

Texture Manipulation

Miguel Costa 84924

Dany Costa 85097

Resumo: Este artigo apresenta uma aplicação de manipulação de texturas apresentadas nas faces de dois cubos distintos, de maneira a que ambos sejam manipulados independentemente permitindo assim a comparação dos resultados obtidos para cada.

Abstract: This article presents an application for texture manipulations on the faces of two similar objects, in a way that both can be independently manipulated in order to be able to compare the results of the changes made to each one of the Objects.

I. INTRODUÇÃO

No âmbito da disciplina de Computação Visual foi-nos proposto a realização de um trabalho usando a ferramenta *WEBGL*, e de acordo com as técnicas aprendidas decidimos realizar uma aplicação para Manipulação do Mapeamento e Repetição de Texturas num dado objeto (Cubo).

Esta aplicação permite-nos visualizar inicialmente dois cubos com as mesmas dimensões e com a mesma textura aplicada. A partir daí, de acordo com as vontades do utilizador, podem ser realizadas várias transformações nas texturas de cada cubo individualmente, como alteração da textura (quer por ficheiro local, quer por *link*), alteração das dimensões do cubo para obter diferentes objetos, escolha do número de faces que cada cubo terá a textura apresentada, manipulação da apresentação da mesma numa mesma face do objeto (NxM) que por sua vez permitirá a adição de mais efeitos na textura desde espelhamento e esticamento da textura na mesma face quer na vertical ou na horizontal. Podemos também colocar um efeitos no sistema de cores permitindo apenas a visualização da textura a preto e branco e o efeito *Blending* que provoca uma espécie de transparência nas texturas do objeto. No final também permitimos ao utilizador escolher o tipo de perspetiva que deseja para a visualização e rotações

automáticas em torno dos de cada um dos três eixos (xyz) individualmente para cada Objeto.

Temos também a funcionalidade de Zoom In/Out dos objetos com limite de dimensões, rotação dos mesmos manualmente com o rato e seleção do cubo a ser alterado com as teclas de setas direita e esquerda do teclado para o objeto da direita e da esquerda respetivamente.

II. DESCRIÇÃO BREVE DO CONTEÚDO DOS FICHEIROS

O projeto apresenta um conjunto de ficheiros necessários

à execução da aplicação. Nesta secção apresentamos uma breve

descrição dos ficheiros mais relevantes:

- Textures.html – Este ficheiro contém o código html necessário para apresentar o viewport e os botões num ambiente Web.
- Texture.js – Este ficheiro contém a lógica de alto nível da aplicação. É aqui que são definidas as várias funções de inicialização do ambiente, dos objetos, texturas e das funções de manipulação das texturas e movimento dos objetos.

III. ESTRUTURA DA APLICAÇÃO

Para correr a aplicação e apresentar uma cena de visualização, precisamos de inicializar o *WebGL*, um programa de Shaders, os Buffers e as respectivas Texturas.

Temos uma lista de funções para aplicar as transformações a serem realizadas nos objetos que que estão contidas num *eventListener*, e temos funções que desenhavam continuamente a respectiva cena na área de desenho e que controlam o teclado para interação com a aplicação.

No desenho da cena, colocamos um background branco, com 2 tipos de projeção de visualização dando a escolha ao utilizador e de acordo com os sinais ativos e o estado das matrizes, *buffers* e

variáveis são desenhados os objetos dentro do espaço designado pela aplicação.

Dentro do desenho dos respectivos objetos existe manipulação das matrizes e aplicação das mesmas nos *Buffers* para desenho dos vértices e das suas faces. Tendo texturas implementadas, as mesmas também serão aplicadas aos respectivos *Buffers* para o seu desenho nas faces dos polígonos e o uso de certas funções para alterar parâmetros que terão efeito nas texturas em certos acontecimentos.

Temos também uma função de animação que vai alterar os parâmetros relacionados com a rotação automática dos objetos em torno dos eixos, que serão usadas pelas funções de desenho dos Objetos.

Todas estas funções são gerais na utilização do *WebGl* e foram adaptadas para a nossa aplicação.

Para o nosso trabalho necessitamos de inicializar 3 tipos de *Buffers*, um para os pontos do cubo, um para as texturas e o terceiro para a formação das faces dos polígonos através dos índices dos pontos a desenhar.

Na inicialização das Texturas temos duas variáveis guardadas num array responsáveis pelo armazenamento relacionado com as texturas de cada um dos Polígonos.

IV.TEXTURAS

Como mencionado acima o objetivo da aplicação é poder manipular o mapeamento e repetição de texturas de um dado objeto neste caso o cubo.

Neste tópico iremos retratar os vários métodos utilizados e a sua respectiva implementação.

