

ATIVIDADE PRÁTICA 4 - MAPEAMENTO E FILTRAGEM DE TEXTURAS

GDSCO0051 - Introdução à Computação Gráfica - 2021.1

Data de entrega: 16/11/2021, 23h59min.

1 Atividade

Nesta atividade os alunos realizarão uma análise comparativa dos diversos tipos de filtros de texturas estudados em aula. Para isso, os alunos deverão inicialmente gerar um conjunto de imagens que ilustrem os fenômenos de *magnificação* e *minificação*. Em um segundo momento, os alunos renderizarão novamente estas imagens, porém agora com o uso de diferentes filtros de texturas. Os resultados finais obtidos deverão ser apresentados e discutidos.

2 Objetivo

O objetivo deste trabalho é familiarizar os alunos com os conceitos de mapeamento e filtragem de texturas utilizados em computação gráfica.

3 O *Framework*

O trabalho deve ser desenvolvido em JavaScript utilizando-se, como ponto de partida, o *framework* disponível no endereço:

<https://codepen.io/ICG-UFPB/pen/BaRqvQR>

O template fornecido com este exercício renderiza um cubo com textura, como ilustrado na Figura 1.

A imagem de textura apresentada na Figura 1a, originalmente em formato PNG, foi codificada em base 64 (<https://en.wikipedia.org/wiki/Base64>) e embutida, na forma de uma *string*, no próprio código JavaScript. O template contém uma segunda textura, também codificada em base 64, disponível para renderização, como pode ser observado na Figura 2.

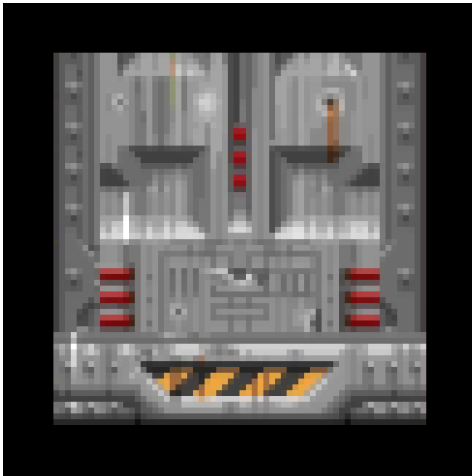
Neste template se optou pelo uso de texturas codificadas em base 64 pelo fato desta abordagem ser autocontida. Entretanto, o `three.js` disponibiliza outros métodos para a carga de texturas como, por exemplo, os *links* para imagens externas.

A geometria do cubo, bem como as duas texturas, foram disponibilizados no template para conveniência, estando os alunos livres para utilizarem outras geometrias e imagens de textura. O filtro de textura utilizado originalmente no template, tanto na magnificação quanto na minificação, é o *nearest neighbor*.

4 Desenvolvimento

Esta atividade consiste na realização de uma análise comparativa entre os diversos tipos de filtros de texturas estudados em aula. Para isso, os alunos deverão construir um conjunto de situações de *rendering* que explorem as diversas capacidades e limitações dos diferentes filtros de texturas.

A análise deve envolver os seguintes filtros estudados em aula:



(a) Textura de um portão do jogo Doom (disponível em <https://twitter.com/romero/status/542945951720038400>). Resolução: 64x64 texels.



(b) Cubo com a textura do portão aplicada em cada face.

Figure 1: Exemplo utilizando a textura do portão do jogo Doom.

- *Nearest neighbor*
- Bilinear
- *Mipmapping*, incluindo suas diversas variações disponíveis no `three.js`
- Anisotrópico, incluindo os diferentes fatores de anisotropia disponíveis no `three.js`

Uma boa introdução ao sistema de mapeamento de texturas do `three.js` está disponível em <https://threejsfundamentals.org/threejs/lessons/threejs-textures.html>.

