TPC 3 – MSSN

Licenciatura em Engenharia Informática e Multimédia



**Realizado**:

Pedro Henriques – A45415

Vasco Cruz – A42837

**Conteúdo**

[Função logística 3](#_Toc61116709)

[Jogo do Caos 5](#_Toc61116710)

[Gramáticas de Lindemayer 7](#_Toc61116711)

[Conjuntos de Julia e Mandelbrot 9](#_Toc61116712)

# **Função logística**

Considere a função logística, f(x) = rx(1-x). Usando uma qualquer ferramenta à sua escolha (máquina de calcular, excel, sagemath, etc.)

* 1. Encontre valores do parâmetro r que resultem em trajetórias (órbitas, sequências temporais) que convirjam para:
     1. Ponto fixo igual a zero;

R: 0.5

* + 1. Ponto fixo diferente de zero;

R: 1.3

* + 1. Ciclo limite com período 2;

R: 3.2

* + 1. Ciclo limite com período 4;

R: 3.5

* + 1. Ciclo limite com período 8;

R: 3.57

* + 1. Ciclo limite com período 3;

R: 3.853

* + 1. Ciclo limite com período 5;

R: 3.74

* + 1. Ciclo limite com período 6;

R: 3.627

* + 1. Atractor aperiódico (caótico).

R: 4.0

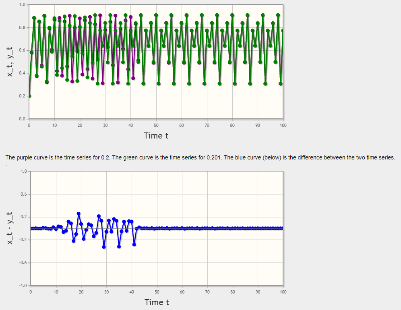
* 1. Mostre que para todos os casos anteriores (exceto o último) a trajetória obtida não é sensível às condições iniciais.

