



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: 2017.1

Data: 06/06/2017.

Prof. Antonio L. Apolinário Junior

Estagiário Docente: Rafaela Alcantara

Roteiro do Laboratório 2 – Transformações Geométricas

Objetivos:

- Compreender de forma prática o conceito de **Transformações Geométricas**.
- Aplicar transformações geométricas, com uso da linguagem *Three.js*, em objetos 3D de acordo com os conceitos apresentados em sala de aula.

Conceitos básicos:

A partir do uso de transformações geométricas, torna-se possível a visualização de uma mesma forma básica de um objeto em diferentes posições, dimensões e orientações. Os tipos mais comuns de transformação são: **rotação**, **translação** e **escala**. A **rotação** modifica a orientação do objeto em relação a um referencial, como a origem do sistema de coordenadas. A **translação** altera a posição do objeto no espaço a partir do deslocamento do objeto. A **escala** redimensiona o objeto a partir do aumento ou diminuição de cada uma das suas coordenadas. Transformações mais complexas podem ser obtidas a partir da aplicação de uma sequência de transformações básicas. Além disso, objetos podem ser organizados de forma hierárquica, sendo que cada transformação aplicada em um nível da hierarquia é propagada para os objetos filhos dessa hierarquia.

Em *Three.js*, transformações geométricas podem ser aplicadas diretamente sobre objetos 3D a partir da manipulação de métodos e atributos do tipo **THREE.Object3D**, tipo implicitamente herdado por qualquer objeto 3D criado em *Three.js*. O método **set** do atributo **position** de um objeto 3D permite o posicionamento absoluto desse objeto na cena 3D. Os métodos **translateX**, **translateY**, **translateZ** permitem a translação relativa de um objeto nos eixos X, Y e Z. Já o método **rotateOnAxis** permite a rotação de um objeto em torno de um eixo e um ângulo específicos.

Objetos podem ser agrupados de forma hierárquica a partir do uso de **grupos**. Um grupo pode ser criado em *Three.js* como uma instância do tipo **THREE.Object3D**. Um grupo pode adicionar um novo objeto na hierarquia a partir do método **add**, que recebe como parâmetro outro objeto do tipo **THREE.Object3D** (uma instância do tipo **THREE.Mesh**, por exemplo). Como um grupo de objetos 3D também consiste em um objeto 3D, ele herda todos os métodos e atributos que permitem a realização de transformações geométricas em *Three.js* (**set**, **translateX**, **rotateOnAxis**, etc.).

O *Three.js* também fornece a possibilidade de criação, manipulação e aplicação explícita das matrizes de transformação geométricas. Uma matriz de transformação geométrica 3D é representada a partir do uso de coordenadas homogêneas como uma instância do tipo **Three.Matrix4**. As matrizes de transformação geométrica de cada objeto podem ser acessadas a partir do atributo **matrix**. Métodos como **makeRotation**, **makeTranslation** e **makeScale**

permitem que matrizes de rotação, translação e escala sejam criadas automaticamente, de acordo com os parâmetros recebidos. Já os métodos **updateMatrix**, **applyMatrix** do tipo *Three.JS* permitem a aplicação mais explícita da transformação geométrica sobre os objetos.

Roteiro:

1. Baixe do repositório da disciplina 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 cada um dos exemplos desse Laboratório.
3. Abra os códigos e analise cada um dos exemplos que compõe esse Laboratório. Entenda as transformações geométricas feitas para cada exemplo.
4. Modifique o exemplo **sistemaPlanetario.html/sistemaPlanetario.js** de forma que a esfera da Terra (em azul) gire, de forma automática, em torno da esfera do Sol no eixo Y. Considere que a Terra está distante do Sol em 0.7 unidades no eixo X. Considere também que a Terra, além de girar em torno do Sol, gira em torno de si mesma no eixo Y. Execute o exemplo no servidor *web* e verifique se a animação foi feita com sucesso.
5. Modifique o *script* gerado no item 4 de forma que a esfera da Lua (em branco) gire, de forma automática, em torno da esfera da Terra no eixo Y, enquanto essa gira em torno da esfera do Sol, também no eixo Y. Considere que a Lua está distante da Terra em 0.15 unidades no eixo X. Considere também que a Lua, além de girar em torno da Terra, gira em torno de seu próprio eixo Y. Execute o *script* modificado no servidor *web* e verifique se a animação foi feita de forma correta.
6. Utilizando o conceito de grupos de objetos, agrupe o sistema planetário desenvolvido no item 5 em grupos hierárquicos e realize as transformações geométricas sobre os grupos e sobre cada objeto individualmente, de forma que a animação gerada seja a mesma obtida no item 5. Execute o exemplo no servidor *web* e verifique se o uso de grupos para a realização de transformações geométricas hierárquicas manteve a animação consistente em relação ao que foi solicitado.
7. Tomando como base o código desenvolvido no item 6, modifique a animação de forma que apenas o Sol gire em torno de si mesmo no eixo X. As rotações no eixo Y se mantêm para todos os objetos. Execute o exemplo no servidor *web* e verifique se a sua aplicação está em conformidade com o que foi proposto neste item.
8. Utilizando o *script* desenvolvido no item 7, crie apenas uma instância do tipo **THREE.SphereGeometry** para representação da esfera e utilize a operação de escala para redimensionar cada objeto (Lua, Terra e Sol) de forma isolada. Lembre-se que o Sol deve ser maior que a Terra, e a Terra maior que a Lua. Execute o exemplo no servidor *web* e verifique se a operação da escala foi feita de forma apropriada sobre os objetos.
9. Altere o código gerado no item 8 de forma a incluir uma nova esfera vermelha (Marte) no sistema solar. Leve em consideração que Marte é maior que a Terra (representada pela esfera azul), que Marte está mais distante do Sol (esfera amarela) que a Terra e que Marte gira em torno do Sol e em torno de si mesma. Lembre-se de utilizar apenas uma única instância do tipo **THREE.SphereGeometry** para representação da esfera. Execute o exemplo no servidor *web* e verifique se a animação com Marte foi feita de forma correta.

¹ no ambiente Windows, utilize o programa USBWebServer (<http://www.usbwebserver.net/en/>) disponível no repositório da disciplina: <http://homes.dcc.ufba.br/~apolinario/Disciplinas/2017.1/MATA65/USBWebserver%20v8.6.zip>. Nos ambientes Linux e MacOS configure o servidor Apache.