roughness** controla a rugosidade do material;
- O campo **Metallic** regula o aspeto metálico/não-metálico do material;
- O campo **Transmission** controla o nível de opacidade do material (valor 0 corresponde a totalmente opaco; valor 1 corresponde a um material tipo vidro). Por omissão, apenas veremos esta superfície tipo vidro a refratar a imagem do ambiente HDRI.

- Atribuir ao material *Laranja* uma **Base Color** cor-de-laranja, um valor de **Metallic** e **Roughness** de 0 e colocar o valor de **Transmission** (para fazer a refração da luz) a 1;
- Ativar a *Collection* (tornando visível a mesa sobre a qual está colocada a *Laranja*) e observar que a superfície da mesa não aparece refratada no vidro;
- No editor **Properties**, seleccionar o ícone **Material Properties** () , painel **Settings** e ativar **Screen Space Refraction**;
- De seguida, ir a **Render Properties** () e ativar **Screen Space Reflections**, observando as diferenças geradas no material;
- Ocultar de novo a *Collection*, voltar a colocar o valor de **Transmission** a 0 e de **Roughness** a 0.5.


O campo *Base Color* permite-nos, para além de definir a cor do material, adicionar uma textura, como a que criámos anteriormente para a *Laranja*.

- Para adicionar uma textura, podemos fazê-lo de duas formas: através do ícone **Material Properties** do editor **Properties**, ou através do **Shader Editor**;
- No editor **Properties**, ícone **Material Properties**, painel **Surface**, clicar no pequeno círculo à esquerda da cor do campo *Base Color* () e, no menu que surge, seleccionar *Image Texture* da coluna *Texture*;
- Em *Base Color*, seleccionar o ícone da imagem () e escolher a textura *Laranja_textura*, observando que a textura é aplicada ao modelo no **3D Viewport**;
- Podemos observar também que, no **Shader Editor**, esta operação criou um novo *node* (*Image Texture*) que ficou ligado ao campo *Base Color* do *node* *Principled BSDF*;
- Apesar da edição de *shaders* poder ser feita através do editor **Properties**, quando estes aumentam de complexidade, este editor pode revelar-se uma interface algo confusa, sendo mais fácil utilizar o **Shader Editor**;
- Para efetuarmos a mesma operação no **Shader Editor**, vamos começar por seleccionar o *node* *Image Texture* e eliminá-lo;
- De seguida, no **Shader Editor**, menu **Add** (ou teclas **SHIFT+A**), seleccionar **Texture** → **Image Texture** (em alternativa, no campo **Search** do menu **Add**, escrever *Image Texture* e seleccionar);
- Clicar com o **LMB** para colocar o *node* no editor. Se for necessário, podemos fazer *zoom in* com a roda do rato;
- No *node* *Image Texture*, clicar no ícone da imagem () e seleccionar a textura *Laranja_textura*;

- Utilizar o **LMB** para ligar a saída *Color* ao campo *Base Color* do *Principled BSDF*.



É possível usar mais do que um material ao mesmo modelo, atribuindo materiais distintos a diferentes partes da malha poligonal.

- No ícone **Material Properties** () adicionar um segundo material carregando no botão **+** junto ao topo;
- De seguida carregar em **+ New** e mudar o nome do novo material para *Caule*;
- Entrar no **modo de edição** e, em modo de seleção de faces, seleccionar as faces do caule que não têm um material atribuído (visto de topo e tecla **C** poderão ser úteis);
- Seleccionar o material *Caule* na lista disponível em **Material Properties** e carregar em *Assign*;
- De seguida, mudar a *Base Color* para uma tonalidade verde;
- Sair do *Edit Mode*.

Com o material *Laranja* seleccionado, no **Shader Editor** podemos observar uma estrutura de *nodes* baseada em *inputs* e *outputs*, com a seguinte organização: um *node* de textura ligado a um *node* de *shader*, que por sua vez está ligado a um *node* *Material Output*. Como a ligação entre estes dois últimos está feita entre o *BSDF* e *Surface*, isto quer dizer que o *shader* calcula a interação da luz com o material e que o resultado é-nos apresentado na superfície do modelo.

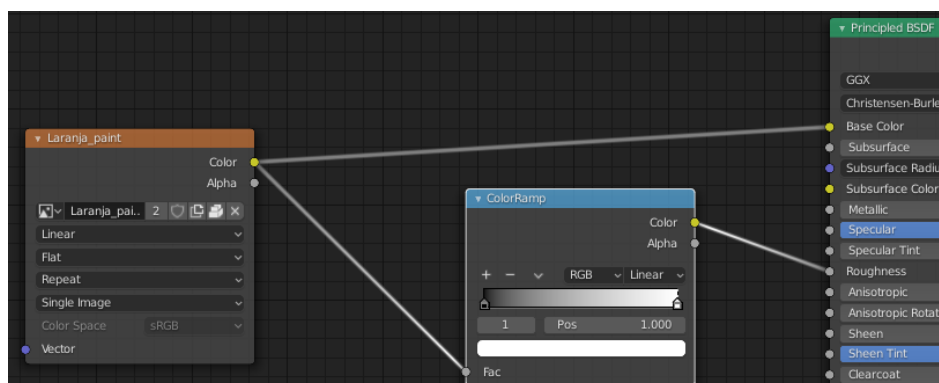
Podemos ligar *nodes* uns aos outros, clicando com o **LMB** num *output* e ligando ao *input* desejado. Para desligar *nodes* podemos utilizar o **LMB** no *input* do *node* de destino e retirá-lo, ou manter pressionada a tecla **CTRL+RMB** e arrastando, cortar a ligação.