Figura 8 - R = 3.627

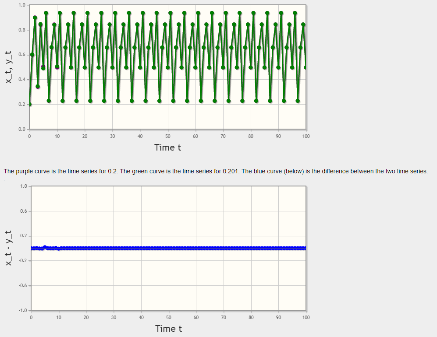


Figura 7 - R=3.74

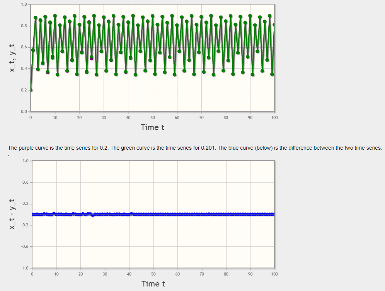


Figura 5 - R = 3.57

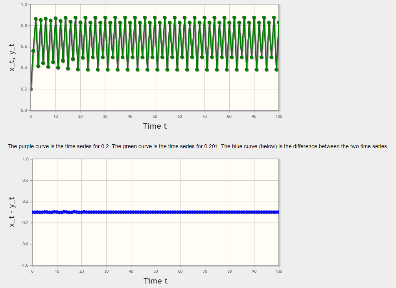


Figura 4 - R = 3.5

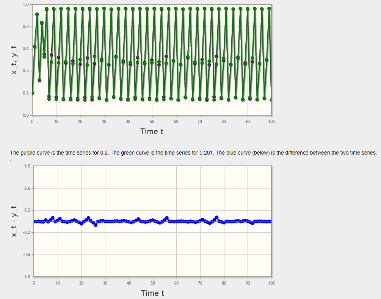


Figura 6 - R = 3.857

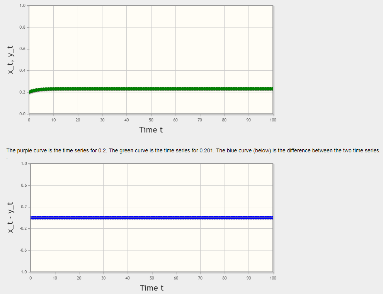


Figura 2 - R = 1.3

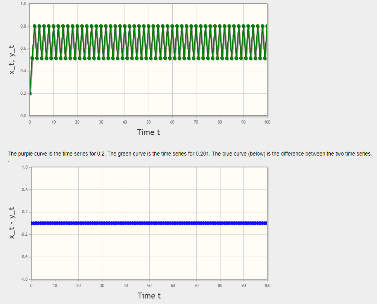


Figura 3 - R = 3.2

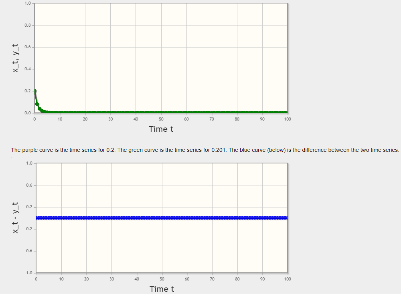


Figura 1 R = 0.5

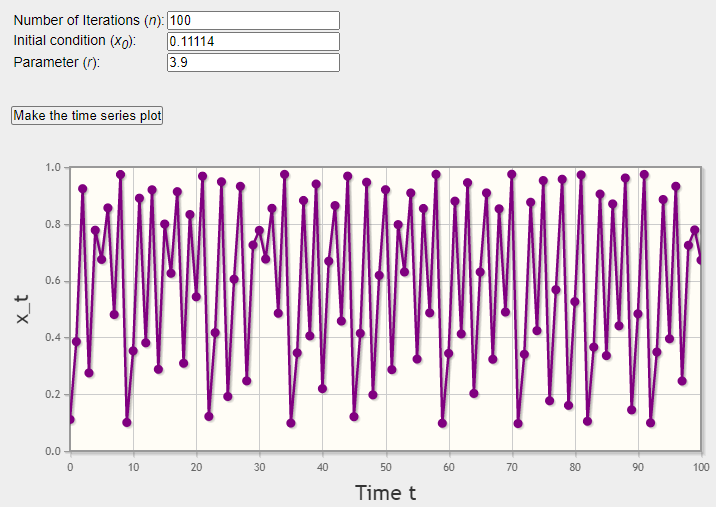
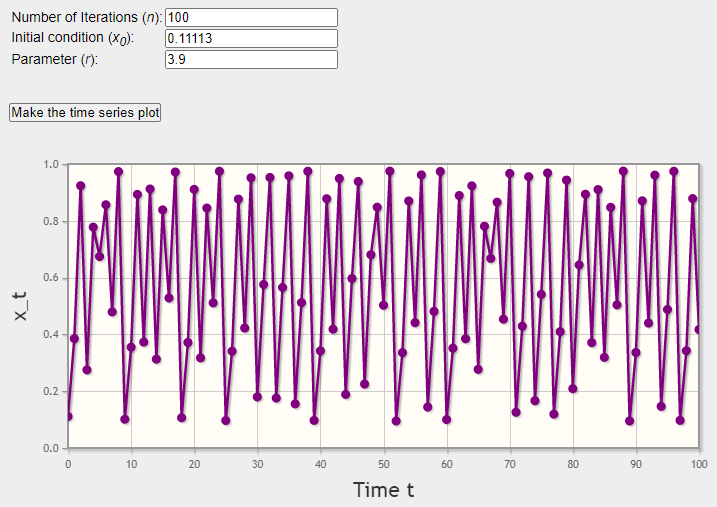
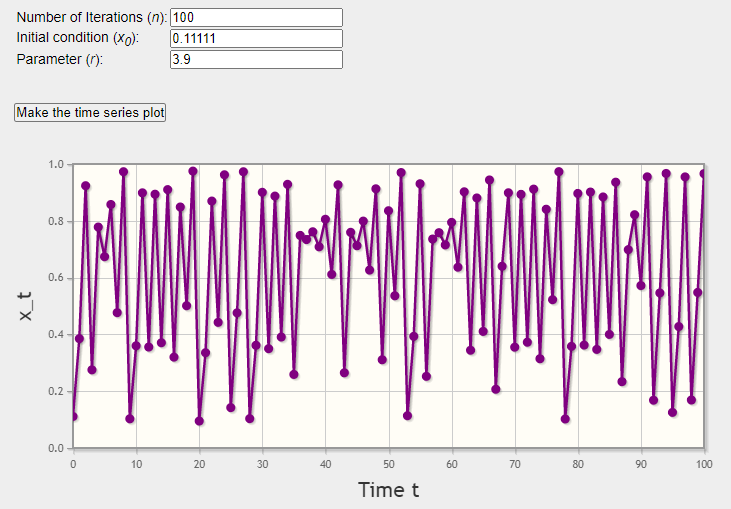
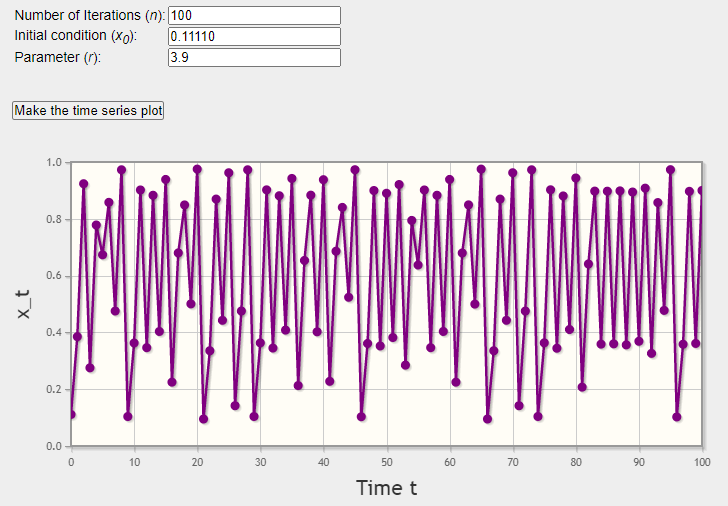
Como é possível verificar a maioria das imagens possui um gráfico de diferenças com uma linha azul horizontal, ou seja, a diferença é muito pequena ou constante. Mesmo naqueles gráficos onde não se verifica a linha, a diferença máxima é muito pequena, cerca de 0.2 valores, convergindo rapidamente para o ciclo com um ponto fixo muito próximo.

* 1. Explique no que consiste o chamado "efeito borboleta" e ilustre esse fenómeno para o caso da função logística

R: O efeito borboleta é um fenómeno natural em que uma pequena alteração às condições iniciais geram consequências imprevisíveis. Tais como uma borboleta a bater as asas em Tóquio, nunca iria prever que uma consequência das suas ações seria a criação de um furacão em Nova Iorque.

Um exemplo da função logística tem em conta valores de R próximos de 4, podendo existir outros. Tendo em conta as imagens que seguem onde se inclui o R e a condição inicial.

Figura 1 - Caos, função logística



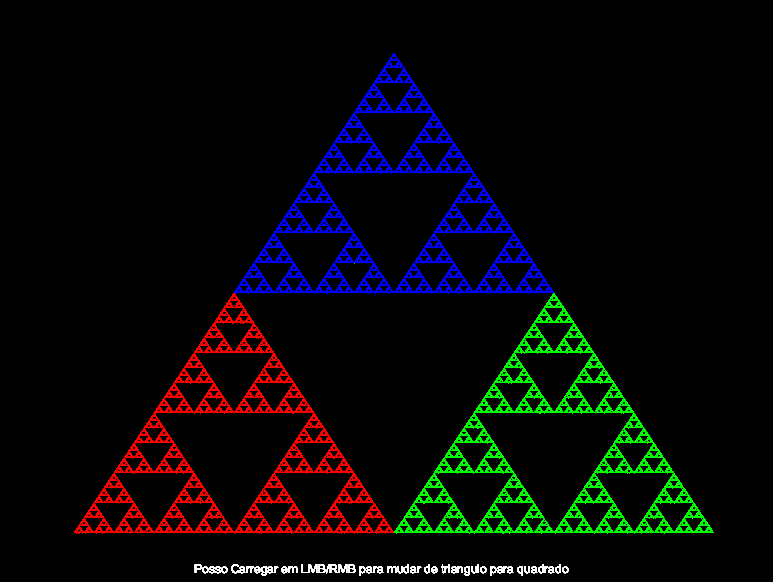
Tendo em conta as imagens de cima é possível verificar que uma alteração mínima na ordem do 10 gera alterações caóticas, levando ao que parecem ciclos limites nalguns casos, mas que não se mantêm, ou puramente caos como é o caso da segunda imagem.

# **Jogo do Caos**

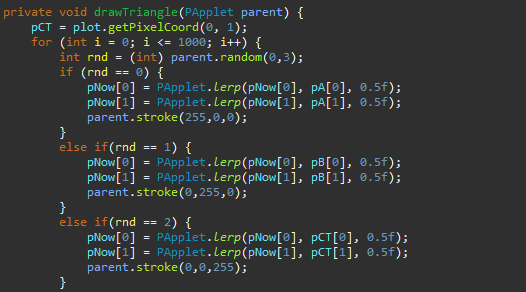
O jogo do caos está dividido em duas classes diferentes:



Como o próprio nome indica, a primeira são duas simulações do jogo do caos, a primeira um triangulo que resulta no famoso triângulo de sierpinski:

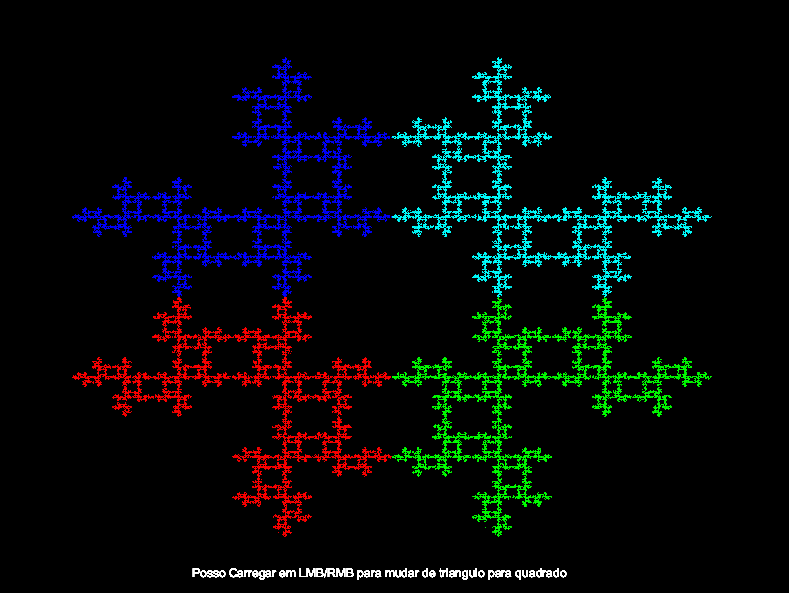


Obtido através do seguinte código:

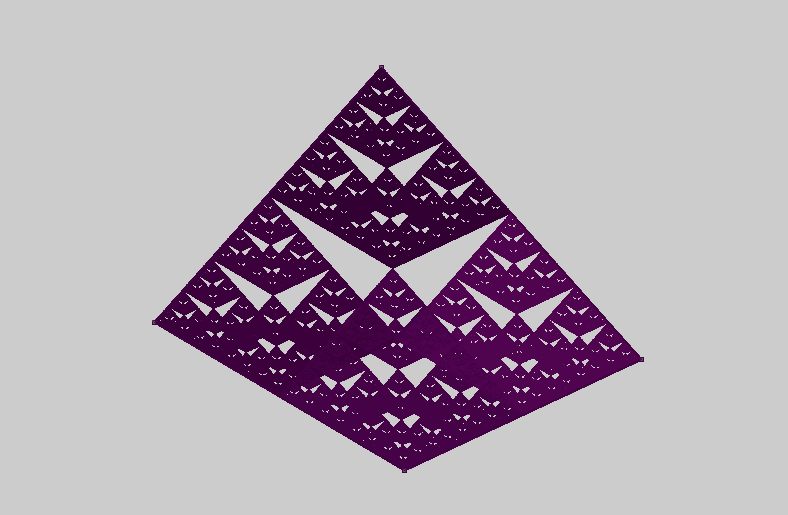


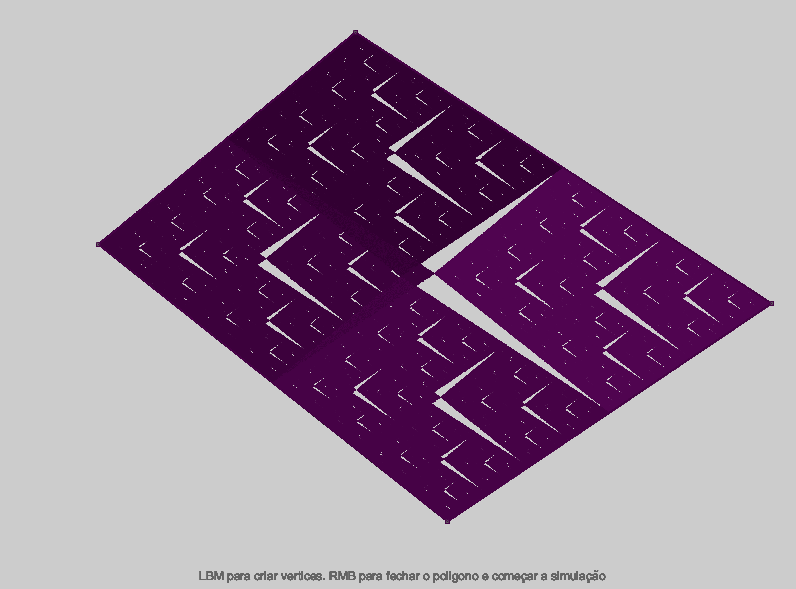
De facto, não há muito mais a dizer pois é uma implementação do algoritmo do jogo do caos descrito no próprio enunciado e por isso se for para pintar no vértice 0, pinto o ponto de uma cor, e repito isso para os vértices do triangulo. Depois é só uma questão de encontrar as melhores cores para o caso.

Como se pode ver, temos ainda outro caso implementado:



Que é implementado de maneira idêntica ao triangulo mas com mais um vértice obviamente.

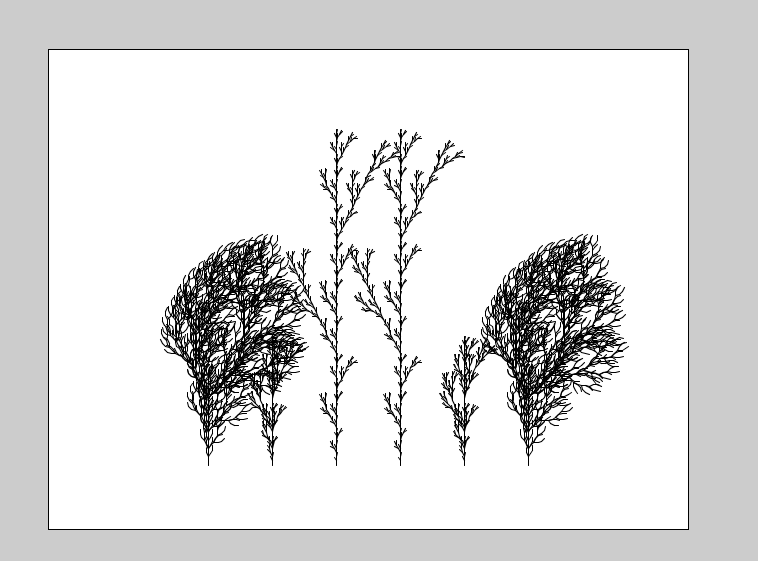
Passaremos agora, para o Jogo do Caos dinâmico. Neste jogo do Caos, pegamos na abordagem que tínhamos tido para as fractais em cima, e aplicamos uma componente dinâmica, ou seja, com cada clique do rato, criamos um vértice, e depois aplicamos o mesmo algoritmo de forma a gerar um valor aleatório e quando calha num dos vértices colori-lo:



A figura acima, mostra dois dos vários possiveis exemplos, agora é uma questão de criatividade!

# Gramáticas de Lindemayer

Neste exercicio, começaremos pelo fim, de modo a demonstrar que o objetivo pedido foi alcançado:



Temos por isso, três tipos de arvores diferentes, num estado em já lhes foram aplicados os axiomas um número confortável de vezes.

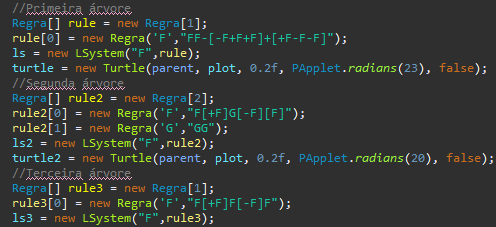


Figura 2 Classe LSystemApp5

O código em cima, dita as regras que cada arvores tem, e em especial, o angulo em que cresce. Depois é feita uma turtle (que funciona como spawn point) para cada árvore.

Na pasta Java, incluem-se outras classes com arvores também interessantes, mas muitas vezes para serem interessantes é necessário um angulo que não comporta tantas arvores quanto se queria e daí a necessidade de se enviar múltiplas classes sobre este tema. Em baixo encontram-se figuras com todas essas arvores:

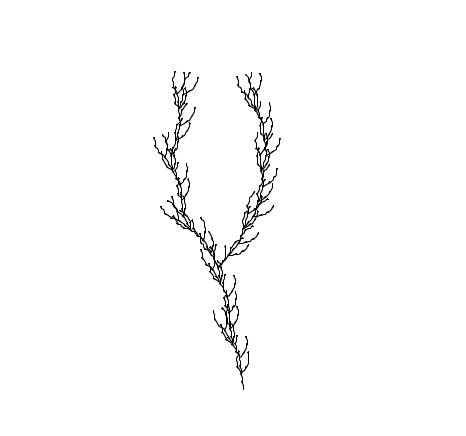
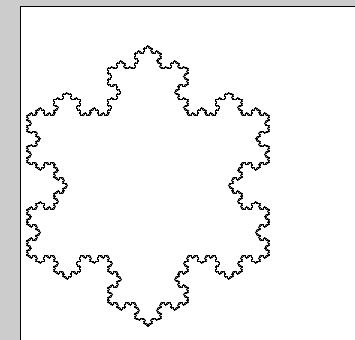
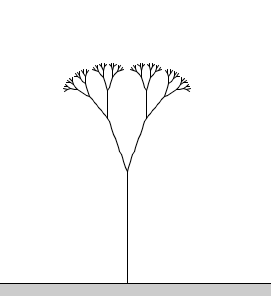
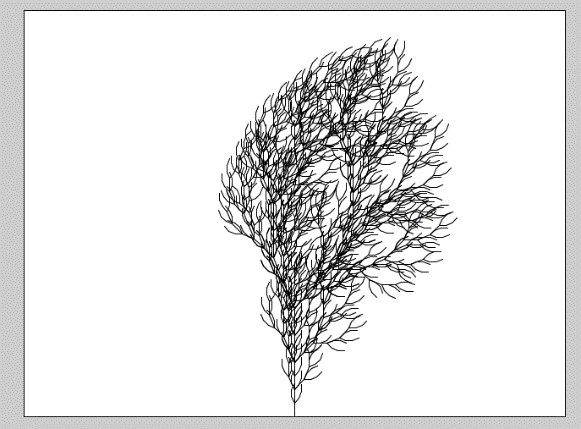


Figura 3 Classes LSystemApp1...4. Sendo que o 4 tem um factor aleatóriors quando se cria a arvore

# Conjuntos de Julia e Mandelbrot

Apesar de não ter sido pedido, implementou-se na mesma o conjunto de mandelbrot.

Relembrado a formula, precisamos de escrever a formula Zn+1 = (Zn)^2 + C, em que Z é um numero complexo e C uma constante que também pode ser complexa. C pode ser exprimido por a + bi, da-nos jeito separar a parte real da imaginária, e depois de desenvolver a formula chegamos a (a^2-b^2) + (2\*a\*b\*i). Para fazermos o conjuto de Mandell-Brott, primeiro precisamos de duas variáveis para guardar as iterações do ciclo anterior:



Agora que essas variáveis ja estão guardadas, podemos dar inicio a construção dos numeros complexos para fazermos toda a matematica necessária...



No proximo passo, construimos já, a parte Real de C, segundo a formula. A componente x é a componente real da interação anterior.



