

Atividade Prática 4 - Mapeamento e Filtragem de Texturas

João Victor Rodrigues Galvão - 20190172620

Yyson Nunes Figueiredo - 20190019716

Introdução

O Trabalho desenvolvido aqui teve como objetivo realizar uma comparação entre os filtros de texturas estudados em sala e as suas respectivas utilidades em situações de magnificação e minificação. Para Tal, os filtros oferecidos pela biblioteca do threeJS (**NearestFilter**, **LinearFilter**, **NearestMipmapNearestFilter**, **NearestMipmapLinearFilter**, **LinearMipmapNearestFilter** e **LinearMipmapLinearFilter**) foram utilizados.

Fenômeno de Magnificação

Em primeiro plano, para a análise da Magnificação, foi utilizada uma imagem de dimensões 125x107 **texels**. A figura 1 ilustra um cubo o qual foi aplicado o filtro **Nearest neighbor** e, por tal razão, temos um resultado bastante pixelado, fruto da magnificação causada pela baixa resolução.

Nearest neighbor



Figura 1: Aku Aku (Game: Crash Bandicoot) - magFilter: NearestFilter

Por outro lado, A figura 2 ilustra o mesmo cubo com a filtragem sendo a Bilinear. Percebe-se que o "pixelamento" diminuiu bastante, contudo o resultado final gerou uma textura mais borrada.

Bilinear



Figura 2: Aku Aku (Game: Crash Bandicoot) - magFilter: LinearFilter

Fenômeno de Minificação

Na análise da Minificação, foi utilizada uma imagem de dimensões 2048x2048 **texels**.

Filtro de MipMapping

O filtro de mipmapping foi variado de acordo com as opções do **three.js**, nesta análise o parâmetro **texture.anisotropy** foi mantido como 1.

As figuras 3, 4, 5 e 6 mostram o uso dos filtros **NearestMipmapNearestFilter**, **NearestMipmapLinearFilter**, **LinearMipmapNearestFilter**, **LinearMipmapLinearFilter**.

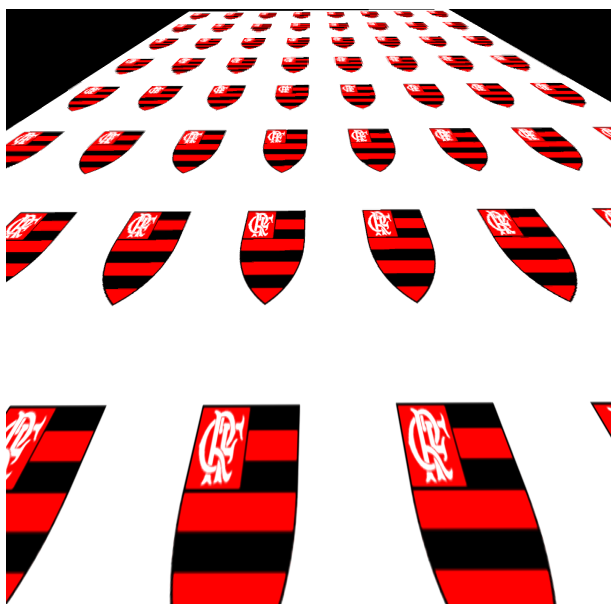


Figura 3: NearestMipmapNearestFilter

O filtro **NearestMipmapNearestFilter** escolhe o mipmap que mais se aproxima do tamanho do pixel a ser texturizado, esse filtro usa o mesmo critério do **NearestFilter**. Como resultado, observamos uma imagem

com uma falta de definição onde ocorre a minificação, principalmente nas linhas finais, com brasões distorcidos e mal definidos.

