



Universidade Federal da Bahia - UFBA

Instituto de Matemática - IM

Departamento de Ciência da Computação - DCC

Curso de Bacharelado em Ciência da Computação

MATA65 - Computação Gráfica

Período: 2018.1

Data: 10/07/2017.

Prof. Antonio L. Apolinário Junior

Estagiário Docente: Rafaela Alcântara

Roteiro do Laboratório 9 – Mapeamento de Textura e Técnicas de Renderização Baseadas em Mapeamento

Objetivos:

- Reforçar os conceitos básicos de mapeamento de textura;
- Entender como o mapeamento de texturas pode ser feito com a biblioteca *Three.js*;
- Verificar como as técnicas de renderização baseadas em mapeamentos podem aumentar o realismo dos algoritmos de iluminação locais;
- Compreender como as técnicas de mapeamento são implementadas na biblioteca *Three.JS*.

Conceitos básicos:

Mapeamento de texturas consiste na projeção de uma imagem de um material sobre um objeto, com o objetivo de simular materiais heterogêneos na aparência do objeto. Para tanto, é necessário definir uma função de mapeamento que leve os pontos da imagem para os pontos da superfície do objeto.

Na biblioteca **Three.js**, o carregamento de uma imagem pode ser feito a partir do método `load` do tipo **THREE.TextureLoader**, que recebe como parâmetro a localização da imagem no diretório de arquivos e retorna a imagem para armazenamento em uma variável do tipo **var**. O mapeamento da textura sobre um objeto pode ser feito a partir do atributo `map` do material do objeto, que deve receber como parâmetro justamente a textura que será mapeada.

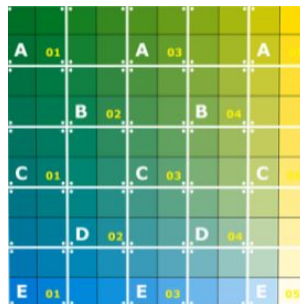
A **Three.JS** fornece ainda, associado aos loaders de modelos geométricos, a possibilidade de ler em modelo e texturas associadas. No caso específico dos modelos que temos utilizado, no formato **.OBJ**, temos um arquivo associado, com extensão **.MTL**, que descreve o material dos objetos cuja geometria e outras propriedades são descritas no **.OBJ**. Os mapeamentos utilizados no modelo são descritos no arquivo **.MTL**, e são carregados a partir de um outro loader da biblioteca **Three.JS**: o **MTLLoader**. Dessa forma, é possível fazer o carregamento “automático” do modelo geométrico e de suas características visuais através dos loaders específicos **OBJ** e **MTL**.

Um exemplo de uma especificação de um arquivo **.MTL** é dado a seguir. Pode-se perceber que os parâmetros de iluminação **Kd**, **Ka**, **Ks** e **Ns** são definidos para esse material de nome **pattern**, bem como o nome da textura (imagem) que será utilizada para mapeamento do parâmetro **Kd**.

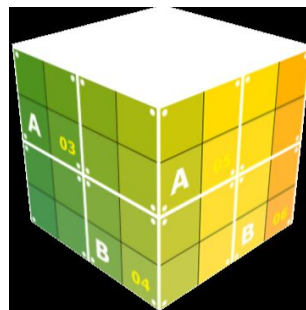
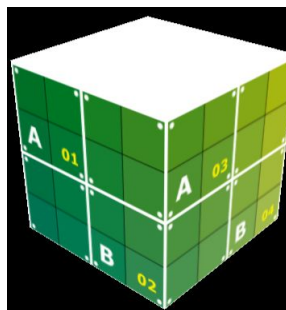
```
newmtl pattern
illum 4
Kd 1.0 1.0 1.0
Ka 0.2 0.2 0.2
Ks 0.0 0.0 0.0
Ns 10.0
map_Kd ash_uvgrid01.jpg
```

Roteiro:

1. Baixe do Moodle os códigos fonte e as dependências base para esse Laboratório. Descompacte no diretório que será visível pelo servidor *web*.
2. Configure o servidor *web*¹ e execute o exemplo desse Laboratório.
3. Abra o exemplo deste laboratório e analise como é possível fazer mapeamento de texturas com o **Three.js**.
4. Abra o arquivo **cube.obj** e verifique como as coordenadas de texturas influenciaram no mapeamento da textura sobre o objeto.
5. Modifique o arquivo **cube.obj** de forma que cada face do cubo tenha a seguinte aparência:

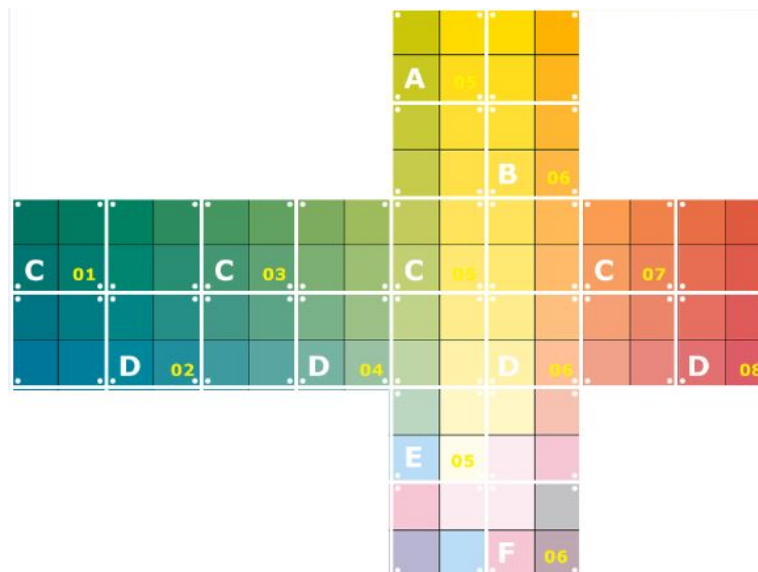


6. Modifique o arquivo **cube.obj** de forma que toda a faixa que vai de A a B da textura seja mapeada nas faces laterais do cubo. As faces de cima e de baixo do cubo devem ser mantidas na cor branca. Um exemplo da aparência resultante pode ser vista na figuras abaixo



¹ necessário para o ambiente Windows. Utilize o programa USBWebServer (<http://www.usbwebserver.net/en/>) disponível no Moodle

7. Modifique o arquivo `cube.obj` que um recorte da textura seja mapeada sobre o cubo como se fosse um dado. Para tanto, baseie o mapeamento no seguinte recorte.



8. Modifique o arquivo `cube.obj` de forma que uma segunda textura (`lena.png`) seja mapeada de forma alternada em cada face do cubo.
9. Abra a página `Earth.html` e analise o código `Earth.js`. Entenda como o processo de mapeamento de textura “automático” do `Three.JS` ocorre. Modifique esse exemplo para que uma camada de nuvens seja incluída sobre o planeta terra (verifique no diretório de `Assets/Texture` a existência de uma textura de nuvens para esse modelo). A camada de nuvens deve estar acima da superfície da Terra e girando em uma velocidade diferente da Terra.
10. No mesmo diretório de `Assets/Texture` existe outro arquivo contendo um mapa de normais para o planeta Terra. Verifique no manual do `Three.JS` como é possível agregar esse novo mapa de normais ao material aplicado a modelo do planeta Terra. Modifique o código `Earth.js` para que carregue e aplique o mapa de normais na renderização do modelo.

DESAFIO: No diretório de `Assets/cubemaps`, com texturas de nome `galaxyxxx.png`. Pesquise na bibliografia do curso como criar um **cubemap** para que a visualização do planeta terra pareça ser feita no espaço.

SUPERDESAFIO: No mesmo diretório de `Assets/cubemaps` existem texturas de outros **environment maps** que registram visões de um ambiente externo. Modifique o exemplo do cubo para que esse pareça estar localizado em um daqueles ambientes externos. E mais, troque a textura do padrão quadriculado por uma textura de **environment mapping**, ou seja, que o cubo pareça espelhado, refletindo o ambiente externo representado pelo **cubemap**.