



**Ciências
ULisboa**

Faculdade
de Ciências
da Universidade
de Lisboa

Computação Gráfica

2023/2024

Trabalho – WebGL

Entrega

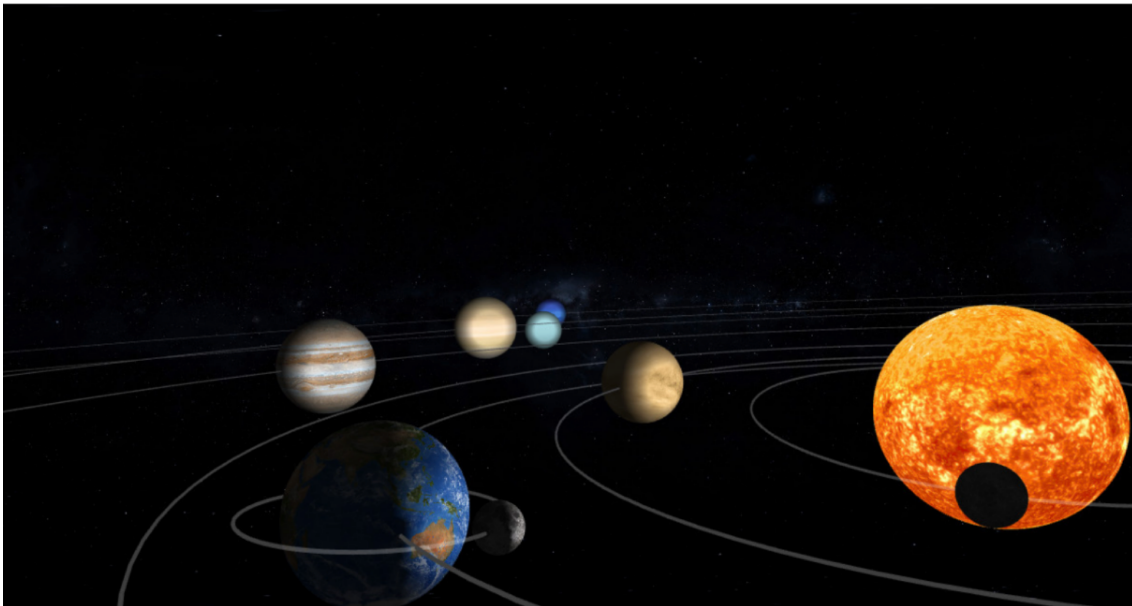
Devem entregar um *zip* com todos os ficheiros do projeto e um relatório sucinto. A entrega é feita na página da disciplina no Moodle.

Todos os ficheiros devem ser identificados com o **número do grupo** (por exemplo, *RelatórioCG01.pdf* para o grupo **CG01**). No relatório deve constar o **número do grupo**, os **nomes** e os **números** dos elementos do grupo.

Prazo de entrega: 03 de dezembro de 2023, às 23 horas.

Enunciado

O objetivo deste trabalho é usar uma biblioteca WebGL para gerar uma representação 3D de um sistema solar. A figura seguinte é um exemplo do que é pedido, embora não tenha de ser exatamente assim.



Exemplo do trabalho.

O trabalho deve implementar os seguintes pontos fundamentais:

1. O sistema de astros deve ter pelo menos uma estrela e 4 planetas.
2. O sistema deve ter vários planetas, incluindo um deles com uma lua. A órbita da lua deve ser inclinada.
3. Aplicar pelo menos uma textura.
4. A iluminação deve ser feita através de uma fonte de luz pontual localizada na posição da estrela.
5. Devem implementar o algoritmo de shading de Gouraud e o de Phong.
6. As órbitas dos planetas devem ter velocidades diferentes.
7. Pelo menos uma da órbita deve ser elíptica.

Interações:

1. Pressionar as teclas '1' e '2' deve fazer reset da posição da câmara para a posição atual da terra e do sol, respetivamente.
2. Pressionar a tecla 'o' deve permitir visualizar as órbitas dos planetas.
3. As setas e o rato devem permitir movimentar a câmara pela cena 3D.

Sugestão de extras:

1. Implementar possíveis sombras que os planetas criam.
2. Implementar outros detalhes que valorizem o trabalho, quanto mais criativos melhor. Alguns exemplos são: anéis no planeta, cintura de asteroides, cometa, realismo do tempo de órbita no caso do sistema solar (à escala), fonte de luz adicional, fonte de luz *spot*, etc.

O relatório deve ser sucinto e responder aos seguintes pontos:

Parte I – Implementação dos objetos 3D

- 1- Mostre uma captura de ecrã em que se veja claramente toda a cena 3D implementada, por exemplo uma vista de cima. Descreva a cena 3D, quantos astros estão representados, identifique a lua, e mencione outros detalhes da implementação que considere importantes.

Parte II – Texturas

- 1- Descreva a implementação das texturas e como estas são aplicadas aos objetos.

Parte III – Animação

- 1- Descreva como são implementadas as animações existentes na cena 3D, incluindo as órbitas dos planetas e luas, rotações sobre o próprio eixo dos planetas, e outras que tenham sido implementadas, mencionando também a parte do pipeline gráfico relevante.
- 2- Inclua o link para um ou dois pequenos vídeos que mostre a animação da lua e dos restantes planetas.