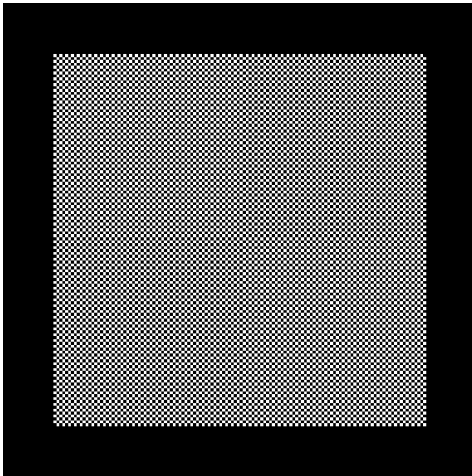
Apesar de apresentar uma ótima introdução ao sistema de mapeamento de texturas do `three.js`, o link acima não aborda a questão de anisotropia. A filtragem anisotrópica, no `three.js`, é controlada através da propriedade `anisotropy`, e maiores detalhes sobre esta propriedade podem ser vistos aqui: <https://threejs.org/docs/#api/en/textures/Texture.anisotropy>.

O site do `three.js` também contém um *demo* sobre anisotropia, disponível em: https://threejs.org/examples/webgl_materials_texture_anisotropy.html.

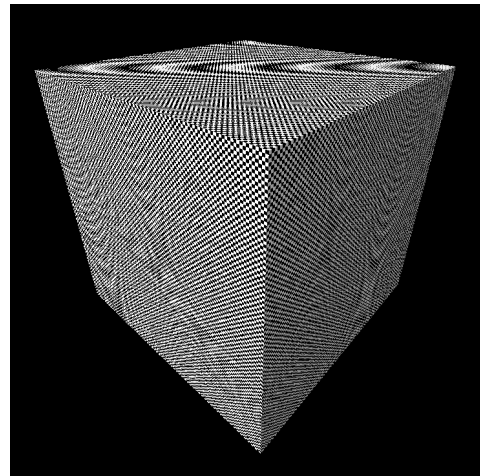
5 Entrega

Os trabalhos devem ser entregues, via atividade específica do SIGAA, até as **23 horas e 59 minutos** do dia **16/11/2021**. A entrega consistirá em um arquivo compactado (*i.e.* ZIP) contendo um relatório e o código fonte:

1. Relatório no formato PDF, contendo:
 - (a) Nome e número de matrícula do(s) alunos(s).
 - (b) Um parágrafo que descreva a atividade desenvolvida.
 - (c) Breve explicação das estratégias adotadas pelo aluno na resolução da atividade.
 - (d) *Printscreens* e discussão dos resultados gerados, dificuldades e possíveis melhoras.



(a) Textura *checkerboard*. Resolução: 512x512 texels. Cada quadrado (preto ou branco) da imagem de textura tem dimensões 4×4 texels.



(b) Cubo com a textura *checkerboard* aplicada em cada face.

Figure 2: Exemplo utilizando a textura *checkerboard*.

- (e) Referências bibliográficas.
- (f) O aluno pode, se assim desejar, disponibilizar seu código fonte também em um repositório *online* (e.g. codepen.io, jsfiddle.net, etc.) e incluir o *link* para este respositório em seu relatório. Entretanto, observa-se que esta disponibilização do código fonte em sites é opcional, não vale nota, e não substitui o envio do código fonte pelo SIGAA.

2. Arquivo contendo o código fonte em JavaScript.

Não serão aceitos trabalhos enviados por outro meio que não o SIGAA.

Este trabalho pode ser desenvolvido em duplas.

Importante:

1. Não serão aceitos trabalhos enviados por outro meio que não o SIGAA.
2. Trabalhos entregues até às **23h e 59min** do dia **16/11/2021** serão contabilizados como presença em aula. Trabalhos que **não forem entregues** até às **23h e 59min** do dia **16/11/2021** serão contabilizados como **falta em aula**, e receberão **nota zero**.