

---

## Introdução à Computação Gráfica

Emmanuella Faustino Albuquerque  
20170002239

### Atividade Prática 4: Mapeamento e Filtragem de Texturas 16 de novembro de 2021

#### VISÃO GERAL

Nesta atividade foram analisados os diferentes tipos de filtragem disponíveis no Three JS, como Nearest Neighbor, Bilinear, Mipmapping, e a filtragem Anisotrópica, com o intuito de utilizá-las para gerar resultados que diminuam os problemas de magnificação e minificação.

#### ESTRATÉGIAS

##### Definindo os Tipos de Filtragem

Primeiramente foi preciso relacionar todos os tipos de filtragem de textura vistos nas aulas, com as identificações que o Three JS utiliza para representá-los.

Nome da Filtragem	Nome da Filtragem no Three JS
Nearest Neighbor	NearestFilter
Bilinear	LinearFilter
Mipmapping	LinearMipmapLinearFilter

O LinearMipmapLinearFilter representa o caso de Mipmapping clássico visto em aula, que realiza uma filtragem bilinear nos dois níveis utilizados e realiza uma interpolação linear entre esses dois níveis.

##### Definindo o conjunto de imagens

Quando a Magnificação ocorre, temos que vários pixels recebem a mesma cor de um único texel, com isso, obtemos uma imagem final pixelada. Quando a Minificação ocorre, temos que um único pixel recebe a cor de vários texels, por esse motivo, obtemos ruído na imagem final.

Logo, para gerar o efeito da Magnificação, foram escolhidas imagens com a resolução mais baixa, pois, assim um único texel irá cobrir uma grande região de pixels. Já para a Minificação, foram escolhidas imagens com alta resolução, pois assim, um único texel irá cobrir uma pequena região.

**Observação:** Na seção Magnificação está sendo usado como padrão para casos de minificação o nearest neighbor e na seção Minificação está sendo usado como padrão para casos de magnificação o nearest neighbor. Portanto, quando não estiver indicado o caso, temos que o filtro utilizado se refere ao caso referenciado pela seção.

---

## RESULTADOS

### Magnificação

Como é possível observar abaixo, temos um efeito pixelado gerado pela filtragem Nearest Neighbor, e o borramento na filtragem Bilinear, gerado pela interpolação das cores. Em função da magnificação exagerada, temos um borramento grande.

