

# Computação Gráfica I - MAB122 (2020-1)

## Professor: João Vitor de Oliveira Silva

### TERCEIRA TAREFA PRÁTICA

*Leia o enunciado todo desta tarefa antes de “colocar a mão na massa”.*

*Para executar o esqueleto dessa tarefa, é necessário rodar um servidor local dentro da pasta em que constam os arquivos do esqueleto. Isso pode ser feito de diferentes formas, a mais simples é tendo o Python instalado em seu computador. Para mais detalhes, veja <https://threejs.org/docs/#manual/en/introduction/How-to-run-things-locally>.*

Seu objetivo nesta tarefa é realizar uma outra animação. Desta vez, deverá animar os movimentos de rotação e revolução dos planetas em torno do sol e os movimentos de rotação e revolução da lua em torno da Terra. No arquivo `main.js`, já está escrita toda lógica de renderização de uma cena usando o ThreeJS, sendo necessário de início criar um objeto esfera para representar a lua e os demais planetas. No caso do planeta Saturno, é necessário também criar um objeto anel, que será filho de uma esfera. Também deve ser adicionada a luz emitida pelo sol.

*Se achar necessário, pode criar classes e/ou funções auxiliares.*

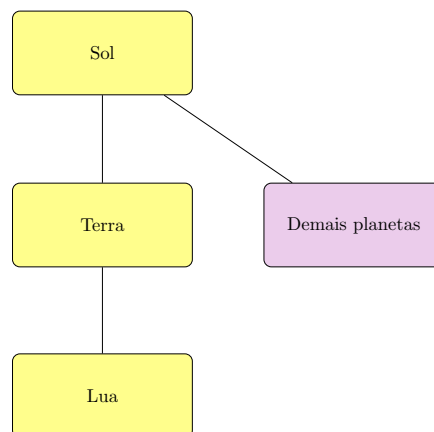


Figura 1: Hierarquia do sistema solar simplificada (sol, terra e demais planetas). Importante destacar que cada planeta será um filho de diferente de sol.

Na Figura 1, é possível vermos a hierarquia esperada para a cena.

Para se criar um objeto lua, é necessário carregar a textura `moon.jpg`. O processo já foi feito para sol e a lua para facilitar esta etapa. O material a ser usado nos demais planetas e na lua deverão ser do tipo Phong. As texturas dos demais planetas pode ser

obtida em <https://www.solarsystemscope.com/textures/>.

*Caso queira fazer um sistema solar “diferente”, pode utilizar outras imagens como textura ☺.*

Após a criação do objeto lua, você deve adicionar uma luz que será emitida pelo sol. Para criar essa luz, você irá criar um objeto do tipo `THREE.PointLight`, que é um tipo de luz que ilumina em todas as direções. A sua cor deve ser branca (`0xffffff`) e ter intensidade 1.5. O decaimento deve fazer com que planetas mais próximos de sol sejam mais facilmente visíveis, entretanto os mais distantes como Netuno e Plutão também devem receber um pouco da iluminação. Adicione também a sua luz criada como um filho do sol.

Para mais detalhes, ver <https://threejs.org/docs/#api/en/lights/PointLight>.

Por fim, faça:

1. Os movimentos de rotação e revolução da terra em torno do sol. Os mesmos ocorrem em aproximadamente 1 dia e 365 dias, aproximadamente.
2. Os movimentos de rotação e revolução da lua em torno da terra. Os mesmos ocorrem de maneira sincronizada em aproximadamente 30 dias.
3. Os movimentos de rotação e revolução dos demais planeta em torno do sol. A tabela a seguir possui o tempo necessário para cada um destes:

Planeta	Tempo de rotação	Tempo de revolução
Mercúrio	58.6 dias	87.97 dias
Vênus	243 dias	224.7 dias
Marte	1.03 dias	1.88 anos
Júpiter	0.41 dias	11.86 anos
Saturno	0.45 dias	29.46 dias
Urânio	0.72 dias	84 dias
Netuno	0.67 dias	164.79 dias
Plutão	6.4 dias	248.6 dias

Os tempos inseridos servem apenas para que a velocidade das rotações que definir sejam próximas da realidade, não é necessário levar estes valores ao pé da letra.

Importante destacar que os planetas Vênus e Urânio realizam uma rotação horária, todos os demais planetas e a lua realizam uma rotação anti-horária.

Você pode usar qualquer um dos métodos e funções presentes na biblioteca ThreeJS para realizar as transformações afins necessárias. A documentação da biblioteca pode ser acessada em <https://threejs.org/docs/>. Se achar necessário, também pode usar a biblioteca TweenJS como na tarefa prática 2.

### CONSIDERAÇÕES FINAIS

- É recomendável o uso do Google Chrome para abrir os arquivos `.html`.
- O trabalho pode ser feito de forma **individual** ou em **grupos** de até 3 pessoas.
- A entrega deve feita pela plataforma Google Classroom. Pode-se enviar um arquivo `.zip` ou um link do repositório com a solução desenvolvida.

**Prazo para entrega:** 09/03 (até 19:59).