

Trabalho C

Cena Interativa com Malhas, Materiais, Luzes, Texturas e Câmara Estereoscópica



Figura 1— Referências para a elaboração de uma paisagem rural visitada por uma nave extraterrestre: (A) casa tipicamente alentejana, (B) sobreiro descortiçado, (C) montado alentejano e (D) disco voador estilo anos 50.

Objetivos

Os objetivos do terceiro trabalho de laboratório são (i) perceber as noções básicas de iluminação, (ii) os conceitos de material, (iii) as fontes de luz direcional e *spotlight*, (iv) a modelação geométrica por instanciação de primitivas e criação de malhas de polígonos, (v) compreender os princípios básicos da aplicação de texturas, e (vi) os princípios fundamentais sobre a câmara estereoscópica.

Todos os grupos submetem o código até ao dia **6 de junho às 23h59** (final da Semana 6). As discussões serão realizadas nos respetivos turnos na **2ª aula** da Semana 7. O Trabalho C corresponde a **4 valores** da nota da componente laboratorial. A realização deste trabalho tem um esforço estimado de **14 horas** por elemento do grupo, distribuído por **duas semanas**.

Não esquecer de comunicar ao docente do laboratório as **horas despendidas pelo grupo (média do grupo)** na realização deste trabalho.

Lista de Tarefas

1. **[0,25 valores]** Criar uma aplicação Three.js dedicada à geração procedimental de texturas que serão usadas para representar um campo floral e um céu estrelado: (i) o campo rural consistirá em centenas de pequeníssimos círculos brancos, amarelos, lilases e azuis-claros sobre um fundo verde-claro; e (ii) o céu estrelado consistirá num pontilhado de centenas de estrelas brancas, representadas por pequeníssimos círculos, sobre um fundo *degradé* linear de azul-escuro para violeta-escuro. O utilizador pode transitar entre gerar a textura do campo floral e o céu estrelado recorrendo às teclas '1' e '2', respetivamente
2. **[0,50 valores]** Criar uma aplicação Three.js para a cena principal. Gerar um terreno com montes e vales para que se assemelhe a um montado alentejano. Para tal, fazer uso de um gerador de *heightmaps* (<https://manticorp.github.io/unrealheightmap/>). Para além do mapa de alturas, aplicar a textura do campo floral gerada na Tarefa 1. Ambas as texturas devem ser aplicadas sobre a malha de um plano.
3. **[0,25 valores]** Criar uma *skydome* por forma a envolver o terreno. Aplicar a textura do céu estrelado gerada na Tarefa 1. Esta textura deve ser aplicada sobre a malha de uma esfera.
4. **[0,50 valores]** Adicionar uma lua cheia modelada por uma esfera. Explorar as propriedades de emissividade do material para que a lua seja brilhante. Criar a iluminação global da cena recorrendo a uma fonte de luz direcional. A fonte de luz direcional deve incidir com um ângulo diferente de zero relativamente à normal do plano xOy do referencial global da cena. Esta fonte de luz deve poder ser ligada ou desligada através da tecla ('D(d)').
5. **[0,50 valores]** Construir um sobreiro descortiçado utilizando apenas cilindros e elipsoides. Um dos troncos deverá ficar ligeiramente inclinado, sendo interposto por um ramo secundário com inclinação oposta. A copa deverá ser modelado com 1, 2 ou 3 elipsoides. O tronco e o ramo deverão apresentar uma cor castanho-alaranjada, enquanto a copa deverá ter uma tonalidade verde-escura. Colocar múltiplas instâncias deste sobreiro sobre o terreno da Tarefa 2, tendo cada instância alturas, posições e orientações distintas.
6. **[1,00 valores]** Construir uma casa alentejana recorrendo a malhas de polígonos. Por forma a facilitar a construção das malhas, sugere-se que definam apenas as faces visíveis de cada peça (portanto, não é necessário modelar as faces não visíveis!). Note-se que as janelas, porta(s) e o telhado devem estar visíveis.

Nota: O resultado deve consistir numa aproximação 'low-poly' de uma casa alentejana.

Nota: As janelas e portas podem ser coloridas com um único tom (e.g., azul), não sendo necessários efeitos decorativos nestas componentes; o telhado deve ser colorido num tom laranja.

Nota: Apesar de ser facultativo, podem modelar o bordado das paredes com barras de cores azuis ou amarelas (i.e., faixa colorida em rodapé que acompanha todo o perímetro da casa).

7. **[0,50 valores]** Sobre o terreno e a casa, deve orbitar um disco voador (i.e., um ovni). Para tal, sugere-se recorrer à utilização de uma esfera achatada para modelar o corpo da nave, a uma calote esférica para definir o cockpit, tendo múltiplas pequenas esferas colocadas radialmente no fundo da nave assim como um cilindro achatado no centro da parte de baixo. Ancorar uma luz pontual a cada pequena esfera e uma luz de holofote (*spot light*) ao cilindro, devendo esta última luz estar apontada para baixo na direção normal à nave. As luzes pontuais e a luz *spotlight* podem ser ativadas ou desativadas através da tecla 'P(p)' e tecla 'S(s)', respetivamente. A nave tem de rodar, sobre o seu eixo de simetria, a uma velocidade angular constante, podendo deslocar-se horizontalmente, a uma velocidade linear constante, recorrendo às teclas das setas. O cálculo do movimento deve ter em consideração que o utilizador pode carregar em várias teclas em simultâneo.

