ADDETC – Área Departamental de Engenharia Eletrónica e Telecomunicações e de Computadores

LEIM -Licenciatura Engenharia informática e multimédia

Modelação e Simulação de Sistemas Naturais

Trabalho para casa 2

Turma:

LEIM-32D

Trabalho realizado por:

Miguel Silvestre Nº45101

Miguel Távora Nº45102

Pedro Dias Nº45170

Docente:

Arnaldo Abrantes

Data de entrega:3/11/2019

Índice

1.1 PARTE A- FRACTAIS **4**
1.1.1 SISTEMAS DE LYNDENMAYER **4**
EXERICIO 1 ..... **5**
EXERCÍCIO 2 ..... **7**
EXERCÍCIO 3 ..... **7**
3.2) **8**
EXERCÍCIO 4 ..... **9**
EXERCÍCIO 5 ..... **10**
2) **10**
1.1.2 CHAOS GAME **11**
EXERCÍCIO 6 ..... **11**
EXERCÍCIO 7 ..... **12**
1.1.3 FRATAIS DE FUGA NO TEMPO – CONJUNTO DE JULIA E MANDELBROT **13**
EXERCÍCIO 8 ..... **13**
EXERCÍCIO 9 ..... **15**
EXERCÍCIO 11 ..... **16**
EXERCÍCIO 12 ..... **18**
1.2 PARTE B - AUTÓMATOS CELULARES **19**
EXERCÍCIO 13 ..... **19**
EXERCÍCIO 14 ..... **21**
EXERCÍCIO 15 ..... **21**
EXERCÍCIO 16 22
EXERCÍCIO 17 ..... ERROR! BOOKMARK NOT DEFINED.

Índice figuras

Figura 1 - L-System Bush 5
Figura 2 - L-System Algae 5
Figura 3 - L-System Bush 2 6
Figura 4 - L-System Erva Daninha 6
Figura 5 - L-System stick 6
Figura 6- L-System Bush 3 6
Figura 7 - L-System fruit tree 7
Figura 8 - L-System Anti-Snowflake 8
Figura 9 - L-System Snowflake 8
Figura 10 - Curva de Koch virada para cima 8
Figura 11 - Curva de Koch virada para baixo 8
Figura 12 - L-System aleatório 2 ..... **Error! Bookmark not defined.**
Figura 13 - L-System floresta 10
Figura 14 - Forma geométrica quadrado interativo 11
Figura 15 - Triangulo sierpinski interativo 11
Figura 16 - Fern jogo do caos 12
Figura 17 - Julia set para 0+0.8i 13
Figura 18 - Julia set para -0.875-0.2321i 14
Figura 19 - Conjunto de Mandelbrot 14
Figura 20 - Julia set para -0.70176- 0.3842i 15
Figura 21 - Julia set para 0.285- 0.01i 16
Figura 22 - Órbita divergente 16
Figura 23 - Órbita de um ponto fixo 16
Figura 24 - Órbita periodica 17
Figura 25 - Órbita caótica 17
Figura 26-Regra 60 com init() 19
Figura 27-Regra 60 com initRandom() 20
Figura 28-Regra 126 com init() 20
Figura 29-Regra 126 com initRandom() 20
Figura 30-Jogo Da Vida 22
Figura 31 - Regra da Maoria 22
Figura 32 - Versão do conjunto de Mandelbrot desenvolvida por nós no exercício 8 ..... **Error! Bookmark not defined.**

1.1 Parte A- Fractais

1.1.1 Sistemas de Lyndenmayer

Um L-System ou LyndenMayer system é um tipo de gramática formal e um tipo paralelo de reescrita de sistemas. Um L-System consiste num alfabeto de símbolos, com um conjunto de caracteres que correspondem a regras,sendo assim possível construir *strings*. A partir de um axioma inicial inicia-se a construção do sistema , do qual gera cadeias em estruturas geométricas.

A natureza recursiva das regras dos L-System leva a uma auto-similaridade e consequêntemente a forma de fratais, sendo assim fácil descrever com um L-System. O comportamento de células de plantas e o processo de crescimento também é simples de descrever com recurso a L-System.

Exericio 1

Para a realização do seguinte exercício foi utilizado o código da aula alterado convenientemente para a realização dos exemplos. Os exemplos são todos correspondentes a L-System *bushes* e *sticks*.

