Trabalho C

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

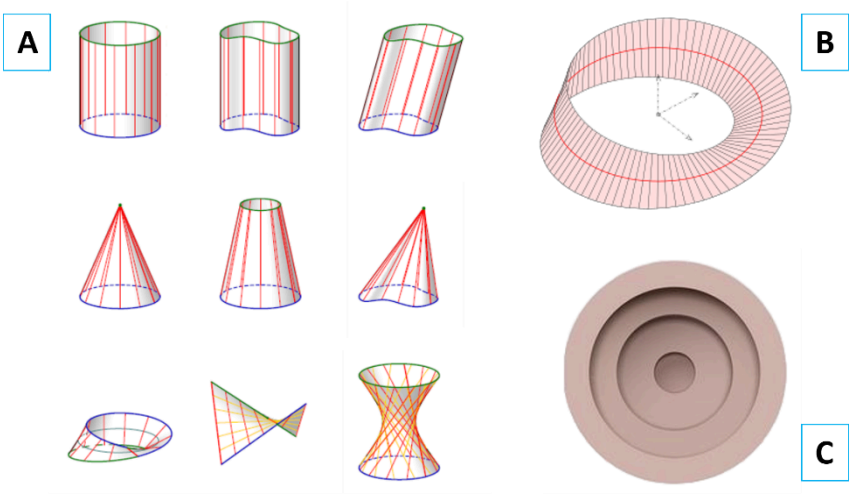

Figura 1– Referências para a elaboração de uma cena como carrossel matemático: (A) exemplos de superfícies paramétricas (e.g., superfícies regradas); (B) fita ou faixa de Möbius; (C) vista de topo da estrutura móvel do carrossel.

Objectivos

Os objectivos do terceiro trabalho de laboratório são (i) perceber as noções básicas de iluminação, (ii) os conceitos de material, (iii) as fontes de luz direccional e *spotlight,* (iv) a modelação geométrica recorrendo a superfícies paramétricas 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 **24 de Maio às 23:59h** (final da Semana 6). As discussões serão realizadas nos respectivos turnos na **1ª ou 2ª aula** da Semana 7, entre 27 a 31 de Maio. 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. **[0,50 valores]** Criar a base de um carrossel como está ilustrada na Figura 1 (C) que é composta por um cilindro central e 3 anéis concêntricos. Enquanto o cilindro central permanece na mesma posição mas roda sobre o seu eixo longitudional, os 3 anéis podem se elevar e descer, de forma independente, mas relativamente ao cilindro central, recorrendo às teclas '1', '2' e '3'. O cálculo do movimento deve ter em consideração que o utilizador pode carregar em várias teclas em simultâneo. Devem recorrer às geometrias THREE.RingGeometry, THREE.TubeGeometry ou THREE.ExtrudeGeometry.
   Nota: Não devem ser notórias quaisquer hiatos ou lacunas entre as quatro peças.

2. **[0,25 valores]** Criar uma *skydome* por forma a envolver o carrossel. Aplicar como textura um *still frame* do vídeo elaborado pelo pioneiro Oskar Fischinger An Optical Poem (1938). Esta textura deve ser aplicada sobre a malha de uma calote esférica.

3. **[1,0 valores]** Assentar em cada anel 8 superfícies paramétricas que devem estar distribuídas radialmente de forma regular (i.e., de 45º em 45º). O carrossel deve conter ao todo 8x3=24 peças. Para tal, recorrer ao THREE.ParametricGeometry. A escolha de superfícies paramétricas é livre, mas devem consistir em 8 formas diferentes, incluindo superfícies regradas (e.g., hiperbolóide de 1 folha). Colocar múltiplas instâncias destas 8 superfícies sobre os restantes anéis, tendo cada instância dimensões e orientações distintas. Cada superfície deve rodar segundo uma velocidade angular constante ao longo de um eixo de rotação que atravessa o centróide da forma, sendo a escolha do eixo e velocidade de rotação livre.