IV.I. Repetição de imagens por face

Com esta funcionalidade permitimos ao utilizador poder escolher o número de vezes que a textura é apresentada em cada face do objeto ($N \times M$). O utilizador pode escolher o cubo em que pretende aplicar as alterações pretendidas e atribuir as repetições (por coluna e linha) para o cubo escolhido. No exemplo que se segue no cubo à esquerda temos uma repetição de 5 faces por linha e 3 por coluna enquanto que no cubo da direita podemos observar o inverso.

Em termos de implementação limitamos o número a quantidade de repetições por cada linha/coluna de [1 a 5] de cada face.

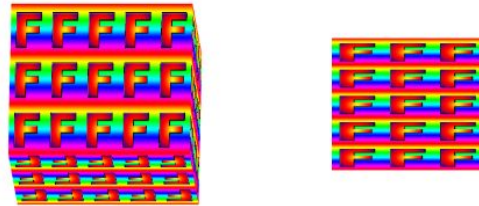


Fig 1 - Exemplo de 5*3 imagens por face e de 3*5 imagens por cada face

Nestes próximos capítulos de repetições pegamos na repetição da imagem em cada face e aplicamos alguns efeitos. Temos o exemplo do efeito de espelho e de *clamp to edge* (que é uma extensão da própria textura aos limites da face do objeto) que estão exemplificados nos próximos exemplos.

Em termos de implementação alteramos o mapeamento dos parâmetros de apresentação das texturas usando certos valores que permitem o seu correto funcionamento. Na figura número 2 podemos observar do lado esquerdo um cubo com o uso do efeito de espelho, em que cada par de texturas na horizontal age como se fossem espelho do seu par e do lado direito temos o uso do *clamp to edge* em que observamos a extensão da textura aos limites das faces do cubo na horizontal. Na figura 3 observamos os mesmos efeitos mas em termos verticais.

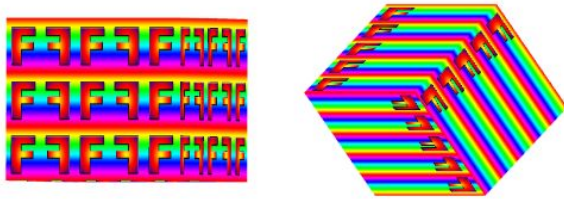


Fig 2 - Horizontal Mirror e Horizontal Clamp to Edge

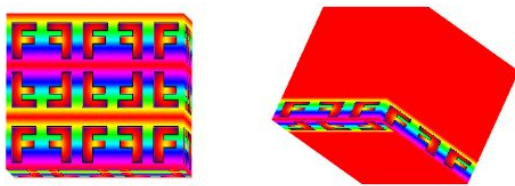


Fig 3 - Vertical Mirror e Vertical Clamp to Edge

IV.II Blending

O *Blending* é um efeito que aplica um certo nível de transparência nos objetos, permitindo a visualização das 6 faces do cubo. Esta função acaba por ser bastante simples de implementar, de acordo com os objetos que têm o efeito aplicado dá-se o *enable* deste mesmo efeito e *disable* do efeito de profundidade.

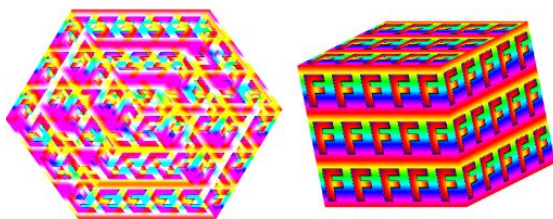


Fig 4 -Blending on vs Blending off

IV.III Luminance

A luminância é uma textura em que só existe um valor por pixel e cada um é repetido para cada um dos canais R,G,B.

Quando é selecionado esta textura, enquanto que um dos objetos permanece com os seus valores inalterados, no selecionado ativamos o efeito resultando numa alteração no sistema de cores que permite visualizar as texturas apenas a preto e branco. Como podemos visualizar na figura, do lado esquerdo temos um cubo com o efeito ativo e do lado direito desativado para podermos comparar.

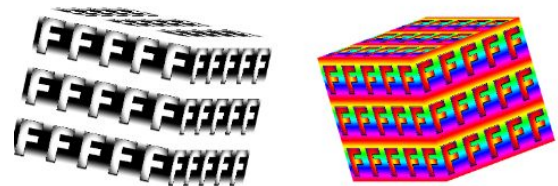


Fig 5- Luminance on vs Luminance off

V. OUTRAS FUNCIONALIDADES

Para além dos efeitos de mapeamento e manipulação das texturas que foram apresentadas anteriormente também implementamos outras funcionalidades tais como a rotação dos objetos (de forma independente) em cada um dos eixos (xyz) ou de forma livre , através do movimento do rato. Para além disso podemos também modificar o tamanho dos objetos (sempre de forma independente), também é possível alterar o tipo de projeção e por fim podemos também carregar uma textura através de um link ou através de um ficheiro local desde que a imagem cumpra os requisitos mínimos (mesmas dimensões de largura e altura).

