Animador de cenas 3D

Pedro Gonçalves 88859
Pedro Silva 89228

***Resumo* – Animador de um Sistema planetário, mais concretamente do Sistema Solar (versão pós 2006), em 3 dimensões. Neste modelo podemos observar a rotação e translação de todos os planetas, podendo escolher ver o Sistema Solar à escala real ou com uma escala adaptada para facilitar a visualização de todos os planetas. O utilizador pode também comparar o tamanho dos planetas e do Sol com a escala real, bem como "ver" uma representação 3D de um asteroide.**

***Abstract* – Planetary system animator, more specifically the Solar System (after 2006 version), in 3 dimensions. In this model we can observe the rotation and translation of all the planets, and either watch the Solar System in the real scale or with an alternative scale that makes it easier to visualize all the planets. The user can also compare the size of the planets and the Sun with the real scale and "see" the 3D representation of an asteroid.**

I. INTRODUÇÃO

Neste relatório, vamos abordar as várias tecnologias que foram desenvolvidas em WebGL e que foram implementadas neste projeto, desde a criação dos planetas através de vértices, à sua normalização e aos seus períodos de rotação e translação.

II. BACKGROUND E INTERFACE

Para o fundo da nossa aplicação utilizámos uma imagem a simular o universo, embutida no body e no *canvas* do documento, desta forma tornando mais realista toda a simulação. Também escolhemos botões e *dropdowns* que ficassem bem com o fundo usado.

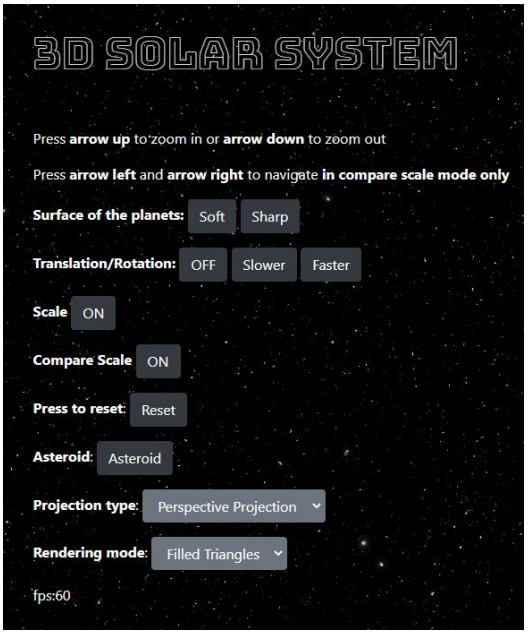

Figura 1 – Botões para manipulação do comportamento dos planetas

III. CRIAÇÃO DO SOL E DOS PLANETAS

O primeiro objetivo deste projeto era a criação de esferas que se comparassem ao Sol e aos planetas do Sistema Solar.
Para isso adaptamos algum código utilizado nas aulas práticas, onde é possível criar um cubo em 3 dimensões

através de um array com a posição dos seus vértices. Após a criação do cubo, desenvolvemos um modelo representando a esfera de raio unitário e centrada na origem, deslocando cada um dos seus vértices para a superfície da esfera de raio unitário e centrada na origem. Para isso, foi usada a função *midPointRefinement(coordsArray, recursionDepth)* que recebe as coordenadas dos vértices de um triângulo e cria uma determinada quantidade de novos triângulos, dependendo do seu nível de recursividade. Usámos igualmente as funções *normalize(v),* que permite normalizar o vetor v, e *computeVertexNormals(coordsArray, normalsArray),* que calcula os vetores normais para cada triângulo.

Todos os planetas foram adicionados um a um no ficheiro "sceneModels.js", no qual definimos os vários atributos, entre estes a sua posição relativa no espaço, a sua escala em relação à janela de *clipping*, os seus ângulos de rotação e controlos de animação, entre outros. Para criar e alinhar todos os planetas utilizámos uma escala irrealista, devido ao facto de o Sol ser muito maior que todos os planetas sólidos, o que iria dificultar a visualização desses mesmos planetas. Todos os planetas ficaram alinhados por ordem, com um espaçamento aproximado entre estes.

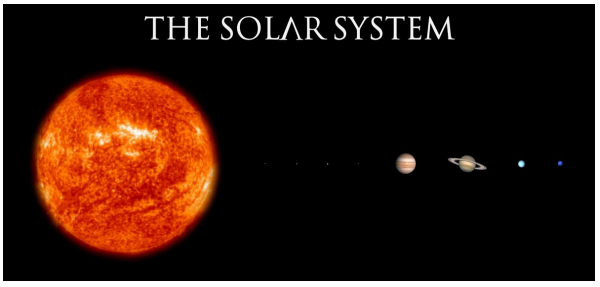

Figura 2 - Escala real do Sistema Solar

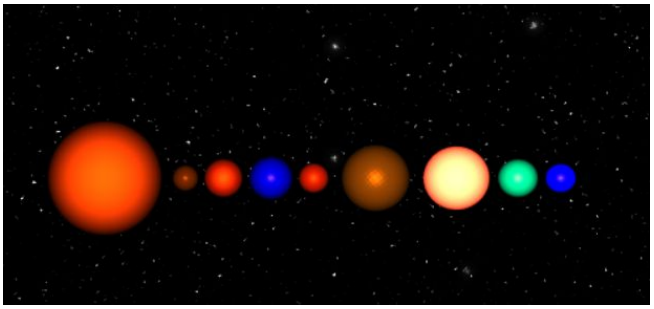

Figura 3 – Escala utilizada para a representação do Sistema Solar

IV. SUPERFÍCIE DOS PLANETAS

A superfície dos planetas pode ser vista de acordo com o nível de recursividade utilizado para formar a esfera. Isto porque, com um nível muito alto de recursividade (mais vértices), torna-se difícil ver a rotação dos planetas e do Sol, além de tornar a aplicação mais lenta. Ao iniciar a aplicação, os planetas têm uma superfície mais irregular, podendo ser alterada através do botão "*soft*".

Figura 4 – Botões para manipulação da superfície

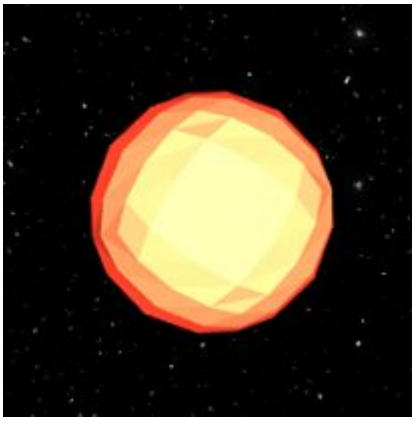

Figura 5 – Representação do planeta Saturno com a superfície irregular

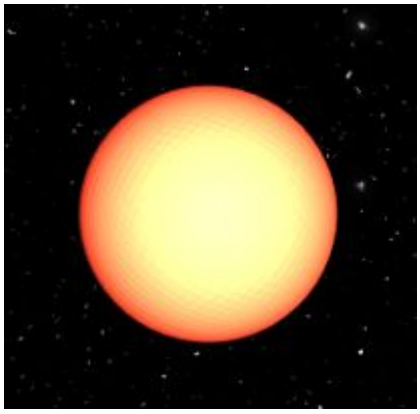

Figura 6 – Representação do planeta Saturno com a superfície plana

V. ROTAÇÃO DOS PLANETAS E DO SOL

Em relação à rotação dos planetas e do sol, utilizámos as variáveis *isRotZZOn*, que define se a rotação está ou não ativa no planeta definido, *rotZZSpeed* que manipula a velocidade de rotação, visto que todos os planetas têm uma velocidade de rotação diferente e por fim *rotZZDir* que permite escolher a direção da rotação. No caso do Sistema Solar, o único planeta que roda numa direção diferente dos restantes é Vénus, que roda na direção dos ponteiros do relógio.
O utilizador da aplicação pode alterar estes valores através dos botões "On / Off", para desligar a rotação dos planetas (e do Sol), "Slower" e "Faster" para desacelerar a acelerar a velocidade de rotação dos mesmos, respetivamente.

