Aula 8

- Definição de mais do que um foco de luz para a mesma cena.
- Armazenamento de modelos poligonais em ficheiro: utilização de uma versão simplificada do formato OBJ.
- Representação exata de modelos poliédricos vs. Representação aproximada de modelos curvos.
- A Técnica de Gouraud para estimação dos vetores normais.
- Análise, para um mesmo modelo, do efeito da associação de diferentes vetores normais aos seus vértices.
- Propriedades dos Materiais: estabelecimento das propriedades de diferentes materiais; efeito na representação final dos modelos.

1.1 Vários focos de luz pontuais

Analise o exemplo incompleto **WebGL_example_24.html**.

O objetivo deste exemplo é instanciar mais do que um foco de luz pontual para iluminar um modelo 3D, usando o modelo de iluminação de Phong.

Identifique as principais alterações relativamente ao exemplo anterior:

- Definição de uma classe que permite instanciar vários focos de luz, com diferentes características. Verifique quais os atributos e métodos associados a cada instância.

- Possibilidade de aplicar transformações de rotação independentes a cada um dos focos, que passam a poder ter um comportamento animado.

- Criação de um *array* global que permite armazenar os focos instanciados.

- Alteração da estrutura global da função **computeIllumination()**, de modo a suportar a existência de mais do que um foco de luz e a poder aplicar transformações de rotação a cada foco. Verifique como é definida e aplicada a matriz de transformação associada a cada foco.

- Alteração da função **animate()**, de modo a animar a rotação independente de cada um dos focos de luz. Verifique como é atualizado o ângulo de rotação associado a cada foco.

Questões:

- Onde estão situados os focos de luz que iluminam a cena? Quais são as suas características?
- Que tipo de movimento estão a efetuar? Qual é o mais rápido?
- Qual é a cor atribuída ao modelo? Quais são as propriedades do material que lhe está associado?

Tarefas:

- Visualizar os modelos do cubo, do tetraedro e da **esfera**, com diferentes níveis de detalhe.
- Alterar os coeficientes de reflexão e verificar o efeito dessas alterações.

Tarefa:

Crie mais focos de luz na cena:

- Um foco frontal estático, emitindo luz verde.
- Um foco situado abaixo do modelo e emitindo luz branca. Acrescente-lhe um movimento de rotação em torno do eixo XX'.

Sugestões:

Acrescente funcionalidades que permitam:

- Alterar os coeficientes de reflexão associados ao material definindo o modelo.
- Alterar a cor de um foco de luz selecionado.
- Alterar a posição espacial de um foco de luz selecionado e a sua distância à cena (foco direcional vs. foco pontual).

1.2 Vários focos de luz pontuais — GPU — *Per Vertex Shading*

Considere agora a versão do exemplo anterior em que o cálculo do modelo de iluminação é efetuado na GPU: **WebGL_example_24_GPU_per_vertex.html**.

Tarefas:

- Instancie os mesmos focos de luz que no exemplo anterior.
- Compare visualmente os resultados de iluminação obtidos, para modelos com baixo nível de detalhe.

Sugestão:

- Crie uma cena com 4 esferas e analise os efeitos de iluminação obtidos.

1.3 Superfícies poliédricas vs. Superfícies curvas — Utilização do formato OBJ

Analise o exemplo incompleto **WebGL_example_25.html**.

O objetivo deste exemplo é permitir ler modelos definidos em ficheiro, usando uma versão simplificada do **formato OBJ**.

Cada um desses ficheiros contém habitualmente, além das **coordenadas dos vértices** definindo a superfície do modelo, um **vetor normal** unitário associado a cada vértice.

Analise a função que permite a leitura de ficheiros definidos na versão simplificada do formato OBJ.

Tarefa:

Carregue os modelos definidos nos ficheiros:

- **cubo.OBJ** e **cuboGouraud.OBJ**
- **tetraedro.OBJ** e **tetraedroGouraud.OBJ**

Compare os efeitos obtidos.

Veja que informação se encontra armazenada em cada um dos ficheiros.

Tarefa:

Carregue os modelos definidos nos ficheiros:

- **prismaTriangular.OBJ** e **cilindroAproxPrismaTriang.OBJ**
- **prismaHexagonal.OBJ** e **cilindroAproxPrismaHexagonal.OBJ**

Rode os modelos em torno do eixo YY'. Compare os efeitos obtidos.

Rode-os também em torno dos outros eixos coordenados.

Veja que informação se encontra armazenada em cada um dos ficheiros.

Questão:

Quais é o erro no modelo que define o prisma triangular?

1.4 Efeito das características dos materiais

Tarefa:

Acrescente ao exemplo anterior a possibilidade de atribuir diferentes características ao material definindo cada modelo.

Use a tabela definida no ficheiro **Caracteristicas_Materiais.pdf**.

1.5 Efeito das características dos materiais — GPU — *Per Vertex Shading*

Considere agora a versão do exemplo anterior em que o cálculo do modelo de iluminação é efetuado na GPU: **WebGL_example_25_GPU_per_vertex.html**.

Tarefa:

- Compare visualmente os resultados de iluminação obtidos, para modelos com diferentes níveis de detalhe.

1.6 Efeito das características dos materiais — GPU — *Per Vertex Shading*

Considere agora o exemplo adicional: **WebGL_example_NEW.html**.

Foram criadas funcionalidades que permitem instanciar modelos 3D básicos (cubo, tetraedro, esfera), e atribuir-lhes diferentes características geométricas (posição, tamanho, etc.) e propriedades materiais.

Para isso, foi

- Criado o ficheiro **sceneModels.js**, onde estão definidas as funções que permitem **instanciar os modelos** desejados e é definido o ***array*** que contém as instâncias dos modelos a visualizar.

- Adaptadas algumas das funções já existentes: **drawScene()**, **drawModel()** e **initBuffers()**.

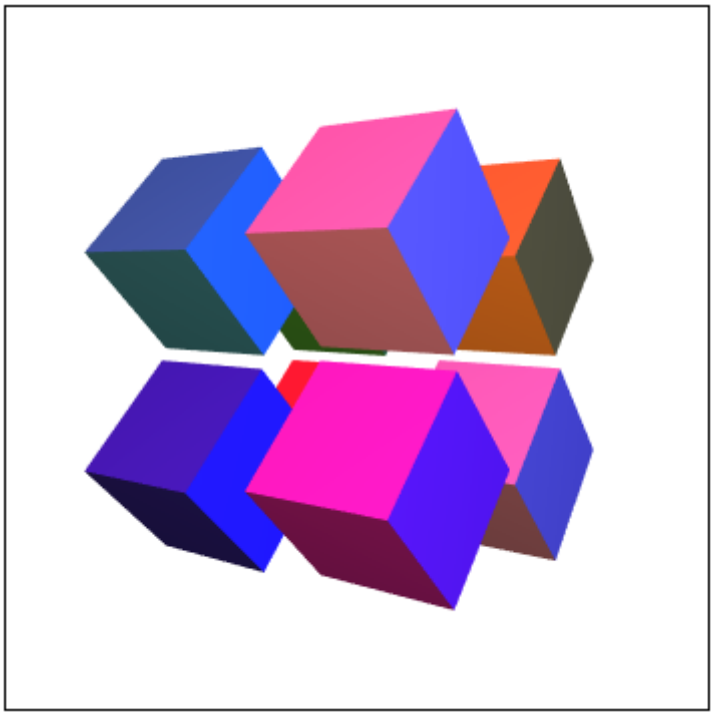
Tarefa:

- Analise o ficheiro **sceneModels.js** e verifique como é possível instanciar vários modelos com diferentes características.

- Note o modo como é possível instanciar um paralelepípedo a partir do modelo que representa um cubo, por alteração simples dos seus fatores de escala.

Sugestões:

- Crie um novo exemplo onde a cena é constituída por **quatro cubos** com diferentes características e posicionados como se mostra na figura. Os cubos deverão rodar em torno de um eixo vertical.



- Crie outro exemplo em que a cena seja constituída por algumas dezenas de modelos cujo tipo e características são definidos de modo aleatório. A figura abaixo ilustra um possível resultado.