

Computação Gráfica (MIEIC)

Trabalho Prático 4


Aplicação de texturas


Objetivos


- Definir coordenadas de textura de forma adequada
- Explorar os diferentes modos de aplicação de textura.
- Combinar o uso de materiais com texturas para obter uma aparência realista

Trabalho prático

Ao longo dos pontos seguintes são descritas várias tarefas a realizar. Algumas delas estão anotadas

com o ícone  (captura de imagem). Nestes pontos deverão, com o programa em execução, capturar uma imagem da execução. Devem nomear as imagens capturadas seguindo o formato "CGFImage-tp4-TtGgg-x.y.png", em que TtGgg referem-se à turma e número de grupo e x e y correspondem ao ponto e subponto correspondentes à tarefa (p.ex. "CGFImage-tp4-T3G10-2.4.png", ou "CGFImage-tp4-T2G08-extra.jpg").

Nas tarefas assinaladas com o ícone  (código), devem criar um ficheiro .zip do vosso projeto, e nomeá-lo como "CGFCode-tp4-TtGgg-x.y.zip", (com TtGgg, x e y identificando a turma, grupo e a tarefa tal como descrito acima).

Quando o ícone  surgir, é esperado que executem o programa e observem os resultados. No final, devem submeter todos os ficheiros via Moodle, através do link disponibilizado para o efeito.

Devem incluir também um ficheiro **ident.txt** com a lista de elementos do grupo (nome e número). Só um elemento do grupo deverá submeter o trabalho.

Preparação do Ambiente de Trabalho

Este trabalho deve ser baseado numa cópia dos trabalhos anteriores (uma sala de aula com dois planos, duas mesas, duas paredes, chão, um cilindro e um prisma, TP2 e TP3 respectivamente). Devem descarregar o código disponibilizado para este trabalho do Moodle e acrescentar a pasta **resources** incluída (que contém as imagens a usar como texturas) no vosso projeto, ao mesmo nível que a pasta **lib**. Substitua também o ficheiro **plane.js** fornecido anteriormente pelo fornecido no novo código.

1. Aplicação de texturas

O mapeamento de texturas é uma forma de atribuir informação armazenada em formato *bitmap* a diferentes zonas das superfícies 3D desenhadas. Um dos seus usos mais comum é o de mapear partes ou a totalidade de uma imagem a uma geometria, de forma a acrescentar detalhe visual sem aumentar o número de vértices e sem acrescentar complexidade à geometria (outros tipos de mapeamento incluem, por exemplo, *bump mapping* e *normal mapping*, mas que não serão explorados neste trabalho).

No contexto de OpenGL/WebGL, uma textura de duas dimensões pode resultar do carregamento de uma imagem *bitmap*, e que é carregada para um buffer, que posteriormente pode ser acedido usando duas dimensões vulgarmente identificadas como *s* e *t* (ou noutros contextos como *u* e *v*), e cujas coordenadas são normalizadas entre 0 e 1 (ver fig. 1).

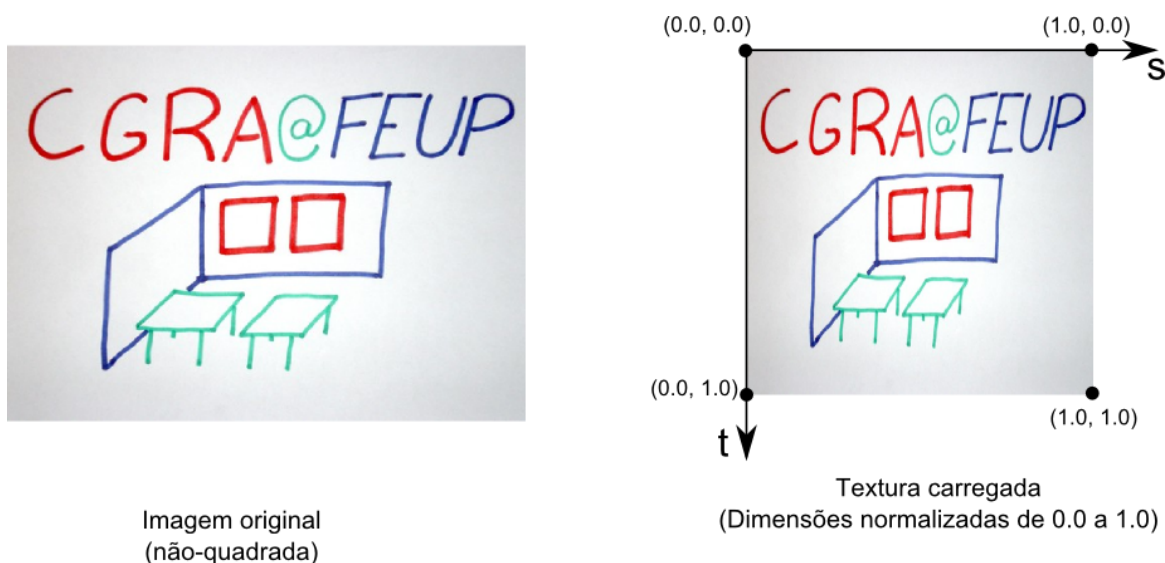


Figura 1: Imagem e correspondente textura carregada

Uma textura previamente carregada pode ser aplicada a uma dada geometria - no caso mais básico um triângulo - fazendo o mapeamento entre os vértices da geometria e os pontos da imagem que lhes estarão associados, definindo para cada vértice uma coordenada de textura (ver fig. 2, a) e b)).

Conceptualmente, podemos considerar que estamos a definir o "recorte da imagem" que será aplicado ao triângulo em questão, sendo que caso o "recorte" não tenha as mesmas proporções do triângulo original, a imagem será distorcida de acordo (ver fig. 2, c) e d)).

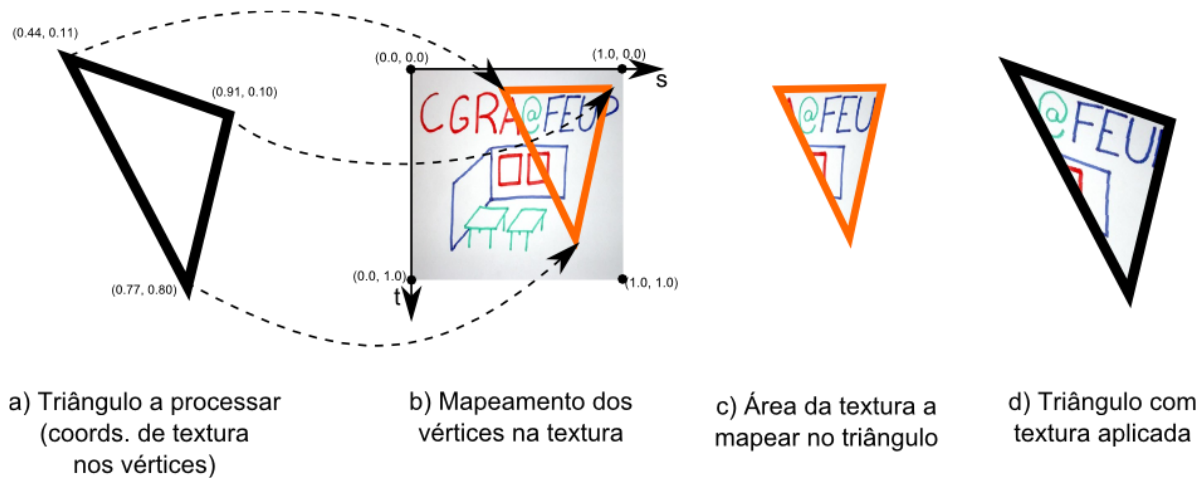


Figura 2: Mapeamento entre triângulo e textura definindo coordenadas de textura por vértice.

Neste primeiro exercício será necessário declarar coordenadas de textura nos objetos em que se pretende aplicar texturas, criar objetos da classe **CGFappearance** que contemplem texturas, e aplicá-los aos diferentes elementos da cena a texturar.

1. Edite o método **initBuffers** da classe **MyQuad**, de forma a definir coordenadas de textura para os seus quatro vértices. Para esse efeito deve acrescentar, à semelhança do que fez para as normais, um **array** de coordenadas de textura (apenas duas coordenadas por vértice):

```
this.texCoords = [
    .../ ... /
    .../ ... /
    .../ ... /
    .../ ...
];
```

Com isto, todos os objetos que usem **MyQuad** poderão ter texturas aplicadas (em particular o **MyUnitCubeQuad** e por inerência a **MyTable**).

2. Na função **init** da cena, ative a utilização de texturas:

```
this.enableTextures(true);
```

3. Crie um objeto **tableAppearance** da classe **CGFappearance** que descreva o material do tampo das mesas de madeira (pouca componente especular e baixo brilho, forte componente difusa), associe-lhe a textura **table.png** (use o método **CGFappearance.loadTexture** e tenha em atenção que o caminho para o respetivo ficheiro é relativo à pasta onde se encontra o ficheiro **index.html**). Aplique este material ao tampo de ambas as mesas. ➡

(1.3 ) (1.3 )

2. Modos de *Wrapping* de Texturas

No exemplo explorado até agora, as coordenadas de textura associadas a cada vértice encontram-se na gama normalizada de 0.0 a 1.0. No entanto, é possível indicar valores fora dessa gama, quando pretendemos, por exemplo, ter várias repetições da mesma imagem num polígono, ou mapear a totalidade da imagem apenas numa parte do polígono. A forma como os valores fora da gama [0..1] são utilizados na aplicação de uma textura é controlada definindo o modo de *wrapping*. Os modos de *wrapping* suportados variam um pouco entre versões de OpenGL, no caso do WebGL 1.0 os modos possíveis são 'REPEAT', 'CLAMP_TO_EDGE' e 'MIRRORED_REPEAT'. Na figura 3 estão ilustrados alguns exemplos de como manipular as coordenadas de textura em cada modo para obter diferentes efeitos. Note que o modo de *wrapping* pode ser diferente nas duas dimensões *s* e *t*.