Para fazer a parte imaginaria de C, definimos

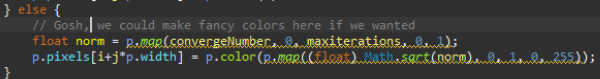


E depois mais a frente somamos a parte y da iteração anterior:



Depois disto tudo podemos dar finalmente cor caso o ponto não chegue a passar do infinito:



Caso não passe do infinito podemos fazer um mapping para convergir a cor num gradiente entre branco e preto, ou seja tons de cinzento, dependendo se converge rapidamente ou não tanto. 

Temos com resultado final:

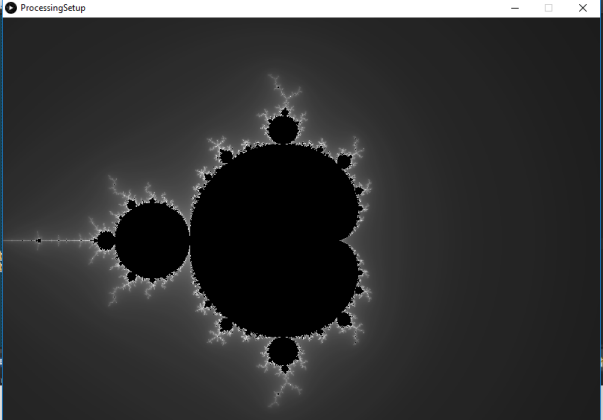
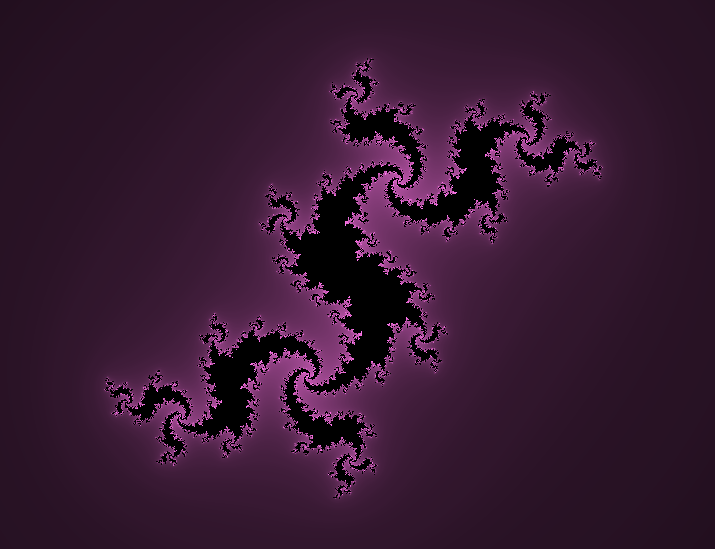
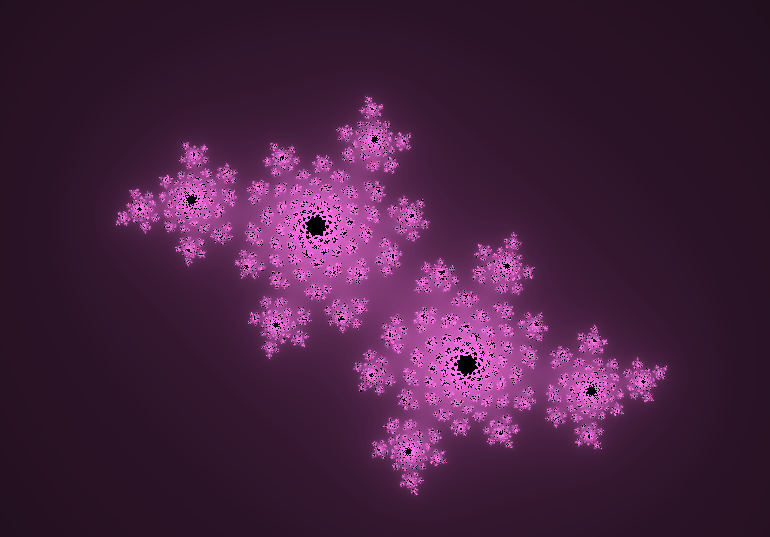


Figura 4 Classe MandelBrot

Para fazer alguns Set's de Julia, basta alterarmos diretamente C para um valor que seja interessante e para isso, podemos deixar o utilizador alterar com o rato esse valor. Com essa funcionalidade temos como resultado:

Figura Classe DynamicJulia



# Fractais

Fractal é uma figura geométrica onde uma parte de si replica o seu todo, ou traços de si, tendo como exemplo o conjunto de Mandelbrot abordado em cima. Assim pode-se dizer que um fractal é um objeto com detalhes infinitos, autossimilar e com perímetro infinito.

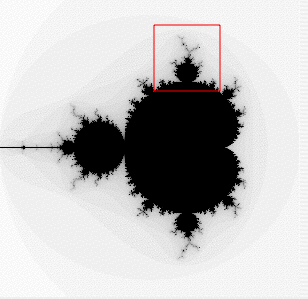


Figura 6 - Conjunto de Mandelbrot

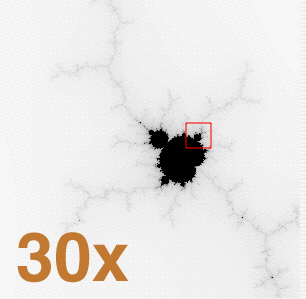


Figura 7 Conjunto de Mandelbrot com 30x zoom



Figura 8 - Conjunto de Mandelbrot com 4x zoom

Como se pode ver pelas imagens o conjunto de Mandelbrot replica-se conforme vamos fazendo zoom, adicionando a si mesmo mais detalhes até ao infinito.

