



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: 10/07/2018.

Prof. Antonio L. Apolinário Junior

Estagiário Docente: Rafaela Alcantara

Atividade 5 - Técnicas de Mapeamento

Motivação:

Técnicas baseadas em mapeamento são bastante utilizadas em CG para aumentar o realismo de um objeto e/ou cena. Com elas podemos simular materiais com textura e rugosidade, produzir sombras e representar efeitos de iluminação global mais sofisticados [1][2]. Sua base é armazenar informações sobre o modelo e/ou cena em uma imagem. A partir da função de mapeamento essa imagem pode ser acessada e informações do fragmento podem ser obtidas no momento do cálculo da iluminação.

Uma possibilidade interessante é que essa imagem seja dinâmica, permitindo que as informações mapeadas mudem ao longo do tempo, como é o caso de mapeamentos de ambiente para simulação de superfícies espelhadas [1][2]. Nesse caso específico, o mapeamento é gerado proceduralmente. No entanto, podemos pensar em aplicações onde a fonte da mudança é externa à aplicação, por exemplo vinda de um fluxo de vídeo capturado por uma *webcam*. A Figura 1 mostra um exemplo da biblioteca *Three.js* [3] que mapeia o fluxo de vídeo de uma *webcam* em um conjunto de planos, formando um painel esférico, a partir de um material denominado *videoTexture*. Esse componente permite então aplicar efeitos de mapeamento com um fluxo do vídeo.

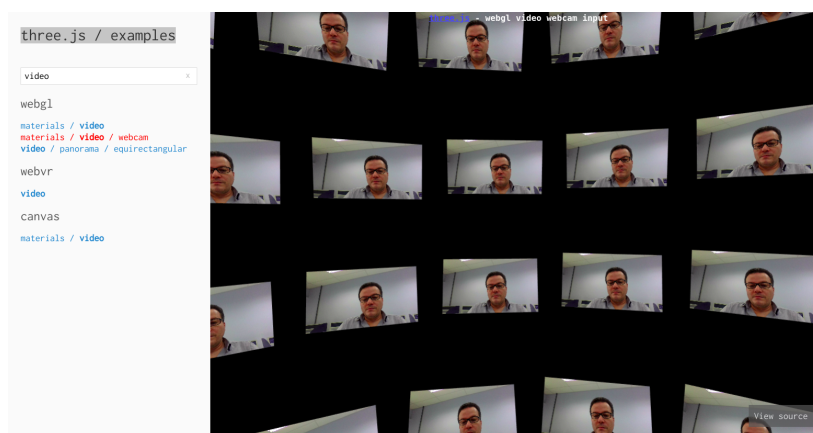


Figura 1: Aplicação com o componente *VideoTexture*¹ em um exemplo da biblioteca *Three.js*².

¹ <https://threejs.org/docs/index.html#api/textures/VideoTexture>

² https://threejs.org/examples/webgl_materials_video_webcam.html

Especificações:

Nessa atividade sua tarefa é gerar uma aplicação em **Javascript/Three.JS** [3] que mapeie o fluxo de vídeo vindo de uma *webcam* nas 6 faces de um cubo. Em cada face deve-se aplicar um tipo de mapeamento diferente a image proveniente do fluxo de video, a saber:

1. ***Normal Mapping*** : gerar uma superfície rugosa sobre a qual o video é projetado;
2. ***Displacment Mapping***: gerar uma superfície com irregularidades sobre a qual o video é projetado;
3. ***Alpha Mapping***: a intensidade da imagem do video controla a transparencia do material;
4. ***Bump Mapping***: a imagem do video controla a rugosidade do material;
5. ***Specular Mapping***: a imagem do video controla o reflexo especular do material. Nesse caso essa face deverá ter uma cor fixa;
6. ***Environment Mapping***: essa face deve refletir, como um espelho, o ambiente ao redor do cubo³.

O cubo deve ficar girando ao redor de um dos seus eixo. O usuário também deve ser capaz de inspecionar todas as suas faces, incluindo aproximar-se, de forma livre.

Entrega da atividade:

- O trabalho deverá ser submetido **somente** via **Moodle**, respeitando a data e hora limite para entrega. Em caso de qualquer problema de arquivos corrompidos ou similar o trabalho será considerado não entregue. Portanto, verifique bem o que for entregar!!
- A entrega no **Moodle** deve ser feita em **um único arquivo compactado (.tgz, .zip ou .rar)** contendo um subdiretório com seu nome e dentro deste **todos** os arquivos necessários para a execução de seu código.
Na falta de algum arquivo (libs, scripts, modelos, texturas, etc), uso de caminhos absolutos, ou qualquer outra "falha" que necessite da edição de seu código fonte a atividade será desconsiderada!!
- A cooperação entre alunos é considerada salutar. No entanto, atividades com alto grau de similaridade serão tratadas como plágio, o que resultará em avaliação **zero** para **todos** os envolvidos.
- Da mesma forma, a internet está cheia de respostas. Use mas não abuse da consulta a códigos prontos. Ctl-C Ctl-V também é plágio!
- Qualquer dúvida, **não suponha** procure o professor⁴ ou o estagiário⁵ para esclarecimentos.

³ Para que esse efeito funcione é preciso definir um mapeamento de ambiente.

⁴ antonio.apolinario@ufba.br

⁵ rafa.alcantara23@gmail.com

Referencias:

- [1] Hughes, John F., Andries Van Dam, James D. Foley, Morgan McGuire, Steven K. Feiner, David F. Sklar, and Kurt Akeley. **Computer graphics: principles and practice**. Pearson Education, 2014.
- [2] Angel, Edward, and Dave Shreiner. **Interactive Computer Graphics with WebGL**. Addison-Wesley Professional, 2014.
- [3] Dirksen, Jos. **Learning Three.JS – the JavaScript 3D Library for WebGL**. Packt Publishing Ltd, 2015.