As cores nos *inputs* e *outputs* indicam-nos entre quais deles podemos estabelecer ligações. Dão-nos ainda informação sobre o tipo de dados que estamos a utilizar. Por exemplo, os pontos amarelos assinalam informação sobre a cor, enquanto os pontos cinzentos dizem respeito à escala de cinzentos ou informação numérica.

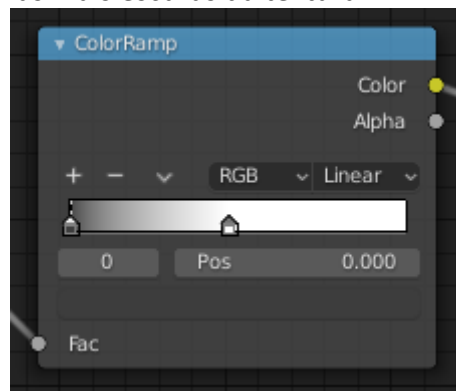
Relativamente ao objeto *Laranja*, observamos ainda que todas as faces da malha poligonal têm agora um material atribuído. O aspeto dos materiais, contudo, é ainda

artificial e pouco realista. Podemos aperfeiçoar o material editando o *shader*, e criando pequenas imperfeições na superfície como seria expectável observar numa laranja real.

- Para adicionar rugosidade ao modelo *Laranja*, poderíamos criar um *Roughness map* recorrendo a ferramentas externas ao Blender. No entanto, este último disponibiliza formas de simular um *Roughness map* de forma rápida e com um resultado final satisfatório;
- Para tal, começamos por adicionar, no **Shader Editor**, um *node ColorRamp* (**Add**→**Converter**→**ColorRamp**);
- Ligar o campo *Color* da textura *Laranja_textura* ao campo *Fac* do *ColorRamp*;
- Ligar o campo *Color* do *node ColorRamp* ao campo *Roughness* do *Principled BSDF*.



- Ao movermos os controladores do gradiente, podemos observar as variações na superfície do modelo. **Mover o controlador branco** mais para o meio do gradiente (**Pos 0.4**). **Selecionar o controlador preto** e, clicando na barra de cor abaixo, subir um pouco o valor do brilho, de forma a aumentarmos ligeiramente a rugosidade nas zonas mais escuras da textura.



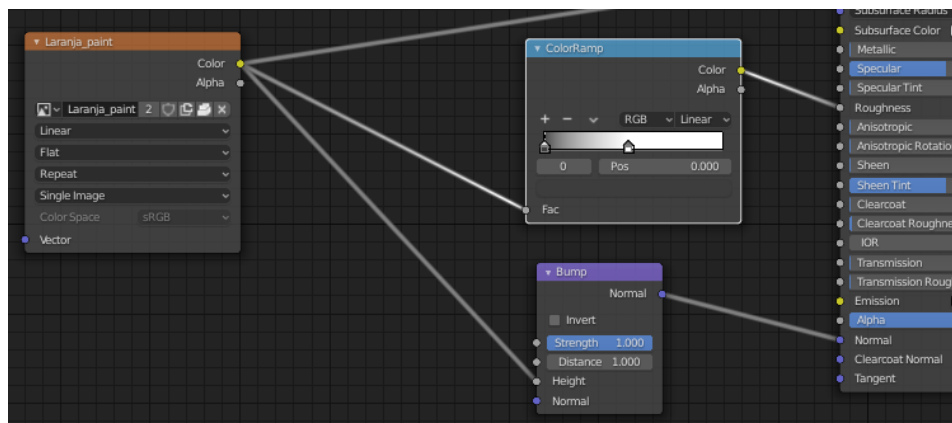
Neste exemplo, estamos a utilizar o *node ColorRamp* para controlar a rugosidade da superfície através de um gradiente (neste caso em escala de cinzentos). O **0** corresponde ao **preto** e o **1** corresponde ao **branco**. Tendo esta informação ligada à textura e ao campo *Roughness*, vamos indicar ao *software* que a rugosidade da superfície não é homogénea, criando diferenças de rugosidade entre as áreas mais escuras e as mais claras da textura (ou seja, um valor mais perto do **0** na informação de cor converte-se num valor mais perto do **0** na informação sobre a rugosidade, e

vice-versa). Esta variação contribui para a obtenção de um resultado final mais realista.

