

3

CAPÍTULO

MODELAGEM PROCEDURAL – MODELOS E APLICAÇÕES

Art challenges technology, technology inspires the art.

John Lasseter

No dia 9 de agosto de 2016, a produtora Hello Games em parceria com a Sony, lançou o jogo *No Man's Sky*¹ (2016), nesta obra de ficção científica, que era aguardada desde dezembro de 2013, os jogadores são exploradores, inseridos em um universo com cerca de 18 quintilhões (2^{64}) de planetas, todos eles com características próprias e únicas de atmosfera, hidrosfera, litosfera e biosfera. O objeti-

¹ *No Man's Sky* foi desenvolvido pela *Hello Games*, empresa indie de Guildford, Reino Unido.
Link: <http://www.no-mans-sky.com/about/>

vo do jogo é que, através da exploração e mapeamento do universo, o jogador possa chegar ao centro deste, e assim, coletar uma maior quantidade de dados e itens e inseri-lo no Atlas.



Figura 3.1. Imagem do jogo *No Man's Sky* (2016).

Durante o anúncio do jogo, no evento *Spike Video Game Awards* em 2013², o CEO da produtora Hello Games³, Sean Murray, explicou que o método que os programadores escolheram para a produção do *game*, seria a de *procedural generation*⁴, método algorítmico matemático. Todos os elementos do jogo, a geografia, a temperatura, a biodiversidade de fauna e flora, estavam no cerne deste algoritmo de *procedural generation*. Esta técnica, apesar de não ser nova, criou um status *hype* para o jogo, devido a façanha do seu poder de criação – o mecanismo desta ferramenta trabalha através de uma função algorítmica matemática no qual é possível unir todas as leis físicas do universo.

Esta técnica, datada de mais de 30 anos, nasce da necessidade dos desenvolvedores da época, pois devido a limitações de memória nos computadores, não

² Pode ser observado no site: <http://joguindie.com.br/artigos/hello-games-no-mans-sky/> Acessado em 2/11/2016.

³ Empresa independente do Reino Unido, fundada em julho de 2009, conhecida pela série *Joe Danger*. Link da empresa: <http://www.hellogames.org/>.

⁴ Técnica de criação de conteúdo digital através de algoritmos de programação.

era possível criar jogos com muitos *layouts*⁵ gráficos e objetos tridimensionais de alta qualidade. Apesar de muitos jogos na época utilizarem desta técnica, o que mais se destaca é o jogo *Rogue* (1980).



Figura 3.2. Imagem do jogo *Rogue* (1980).

Lançado em 1980, neste jogo, os programadores criaram uma função algorítmica para a produção de cenários e desafios, criando um jogo com inúmeras possibilidades e assim, o jogador dificilmente encontraria a mesma cena duas vezes. Dentre as características que tornam este jogo marcante, e que influenciou na produção de jogos da posteridade, é a utilização de caracteres ASCII⁶ para a criação de elementos gráficos, assim como, a aleatoriedade na geração de layouts de masmorras e a localização de objetos no cenário, que resulta numa experiência de (quase) infinitos panoramas.

Ainda na década de 1980, jogos como *River Raid* (1982), lançado para *Atari 2600*, e *Elite* lançado para computador, utilizaram da mesma técnica, para a produção das fases bilaterais assim como para criar oito galáxias, cada uma com 256 planetas.

⁵ Disposição dos elementos que são mostrados na tela do computador.

⁶ *American Standard Code for Information Interchange* código binário que codifica um conjunto de 128 sinais: 95 sinais gráficos (letras do alfabeto latino, sinais de pontuação e sinais matemáticos) e 33 sinais de controle, utilizando, portanto, apenas 7 bits para representar todos os seus símbolos.

Mais recentemente o jogo *Minecraft* (2011) provou porque esta técnica tem se mostrado a favorita entre desenvolvedores independentes, pois a mesma se apresenta como uma técnica de baixo custo permitindo aos desenvolvedores construir jogos repletos de conteúdo, mas sem demandar milhares de horas de trabalho ou muita mão-de-obra, os códigos de programação agora lidam com esta tarefa.



Figura 3.3. Imagem do jogo *Minecraft* (2011).

Utilizar de *procedural generation*, ou modelagem procedural como um método de automatização na geração de conteúdos assim como de objetos, salas, cenários, *levels*, ou mesmo um planeta, vai de encontro com os métodos já tradicionalizados na produção de *games*, que consiste na manufatura de modelos e maquetes, na criação de mapas para o jogo, na pintura e no design gráfico. Por outro lado, os jogos que necessitam de conteúdos originais são limitados a um número finito de *levels* ou uma área predefinida de jogabilidade.

O fator inovador inserido pelos novos criadores de jogos é a junção de dados à função algorítmica. Se antes não era possível desenvolver *games* com riqueza de detalhes por falta de memória e capacidade de geração de qualidade gráfica, hoje já é possível utilizar a capacidade de processamento dos computadores inserindo princípios físico-químicos, ou mesmo dados fictícios a funções algorítmicas, de modo que cada experiência de cada jogador se torne única. Se antes, um jogo seria um conjunto predelineado e previsível, agora temos uma narrativa experiencial aberta.

DO DESENHO A MODELAGEM TRIDIMENSIONAL

A utilização de desenhos para representar uma história ou um acontecimento existe desde que o homem desenvolveu a habilidade de representação abstrata. Imagens nas pedras em cavernas pré-históricas demonstravam acontecimentos cotidianos, a caça como tema principal. Com o passar dos séculos, o homem foi refinando sua capacidade de representação pelos desenhos e passa a representar especialmente figuras sagradas, fato visto pelos desenhos que representam os deuses do antigo Egito, e que eram utilizados para ornamentar tumbas e contar a história do imperador falecido.

Durante o renascimento, os artistas começam a utilizar a perspectiva permitindo um arrojamento espacial muito melhor do que se tinha anteriormente. Além da perspectiva, também se obteve um grande avanço do estudo de anatomia e com isso os artistas conseguem dar mais realidade aos seus desenhos e esculturas.

Com o advento da revolução industrial surge a necessidade de criar projeções de modelos de máquinas e equipamentos. Com isso o desenho técnico ganha um avanço significativo para atender a demanda exata de dimensões para a produção.

Com a evolução tecnológica, chegamos a invenção do computador, cujo funcionamento era estritamente baseado na programação, não demorou muito para que se percebesse a necessidade de criar um computador com habilidades gráficas. Com isso, temos Lisa (1983)⁷, considerado o primeiro computador visual com interface gráfica, portanto não era mais necessário o uso de comandos de texto e sim via ícones.



Figura 3.4. Imagem do Computador *Apple Lisa*, da Macintosh (1984).

⁷ Computador pessoal desenvolvido pela empresa Apple, foi o primeiro computador a ter um *mouse* e uma interface gráfica. Mais informações em: <http://www.dmoz.org/Computers/Systems/Apple/Lisa/>.

O resto dessa evolução, já apresentada no capítulo anterior remete-nos aos gráficos utilizados hoje em dia, com modelagens tridimensionais e efeitos especiais capazes de criar visuais muito realistas, entretanto ainda se utiliza de todo o conhecimento que foi aprendido por toda a história para a criação de peças visuais como a perspectiva, a anatomia e diversos conhecimentos pictóricos.

Para se estabelecer uma relação mais aprofundada, ao observarmos como são os processos de criação para uma obra tridimensional podemos perceber todas as etapas da evolução artística, primeiramente são feitos os croquis, ou seja, rascunhos dos personagens sem adição de detalhe para posicionamento inicial e caracterização do mesmo. Como podemos ver no trabalho de Jon Diesta⁸ (2010):



Figura 3.5. *Concep Art* de Jon Diesta, *Monstros S. A.* (2010).

Após isso são criados as *concept arts*, como previamente descrito, coloca-se o personagem em uma situação ou paisagem ao qual o artista imagina que possa existir no trabalho final, então se tem a criação de vistas diferentes do personagem, como referência para a modelagem tridimensional, nesse momento a profundidade e anatomia do personagem se tornam de suma importância, pois será necessário planejar a estrutura da malha do objeto para que, no momento em que este for animado não ocorram defeitos. Após a modelagem, é feito o material e textura do personagem se utilizando de uma imagem plana como referência. Precisão é essencial para que não se criem bordas, no momento de criação do personagem, e nas paisagens é feito um estudo de iluminação para criar a atmosfera desejada.

⁸ Designer que trabalha para os estúdios Disney desde 2006. Link: <http://jondiesta.blogspot.com.br/>.

MODELAGEM PROCEDURAL

As técnicas para modelagem podem ser divididas em três classes, modelagem interativa, reconstrução por uso de dados do mundo real e modelagens procedurais baseadas em regras. Mesmo que todos esses processos possam levar a um nível de realismo alto, cada um deles apresenta deficiências. Modelos recriados com dados do mundo real, ou seja, tamanho, largura, aspereza, brilho entre outros, ou criados de modo interativo, copiando mentalmente ou através de imagens utilizando um programa de modelagem, não reagem com as diferenças do ambiente (neve, tempo seco, vento, etc.), esse aspecto é possível com a modelagem procedural, entretanto, esta se limita muito quanto ao controle do formato e, por ser um trabalho extenso de utilizações de parâmetros numéricos e regras matemáticas para os modelos tridimensionais.

O método de modelagem interativa como apresentado em *3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation* de Ami Chopine (2011, p. 21) é geralmente associado a utilização de uma imagem bidimensional ou mais e as transforma em modelos tridimensionais, conhecido também como modelagem tradicional. Essa técnica consiste na utilização de desenhos, de preferência de ângulos de visão diferentes aonde o modelador posiciona os pontos em posições estratégicas do desenho em todas as suas perspectivas adicionando profundidade e forma ao objeto.

Essa técnica permite um maior controle da aparência do objeto, entretanto se for necessário um alto número de criações pode-se ocupar muito tempo na produção e deixar o produto final muito caro.

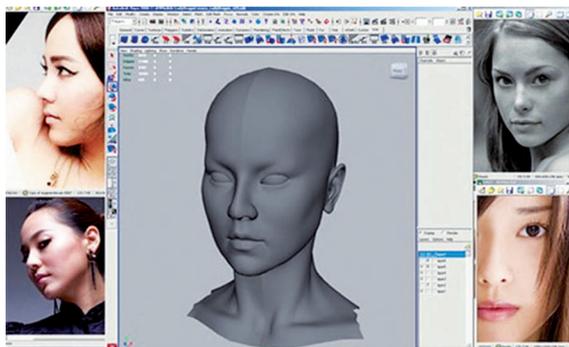


Figura 3.6. Imagem ilustrativa do processo de modelagem com imagens referenciais.

Além disso, essa técnica é muito intrínseca à habilidade do modelador, dado que muitas vezes o número de referências pode ser extremamente escasso caso o próprio artista não consiga criar suas próprias imagens. Esse problema e sua solução são tratados no texto *Structured Annotations for 2D-to-3D Modeling*, do pesquisador Yotam Gingold e Denis Zorin (2009)⁹, da Universidade de Nova York e Takeo Igarashi, da Universidade de Tóquio. Os autores do texto demonstram uma forma simplificada de trabalhar com as formas vetoriais tridimensionais, também conhecidas como primitivos, onde estas são deformadas de acordo com a necessidade do modelador para alcançar o formato desejado.

É criado uma base simplificada do objeto e então são criadas partes diferentes uma por vez, sempre obedecendo regras que foram anotadas no desenho quanto a sua estrutura e comportamento, dessa forma, se torna possível a criação de um objeto tridimensional somente utilizando de uma referência bidimensional. Outra ideia, pode ser observada no trabalho dos pesquisadores B. Benes e seus colegas da *Purdue University* (2014)¹⁰, que demonstraram uma maneira de resolver um problema recorrente na modelagem procedural. Ao automatizar a computação dos parâmetros da modelagem de um determinado tipo de árvore, utilizando valores característicos de uma determinada árvore já previamente trabalhada, assim, uma árvore que tenha sido utilizada anteriormente pode ser reconstruída e ao mesmo tempo, permitir todas as vantagens de uma modelagem procedural ao adicionar novos parâmetros aos que já existiam.

Eles demonstram um *framework* (Demonstração gráfica que une informações entre vários objetos promovendo uma funcionalidade geral) para modelagem procedural inversa de árvores reais. O *framework* promove uma nova modelagem procedural paramétrica que descreve uma grande diversidade de árvores, sendo esta uma forma de comparar as semelhanças entre duas árvores de forma eficiente e um método para determinar automaticamente a inclusão de parâmetros no modelo procedural para uma determinada árvore a partir de modelos pré-existentes.

⁹ Link: <http://mrl.nyu.edu/~dzorin/papers/gingold2009sa2.pdf>.

¹⁰ Link para o texto: http://web340.server8.webgo24.de/pirk_info/paper/Stava.etal-2014-Inverse-Modelling.pdf

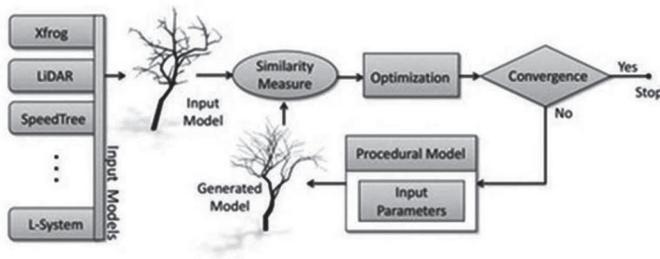


Figura 3.7. Framework para a criação de novas árvores.

No trabalho *Reconstructing 3D Tree Models from Instrumented Photographs* de Ilya Shlyakhter, Max Rozenoer, Julie Dorsey e Seth Teller (2001), pesquisadores do Instituto de Tecnologia de Massachusetts, propõem uma técnica híbrida para resolver o problema da reconstrução de uma árvore a partir de um modelo real encontrado na natureza.

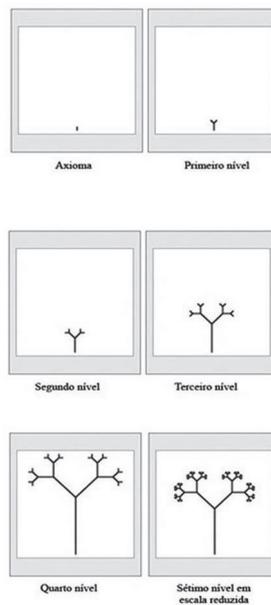


Figura 3.8. Etapas de um sistema L simples proposto por Lindenmayer e Prusinkiewicz.

O método envolve primeiramente a construção da base da árvore (tronco e os galhos maiores) e então é adicionado um *sistema L* por toda essa base. O sistema L foi proposto por Lindenmayer e Prusinkiewicz no texto *The Algorithmic Beauty of*

Plants (1968)¹¹, no qual consiste em uma linha paralela para reescrever um sistema em que a simulação começa com uma sequência de dados iniciais, chamados de axiomas. Em cada passo da simulação, reescreve-se regras substituindo todos os módulos predecessores pelos módulos seguintes, deixando até mesmo o mais simples sistema com uma estrutura parecida com a das plantas. Para a realização do processo de reconstrução, são necessários quatro passos: o primeiro consiste em segmentar as imagens da árvore e do fundo. Isso geralmente é realizado manualmente usando um editor de imagens, devido a uma falta de um algoritmo de segmentação automatizado.

Ao ser realizada a segmentação, é formada uma silhueta da imagem original, na segunda etapa as linhas são utilizadas em conjunto com informações de posicionamento para cada imagem para calcular uma *mesh* (malha tridimensional) aproximada da árvore. Após isso, o formato da árvore é trabalhado utilizando um *Visual Hull* (Entidade geométrica criada pela técnica de usar uma silhueta para construção de um modelo 3D. Funciona com a interpolação da extensão de duas silhuetas dando a capacidade de criação de uma profundidade). Após alinhar a forma do modelo tridimensional com base na *Visual hull*, é criado um eixo mediano para se unir a imagem formada pela *visual hull*¹². Após todo esse processo, é realizada a implementação do sistema L para a geração de galhos secundários e terciários. Podendo então ser proceduralmente criadas as folhas em cada galho menor.

Outra forma de trabalharmos a modelagem a partir de imagens, pode ser visto no trabalho de Ping Tan, Gang Zeng, Jingdong Wang Long Quan (2007)¹³, da Universidade de Ciências e Tecnologia de Hong Kong e Sing Bing Kang do departamento de pesquisa da Microsoft nos Estados Unidos. Os pesquisadores tentam propor uma forma de gerar modelos tridimensionais de aparência natural a partir de imagens fixas.

