Aula 5

- Animação
- Instanciação de mais do que um modelo na cena
- Composição de transformações locais a um modelo com transformações globais à cena
- Remoção de faces ocultas usando o *"Depth-Buffer"*

1.1 Animação

Analise o exemplo incompleto **WebGL_example_18.html**.

Identifique as principais alterações relativamente ao exemplo anterior:

- As funções auxiliares adicionais, do ficheiro **webgl-utils.js**, que permitem desenvolver código de modo independente do *browser* usado.

- A função **animate()**, que permite atualizar os parâmetros das transformações aplicadas em função do tempo decorrido.

- Os parâmetros controlando as transformações aplicadas estão definidos como variáveis globais.

- A função **tick(),** que permite redesenhar periodicamente o conteúdo do *canvas*.

- Deste modo, a função **drawScene()** deixa de ser invocada após o processamento de cada evento.

- Quando o modelo é representado em projeção perspetiva, é necessário efetuar uma translação para colocá-lo no interior do volume de visualização – ver **drawScene()**.

- O controlo da rotação em torno de YY usando os botões definidos – ver os respetivos *event listeners*.

Tarefa:

- Acrescente a possibilidade de rodar em torno dos eixos XX e ZZ, e de controlar essas transformações usando botões.

Sugestões:

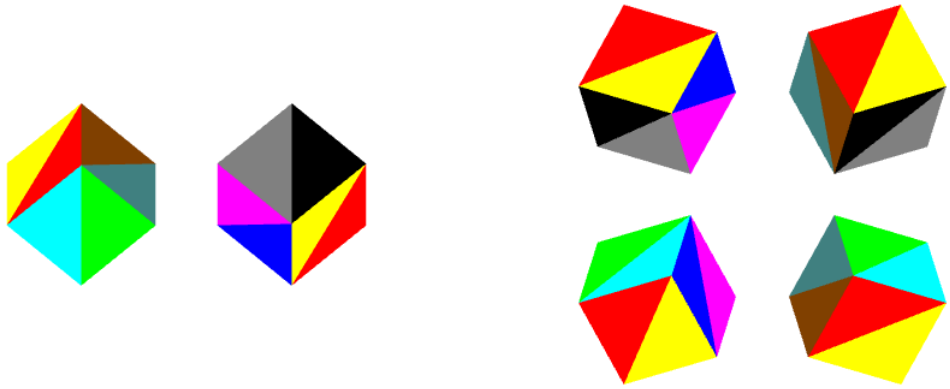
- Animar um movimento de translação na horizontal.

- Usar o teclado para controlar as várias transformações que podem ser aplicadas ao modelo.

1.2 Instanciação de mais do que um modelo na cena

Com base no exemplo anterior, desenvolva um novo exemplo: **WebGL_example_19.html**.

Deve ser possível visualizar mais do que uma instância do cubo: por exemplo, dois cubos, numa primeira versão, e quatro cubos, numa segunda versão.



No caso da projeção perspetiva, deve ser assegurado que os vários modelos se encontram no interior do volume de visualização.

Tarefas:

- Modifique, de modo apropriado, a função **drawScene().**
- Desenvolva uma animação da cena, em que cada cubo tem um comportamento diferente.
- Comece pelo caso mais simples, em que cada cubo roda em torno do seu eixo vertical, possivelmente com direções e velocidades de rotação diferentes.
- Acrescente, depois, a possibilidade de rotação em torno dos outros eixos coordenados.

Questão:

- A utilização de ***"Back-Face Culling"*** é suficiente para que os vários cubos sejam corretamente visualizados? Porquê?

Sugestão:

- Desenvolva uma função auxiliar que permita instanciar um cubo e aplicar-lhe as suas transformações próprias.

1.3 Composição de transformações locais com transformações globais à cena

Analise o exemplo **WebGL_example_20.html**.

Identifique a principal alteração relativamente ao exemplo anterior:

- Além das suas transformações próprias, os dois modelos apresentam um comportamento global.

Questões:

- Que movimento têm os modelos?
- Algum dos modelos se encontra sempre mais próximo do observador?
- Há alguma coisa "estranha" no desenrolar da animação?

Tarefas:

- Ativar/desativar o **teste de profundidade** (*"Depth-Buffer"*), de modo a que a visibilidade das faces dos modelos seja corretamente determinada.
- Perceber a utilidade do **teste de profundidade** (*"Depth-Buffer"*) para que os modelos sejam corretamente representados.
- Perceber em que situações se deve efetuar a remoção das faces ocultas usando ***"Back-Face Culling"***, e em que situações de deve usar o **teste de profundidade** (*"Depth-Buffer"*).

Analise o código deste exemplo, em particular:

- A função **drawScene()**, que permite visualizar os dois modelos e aplicar-lhes várias transformações.
- A função **drawModel()**, que permite visualizar um modelo e aplicar-lhe a transformação que resulta da concatenação das transformação global à cena com as suas transformações próprias.
- Note o modo como são definidas e aplicadas as transformações globais à cena.

Tarefas:

- Controlar a velocidade e a direção de rotação em torno do eixo vertical.
- Aplicar outras transformações globais à cena e controlar os seus parâmetros. Por exemplo, permitir rotações em torno de outros eixos.