Outro exemplo é o triangulo de floco de neve de Koch (desenhado em cima, com recurso a uma gramática de Lindemayer). Onde através das sucessivas iterações o número de lados aumenta, e é possível verificar que o número de lados do triangulo se rege pela fórmula matemática , assim sendo quando n tende para infinito, o número de lados também tenderá para infinito e por consequência o seu perímetro, no entanto o tamanho de cada lado, e por consequência a área adicionada por cada novo triangulo irá tender para zero. Ou seja, um objeto fractal tem área finita, mas perímetro infinito.

Agora transpondo o floco de neve de Koch para um quadrado, e para 3 dimensões temos uma esponge de Menger.

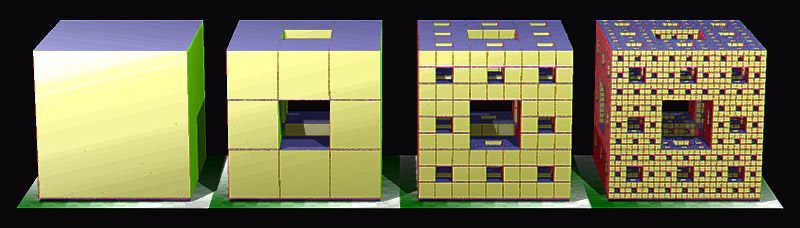


Figura 9 Esponja de Menger

Começando com um cubo perfeito, dividimos cada uma das faces em 9 e removemos o bloco central de cada uma das faces. Por fim removemos o bloco central do cubo. Para os cubos que sobram repetimos o processo infinitamente. Desta forma cada um dos blocos perfeito é dividido em 27 partes e são lhe removidas 7 partes, para um total de 20, permitindo a generalização para a fórmula de onde n é o número de iterações.

Este fractal é relevante pois o seu perímetro cresce muito em relação á sua área, sendo muito útil para a construção de antenas que captam todos os tipos de frequência, em vez das antenas tradicionais que apenas captam ondas cujo comprimento seja múltiplos do espaçamento de suas hastes. Assim este fractal é muito útil para antenas de telemóvel devido á sua versatilidade e ao pouco espaço necessário para a sua utilização.

Utilizando uma das ferramentas recomendadas (Fractal Grower) pelos docentes, reproduzimos os seguintes fractais que se encontravam já definidos por defeito



Figura 10 Árvore de Pitágoras 30/60

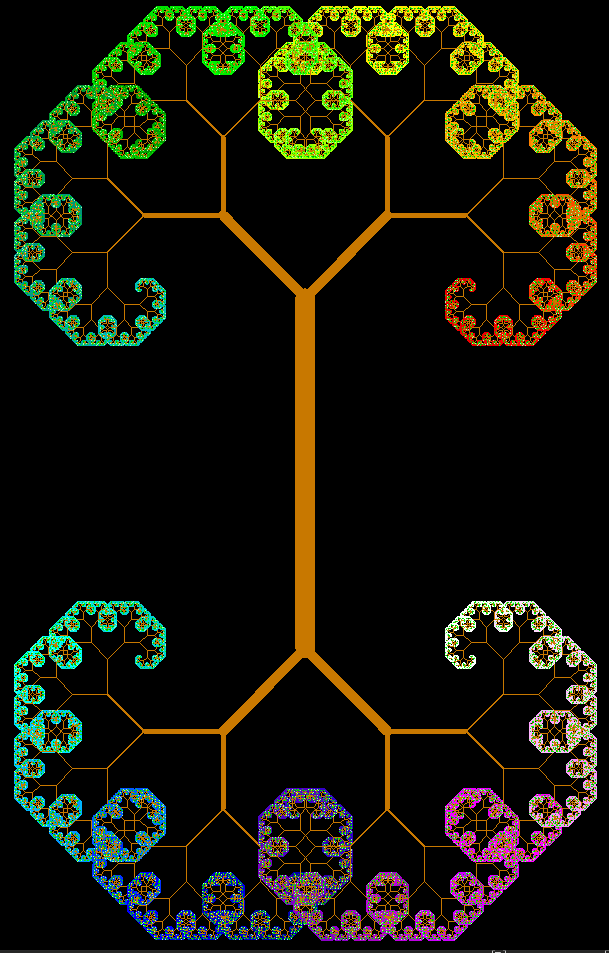


Figura 11 Ramo duplo

E alguns com uma gramática definida por nós.

