Ciências ULisboa
Faculdade de Ciências da Universidade de Lisboa

Computação Gráfica

2020/2021

Trabalho – WebGL

Entrega

Devem entregar 4 ficheiros (um para cada pergunta) e um relatório sucinto.
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 deste exercício: 09 de dezembro de 2020, às 23 horas. **(este enunciado é fornecido com um ficheiro html que deve usar-se como ponto de partida para a sua resolução.)**

Enunciado

O objetivo deste exercício é usar uma biblioteca WebGL para efetuar alterações numa cena 3D e gerar representações e animações específicas.

Este anunciado é acompanhado de uma cena 3D que exemplifica a importação de objetos do Blender para o WebGL – OfficeStart.html. Podem usar este ficheiro como base, ou usar os ficheiros dados nas aulas, nomeadamente o sample7 do tutorial abordado nas aulas TP.

É importante que refiram no relatório onde estão resolvidos cada exercício / alínea, *i.e.* em que ficheiro e linha de código.
Se por razões de organização de código ou outras for mais conveniente entregar mais ficheiros, por favor descrevam a situação no relatório.

É pedido que realizem as seguintes alterações:

1. Modificações na cena original (1.5+1.5+1.0+1.0 valores cada alínea respetivamente).

Este exercício pode ser implementado tendo como base tanto o *Sample7* como o *OfficeStart*. Estas alíneas devem, idealmente, ser implementadas no mesmo ficheiro.

    1. Adicionar três novos objetos à cena: uma pirâmide amarela, um cubo verde, e uma esfera azul.
        1. Nota: O sample7 já contém o cubo, pelo que devem apenas mudar a cor do objeto.
    2. Modificar a posição da fonte de luz de modo a iluminar a parte do cubo de frente para a câmara.

3. Modificar a posição da câmara de modo a apresentar uma visão da lateral da cena.
4. Modificar a posição da câmara de forma a apresentar uma visão de topo da cena.

2. Iluminação (2.0 + 1.5 + 1.0 + 1.0 + 1.5 valores cada alínea respetivamente).
Este exercício pode ser implementado tendo como base tanto o *Sample7* como o *OfficeStart*. Estas alíneas devem, idealmente, ser implementadas no mesmo ficheiro.

1. Implementar o método de Phong Shading, considerando apenas as componentes de luz ambiente e difusa. O relatório deve comparar este resultado com o resultado obtido com o método de Gouraud, implementado nas aulas TP, e analisar qual produz resultados mais satisfatórios.
2. Implementar a componente de luz especular. O relatório deve comparar este resultado com o resultado obtido na alínea anterior.
3. Incluir 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.
5. Implemente uma fonte de luz pontual, de cor diferente da luz direcional e ambiente.

3. Animação (2.0+2.0+2.0 valores).
Este exercício pode ser implementado tendo como base tanto o *Sample7* como o *OfficeStart*. Estas alíneas devem, idealmente, ser implementadas no mesmo ficheiro.

1. Produza uma animação da cena iluminada pelo script inicial (*Sample7* ou *OfficeStart*) ou por uma das alíneas do ponto 2, movimentando apenas a fonte de luz direcional em torno da esfera.
2. Produza uma animação da cena original (*Sample7* ou *OfficeStart*) movimentando apenas a câmara em torno da cena.
3. Considere os objetos adicionados no Exercício 1. Considere que a pirâmide é um sol, e que portanto permanece imóvel. O cubo é um planeta, que deve orbitar em torno do sol, e a espera é a lua do planeta. Implemente estes movimentos para o cubo e o planeta.

4. Novos modelos de objectos (1.0+1.0 valores)

1. No Blender modifiquem e simplifiquem significativamente o modelo da estátua ou do arco (em termos de dimensão, complexidade, fusão de todos os objectos) e gerem o modelo em formato .JSON recorrendo ao script Python 2.7 (obj_parser.py) fornecido na 3a aula de WebGL (ver com especial atenção nos slides desta aula as opções a usar no Blender na altura da conversão para .OBJ).
2. Adicionem o objeto à cena de forma a que os objetos adicionados no exercício 1 (pirâmide, do cubo, e da esfera) e o objeto importado do *Blender* estejam na mesma cena.
    1. O ficheiro *OfficeStart* exemplifica como importar um ou múltiplos ficheiros *json*. Podem usar este código ou podem usar outra ferramenta / biblioteca da vossa preferência.

Importante!
Entrega:
O relatório entregue deve descrever exatamente as modificações que efetuaram para cada alínea, adicionando imagens que mostrem o resultado da modificação efectuada e as alterações realizadas ao código.
Cada alínea do trabalho é avaliada de acordo com o que é descrito no relatório, com especial incidência na correção e completude das explicações e justificações.

Referências:

Como implementar Phong

- Aula da semana de 23/Nov – O sample 7 implementa Gouraud. Para implementar Phong, devemos interpolar as Normais em vez de interpolar as intensidades. Isto significa implementar o cálculo das cores para cada fragmento, em vez de para cada vértice.
- 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

Como implementar Esfera com normais

- Sugestão de algoritmo

- http://www.songho.ca/opengl/gl_sphere.html
    - A primeira caixa de código cria os vértices e as normais. Podem ignorar a parte das texturas.
    - A segunda caixa de código cria os índices.
    - Este código está em C++, sendo que a tradução para javascript exige poucas alterações.
    - É necessário criar um vetor de cores, e adicionar uma cor para cada vértice, assim como foi feito para o cubo.
- https://stackoverflow.com/a/58672161

- 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

Implementar Pirâmide com normais

- As normais das faces da pirâmide podem ser aproximadas pelas normais usadas para o cubo.
- Para ser mais preciso no cálculo, recordem o uso do produto vetorial:

    - https://math.stackexchange.com/a/3080534
    - https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-rendering-a-triangle/geometry-of-a-triangle

Sobre desenhar vários objetos

- Cada objeto deve ter a sua própria matriz *model*, os seus vértices, normais, *etc.* Estes dados devem ser guardados em buffers, tal como foi feito para o cubo.
- https://webgl2fundamentals.org/webgl/lessons/webgl-drawing-multiple-things.html

Como implementar fonte de luz pontual e componente especular

- https://webgl2fundamentals.org/webgl/lessons/webgl-3d-lighting-point.html