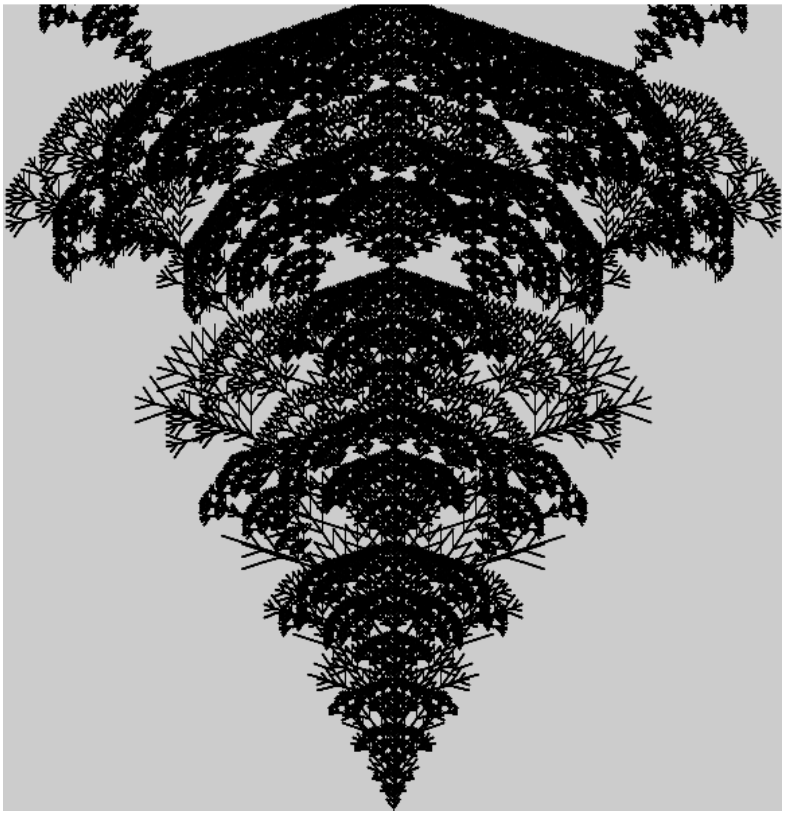

Figura 1 - L-System Bush

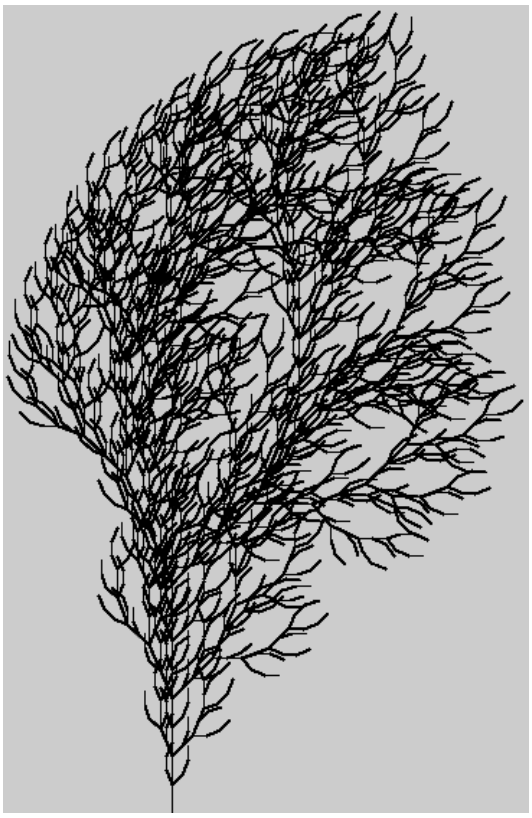

Figura 2 - L-System Algae

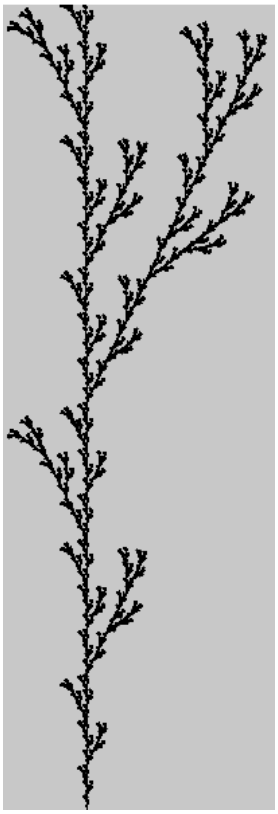

Figura 4 - L-System Erva Daninha

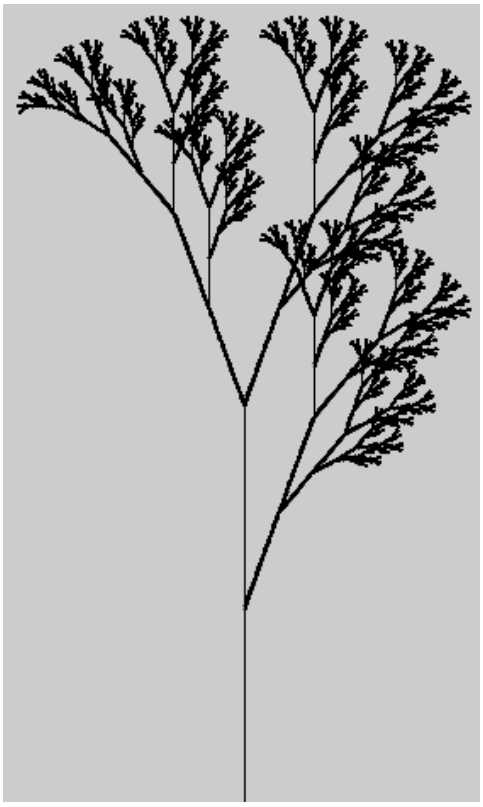

Figura 3 - L-System Bush 2

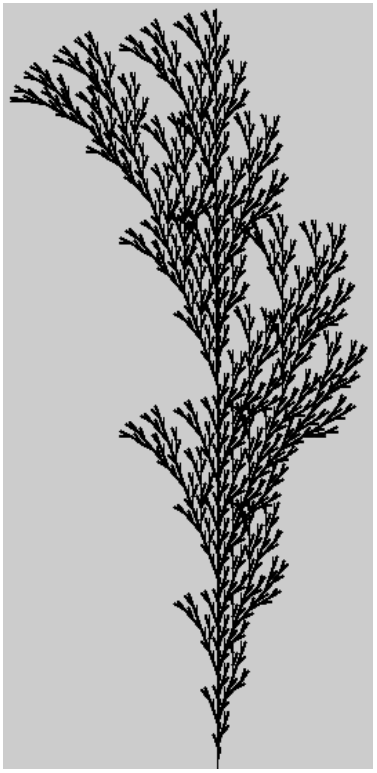

Figura 6- L-System Bush 3

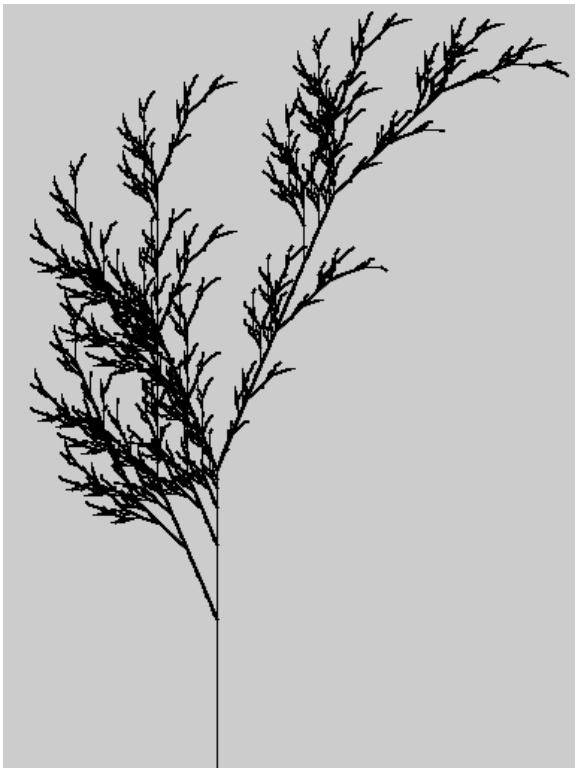

Figura 5 - L-System stick

Exercício 2

Para a implementação do exercicío foi criado um objeto Rule com duas regras. Com os caracteres e os seus correspondentes comandos foi possível construir a seguinte árvore de frutos.

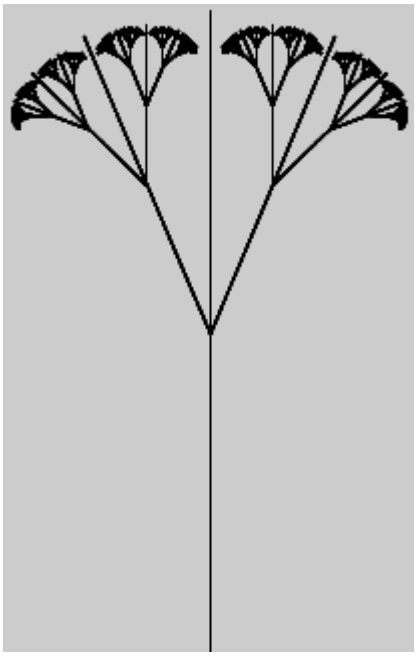

Figura 7 - L-System fruit tree

Exercício 3

Para a implentação do presente exercício foram utilizadas ferramentas aprendidas no primeiro trabalho tais como a rotação, repintar o fundo entre outros. Para centrar as imagens no canvas foi-se variando os valores dos pontos iniciais até se obter um valor que correspondesse ao centro. As imagens dos resultados exibidos são correspondentes apenas à última geração. As variações da curva de Koch foram as que se segue:

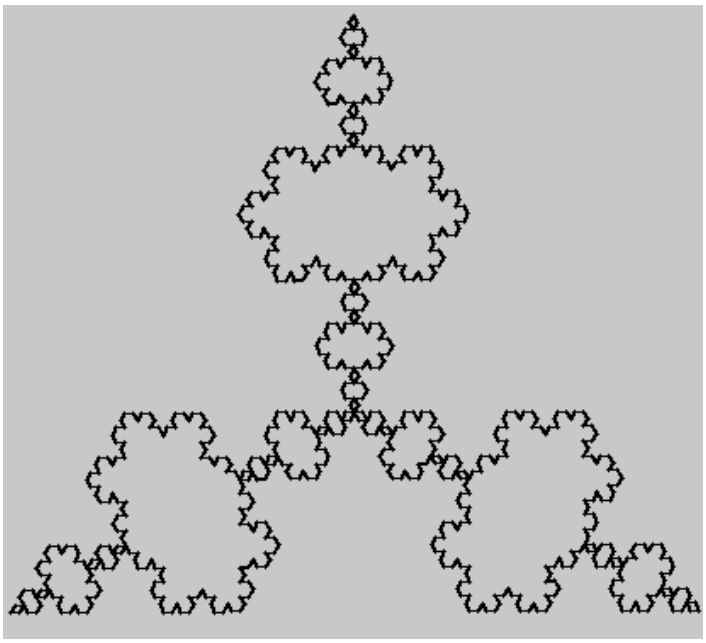

Figura 8 - L-System Anti-Snowflake

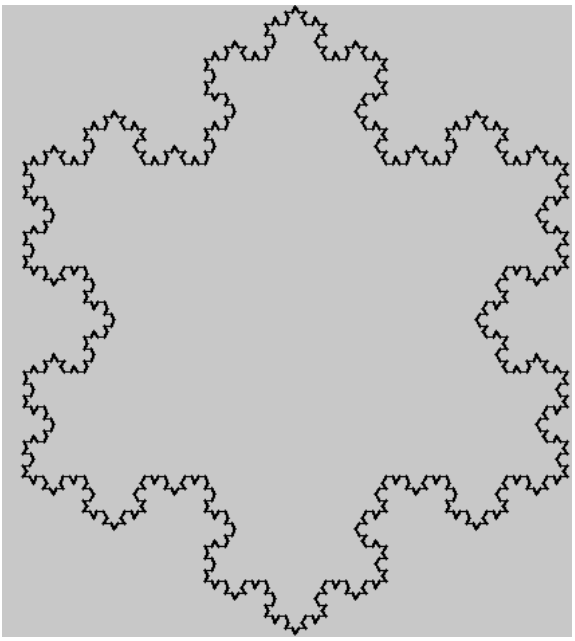

Figura 9 - L-System Snowflake

3.2) Para gerar a probabilidade foi utilizado o método random() da classe PApplet, para realizar a viragem para cima ou para baixo antes de chamar o método render da turtle é aplicado um rotate de -90º ou de 90º, para cima e para baixo respetivamente. Para ajustar a imagem foi também utilizado a posição inicial e redefine-se os pontos.

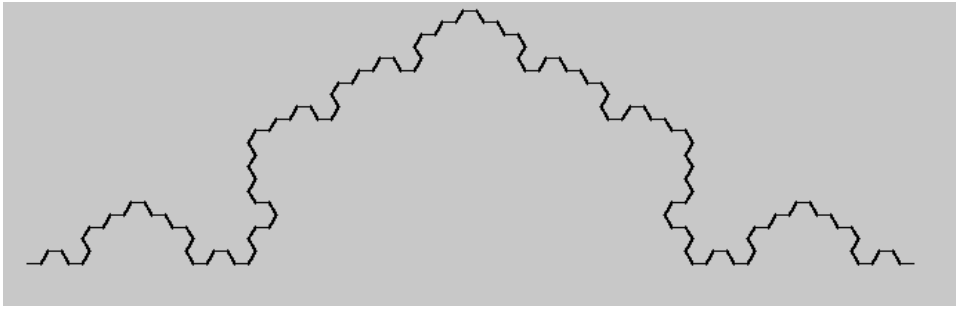

Figura 10 - Curva de Koch virada para cima

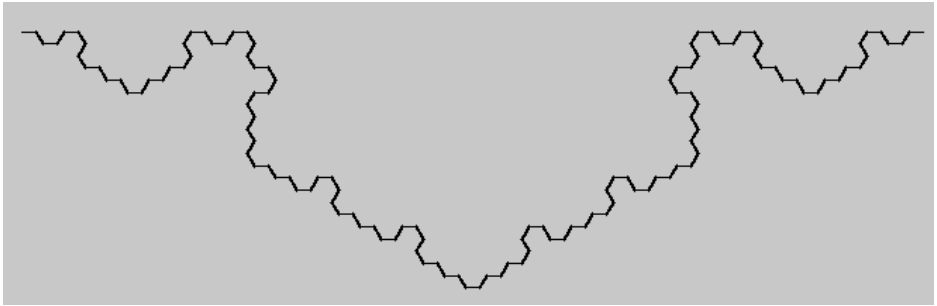

Figura 11 - Curva de Koch virada para baixo

Exercício 4

Para a realização do exercício foi criada uma classe parecida ao L-System, porém com diferenças no construtor e no método nextGeneration(). Onde é utilizada a classe Random do java e se cria um número float entre 0 e 1 caso o random seja inferior a 0.5 gera o L-System Pentaplexity normalmente, caso contrario gera um L-System completamente aletório. O aleatório foi criado começando sempre com o axioma F e os caractéres que se segue ele gera 1 caracter aleatório e acrescenta esse caracter á string, sempre que é chamado o método nextGeneration ele acrescenta esse caracter além disso também guarda sempre o caracter que está a verificar. Os fratais obtidos foram:

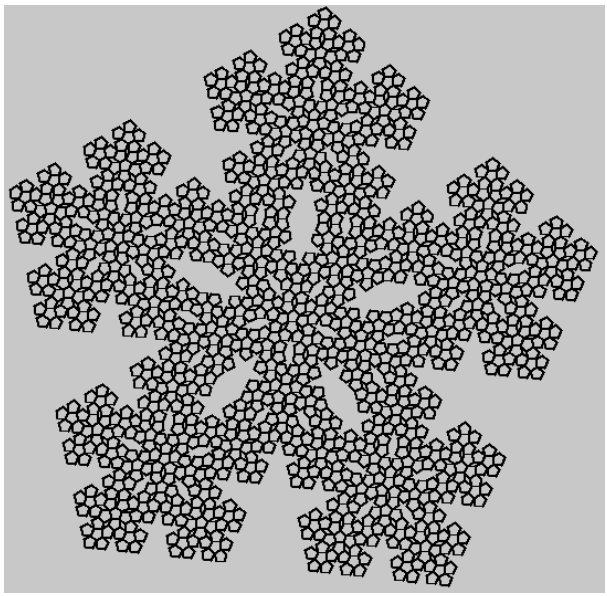

Figura 12 - L-System Pentaplexity

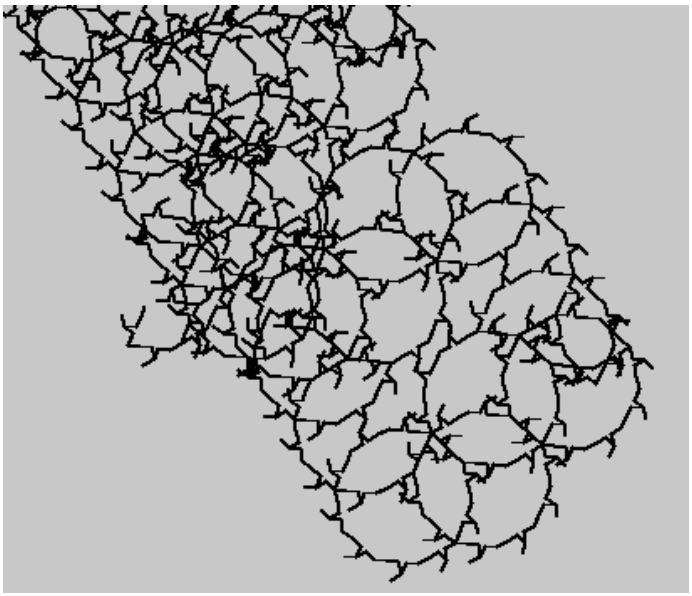

Figura 13 - L-System aleatório 1

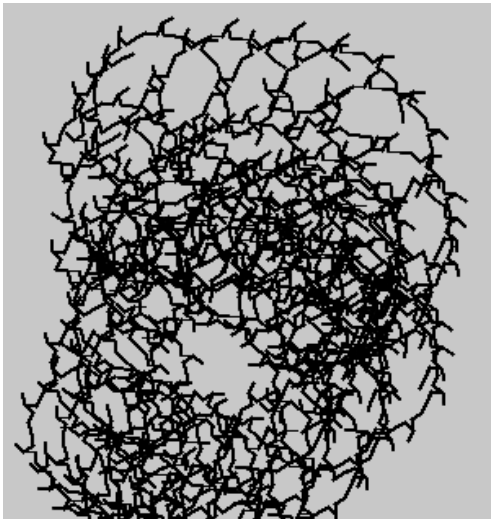

Figura 14 - L-System aleatório 2

Figura 15 - L-System aleatório 3

Exercício 5

Para a criação da floresta foram utilizados apenas L-Systems de *bushes*. No qual são dispostas de forma regular na janela. Durante a evolução das plantas as mais pequenas mudam de posição para cima para simular o crescimento excepto a maior pois sairia da janela. Obtendo assim a seguinte floresta:

Figura 12 - L-System floresta

2) Para animar a floresta iria por as plantas num *loop* onde não seria necessário clicar com o rato na tela para as plantas crescerem e onde em cada novo crescimento das plantas seria gerada uma nova cor aleatória para a planta. Em relação ao meter as árvores num loop a crescerem parece uma ideia trabalhosa mas possível de concretizar, trabalhosa pois a classe Florest trabalha a base de objetos de cada planta tendo de alterar todas as classes instânciadas. Em relação em mudar as cores parece mais simples pois era criado um método que receberia um array de três inteiros e devolveria novamente um array com as cores aleatórias.

1.1.2 Chaos Game

O jogo do caos é um método de criação de um fractal, criado a partir de um poligôno e um ponto dentro do mesmo. O fractal é criado iterativamente por uma sequência de pontos, iniciando num ponto inicial aleatório. Cada ponto aleatório representa uma fração da distância entre o ponto anterior e um vértice do poligôno. Repetindo a interação um grande número de vezes por vezes cria um formato de uma fractal.

O método do jogo do caos desenha pontos em ordem aleatória em todo o atrator. Sendo diferente em relação aos fratais que testam cada pixel na tela para ver se pertence ao fractal. Um fractal pode ser desenhado rapidamente com o jogo do caos porém é dificil ter um bom detalhe do mesmo.

Exercício 6

Para a realização do presente exercicio foi utilizado o trianglo de sierpinski. Para guardar os pontos onde se clica com o rato foi utilizado um array de ints de dimensão 8, do qual quando se clica com o lado esquerdo do rato ele guarda essa posição de pixeis em x e y. Para ele guardar a nova posição dos pixeis seguintes tem de clicar no lado direito, depois no lado esquerdo novamente formando assim o triangulo de sierpinski. Caso se clique novamente no lado direito ele cria uma forma irregular de um quadrado.

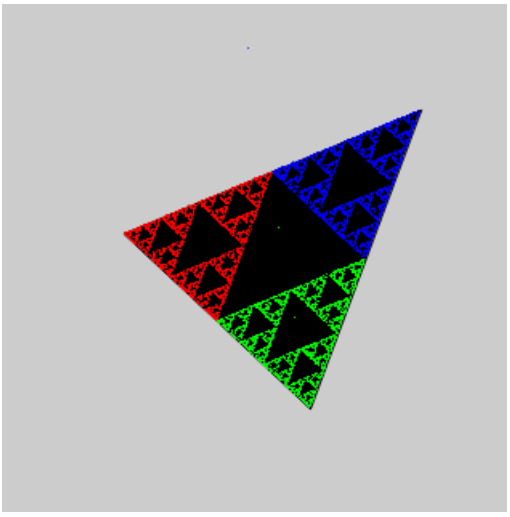

Figura 14 - Triangulo sierpinski interativo

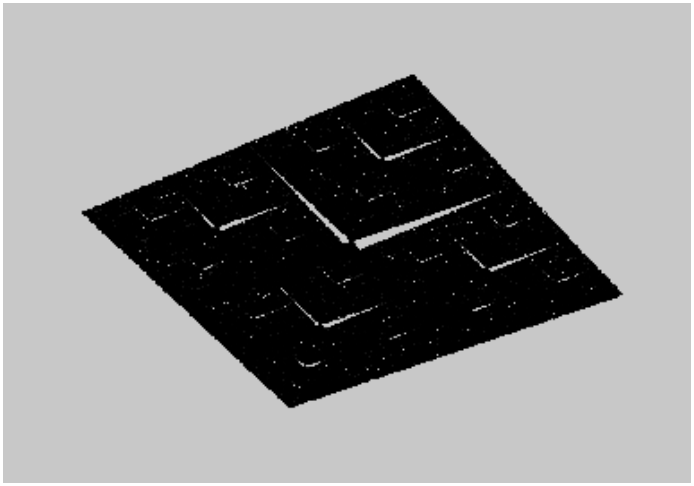

Figura 13 - Forma geométrica quadrado interativo

Exercício 7

Para criar a fractal com restrições foi utilizada um que tira como proveito uma matriz e probabilidades. O fractal escolhido foi o fern onde com um valor de uma matrix para x e y e com uma probabilidade de valores para cada valor de x ou y é construído o fern.

Figura 15 - Fern jogo do caos

1.1.3 Fratais de fuga no tempo – Conjunto de Julia e Mandelbrot

No plano dinâmico dos números complexos, o conjunto de Júlia e o conjunto de Fatou são dois conjuntos complementares definidos por uma função. O conjunto de Fatou da função consiste em valores com propriedades de tal forma que se comportam de forma similar dentro das iterações. O conjunto de Júlia dos valores consiste em valores que sofrendo uma pequena perturbação podem causar drásticas mudanças na sequência dos valores iterados. Estes comportamentos consistem de que o conjunto de Fatou é regular, enquato o de Julia é caótico.

O conjunto de Mandelbrot é um fractal definido como o conjunto de pontos C no plano complexo para um sucessão. Sendo uma representação gráfica, que pode ser divido em um conjunto infinito de figuras pretas, sendo o maior o cardiódie localizado no centro do plano complexo.

Exercício 8

No presente exercício foi pedido para produzir dois conjuntos de Julia e o conjunto de Mandlebrot. Para criar os conjuntos de Julia, foram criadas duas variáveis para guardar a parte real e a parte imaginaria da constante C. São utilizadas variáveis auxiliares para o número máximo de iterações e para guardados os vários pontos pintados no ecrã.

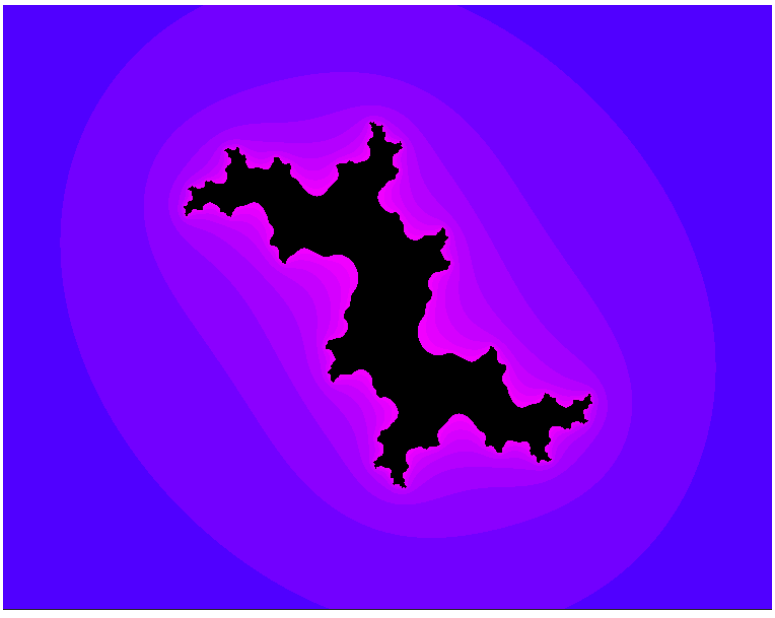

Figura 16 - Julia set para 0+0.8i

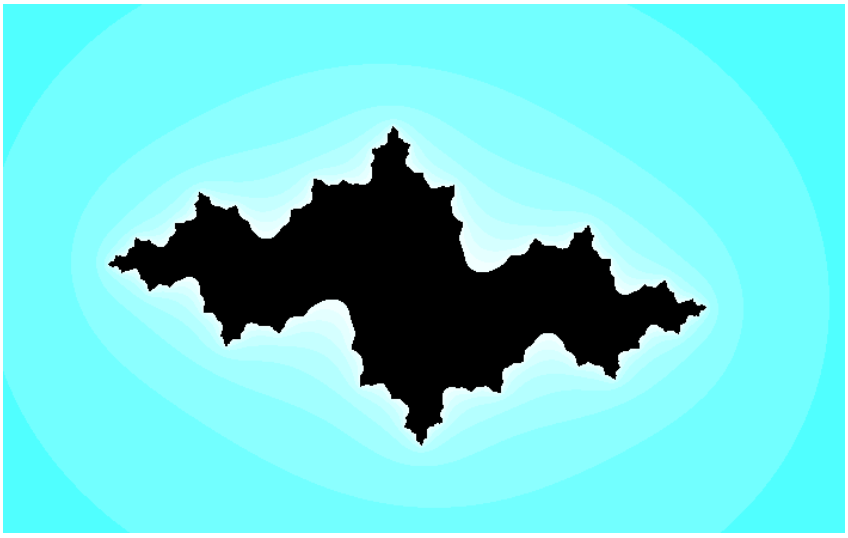

Figura 17 - Julia set para -0.875-0.2321i

Para realizar o conjunto de Mandlebrot o raciocínio foi semelhante ao de conjunto de Julia diferenciando na constante depender do ponto a ser iterado.

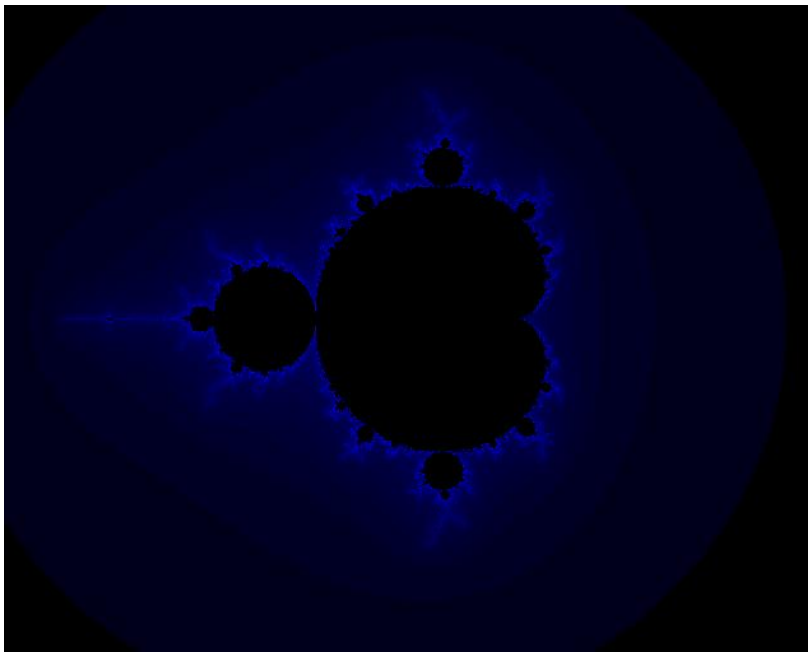

Figura 18 - Conjunto de Mandelbrot

Exercício 9

Para a realização do exercicío visto que já os conjuntos anteriores tinham sido coloridos então o grupo decidiu então acrescentar mais dois conjuntos de Julia e pintar de forma mais diversificada. Para pintar o fractal visto que este é estático, então pintamos de preto pois apenas possui um tipo de cor. O mais divertido é em volta do fracal que possuí varias camadas de pontos mais distintos. Para pintar o seu redor foi utilizada a raiz quadrada do número de iterações e o resto foi utilizado cores de caracter estático no formato de float. As coisas mais interessantes eram somente quando o que varia com o número de iterações é o vermelho, e vai-se variando a quantidade de green e blue.

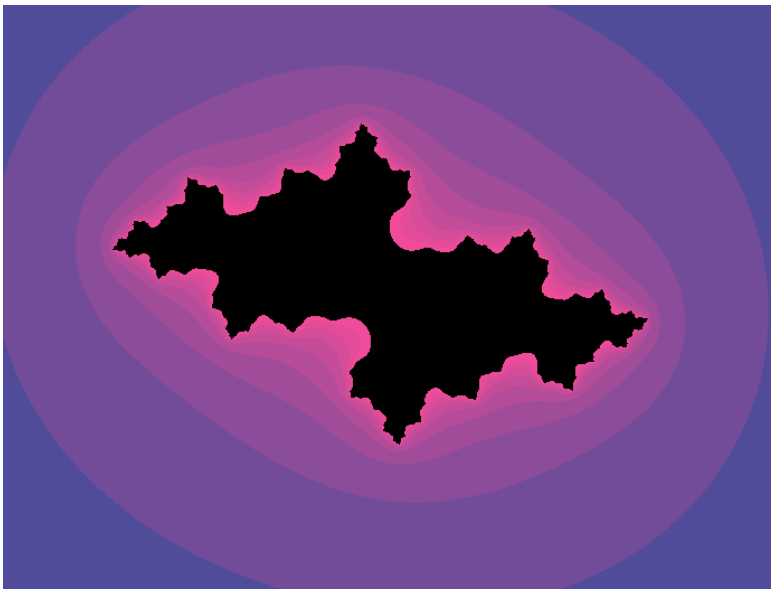

Figura 19 - Julia set para -0.70176- 0.3842i

Figura 20 - Julia set para 0.285- 0.01i

Exercício 11

1. Para uma orbita convergente é:

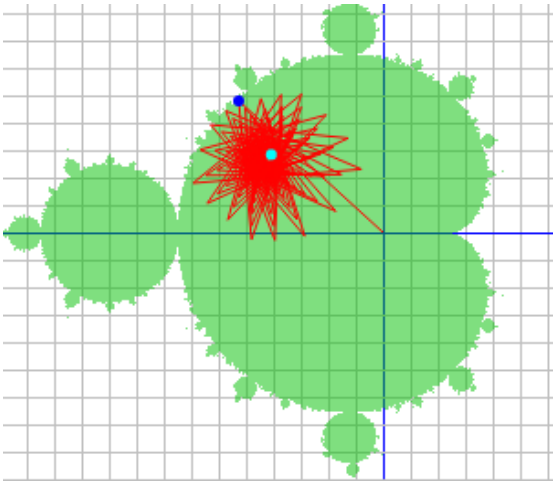

Figura 21 - Órbita divergente

Para um ponto fixo:

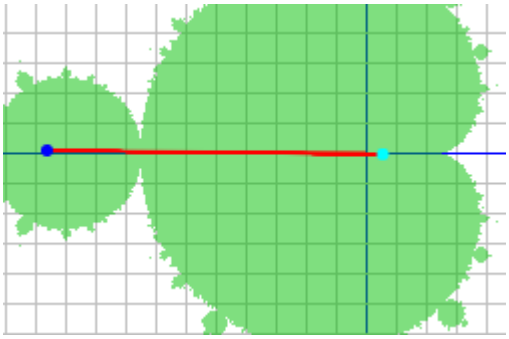

Figura 22 - Órbita de um ponto fixo

Periodico:

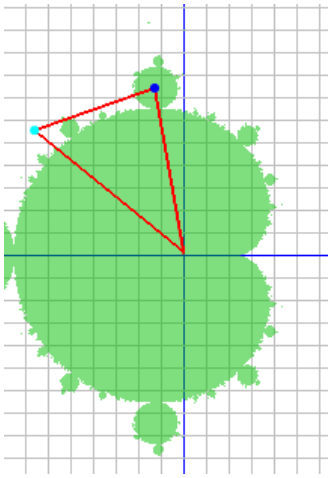

Figura 23 - Órbita periodica

Caótico:

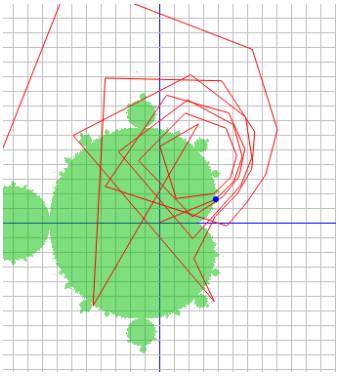

Figura 24 - Órbita caótica

2) Uma funcionalidade adicional que poderia ser introduzida seria o de poder fazer zoom in e zoom out com a roda do rato, através disso era possível visualizar mais detalhamente o que está a acontecer nas órbitas. Além disso seria util ter a possívilidade de mudar a cor do fractal como o background e a órbita. Isso facilitaria a visualização visto que as cores escolhidas na opinião do grupo não são as mais adequadas.

Exercício 12

A ferramenta que o grupo utilizou foi o fractal grower. Devido á grande complexidade de utilização da aplicação e por falta de explicação de como se deve utilizar a aplicação por parte dos produtores da aplicação, o grupo não conseguiu realizar nenhuma aplicação.

1.2 Parte B - Autómatos Celulares

Um áutomato celular consiste numa grelha de células. Cada um deles com um número finito de estados. Por cada célula existe um conjuto de células designhados de vizinhança que interage com a célula em questão. Um estado inicial, para t = 0, é selecionado atribuindo um estado para cada célula. Sempre que é incrementado o t surge uma nova geração, avançando para t = 1 de acordo com uma regra fixa. Regra essa que determina o estado da nova célula em termos do estado corrente das células na sua vizinhança. Tipicamente a regra mesma regra aplica-se a todas as células, a regra não varia ao longo do tempo e a regra aplica-se a grelha inteira simultâneamente.

Exercício 13

Para a realização do exercício são usadas as classes Rule e EAC desenvolvidas em aula. A classe EAC possui atributos para representar o número de células. Através do método designado como init() que inicia a célula localizada no centro e um initRandom() que inicia células de uma forma aleatória. A partir do método getNextstate() são gerados sucessivamente as seguintes gerações. No exercício foram utilizadas regras de 60 e 126.

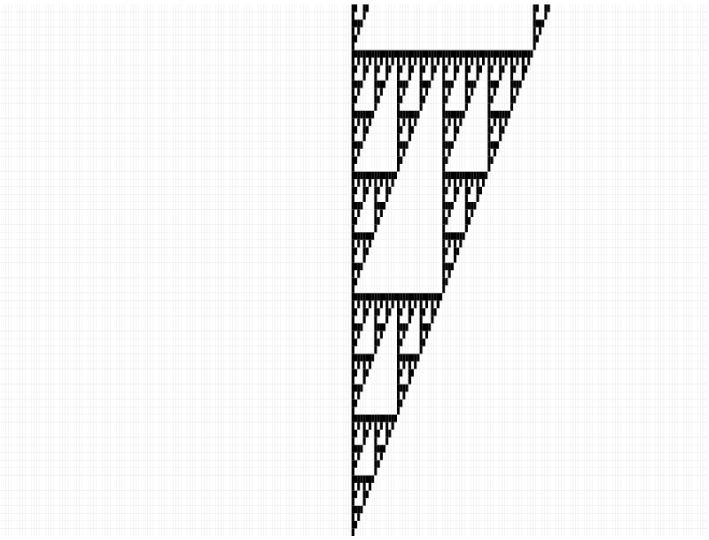

Figura 25-Regra 60 com init()

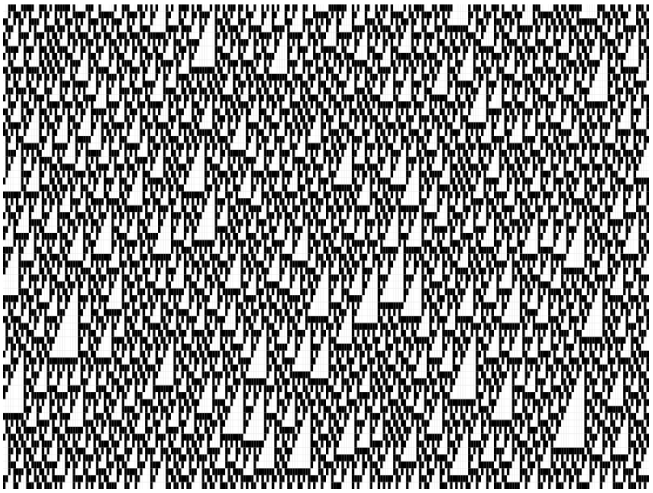

Figura 26-Regra 60 com initRandom()

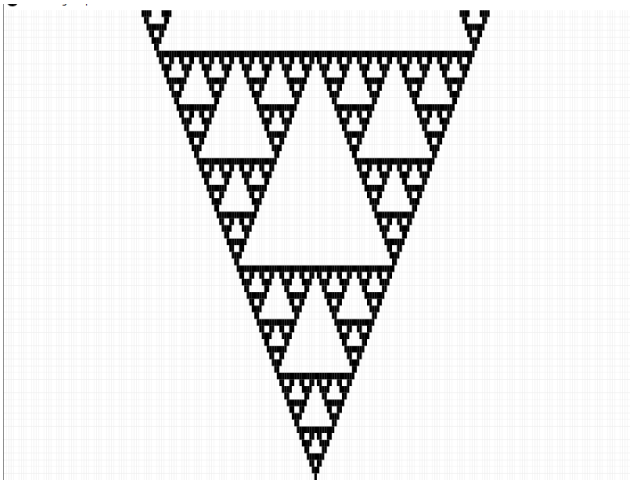

Figura 27-Regra 126 com init()

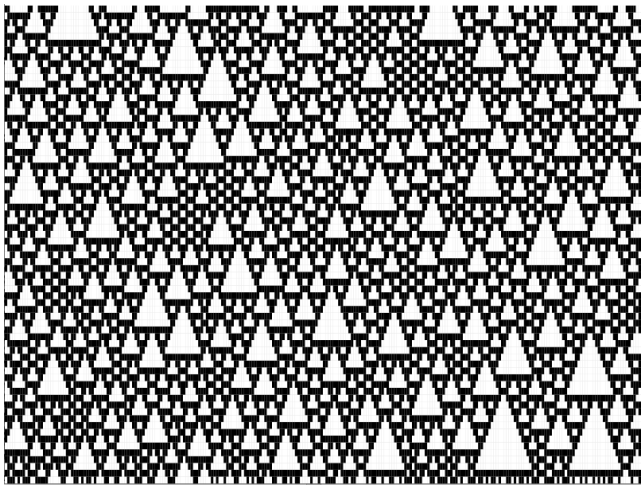

Figura 28-Regra 126 com initRandom()

Perante as figuras obtidas conclui-se que o número da regra altera a rotação do sólido geométrico e que o método chamado sendo ele o init() ou o initRandom faz variar a representação dos sólidos pela grelha.