Para adicionarmos imperfeições à superfície da *Laranja* podemos usar um *Bump map*, uma vez mais com recurso a ferramentas fora do Blender. No entanto, tal como para o *Roughness map*, também podemos simular um *Bump map* no Blender.

Um *Bump map* é uma imagem em escala de cinzentos, indicando ao *software* que os pontos mais próximos do branco devem ser puxados para fora da superfície, enquanto os pontos perto do preto devem ser empurrados para dentro desta, criando imperfeições e micro-relevos na superfície.

- No **Shader Editor**, seleccionar **Add** (ou **SHIFT+A**) → **Vector** → **Bump** e adicionar o *node*;
- Ligar o campo **Color** da textura *Laranja_textura* ao campo **Height** do *node* criado;
- Ligar o *output Normal* deste *node* ao *input Normal* do *Principled BSDF*;

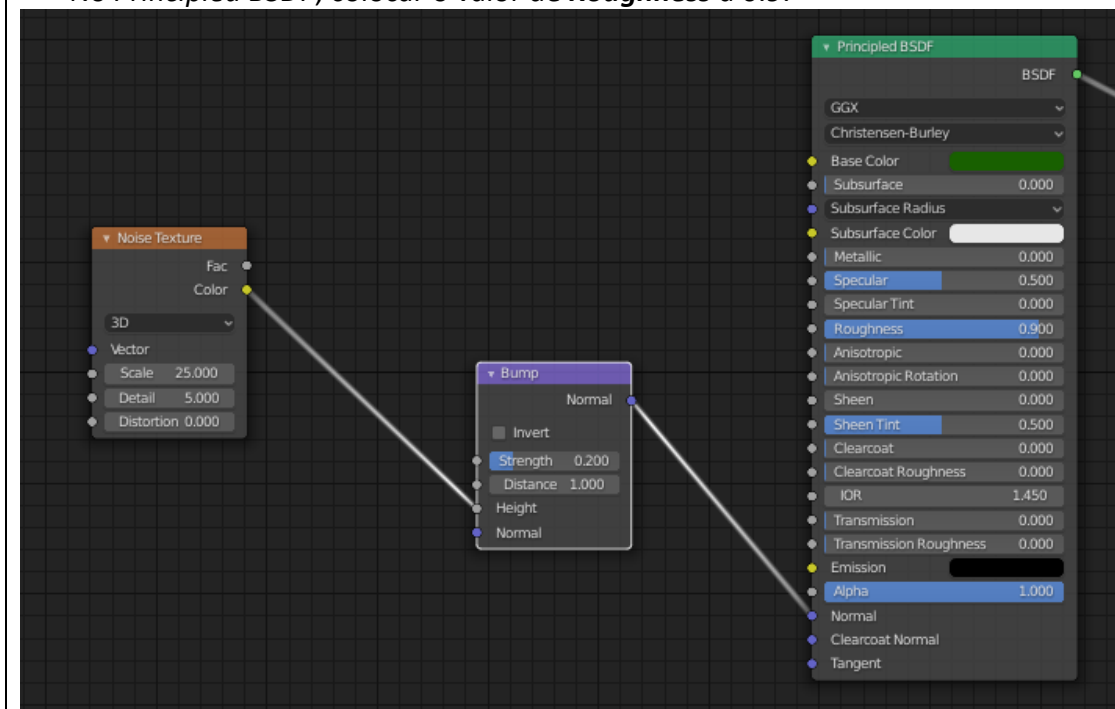


Desta forma o *software* calcula uma altura (*height*) diferente para cada pixel da textura com base na sua cor, simulando os micro-relevos da superfície. Contudo, podemos observar que o resultado obtido não é realista, uma vez que a deslocação destes pontos está muito exagerada.

- No *node Bump* reduzir o valor de **Strength** para 0.15, de forma a se conseguir atingir assim um aspeto mais realista.

Tendo conseguido um aspeto semi-realista para a casca da Laranja, podemos rapidamente usar uma textura de tipo *Noise* (“ruído”) para simular um *Bump map* e quebrar o aspeto homogéneo do material do Caule, que neste momento é constituído apenas por uma cor verde.

- Em **Material Properties** seleccionar o material *Caule*;
- No **Shader Editor**, adicionar um *node Noise texture* (**Add** → **Texture** → **Noise Texture**);
- Neste *node*, alterar os valores de **Scale** para 25 e de **Detail** para 5;
- Adicionar um *node Bump* (**Add** → **Vector** → **Bump**) e colocar a **Strength** a 0.2;
- Ligar o **output Color** do *Noise Texture* ao **input Height** do *node Bump*;
- Ligar o **output Normal** do *node Bump* ao **input Normal** do *Principled BSDF*;
- No *Principled BSDF*, colocar o valor de **Roughness** a 0.9.



Resultado final:

