3D Fractais

André Cardoso, 65069, marquescardoso@ua.pt

Jorge Faustino, 64441, jorgebalseiro@ua.pt

***Resumo* – Este artigo apresenta fractais em 3D, uma breve análise e como foi feita a sua implementação.**

Começa por definir o problema de um ponto de vista geral e teórico, seguindo para uma abordagem mais detalhada relativamente a cada fractal implementado. Por último é abordada a arquitectura do projecto relativamente à interface e implementação WebGL e as várias técnicas do mesmo utilizadas no projecto.

I. INTRODUÇÃO

No âmbito da Unidade Curricular Computação Visual do 4º ano do curso Mestrado Integrado em Engenharia de Computadores e Telemática escolhemos a realização de um projecto em Web Graphics Library (WebGL)[1] no qual desenvolvemos uma aplicação de manipulação de Fractais [2] em 3D.

WebGL é uma Interface de Programação de Aplicações (API) [3] desenvolvida em JavaScript[4] para renderizar gráficos 2D e 3D em web browsers. Actualmente encontra-se integrado em todos os web browsers permitindo a páginas web mostrarem gráficos gerados pela GPU.[5] Permitindo assim uma maior facilidade na criação e desenvolvimento de aplicações Web do que utilização da framework Open Graphics Library (OpenGL)[6] no qual WebGL é baseado.

Fractais são figuras geométricas que exprimem um padrão repetido representado em várias escalas, também conhecidos por *"self-similar"*.[2] Em que cada parte do fractal contém o mesmo padrão sucessivamente. Quando a simetria é exacta em várias escalas. Alguns exemplos conhecidos desse tipo de simetria são apresentados na Figura 1 como é o caso das folhas das plantas no meio natural.

Figure 1- Folha da planta Fetos.

II. IMPLEMENTAÇÃO

Nesta secção vamos apresentar todos os fractais implementados e podem ser visualizados no conteúdo dentro do ficheiro do tipo Winrar[7] que é acompanhado com este artigo.

Como esta solução usa extensivamente o GPU devido á recursividade, para os cálculos da iluminação das figuras, decidiu-se usar os fragment shader. Pois utiliza iluminação para cada pixel em vez de ser por vértices. Para tal foi criada uma matriz nova que contém a transposição da matriz inversa da matriz.

Durante toda a implementação foi usada os recursos disponíveis no site da unidade curricular dedicados às aulas práticas.[8]

As próximas subsecções contêm os vários modelos explorados: Sierpinki Gasket[9], Koch Snowflake[10], Menger Sponge[11], Mosely Snowflake[12]. Assim como um fractal desenvolvido por nós.

A. Sierpinski Gasket

Este fractal é baseado na subdivisão recursiva feita por tetraedros, como demonstra a Figura 2, que contém o estado original, com uma e com duas incrementações sucessivamente.

Figure 2- Representação de Sierpinski Gasket

Para obter o resultado pretendido são selecionados os vértices originais da figura, e é calculado um ponto intermédio cada par. Como é apresentado na Figura 3.

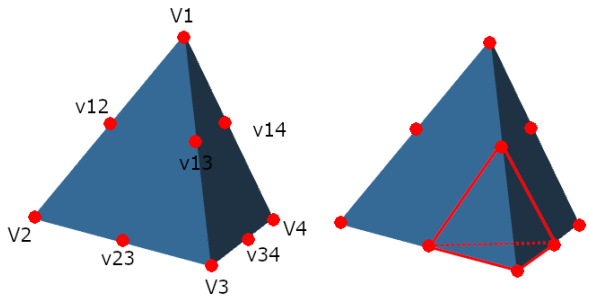

Figure 3- Representação do método de cálculo.

Por fim a nova lista de vértices com os pontos intermédios é introduzida, formando assim diferentes arestas que dão forma a quatro tetraedros no lugar de um.

B. Koch Snowflake

Com a mesma estrutura inicial do exemplo anterior, é possível criar um outro tipo de fractal. consiste em acrescentar um novo tetraedro mais pequeno, perpendicularmente, cada uma das faces como mostra a Figura 4.

Figure 4 - Representação de Kock Snowflake.

Na Figura 5, para cada face, são selecionados os vértices e é calculado o ponto intermédio entre cada par.

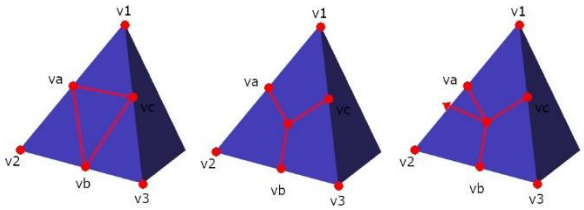

Figure 5 - Representação do método de cálculo.

É calculado o ponto central entre esses 3 novos vértices, e nesse ponto central é usado o cálculo do vector normal através dos vértices originais (v1,v2 e v3), que como estão no sentido anti-horário vão gerar um vector normal a "apontar para fora". As coordenadas da extremidade desse vector irão formar o 4º vértice do novo tetraedro.

C. Menger Sponge

A *sponge* é formada através da divisão recursiva de um cubo em 27 cubos, removendo os cubos no centro de cada face, e no seu interior. Figura 6, representa o estado do fractal nas várias incrementações.

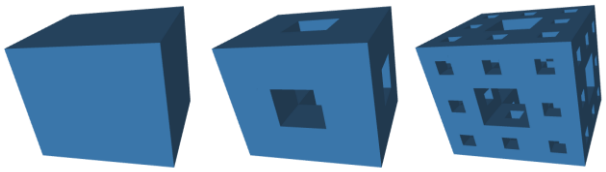

Figure 6 - Representação de Menger Sponge.

Para tal são selecionados os seus 8 vértices, é calculada a distância entre 2 vértices da mesma face, por exemplo o v3 e o v4, e é usado o terço dessa distância para calcular a posição dos novos vérticies.

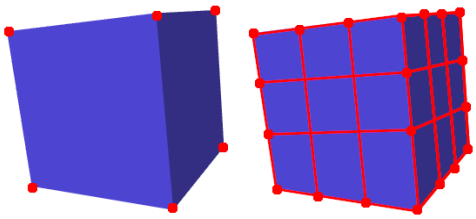

Figure 7 - Representação do método de cálculo.

D. Mosely Snowflake

Neste fractal, divide-se um cubo em 27 cubos mais pequenos e remove-se os cubos dos cantos. Figura 8, representa o estado do fractal nas várias incrementações.

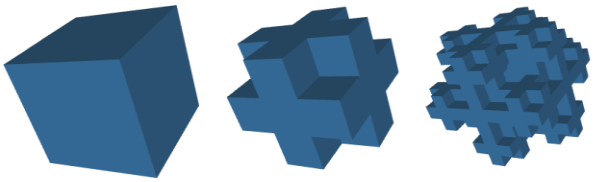

Figure 8 - Representação de Mosely Snowflake.

São selecionados os vértices originais, é calculado o ponto que fica a um terço da distância até a todos os outros vértices, como demostra a Figura 9.

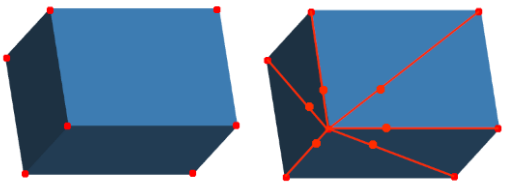

Figure 9 - Representação do método de cálculo.

Depois é calculado o ponto que fica a dois terços de distância até a todos os outros vértices.

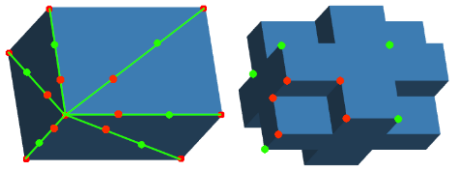

Figure 10 - Continuação da representação do método de cálculo.

Os vértices originais são descartados e a renderização é feita com os novos vértices.

E. Custom Design

Foi um fractal que inventámos, baseado no *Menger Sponge*,

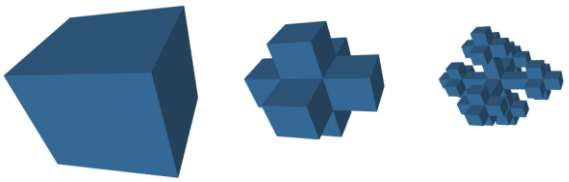

Figure 11 - Representação de Custom Design.

O seu cálculo é também semelhante ao do fractal *Menger Sponge*, o cubo original é dividido em 27, e são escolhidos os cubos centrais.