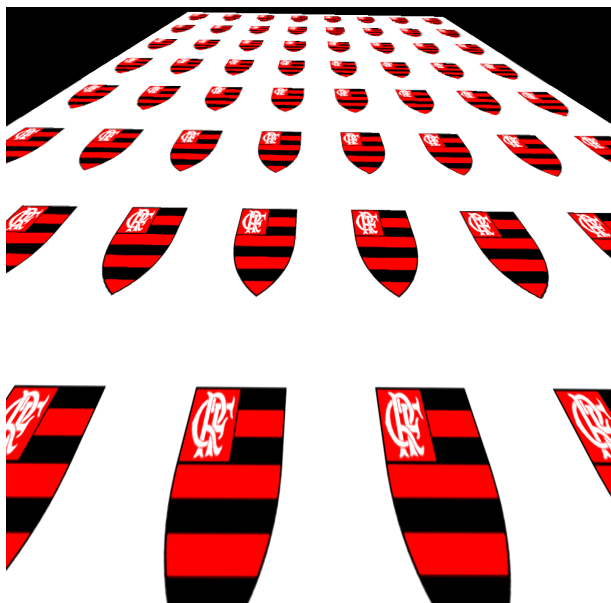


Figura 4: NearestMipmapLinearFilter

Já o filtro **NearestMipmapLinearFilter**, escolhe dois mipmaps que mais se aproximam do tamanho apropriado para texturizar o pixel, esse filtro também usa o critério do **NearestFilter** para escolher a textura de cada mipmap. A textura final é uma média ponderada das duas texturas anteriormente selecionadas. Podemos observar que devido ao uso de duas texturas e da média dela, o resultado é um pouco mais suave que o anterior.

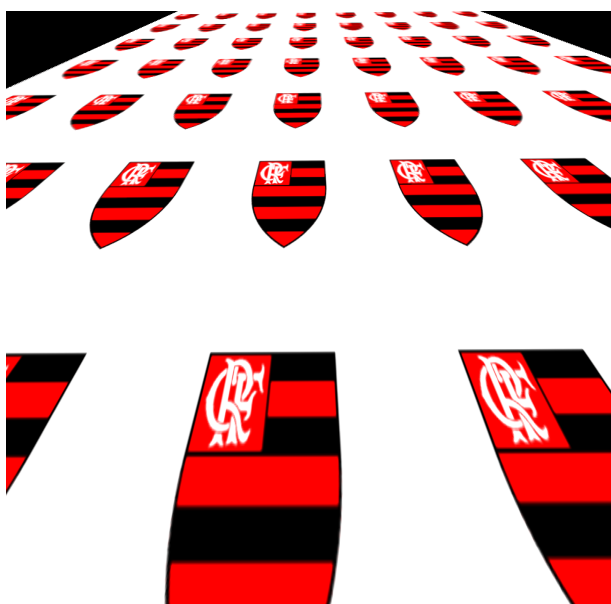


Figura 5: LinearMipmapNearestFilter

O **LinearMipmapNearestFilter** escolhe o mipmap que mais se aproxima do tamanho do pixel a ser texturizado, mas nesse caso utiliza o critério do **LinearFilter** para selecionar o valor da textura. Aqui, assim

como na comparação entre o bilinear e nearest, podemos ver a diminuição do pixelamento e a suavização da imagem.

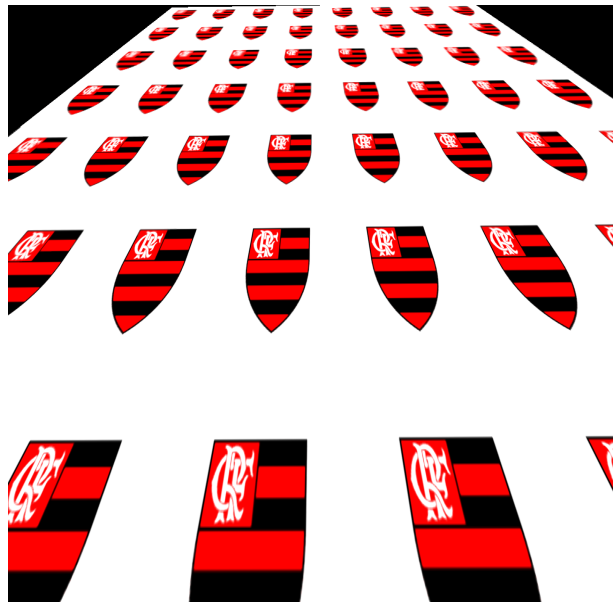


Figura 6: LinearMipmapLinearFilter

Já o **LinearMipmapLinearFilter**, escolhe dois mipmaps que mais se aproximam do tamanho apropriado para texturizar o pixel, em seguida aplica o **LinearFilter** para definir um valor de textura para cada mipmap, a textura final será a média ponderada dos dois valores encontrados. Essa filtragem também é chamada de **trilinear**. Podemos observar que esse filtro exibe o melhor resultado, por mais que ele seja mais custoso computacionalmente, em relação aos anteriores, visualizamos a imagem com uma maior definição.

Filtro de Anisotropia

Ainda no contexto da minificação, a análise a seguir foi feita levando em consideração os parâmetros padrões de magnificação e minificação do ThreeJs(LinearFilter e LinearMipmapLinearFilter, respectivamente). Nesse âmbito, a metodologia consistiu em variar o parâmetro **texture.anisotropy** de modo a atribuí-lo com os valores: 2,4,8 e 16 (valores estes citados em sala de aula).