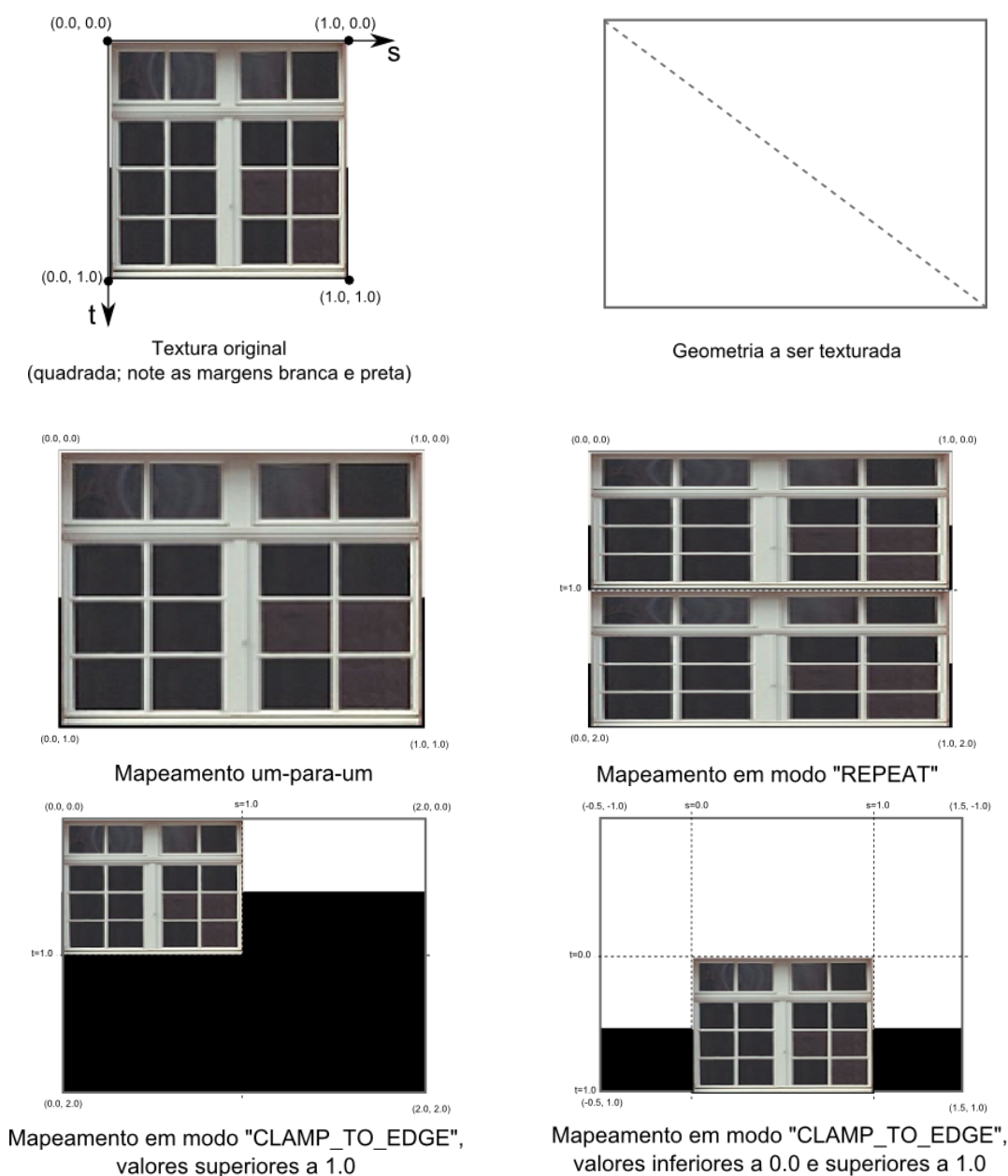


Figura 3: Aplicação de uma textura utilizando repetição ou *clamping*.

Note no modo de clamping como as margens da imagem são estendidas ao longo das zonas de coordenadas fora da gama [0..1]

1. Pretende-se que altere o construtor da classe `MyQuad`, para permitir indicar como parâmetros os valores **minS**, **maxS**, **minT**, **maxT** (mínimos e máximos de **s** e de **t**) de forma a poder parametrizar as coordenadas de textura. Por exemplo, o último caso da figura 3 poderia ser obtido por:

```
new MyQuad(scene, -0.5, 1.5, -1.0, 1.0);
```

2. Crie a `CGFAppearance` **floorAppearance** com a textura **floor.png**. Aplique-a ao chão de forma a esta se repetir 10 vezes na direção **s** e 12 vezes na direção **t**. (Nota: poderá ter de ajustar a inicialização do **MyQuad** de **floor** para esse efeito)



3. Substitua a parede da esquerda por um **MyQuad** em vez de um **Plane**. Crie a `CGFAppearance` **windowAppearance** com a textura **window.png**. Aplique-a na parede da esquerda de forma a que a janela apareça só uma vez, mas centrada na altura e na largura, usando o modo **'CLAMP_TO_EDGE'** (consulte a documentação da **CGFAppearance.setTextureWrap**). Adicione uma fonte de luz na posição da janela.

(2.3 ) (2.3 ) 

3. Partição de uma textura por várias primitivas

A flexibilidade na atribuição de coordenadas de textura permite manipular de diferentes formas a aplicação de texturas. Por exemplo, se tivermos uma superfície, constituída por vários polígonos sobre a qual queremos aplicar uma única textura, como o plano usado para os quadros, podemos fazê-lo distribuindo as coordenadas de textura de forma adequada pelos seus vértices, como ilustrado na fig. 4.