Figura 7 – Botões de "On / Off", "Slower" e "Faster", relativamente à translação e rotação dos planetas

VI. TRANSLAÇÃO DOS PLANETAS

Na parte da translação, simulámos os períodos de translação de todos os planetas em torno do Sol. Nos exemplos apresentados nas aulas, aplicávamos translação uniforme a toda a cena, porém, na nossa simulação tivemos que utilizar uma translação com velocidades diferentes para cada planeta. Para esse efeito, tivemos que criar um array de ângulos, inicialmente com todos os elementos a 0.0, onde cada um desses ângulos corresponde a cada planeta. Os valores desses ângulos atualizam de forma diferente, de modo a simular a translação com escala real entre todos os planetas, como por exemplo, 1 volta do planeta Marte em torno do Sol equivale a aproximadamente 7,8 voltas do planeta Mercúrio em torno do Sol.
É possível parar e recomeçar a translação com o botão "*On/Off*", que altera o valor da variável *globalRotationZZ_ON*. É também possível fazer com que a translação de todos os planetas seja mais rápida, com o botão "*Faster*", ou que seja mais lenta, com o botão "*Slower*".

VII. ILUMINAÇÃO DOS PLANETAS

Na iluminação dos planetas utilizámos várias *lightsources*. Inicialmente pretendíamos dar uma cor a cada

planeta, mas optámos por não lhes dar cor e apenas utilizar a incidência de *lightsources*. Definimos então vários parâmetros para cada planeta, como *kAmbi*, *kDiff*, *kSpec* e *nPhong*, com valores diferentes para cada planeta, de modo a simular cores diferentes. Depois de definidas as cores, tivemos que definir a posição e a intensidade, com os atributos *position*, *intensity* e *ambIntensity*. Definimos uma posição inicial lateral, com a coordenada x negativa, para simular a incidência do Sol nos planetas. Com isto surgiu o problema de como simular a rotação do foco de luz no planeta quando este efetua o movimento de translação em torno do Sol. O nosso principal objetivo seria colocar a *lightsource* com um ponto fixo, o Sol, e iluminar a partir desse ponto em todas as direções, mas acabamos por apenas fazer a rotação da *lightsource* em torno de Z, à medida que o planeta faz a translação, com valores aproximados de forma a que a luz rode com a mesma velocidade da translação do planeta. Por vezes o número de *fps* desce drasticamente, principalmente a utilizar o *Google Chrome*, e acaba por desconfigurar o ângulo da luz.

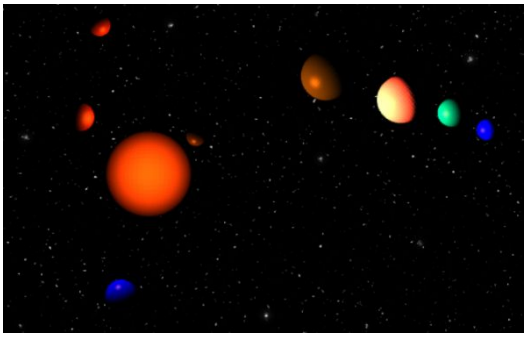

Figura 8 – Rotação dos focos de luz

VIII. *SCALE*

Como já foi referido no ponto II, criámos o Sistema Solar com uma determinada escala de modo a facilitar a visualização de todos os planetas. Posteriormente, desenvolvemos um botão que possibilita alterar a escala. Quando este botão está OFF, aparece a primeira escala, que facilita a visualização e quando está ON aparece a escala real com o tamanho de todos os planetas à escala, porém, a distância entre eles continua a não estar à escala. Com este modo ativado, é possível continuar a ver a translação e a rotação de todos os planetas e respetivas *lightsources*.