Parte IV – Iluminação

1. Implemente o método de Shading Phong, considerando apenas as componentes de luz ambiente e difusa. O relatório deve comparar este resultado com o resultado obtido com o método de Gouraud.
2. Implemente a componente de luz especular no método de Shading Phong. O relatório deve comparar este resultado com o resultado obtido na alínea anterior.
3. Inclua no relatório a visualização das diferentes componentes separadamente (ambiente, difusa, e especular).
4. Experimente com dois valores diferentes de coeficiente de especularidade, e inclua a comparação no relatório – se não se notar bem a diferença, experimente desligar as texturas.

Parte V – Extras

- 1- Descreva todas as implementações ou detalhes adicionais que considere relevantes.

Referências

Algumas sugestões para a implementação das várias partes.

Como implementar vários objetos na mesma cena 3D

Na 2ª aula de WebGL foi visto como usar a biblioteca *twgl* para criar objetos 3D, incluindo esferas. Também foi estudada a implementação de um pipeline gráfico para a visualização de objetos 3D. Esta tarefa é uma extensão desta aula para obter vários objetos simultaneamente na cena 3D.

Uma possível sugestão para implementar vários objetos na mesma cena 3D é utilizar um vetor que contém vários *bufferInfo* gerados pelas funções da *twgl*. Cada objeto deve ter a sua *matriz modelo*, que o irá colocar na sua posição na cena 3D. Existem outras formas para alcançar o mesmo objetivo, pelo que isto deve ser visto apenas como uma sugestão.

Links:

- <https://webgl2fundamentals.org/webgl/lessons/webgl-drawing-multiple-things.html>
- <https://github.com/greggman/twgl.js/blob/master/examples/primitives.html>
- <https://twgljs.org/examples/itemlist.html>
- <https://webgl2fundamentals.org/webgl/lessons/webgl-drawing-multiple-things.html>

Como implementar o algoritmo de shading Gouraud e Phong

Na 3ª aula TP de WebGL foi visto como calcular vetores, e como passar valores para dentro dos shaders. Também foi visto como implementar a componente de luz ambiente. Esta tarefa é uma extensão desta aula para a componente de luz especular e difusa.

Relembrem a diferença entre os dois métodos: um implementa o modelo de iluminação utilizando a interpolação dos valores de intensidade, enquanto que o outro faz interpolação dos valores das normais para calcular os valores de intensidade.

Links:

- https://www.youtube.com/watch?v=33gn3_khXxw&list=PLjcVFFANLS5zH_PeK_C6l8p0Pt1hzph_rt&index=5
- <https://webgl2fundamentals.org/webgl/lessons/webgl-3d-lighting-directional.html>
- <https://github.com/greggman/twgl.js/blob/master/examples/twgl-cube.html>
- <http://www.cs.toronto.edu/~jacobson/phong-demo/>
- <https://www.geertarien.com/blog/2017/08/30/blinn-phong-shading-using-webgl/>
- <https://www.mathematik.uni-marburg.de/~thormae/lectures/graphics1/code/WebGLShaderLightMat/ShaderLightMat.html>
- https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Lighting_in_WebGL
- http://www.webglacademy.com/courses.php?courses=0_1_20_2_3_4_23_5_6_7_10#7

Como implementar as várias fontes de luz e as componentes do modelo de iluminação

- Seção de Lighting do <https://webgl2fundamentals.org/>

Ler mais

Javascript

- <https://www.cs.cmu.edu/~aldrich/courses/17-396/js/>
- <https://www.w3schools.com/js/DEFAULT.asp>
- <https://javascript.info/>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Como o método createSphereBufferInfo() funciona:

- http://www.songho.ca/opengl/gl_sphere.html

Como o pipeline gráfico funciona:

- https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/WebGL_model_view_projection

Como implementar interatividade com as teclas e rato (a ser usado para um extra):

- <https://webgl2fundamentals.org/webgl/lessons/webgl-tips.html#tabindex>
- <https://stackoverflow.com/questions/5512127/how-can-i-rotate-around-a-scene-in-webgl-on-mousedrag-emulating-a-camera-moving>
- <https://learnopengl.com/Getting-started/Camera>
- <https://javascript.info/mouse-events-basics>
- <https://cran.r-project.org/web/packages/rgl/vignettes/WebGL.html>
- http://www.webglacademy.com/courses.php?courses=0_1_20_2_3_4_23_5_6_7_10#4
- <https://github.com/mikolajsenko/3d-view-controls>

- <https://www.youtube.com/watch?v=rd9HbHS1H6I>

Sobre matrizes de transformações:

- https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Matrix_math_for_the_web

Como importar do Blender:

- http://www.webglacademy.com/courses.php?courses=0_1_20_2_3_4_23_5_6_7_10#6

Biblioteca Math, útil para funções trigonométricas, etc.

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math
- <https://stackoverflow.com/questions/38416492/how-to-include-and-use-math-js>

Outros:

- <https://www.youtube.com/playlist?list=PLPqKsyEGhUnaOdIFLKvdkXAQWD4DoXnFI>
- <https://sites.google.com/site/csc8820/educational/webgl-basics>