Em primeiro plano, com a anisotropia igual a 2 (2 amostras x 8 consultas implicando em até 16 consultas), pôde-se perceber um borramento na fronteira superior da figura.

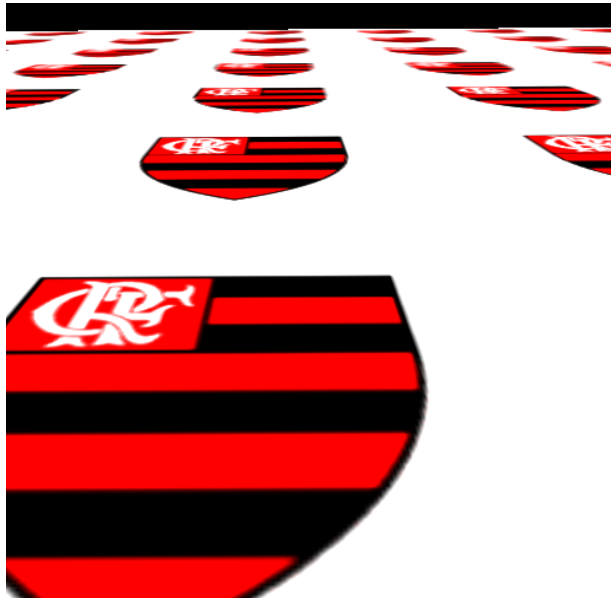


Figura 7: texture.anisotropy: 2

Aumentando a anisotropia para 4 (4 amostras x 8 consultas implicando em até 32 consultas) houve um melhoramento da qualidade da imagem, porém ainda com leves borramentos.

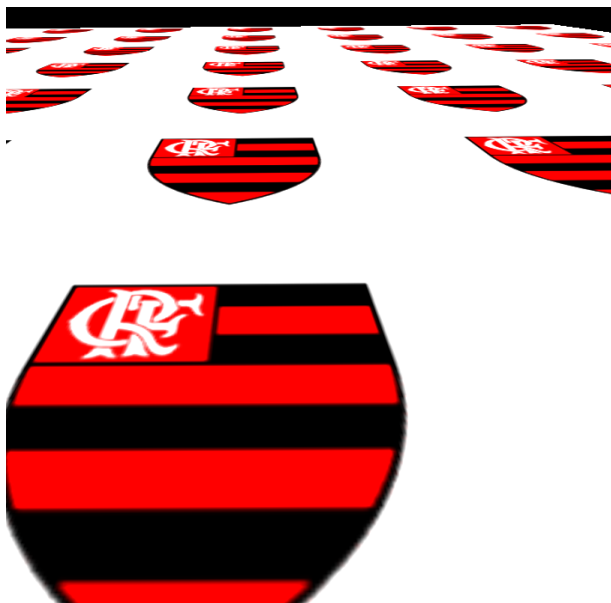


Figura 8: texture.anisotropy: 4

A partir da anisotropia em 8 (8 amostras x 8 consultas implicando em até 64 consultas), a qualidade da imagem teve um nítido melhoramento, de modo que não houve muita diferença entre a anisotropia ser 8 ou 16 (com 16 amostras x 8 consultas podendo implicar em até 128 consultas).

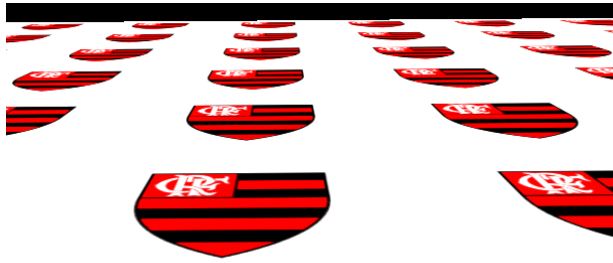


Figura 9: texture.anisotropy: 8

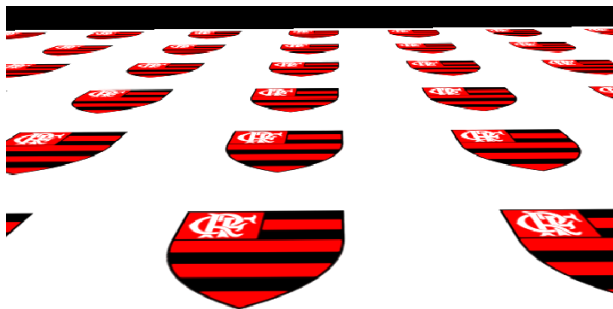


Figura 10: texture.anisotropy: 16

Referências

<https://sbcode.net/threejs/mipmaps>

<https://threejsfundamentals.org/threejs/lessons/threejs-textures.html#loading>