V.I Rotação

Para além da rotação livre, através do rato, também podemos fazer rotações ao nível de cada eixo xyz, em que temos 1 botão para cada eixo individual tal como podemos observar na figura 6.

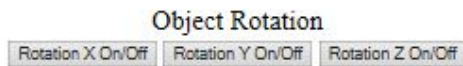


Fig 6 - Botões das Rotações Automáticas dos Objetos

V.II Tamanho do Objeto

Esta função permite ao utilizador alterar o tamanho do modelo 3D ao nível de cada uma das suas direções, quer em x, y e em z. No exemplo a seguir temos uma demonstração em que à esquerda temos um cubo normal de aresta=3 ($s_x=3$, $s_y=3$, $s_z=3$) e à direita ao alterar apenas o valor de y obtemos um paralelepípedo em vez de um cubo. Os tamanhos estão limitados de [1 a 3].



Fig 7 - Objetos independentes com diferentes tamanhos aplicados

V.III Tipo de projeção

Na nossa aplicação permitimos ao utilizador escolher o modo de visualização que pretende para observar os dois objetos apresentados. Temos a projeção ortogonal que permite uma visualização do estilo 2D num ecrã, sem noção de profundidade, e temos também a projeção em perspectiva que permite observar o objeto ao estilo 3D, desta vez com noções de profundidade.

Na Figura 8 podemos analisar ambos os modos de visualização para comparar as diferenças entre ambos, tendo à esquerda a projeção ortogonal e à direita em perspectiva.

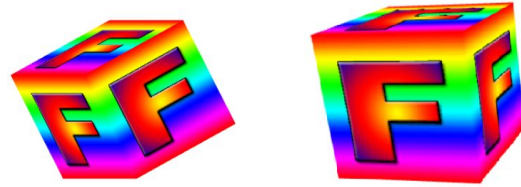


Fig 8 - Projeção Ortogonal vs Perspectiva

V.IV Carregamento de imagens

Como foi supramencionado na nossa aplicação é possível carregar diferentes tipos de imagens quer através de um link quer através de um ficheiro local.

Ao nível de implementação, temos diferentes cuidados no que toca a upload por link ou por upload local. Para o upload por link funcionar devidamente há cuidados necessários pois é necessário requisitar a utilização da imagem. Para tal fazemos esse mesmo requisito à entidade *CORS* que assim nos permitirá o uso de imagens diretamente da Internet. Já por ficheiros locais não é necessário qualquer tipo de cuidado especial pois são ficheiros que o próprio utilizador possui.

VI.CONCLUSÕES

A realização deste trabalho permitiu integrar diversos conhecimentos aplicados nas aulas, tal como a necessidade de exploração dos mesmos e integração de diversas componentes que permitem a interação e um nível bastante razoável em termos de interação com texturas em objetos 3D. Achemos que atingimos o objetivo a que nos propusemos no início do trabalho, conseguimos pegar no que aprendemos e aplicar esses conhecimentos na realização do trabalho. Obviamente que houve alguma procura de conhecimentos Online, certas vezes para ter mais ideias para funcionalidades no trabalho.

Tivemos algumas dificuldades inicialmente em pôr as funcionalidades a correr devidamente para apenas um objeto, mas mais tarde, com a necessidade de apresentar dois objetos distintos

houve problemas maiores, tanto no controlo do manipulamento individual como na execução das funções, mesmo já estando a funcionar apenas para um dos objetos. Apesar destas dificuldades, no final conseguimos ultrapassar tudo e resolver todos os problemas que tínhamos e sentimos que conseguimos cumprir o que nos fora proposto neste trabalho e que houve uma boa consolidação dos conhecimentos aprendidos e usados ao longo da realização deste trabalho. Foi uma experiência bastante apelativa, deu bastante motivação e foi algo enriquecedor em termos de conhecimentos.

VII.REFERÊNCIAS

- [1] WebGL Fundamentals: Uma página Online dedicada a apresentar tutoriais e ferramentas de utilização para o *WebGL*
- [2] <http://sweet.ua.pt/jmadeira/WebGL/> :
Plataforma da disciplina de Computação Visual administrada pelo professor Joaquim Madeira da Universidade de Aveiro que contém Documentos escritos pelo mesmo e pela professora Beatriz Sousa Santos sobre técnicas e ferramentas de *WebGL*