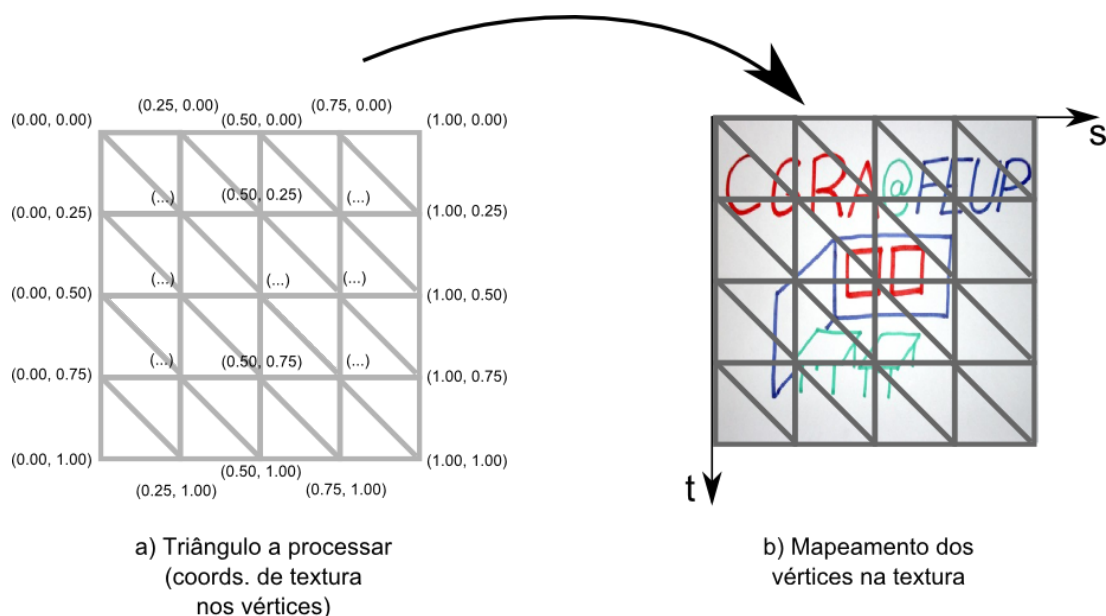



Figura 4: Aplicação de uma textura sobre várias primitivas

1. Edite a classe **Plane** fornecida com o código deste trabalho, de forma a acrescentar coordenadas de textura aos vértices de forma semelhante à ilustrada na fig. 4.
2. Crie agora dois materiais: **slidesAppearance** e **boardAppearance**. O primeiro irá descrever um ecrã projetado (representado pela textura **slides.png**) com pouca componente especular e baixo brilho, forte componente difusa. O segundo será um quadro branco (representado pela textura **board.png**) com alguma componente especular, bastante brilho e uma menor componente difusa. Aplique os materiais sobre os quadros (visto de frente, o da esquerda será o ecrã de projeção). 
3. Note que a textura tem uma proporção diferente da do quadro. Corrija esta situação ao nível da aplicação da textura, podendo para o efeito passar parâmetros adicionais ao construtor da classe Plane para permitir alterar a proporção da textura.

(3.3 ) (3.3 )



Extra: Completar a cena

Complete a cena com mais texturas e materiais. Crie colunas com o cilindro efetuado no TP3 e aplique uma textura adequada.

(extra ) (extra )

Checklist

Até ao final do trabalho deverá submeter as seguintes imagens e versões do código via Moodle, respeitando estritamente a regra dos nomes, bem como o ficheiro ident.txt com a identificação dos membros do grupo:

-  Imagens (4): 1.3, 2.3, 3.3, extra (nomes do tipo "CGFImage-tp4-TtGgg-x.y.png")
-  Código em arquivo zip (4): 1.3, 2.3, 3.3, extra (nomes do tipo "CGFCode-tp4-TtGgg-x.y.zip")