

## Aula 4

- Desenvolvimento modular usando WebGL
- Leitura de modelos a partir de ficheiro
- Tipo de projeção e definição do volume de visualização

### 1.1 Leitura de modelos a partir de ficheiro

Analise o exemplo incompleto **WebGL\_example\_16.html**.

Repare no modo como o código se encontra organizado em vários ficheiros. Analise, em particular:

- O ficheiro **WebGL\_example\_16.html**. Note: (1) por comodidade de desenvolvimento, o código dos *shaders* continua a ser definido neste ficheiro; (2) o modo como é feita a inclusão dos ficheiros **.js**.
- O ficheiro **WebGL\_example\_16.js**. Verifique as novas funcionalidades definidas: de modo particular, a possibilidade de ler o conteúdo de um **ficheiro** de texto, com um formato simples, definindo um **modelo 3D**.
- O ficheiro **maths.js** que permite instanciar e operar com matrizes e vetores para, por exemplo, definir a matriz global de transformação e aplicar transformações elementares.
- O ficheiro **initShaders.js** que permite ler o código e instanciar os *shaders*.

Identifique as novas funcionalidades disponibilizadas:

- Leitura do ficheiro contendo o modelo a representar. Analise o formato de ficheiro utilizado.
- Representação do modelo usando três modos distintos: apenas os vértices, as arestas ou as faces triangulares. **Atenção: esta funcionalidade tem de ser terminada, completando o código na função drawScene().**

**Tarefas:**

- Complete o código e teste os três modos de visualização de um modelo.

- Crie ficheiros contendo a definição de modelos simples: prisma triangular, pirâmide quadrangular, etc.
- Visualize os modelos desenvolvidos; verifique se todas as faces triangulares estão corretamente definidas.

Sugestão:

- Representar os modelos em ficheiro usando JSON.

## 1.2 Projeções 3D e definição do volume de visualização

Analise o exemplo [incompleto WebGL\\_example\\_17.html](#).

Identifique as principais alterações relativamente ao exemplo anterior:

- As funções adicionais do ficheiro **maths.js** que permitem instanciar a matriz de projeção paralela ortogonal e a matriz de projeção perspetiva, por definição dos volumes de visualização associados.
- A possibilidade de escolher o tipo de projeção usado para a visualização dos modelos.

Analise o código deste exemplo, em particular:

- O modo como é efetuada a escolha do tipo de projeção e instanciada a matriz de projeção.
- A passagem da matriz de projeção como argumento ao *vertex-shader*.
- A concatenação da matriz de projeção com a matriz global de transformação – note que a projeção é a última transformação a ser aplicada.

**Tarefas:**

- Visualize os modelos usando projeção paralela ortogonal e projeção perspetiva.
- Assegure, no caso da projeção perspetiva, que o modelo representado se encontra **no interior** do volume de visualização.
- Aplique-lhes diferentes transformações e verifique os seus efeitos.
- Desenvolva funcionalidades que permitam alterar as características do volume de visualização associado à projeção selecionada.

Sugestão:

- Modificar o código de modo a instanciar dois ou mais modelos na cena.