

Aula 2

- *Shaders: Vertex-Shader e Fragment-Shader*
- A linguagem GLSL (*OpenGL Shading Language*)
- Estrutura básica de uma aplicação usando *shaders*
- Desenho de primitivas 2D

1.1 Desenhar um triângulo — Primeiro exemplo de utilização de *shaders*

Analise o exemplo **WebGL_example_06.html**.

Repare no modo como o conteúdo do ficheiro se encontra organizado.

Identifique as principais modificações relativamente aos exemplos anteriores:

- Organização do código.
- Definição, compilação e “*linkagem*” dos *shaders*.
- Invocação dos *shaders* e passagem do *array* de vértices como argumento ao *vertex-shader*.

Algumas questões:

- Quais são as coordenadas dos vértices do triângulo?
- Como é atribuída uma cor ao triângulo?
- Como está definida a janela de *clipping*?
- O que acontece se alterarmos as dimensões do *viewport*?

Sugestões:

- Desenhar um retângulo usando dois triângulos.
- Desenhar uma figura mais complexa, composta por vários triângulos.
- Desenhar triângulos que estejam apenas parcialmente contidos no interior da janela de *clipping*.

1.2 As primitivas gráficas do WebGL

Analise o exemplo incompleto **WebGL_example_07.html**.

Note o modo como é apresentada a informação relativa a cada primitiva gráfica do WebGL.

Algumas questões:

- Quais são as coordenadas dos vértices?
- Qual é a ordem da sua definição?
- Como são definidas as sucessivas primitivas?

Tarefa:

- Desenhar os vários tipos de primitivas gráficas, mas usando sempre o mesmo conjunto de vértices.
- O utilizador seleciona, o tipo de primitiva a desenhar: `LINES`, `LINE_STRIP`, `LINE_LOOP`, `POINTS`, `TRIANGLES`, `TRIANGLE_STRIP` ou `TRIANGLE_FAN`.

1.3 Desenhar um triângulo — Atribuir diferentes cores aos vértices

Analise o exemplo incompleto **WebGL_example_08.html**.

Identifique a principal modificação relativamente aos exemplos anteriores:

- Atribuição de uma cor a cada vértice.
- Invocação dos *shaders* e passagem de dois *arrays* (globais) armazenando as coordenadas e o atributo de cor de cada vértice, como argumentos, ao *vertex-shader*.

Algumas questões:

- Como é atribuída uma cor a cada vértice do triângulo?
- O que acontece se forem atribuídas cores diferentes a cada vértice de um triângulo?

Tarefa:

- Completar o exemplo, de modo a poder atribuir uma cor diferente a cada um dos vértices do triângulo.

1.4 Desenhar vários triângulos: O Triângulo de Sierpinski (*The Sierpinski Gasket*)



[Wikipedia]

Analise o exemplo **WebGL_example_09.html**.

Repare no modo como o conteúdo do ficheiro se encontra organizado.

Analise o algoritmo recursivo que permite definir o fractal.

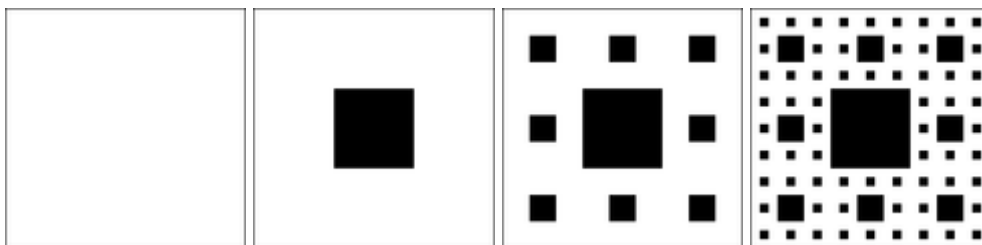
Algumas questões:

- Como são calculadas as coordenadas dos sucessivos vértices?
- Que coordenadas são armazenadas no *array* de vértices?

Sugestões:

- Pedir ao utilizador a altura da árvore de recursão.
- Pedir ao utilizador as coordenadas dos vértices do triângulo inicial.
- Armazenar em ficheiro as coordenadas dos vértices definindo o fractal construído, para permitir a sua reutilização.

1.5 O Quadrado de Sierpinski (*The Sierpinski Carpet*) – OPCIONAL

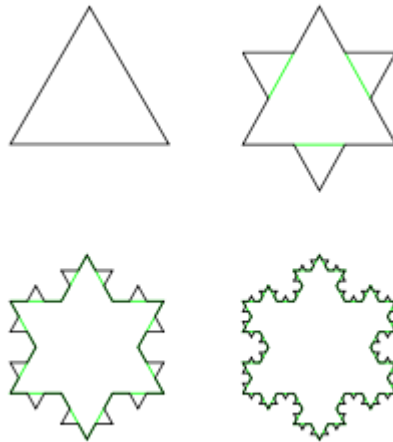


[Wikipedia]

Com base no exemplo anterior, desenvolva um novo exemplo (**WebGL_example_10.html**) que permita visualizar o Quadrado de Sierpinski, com diferentes níveis de recursividade.

Em primeiro lugar, estabeleça as regras para o processo de subdivisão. Note que cada quadrado é representado por dois triângulos

1.6 Desenhar segmentos de reta: A Curva de Koch (*The Koch Snowflake*)



[Wikipedia]

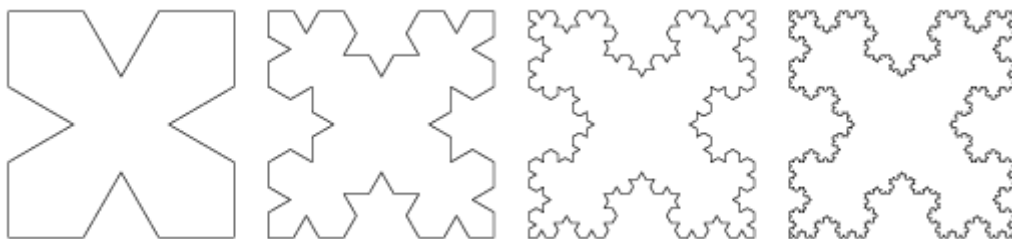
Analise o exemplo **WebGL_example_11.html**.

Analise o algoritmo recursivo que permite definir o fractal.

Algumas questões:

- Como são calculadas as coordenadas dos sucessivos vértices?
- Que coordenadas são armazenadas no *array* de vértices?

1.7 O Fractal de Cesàro – OPCIONAL



[Mathworld]

Com base no exemplo anterior, desenvolva um novo exemplo (**WebGL_example_12.html**) que permita visualizar o Fractal de Cesàro, com diferentes níveis de recursividade.

Em primeiro lugar, estabeleça as regras para o processo de subdivisão.