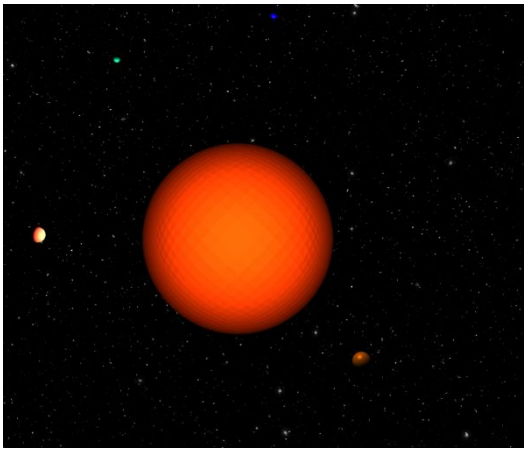

Figura 9 – Sistema solar à escala real

IX. NAVEGAÇÃO ATRAVÉS DO TECLADO

Para ajudar a visualização e a interação com o modelo 3D, implementámos Zoom In e Zoom Out e ainda a possibilidade de mover lateralmente através do uso do teclado. A técnica de Zoom In e Zoom Out está disponível em todos os modos, enquanto que a

mobilidade lateral apenas está disponível no modo Compare Scale que será abordado no próximo tópico.

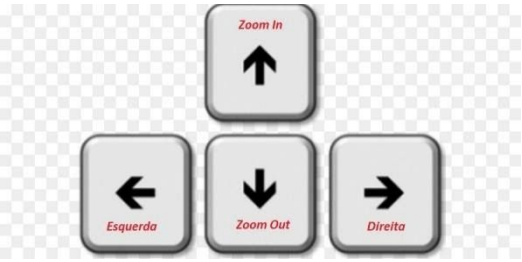

Figura 10 – Movimento associado a cada tecla

X. *COMPARE SCALE*

Este modo permite ao utilizador visualizar o Sistema Solar à escala, sem translação. Ao carregar no botão "Compare Scale" podemos utilizar todos os modos de navegação através do uso do teclado, como se pode ver na Figura 10, o que permite comparar o tamanho de todos os planetas e perceber as diferenças de dimensão entre os planetas sólidos e gasosos.

XI. *RESET*

Premindo o botão "*Reset",* é possível reiniciar o modelo, colocando os planetas nas posições e tamanhos originais e recomeçando a translação.

XII. ASTERÓIDE

O nosso projeto contém também texturas. Ao clicar no botão de asteroid "ON", o utilizador poderá ver a representação de um objeto similar a um asteróide. Adaptámos os vértices de um ficheiro utilizado nas aulas práticas, "modeloEsferaV1.txt", e definimos várias coordenadas de texturas e também os índices dos vértices do cubo. Por último, importámos a textura do ficheiro "meteor.jpg".
Infelizmente, o resultado obtido não foi o esperado, pois no exemplo da aula conseguimos obter o seguinte asteróide:

Figura 11 – Asteróide esperado

Porém, na sua implementação com o sistema solar foi necessário adaptar código e instanciar objetos em ficheiros diferentes e acabou por sair um objeto estranho que se pode ver na seguinte figura:

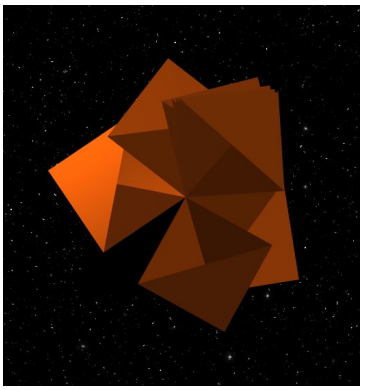

Figura 12 – Asteróide obtido

XIII. CONCLUSÃO

Neste projeto melhorámos as nossas competências na utilização de WebGL e JavaScript. Foi particularmente desafiante a implementação dos focos de luz em cada planeta, uma vez que foi necessário sincronizar a rotação da *lightsource* com a translação de cada planeta. Foi interessante também aprender alguns factos sobre o Sistema Solar que desconhecíamos.

XIV. REFERÊNCIAS

https://keycode.info/
https://webglfundamentals.org/
https://www.infoplease.com/math-science/space/solar-system/basic-planetary-data

Código das aulas práticas 3-9 de Computação Visual