4. **[0,25 valores]** Criar a iluminação global da cena recorrendo a uma fonte de luz direccional. A fonte de luz direccional deve incidir com um ângulo diferente de zero relativamente à normal do plano xOz do referencial global da cena. Esta fonte de luz deve poder ser ligada ou desligada através da tecla ('D(d)'). Definir também uma luz ambiente com baixa intensidade com tom alaranjado.

5. **[0,50 valores]** Construir uma faixa de Möbius (Figura 1 (B)) 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!). Esta superfície deve ser colocada sobre o cilindro central como que a levitar sobre este.
   Nota: O resultado deve consistir numa aproximação 'low-poly' de uma faixa de Möbius.

6. **[0,50 valores]** Ancorar oito luzes pontuais à faixa de Möbius e uma luz de holofote (*spot light*) na base de cada superfície paramétrica, devendo estas luzes estarem apontadas de baixo para cima de modo a iluminar a forma. As luzes pontuais e as luz *spotlight* devem ser activadas ou desactivadas através da tecla 'P(p)' e tecla 'S(s)', respectivamente.
   Nota: A iluminação com o holofote deve ser suficiente para se conseguir visualizar os objectos da cena, mas não necessita de os iluminar na íntegra.

7. **[0,50 valores]** Definir quatro tipos de materiais (*MeshLambertMaterial*, *MeshPhongMaterial*, *MeshToonMaterial, MeshNormalMaterial*) por cada objecto da cena. Deve ser ainda possível alternar o tipo de sombreamento entre *Gouraud* (*diffuse*), *Phong*, *Cartoon* e *NormalMap* usando

as teclas 'Q(q)', 'W(w)', 'E(e)' e 'R(r)', respectivamente. Deve ser ainda possível activar e desactivar o cálculo da iluminação usando uma tecla 'T(t)'.

8. **[0,50 valores]** Definir uma câmara fixa com uma vista sobre toda a cena utilizando uma projecção perspectiva. 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 (*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 projecto (i.e., index.html, sub-pasta com código JavaScript, sub-pasta com texturas) numa página online (e.g., homepage pessoal do Técnico). O programa será testado em aula usando o Meta Quest 1 ou 2.

Notas Importantes:

1. A biblioteca Three.js já contém as classes principais que necessitam para desenvolver os projectos desta cadeira. É por isso aconselhável que os alunos devam adoptar uma programação orientada a objectos 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. A malha da fita de Möbius deve ser modelada 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 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 respectivas 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 interactiva, separa o desenho da cena no ecrã da

actualização 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 objectos do jogo são actualizados de acordo com a física inerente. É ainda nesta fase que se processa a detecção de colisões e implementação dos respectivos comportamentos.

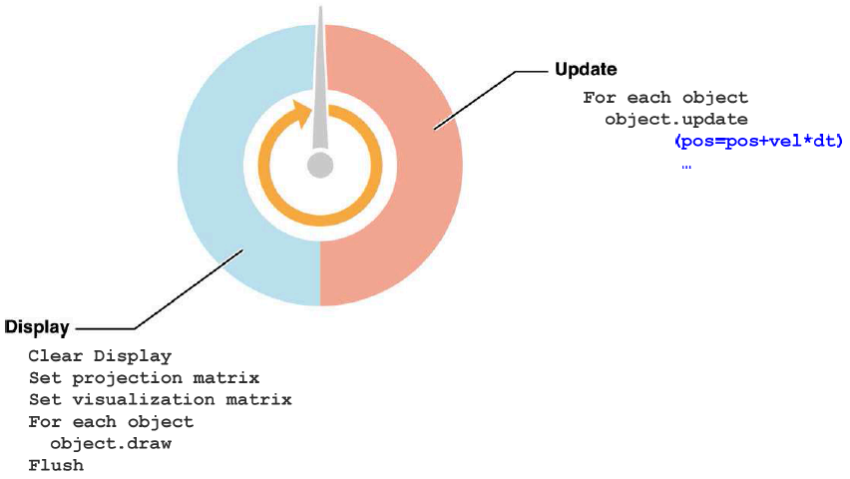

Figura 2 – Ciclo de animação com as fases de *update* e *display*.