

Aula 6

- Modelação usando malhas de triângulos
- Refinamento de modelos por subdivisão sucessiva
- Representação exata de modelos polidédricos
- Representação aproximada de modelos curvos
- Cálculo e representação do vetor normal a cada um dos triângulos de uma malha

1.1 Malhas de triângulos: refinamento por subdivisão sucessiva, usando o ponto médio de cada aresta

Analise o exemplo [incompleto WebGL_example_21.html](#).

Identifique as principais alterações relativamente ao exemplo anterior:

- Ficheiro **models.js**, para implementação de funções operando sobre a malha de triângulos definindo a superfície de cada modelo.
- Definição de funções adicionais (incompletas) no ficheiro **maths.js** para operar com pontos e vetores 3D.
- Botões adicionais, na interface, para aplicar algumas operações sobre a malha poligonal definindo cada modelo.
- Note que estas novas funcionalidades não produzem, ainda, qualquer efeito: é necessário **completar o código** de algumas funções para que isso aconteça.

Tarefas:

- No ficheiro **maths.js**, complete a função que permite determinar o ponto médio do segmento de reta definido pelos pontos **p1** e **p2**:

computeMidPoint(p1, p2)

- Analise o código, no ficheiro **models.js**, das duas funções que permitem efetuar o refinamento de cada um dos triângulos da malha, usando o **ponto médio** de cada uma das suas arestas.

- Experimente aplicar essa operação recursiva de refinamento a diferentes modelos e com diferentes níveis de recursão. Analise as características dos vários modelos obtidos, visualizando-os em modo *wireframe*.

Questão:

- Se uma malha inicial for definida por n triângulos, e sendo k o valor estabelecido para a profundidade de recursão, quantos triângulos definem a malha resultante da subdivisão recursiva?

1.2 Malhas de triângulos: refinamento por subdivisão sucessiva, usando o centróide de cada triângulo

Pretende-se desenvolver uma operação alternativa de refinamento por subdivisão recursiva: cada triângulo será subdividido em três triângulos, que têm o **centróide** do triângulo original como vértice comum.

Tarefas:

- No ficheiro **maths.js**, complete a função que permite determinar o centróide do triângulo definido pelos pontos **p1**, **p2** e **p3**:

computeCentroid(p1, p2, p3)

- Desenvolva, no ficheiro **models.js**, as duas **funções incompletas** que permitem efetuar o refinamento de cada um dos triângulos da malha, usando o respetivo centróide.
- Experimente aplicar essa operação recursiva de refinamento a diferentes modelos e com diferentes níveis de recursão. Analise as características dos vários modelos obtidos, visualizando-os em modo *wireframe*.

Questões:

- Se uma malha inicial for definida por n triângulos, e sendo k o valor estabelecido para a profundidade de recursão, quantos triângulos definem a malha resultante da subdivisão recursiva?
- Atendendo à distribuição espacial dos vértices definindo a malha final, qual dos dois processos de refinamento por subdivisão lhe parece produzir um “melhor” resultado?

1.3 Representação aproximada da esfera de raio unitário

Um modelo representando a esfera de raio unitário e centrada na origem pode ser criado do seguinte modo:

- Com base num modelo poliédrico, convexo, centrado na origem, e com um nível de detalhe apropriado.

- Deslocando cada um dos seus vértices para a superfície da esfera de raio unitário e centrada na origem; cada um dos vértices P desloca-se sobre a semirreta OP .

Tarefas:

- No ficheiro **maths.js**, complete a função que permite normalizar o vetor v :

normalize(v)

- Desenvolva, no ficheiro **models.js**, a **função incompleta** que permite deslocar cada um dos vértices da malha para a superfície da esfera unitária.
- Experimente aplicar essa operação a diferentes modelos e com diferentes níveis de detalhe. Analise as características dos resultados obtidos, visualizando-os em modo *wireframe*.

1.4 Representação do versor normal a cada triângulo

Com base no exemplo anterior, desenvolva um novo exemplo: **WebGL_example_22.html**.

Pretende-se determinar e visualizar o versor (i.e., vetor unitário) perpendicular a cada um dos triângulos da malha definindo a superfície de um modelo.

Desenvolva as funcionalidades necessárias para que isso seja possível.

Sugestões:

- No ficheiro **maths.js**, crie uma função para calcular, dados dois vetores, o resultado do seu produto vetorial.
- Armazene as componentes dos versores calculados num *array* unidimensional, à semelhança do que é feito para as coordenadas dos vértices definindo a malha.
- No ficheiro **models.js**, crie uma função para calcular e armazenar os versores pretendidos.
- Desenvolva um método para efetuar a visualização dos versores sobrepostos à malha de triângulos.