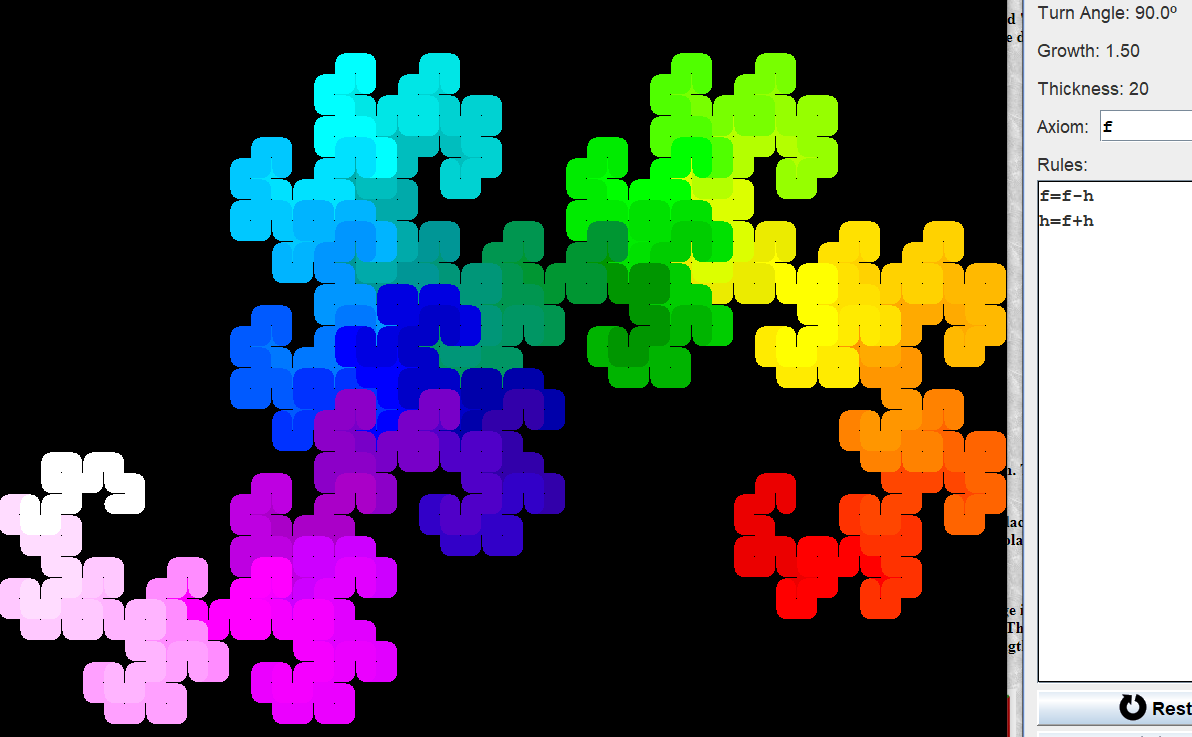


Figura 12 - Curva do dragão

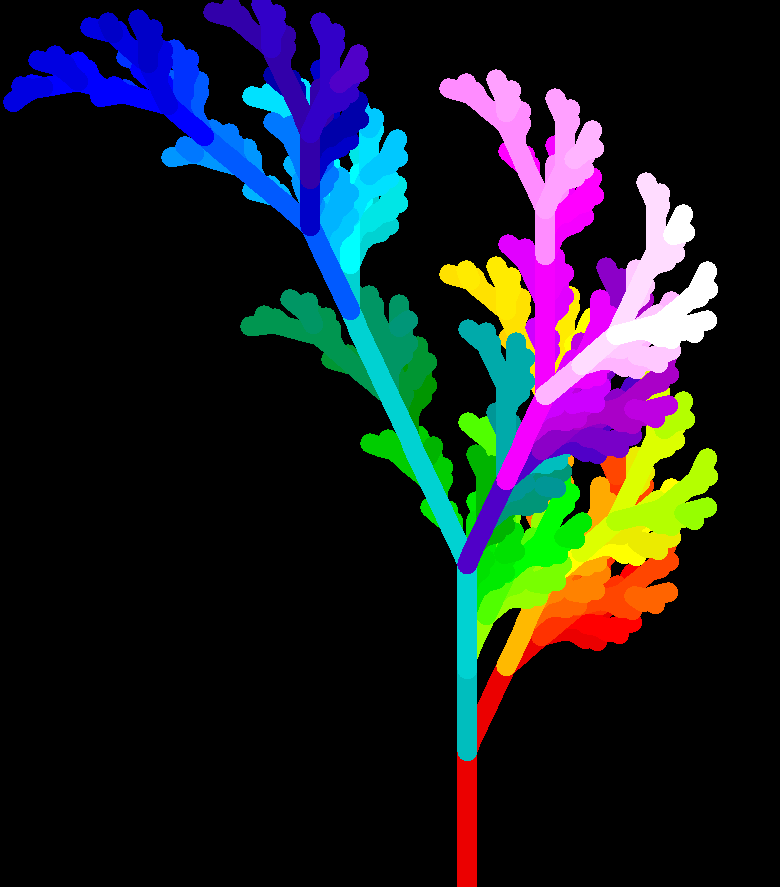


Figura 13 - Árvore fratal