Exercício 14

1) $2^5 = 32$

2) $2^{32} = 4294967296$

$1\ ano = 31536000\ segundos$

$\frac{4294967296}{31{,}536{,}000}$ =136

demoraria cerca de 136 anos

3) $(2 * 1 + 1)^2 - 1 = 8$

dimensao da tabela (moore): $2^9 = 512$

regras $2^{512} = 1.340781e^{154}$

$\frac{1.340781^{154}}{31{,}536{,}000} = 4.248678606738155e^{146} anos$

4) Conclui-se que a vizinhança de Moore trabalha com 8 vizinhos em vez 4, sendo mais flexível do que a vizinhança de Von Neumann.

Exercício 15

No presente exercício fomos incumbidos de desenvolver uma versão nossa do jogo da vida. Nesta classe são utilizadas variáveis para representar o tamanho das células, para guardar as células e para guardar células que vão mudar em cada iteração. Foi também necessário a criação de variáveis de tempo para determinar aquando da iteração das células. Para adicionar interatividade no jogo quando se clica no espaço o jogo para a execução e quando se clica novamente ele retorna ao jogo. O jogo da vida funciona de tal maneira que se a célula tiver três ou mais vizinhos ela sobrevive caso contrário esta morre. Por fim quando premindo na tecla R esta reinicia o jogo.

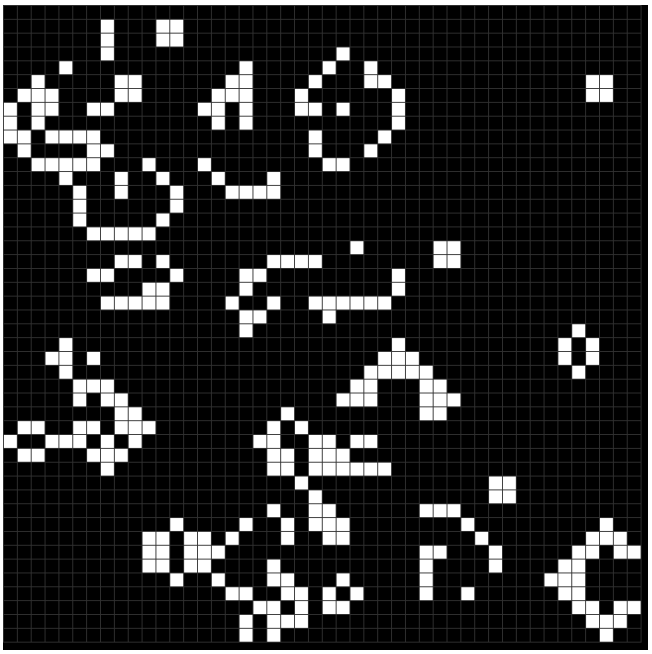

Figura 29-Jogo Da Vida

Exercício 16

Para a relizaçao do presente exercício a lógica por de trás da Regra da maioria é parecida ao do jogo da vida, diferenciando na parte de não se encontrar num loop, com regras diferentes e aplica-se a regra 8 vezes. Sempre que se clica no ecrã este gera uma grelha é desenhado novas componentes.

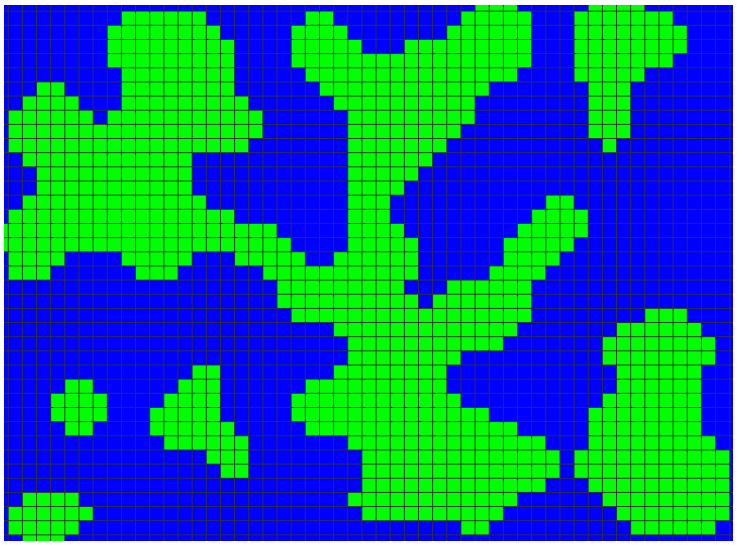

Figura 30 - Regra da Maoria

Exercício 17

Neste exercício é pedido para fazer um pequeno ensaio sobre fractais ou autômatos celulares. Nos decidimos fazer sobre fractais e autómatos celulares, vamos falar sobre o que são a sua origem e alguns exemplos do que se consegue fazer com os mesmos.

Fractais

Um fractal é um objeto da que pode ser dividido em partes, cada uma das quais se assemelha ao objeto original. Fractais tem infinitos detalhes são geralmente auto similares e de escala. O termo fractal foi criado em 1975 pelo o matemático Benoit Mandelbrot, o mesmo que descobriu a geometria fractal na década de 70.

Fractais podem ser gerados por um sistema de funções iteradas, ou seja, consiste em selecionar uma figura inicial aleatória e aplicar-lhe uma série de transformações que vão gerando infinitamente copias menores da a imagem original. Este processo foi usado neste mesmo trabalho, por exemplo na nossa versão da curva de Koch.

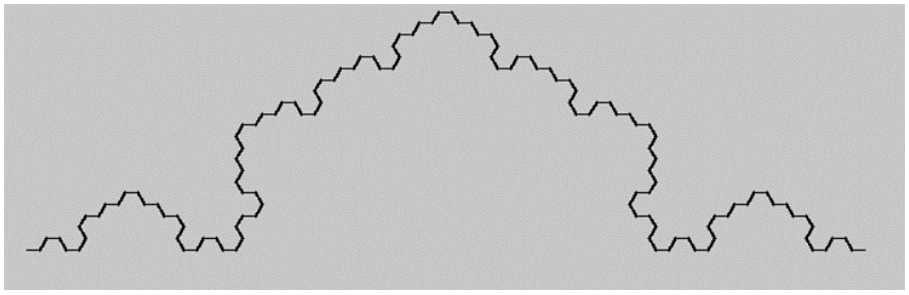

Figura 31-Curva De Koch Desenvolvida No Exercício 2

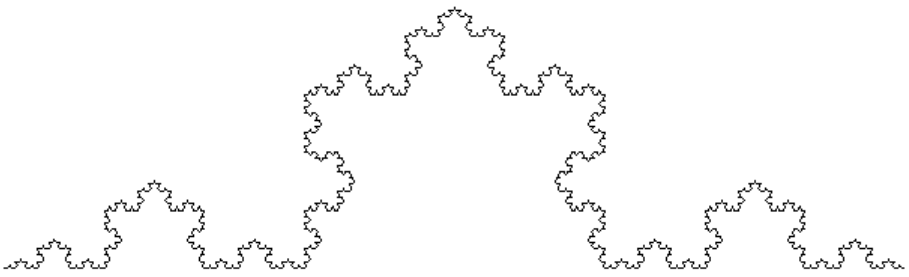

Figura 32-Curva De Koch

Outros exemplos de figuras que podem ser obtidas com este processo são o triangulo de Sierpinski e o tapete de Sierpinski.

