

Computação Gráfica — Trabalho 1

Professor: Waldemar Celles
Aluno: Antenor Barros Leal

29 de setembro de 2024

1 Resumo

Este trabalho tem como objetivo fazer uma representação simplificada do sistema solar, onde são mostrados o Sol, Mercúrio, a Terra e a Lua. A simulação implementa transformações, texturas e uma hierarquia de nodes que representam os corpos celestes, bem como suas órbitas, com base no tempo decorrido.

2 Escalas e períodos

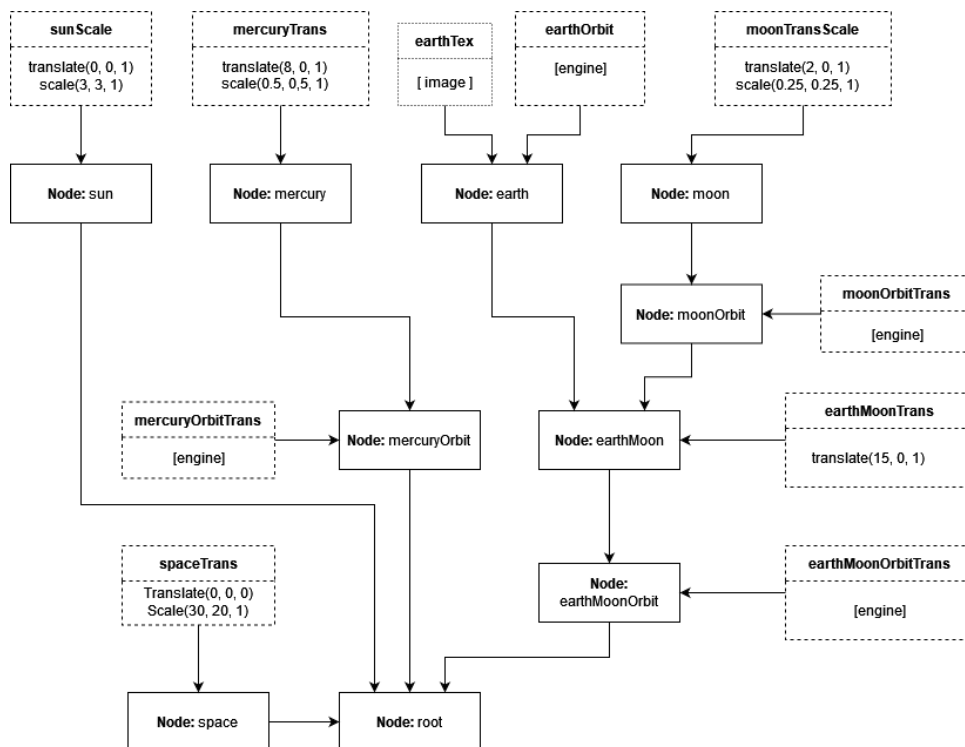
As escalas e distâncias dos planetas em relação ao Sol não são realistas, pois, para isso, os planetas teriam que ser muito pequenos e "distantes", tornando-os difíceis de visualizar. Entretanto, as relações entre os períodos orbitais são realistas.

Na classe da engine, vemos um valor padrão de 365, representando o período orbital da Terra em dias. Multiplicamos por 1000 para que o movimento fique perceptível e multiplicamos por $t-t_0$ (o intervalo de tempo entre chamadas da função `update`) para que o movimento seja agnóstico em relação à velocidade do computador. Este valor de $t-t_0$ assegura que as atualizações ocorram com base no tempo real.

Para a rotação da Terra (transformação `earthOrbit`), usei o valor 100. Embora isso não seja realista, foi necessário utilizar um valor alto para que o movimento de rotação da Terra seja perceptível na simulação.

3 Diagrama

O diagrama a seguir apresenta o grafo de cena, mostrando os objetos criados e suas conexões hierárquicas.



4 Nodes e Transformações

Os planetas e o Sol são representados por formas primitivas do tipo `Disk`, instanciadas nos nodes `sun`, `mercury`, `earth` e `moon`. Abaixo, o papel de cada node e suas respectivas transformações:

4.1 Sol

O node `sun` contém a transformação `sunScale`, que ajusta o tamanho e a posição do Sol. Esta transformação é usada para centralizar o Sol e mudar seu tamanho.

4.2 Mercúrio

O node `mercury` contém a transformação `mercuryTrans`, responsável por posicionar Mercúrio em sua órbita em torno do Sol. Ele é escalado para ser menor que a Terra e o Sol. O node `mercuryOrbitTrans` envolve Mercúrio, servindo para rotacioná-lo em torno do Sol. Este node é controlado pela classe `MovePointer`, que atualiza sua rotação com base no período da órbita de Mercúrio (88 dias, aproximadamente).

4.3 Terra e Lua

O node `earth` contém a transformação `earthOrbit`, que faz a Terra girar ao redor do Sol. Para a Lua, temos dois nodes de transformação: - `moonTransScale`: responsável por posicionar a Lua em relação à Terra, ajustando sua escala e distância. - `moonOrbitTrans`: é um "container"

que faz a Lua orbitar a Terra. Se aplicássemos o engine diretamente no `moonTransScale`, a Lua rotacionaria em torno de si mesma, em vez de orbitar a Terra.

O node `earthMoon` agrupa a Terra e a Lua, permitindo que ambas sejam posicionadas como um sistema em torno do Sol. Já o node `earthMoonOrbitTrans` é responsável pela rotação do sistema Terra-Lua em torno do Sol.

4.4 Espaço

O fundo da cena é representado por uma textura estrelada, aplicada a um quadrado escalado (primitiva `Square`). A transformação `spaceTrans` ajusta o tamanho e a posição desse quadrado para ocupar todo o plano de fundo. Para garantir que o fundo fique atrás dos planetas e do Sol, a componente `z` da transformação de escala é ajustada para um valor inferior (0).

5 Composição de Cena

A cena completa é composta pela árvore de nodes descrita acima: - O `root` contém os nodes do Sol, Mercúrio e o sistema Terra-Lua, além do fundo. - As engines de movimentação (`MovePointer`) são responsáveis por atualizar as órbitas com base no tempo.

Essa composição hierárquica permite que a simulação funcione de maneira modular, onde cada corpo celeste tem suas transformações e atualizações associadas à sua posição e movimento dentro do sistema solar.