Nota: A iluminação com o holofote deve ser suficiente para se conseguir visualizar os objetos da cena, mas não necessita de os iluminar na íntegra.

8. **[0,25 valores]** Definir três tipos de materiais (*MeshLambertMaterial*, *MeshPhongMaterial* e *MeshToonMaterial*) por cada objeto da cena. Deve ser ainda possível alternar o tipo de sombreamento entre *Gouraud (diffuse)*, *Phong* e *Cartoon* usando as teclas 'Q(q)', 'W(w)' e 'E(e)', respetivamente. Deve ser ainda possível ativar e desativar o cálculo da iluminação usando uma tecla 'R(r)'.
9. **[0,25 valores]** Definir uma câmara fixa com uma vista sobre toda a cena utilizando uma projeção perspetiva que pode ser selecionada usando a tecla '7'. Adicionar uma *THREE.StereoCamera* à cena para que a aplicação suporte visualização estereoscópica em dispositivos de Realidade Virtual (VR). Para tal, devem seguir a documentação oficial em como criar conteúdo VR para uma aplicação web (<https://threejs.org/manual/#en/how-to-create-vr-content>). Por forma a correr a vossa aplicação num VR browser ou nos vossos smartphones, devem colocar os conteúdos do vosso projeto (i.e., index.html, sub-pasta com código JavaScript, sub-pasta com texturas) numa página online (e.g., homepage pessoal do Técnico).

Notas Importantes:

1. A biblioteca Three.js já contém as classes principais que necessitam para desenvolver os projetos desta cadeira. É por isso aconselhável que os alunos devam adotar uma programação orientada a objetos recorrendo às classes desta biblioteca, devendo sempre seguir boas práticas de programação que permitam a reutilização do código em entregas posteriores e facilitem a escalabilidade.
2. Não podem usar ferramentas de modelação 3D. As malhas das peças devem ser modeladas manualmente, vértice a vértice, face a face.
3. Para obter bons resultados na iluminação de grandes superfícies, estas devem ser subdivididas em polígonos mais pequenos.
4. Todas as texturas devem reagir à iluminação.
5. Para a utilização de texturas em modo local é necessário configurar as permissões do navegador. O problema e a solução encontram-se descritos na documentação do Three.js.

<https://threejs.org/docs/#manual/en/introduction/How-to-run-things-locally>

Alternativamente, e caso usem o Visual Studio Code como vosso editor de texto, podem instalar a extensão *Live Server* que permite criar rapidamente um servidor local:

<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>

6. Para além de dos acontecimentos de *update* e *display* existem mais um conjunto de acontecimentos, tais como teclas pressionadas ou soltas, temporizadores e redimensionamento da janela. Sugerimos vivamente que tais acontecimentos sejam tratados pelas respetivas funções de *callback* de forma independente. **Tenham em atenção que neste trabalho é requerida a implementação devida dos acontecimentos de redimensionamento da janela.**
7. A implementação de todos os trabalhos desenvolvidos nos laboratórios de Computação Gráfica deve usar o ciclo de animação (*update/display cycle*). Este padrão de desenho, usado nas aplicações de computação gráfica interativa, separa o desenho da cena no ecrã da atualização do estado do jogo em duas fases distintas. Na fase de *display* são cumpridos três passos base: limpar o buffer; desenhar a cena e forçar o processamento dos comandos. Na fase de *update* todos os objetos do jogo são atualizados conforme a física inerente. É ainda nesta fase que se processa a deteção de colisões e implementação dos respetivos comportamentos.

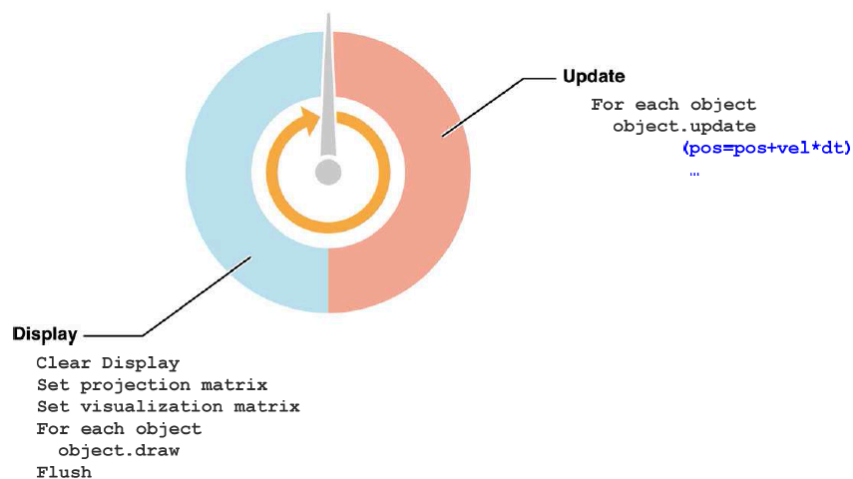


Figura 2 – Ciclo de animação com as fases de *update* e *display*.