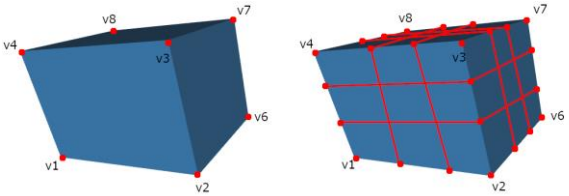

Figure 12 - Representação do método de cálculo.

III. RESULTADO FINAL

O nosso projecto baseia-se em tecnologias web já mencionadas, como tal para o executar apenas é necessário um browser. Para obter uma melhor experiência de utilização foi usado o Bootstrap para customização de estilos.

A Figure 13 apresenta interface gráfica é composta no lado direito pelo canvas que renderiza o Fractal escolhido com o devido número de incrementações. No lado esquerdo é possível controlar a interação com esse mesmo Fractal escolhido. É possível controlar a rotação e a respectiva velocidade de todos os eixos. Abaixo temos uma tabela de controlo onde é possível escolher o fractal, o tipo de projeção e o modo de renderização. Assim como um botão para voltar ao estado inicial.

Figure 13 - Interface da aplicação final.

Os tipos de projecções possíveis são Orthogonal Projection[13] e Perspective Projection.[14]

O modos de renderização são Filled Triangles[15], Wireframe e Vertices.

IV. CONCLUSÃO

Os fractais são em si objetos complexos, mas com utilidade em vários campos, para compreender certos objetos e padrões encontrados na natureza também são usadas em nas áreas cinemáticas e jogos.

Este trabalho foi muito importante para o nosso conhecimento, o aprofundamento deste tema fez-nos aprender e compreender o método utilizado na manipulação de objetos 3D utilizando algoritmos recursivos. Cumprimos todos os objetivos que nos foram propostos neste projeto em que o principal era a projeção de um Fractal 3D.

V. REFERENCES

[1] "WebGL." [Online]. Available: https://www.khronos.org/webgl/. [Accessed: 20-Nov-2018].

[2] "Fractal." [Online]. Available: https://en.wikipedia.org/wiki/Fractal. [Accessed: 20-Nov-2018].

[3] Margaret Rouse, "API." [Online]. Available: https://whatis.techtarget.com/definition/API-gateway-application-programming-interface-gateway. [Accessed: 20-Nov-2018].

[4] Margaret Rouse, "JavaScript," 2018. [Online]. Available: http://searchmicroservices.techtarget.com/definition/JavaScript. [Accessed: 02-May-2018].

[5] Margaret Rouse, "GPU." [Online]. Available: https://searchvirtualdesktop.techtarget.com/definition/GPU-graphics-processing-unit. [Accessed: 20-Nov-2018].

[6] "OpenGL." [Online]. Available: https://www.opengl.org/. [Accessed: 20-Nov-2018].

[7] "Winrar." [Online]. Available: https://www.winrar.com/start.html?&L=0. [Accessed: 20-Nov-2018].
[8] J. Madeira, "Recursos das aulas." [Online]. Available: http://sweet.ua.pt/jmadeira/WebGL/. [Accessed: 20-Nov-2018].
[9] "Sierpinki Gasket." [Online]. Available: https://en.wikipedia.org/wiki/Sierpinski_triangle. [Accessed: 20-Nov-2018].
[10] "Koch Snowflake." [Online]. Available: https://en.wikipedia.org/wiki/Koch_snowflake. [Accessed: 20-Nov-2018].
[11] "Menger Sponge." [Online]. Available: https://en.wikipedia.org/wiki/Menger_sponge. [Accessed: 20-Nov-2018].
[12] "Mosely Snowflake." [Online]. Available: https://en.wikipedia.org/wiki/Mosely_snowflake. [Accessed: 20-Nov-2018].
[13] "Orthogonal Projection." [Online]. Available: https://en.wikipedia.org/wiki/Projection_(linear_algebra). [Accessed: 20-Nov-2018].
[14] "Perspective Projection." [Online]. Available: http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/projection/perspective_projection.html. [Accessed: 20-Nov-2018].
[15] "Filled triangles." [Online]. Available: http://www.gabrielgambetta.com/computer-graphics-from-scratch/filled-triangles.html. [Accessed: 20-Nov-2018].