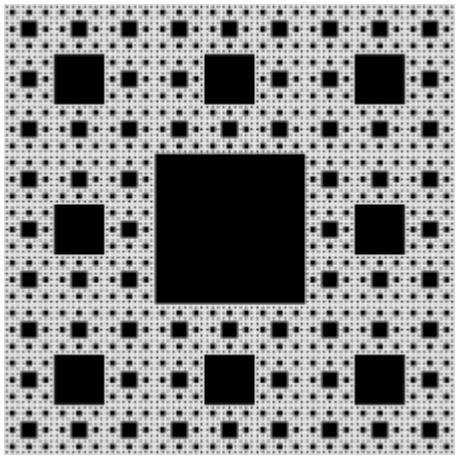

Figura 34--Tapete de Sierpinski

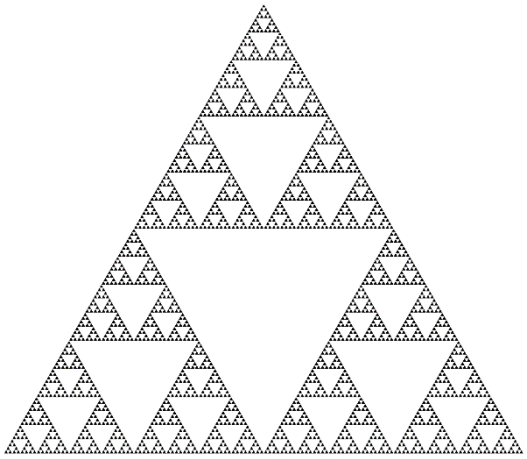

Figura 33-Triangulo de Sierpinski

Fractais podem também ser gerados por forma de recorrência em cada ponto do espaço. Este tipo de fractais é também conhecido por fractais de fuga do tempo. Um exemplo deste tipo de fractais é o conhecido conjunto de Mandelbrot.

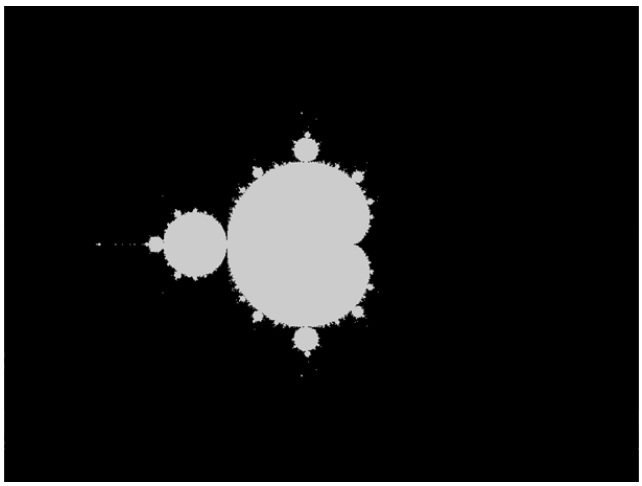

Figura 35-Conjunto De Mandelbrot Desenvolvido no Exercício 8

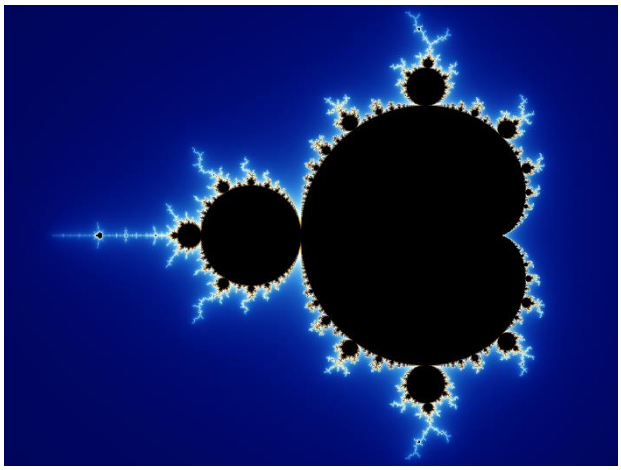

Figura 36-Conjunto De Mandelbrot

Por fim, fractais podem ser gerados de forma aleatória em vez de determinística, ou seja, são gerados através de processos estocásticos. Exemplos deste tipo de fractais são os terrenos fractais e o voo de Levy.

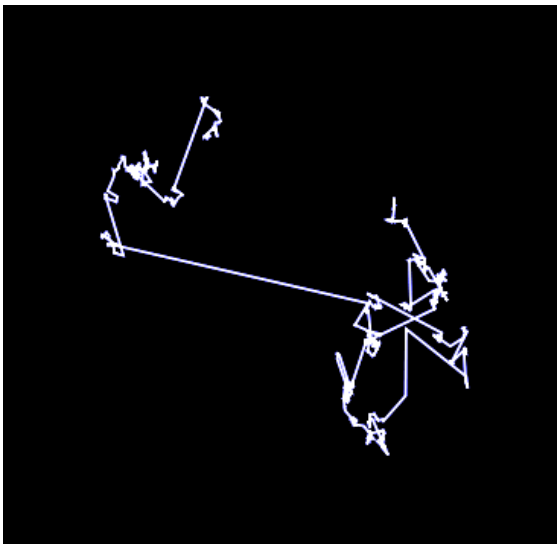

Figura 37-Voo De Levy

Fractais podem ser caracterizados de acordo com a sua autossimilaridade. Fractais podem ter uma autossimilaridade exata, ou seja, o fractal e idêntico em diferentes escalas. Fractais gerados por sistemas de funções iteradas apresentam geralmente autossimilaridade exata.

Fractais podem ter uma parcial autossimilaridade, ou seja, o fractal e semelhante em diferentes escalas, mas não exatamente igual. Fractais deste tipo geram pequenas copias do fractal inteiro de maneira distorcida.

Por último, fractais podem ter uma autossimilaridade estatística, ou seja, o fractal e diferente em diferentes escalas. Fractais deste tipo possui medidas numéricas que são preservadas em diferentes escalas. Fractais estocásticos possuem geralmente este tipo de autossimilaridade.

Autómatos celulares

Os autómatos celulares são modelos de evolução natural mais simples com capacidade para exibir comportamentos mais complicados.

Os autómatos celulares podem ser unidimensionais, bidimensionais ou tridimensionais.

Os autómato unidimensionais tem apenas dois estados possíveis por célula. Uma célula e as suas duas vizinhas forma uma vizinhança de 3 células, por isso existem 2^3 combinações por vizinhança, ou seja, existem 2^8 regras possíveis para este tipo de autómato.

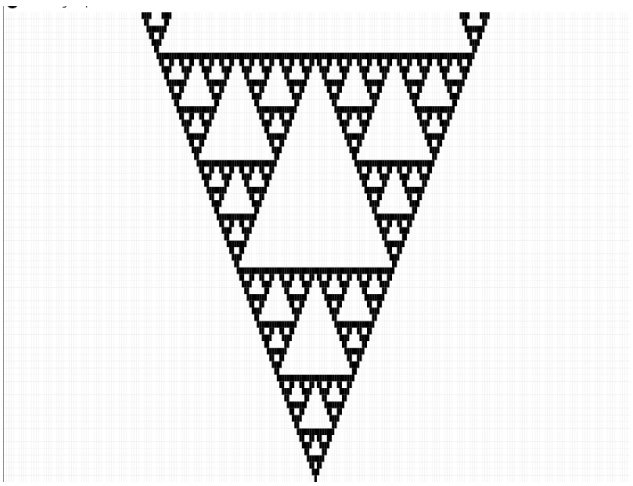

Figura 38-Automato Unidimensional Desenvolvido Neste Trabalho Com a Regra 126

Os autómatos bidimensionais tal como os unidimensionais possuem apenas dois estados por célula com a diferença que cada célula possui 8 em vez de 3 vizinhos. Ou seja, vai ter 2^9 combinações possíveis para este tipo de autómato. O autómato bidimensional mais conhecido é o jogo da vida inventado em 1970 por John Horton Conway.

Figura 39-Jogo Da Vida

Os autómatos tridimensionais são autômatos que permitem simulações mais complexas e são normalmente usados na área da biologia, física etc.