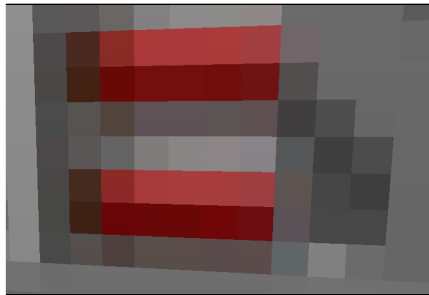


Imagem 1: Uma parte do Portão do Doom com filtragem Nearest Neighbor

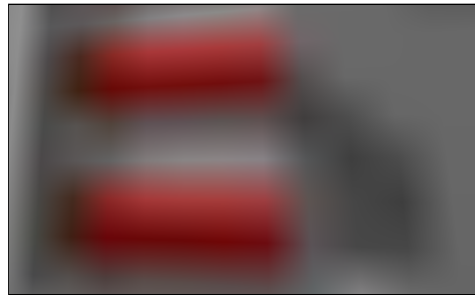


Imagem 2: Uma parte do Portão do Doom com filtragem Bilinear

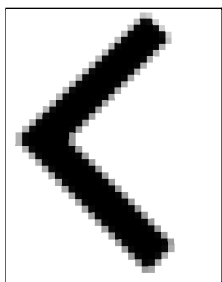


Imagem 3: Seta com filtragem Nearest Neighbor

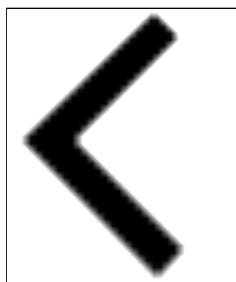


Imagem 4: Seta com filtragem Bilinear

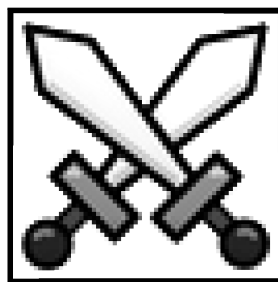


Imagem 5: Espada com filtragem Nearest Neighbor



Imagem 6: Espada com filtragem Bilinear

Nos exemplos acima, foi utilizada a geometria plana (PlaneGeometry) e no exemplo abaixo, a geometria do Cubo (BoxGeometry) no Three JS. Com o objetivo de observar melhor o fenômeno da magnificação foi utilizado o OrbitControls para melhor posicionamento e visualização da textura na geometria em questão.



Imagem 7: Cubo com textura de um Deck de Madeira e com filtragem Nearest Neighbor



Imagem 8: Cubo com textura de um Deck de Madeira e com filtragem Bilinear

### Minificação

Como é possível observar abaixo, obtemos ruído no fundo da cena utilizando a filtragem Nearest Neighbor, e borramento quando utilizando a filtragem Mipmapping.

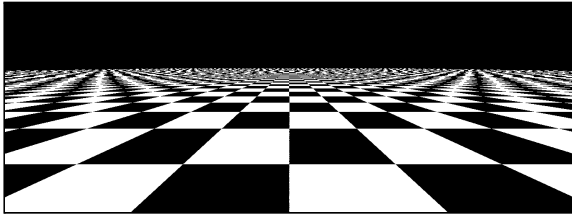


Imagem 9: Textura do padrão xadrez  
com filtragem Nearest Neighbor

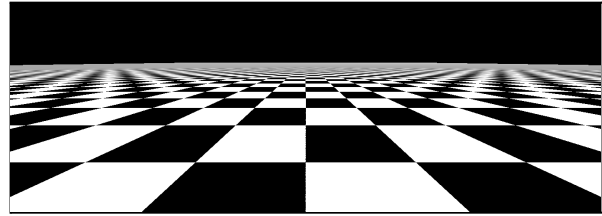


Imagem 10: Textura do padrão xadrez  
com filtragem Mipmapping

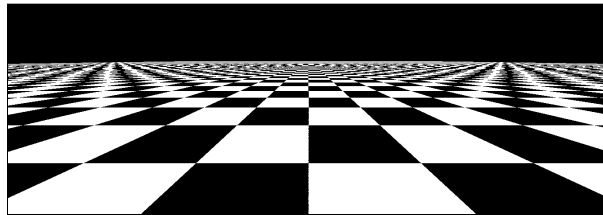


Imagem 11: Textura do padrão xadrez  
com filtragem Anisotrópica de 16 samples

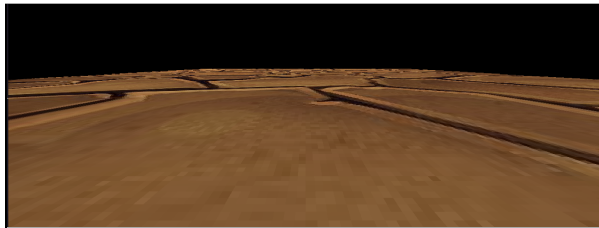


Imagem 12: Textura de um Plano Rochoso  
com filtragem Nearest Neighbor

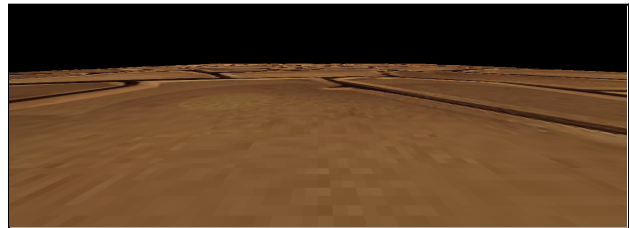


Imagem 13: Textura de um Plano Rochoso  
com filtragem Bilinear

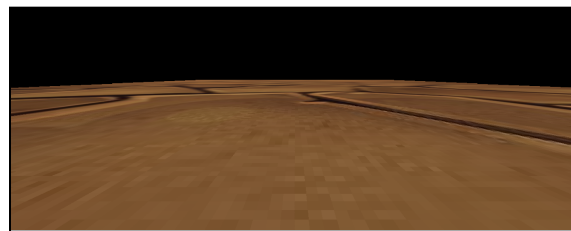


Imagem 14: Textura de um Plano Rochoso  
com filtragem Mipmapping

Nas imagens abaixo, será utilizado (min), para se referir ao fenômeno da minificação e (mag) para o fenômeno da magnificação.

### Variações do Mipmapping

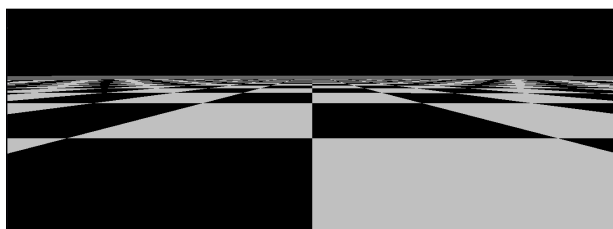


Imagem 15: Textura do padrão xadrez  
(mag) Nearest Neighbor, (min) NearestMipmapNearestFilter

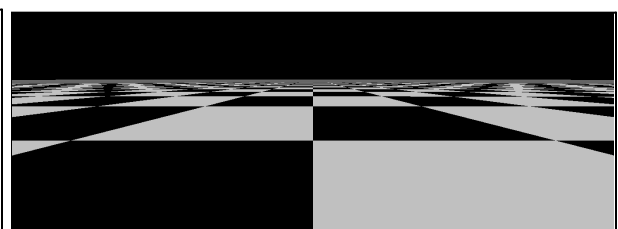


Imagem 16: Textura do padrão xadrez  
(mag) Nearest Neighbor, (min) NearestMipmapLinearFilter



Imagem 17: Textura do padrão xadrez  
(mag) Nearest Neighbor, (min) LinearMipmapNearestFilter

\*\* Na filtragem da Imagem 17, obtemos um resultado mais borrado, pois é escolhido um mip e depois é feita a interpolação de cor de 4 texels. Já na da Imagem 16, obtemos menos borramento, pois ela utiliza 2 mips, e interpola a cor de 2 texels. Por fim, a filtragem da Imagem 15, escolhe apenas 1 mip e seu valor de cor mais apropriado para o pixel. Logo, como é possível observar, quanto maior a quantidade de mips utilizados(se necessário), mais nítida a textura fica.

## Variações do fator de Anisotropia

Quanto maior o fator de anisotropia, menos borramento obtemos, pois, temos uma maior quantidade de mips sendo escolhidos e com isso uma maior quantidade de amostras sendo consultadas para assim gerar através da interpolação o melhor valor de cor para aquele pixel.

Utilizando a função `renderer.getMaxAnisotropy()` foi possível verificar a anisotropia máxima disponível para o meu Hardware, que no caso é de 16 amostras.

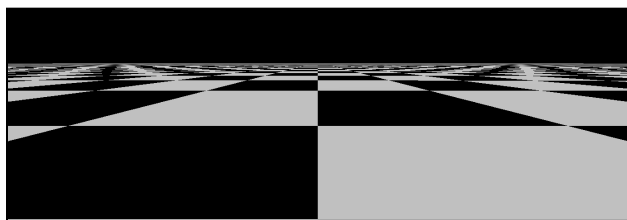


Imagem 18: Textura do padrão xadrez  
(mag) Nearest Neighbor, (min) Anisotrópica de 8 samples

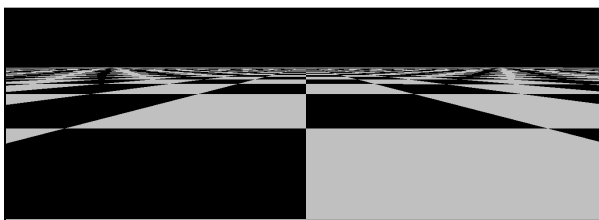


Imagem 19: Textura do padrão xadrez  
(mag) Nearest Neighbor, (min) Anisotrópica de 16 samples

**Observação:** De (2 até 16 samples) obtemos um resultado menos borrado do que um mipmap básico, ao custo de mais amostras de textura sendo usadas.

## Outros Exemplos

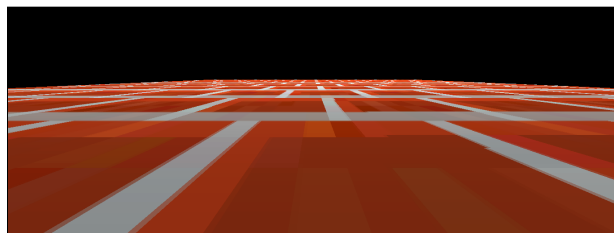


Imagem 20: Textura de tijolos  
(mag) Nearest Neighbor, (min) Nearest Neighbor

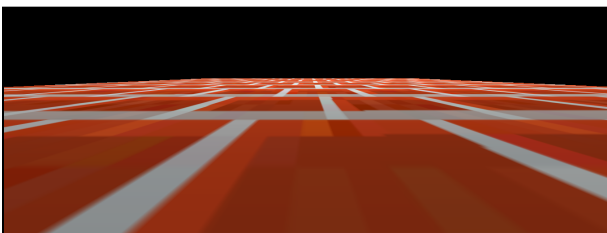


Imagem 21: Textura de tijolos  
(mag) Bilinear, (min) Anisotrópica de 16 samples

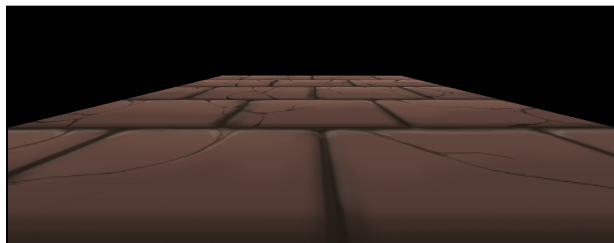


Imagem 22: Textura do telhado cartoon  
(mag) Nearest Neighbor, (min) Nearest Neighbor

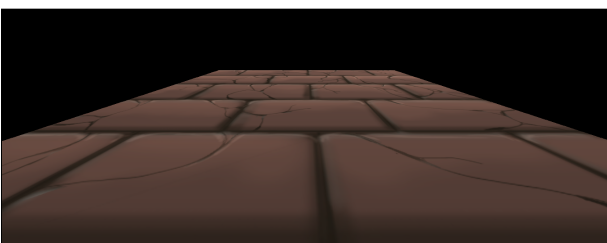


Imagem 23: Textura do telhado cartoon  
(mag) Bilinear, (min) Anisotrópica de 16 samples

---

## REFERÊNCIAS BIBLIOGRÁFICAS

[1] Three.js Textures. ThreeJS Fundamentals. Disponível em:

<https://threejsfundamentals.org/threejs/lessons/threejs-textures.html>. Acesso em: 12 de novembro de 2021.

[2] Texture. Threejs. Disponível em: <https://threejs.org/docs/#api/en/textures/Texture.anisotropy>.

Acesso em: 14 de novembro de 2021.

[3] Texture Mipmaps. Sbcode. Disponível em: <https://sbcode.net/threejs/mipmaps>. Acesso em: 15 de novembro de 2021.