¹¹ Link para pdf: <http://algorithmicbotany.org/papers/abop/abop.pdf>

¹² Entidade geométrica tridimensional criada a partir de silhuetas.

¹³ Link para PDF: <http://research.microsoft.com/en-us/um/people/sbkang/publications/treemodelling07.pdf>

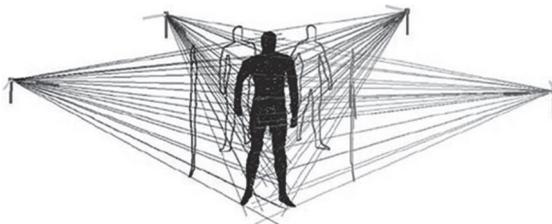


Figura 3.9. Construção de uma forma a partir de suas silhuetas.

Eles explicam que as árvores são difíceis de modelar, por conta da sua geometria complexa. A proposta do sistema deles de modelagem é simplificar todo o processo da modelagem.

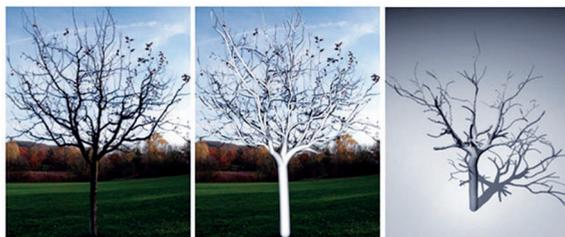


Figura 3.10. Modelagem baseada em figura única para estruturação.

Para tal finalidade eles criaram um sistema baseado em imagens, onde somente é necessária uma câmera manual. Estrutura esta que é utilizada em técnicas de *Motion Capture*¹⁴ (captura de movimento). Utilizando de pontos chave detectadas pelo mecanismo do *Motion Capture* se torna possível o cálculo de profundidade, próximo da estereoscopia que nosso cérebro realiza naturalmente para criar a percepção de distância.

Pensando na proposta de Masahiro Mori (1927-2005), cujo gráfico na figura a seguir, nos permite entender que quanto mais tentamos reproduzir a aparência de algo que nos é familiar, mais esta representação se torna sintética, podemos conjecturar que as técnicas existentes de modelagem de vegetação estão restritas ao nível de *hardware* gráfico devido a necessidade de economizar espaço

¹⁴ Processo de capturar de movimento de um ser real e de processamento digital, através de câmeras capazes de captá-los para animar um personagem tridimensional.

de processamento de polígonos para uma alta densidade de vegetação. Em termos estéticos, entre as técnicas de modelagem existentes, a que melhor se ajusta a curva do *Uncanny Valley* é a modelagem *poly by poly*¹⁵, devido a liberdade artística do desenvolvedor, caso este não tenha limitações com relação ao peso da malha tridimensional.

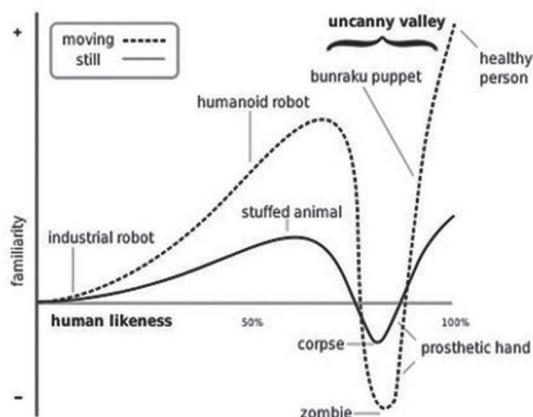


Figura 3.11. *Uncanny Valley*, por Masahiro Mori (1970).

Entretanto, como em *games* é necessário a renderização em tempo real, a utilização de modelos tridimensionais de alto nível de detalhamento não se faz possível quando são exibidos em alta densidade.

Dito isto, a modelagem procedural se apresenta como mais atraente aos desenvolvedores por ser um método que permite uma maior automação no desenvolvimento de objetos. Por se tratar de um método que utiliza de parâmetros matemáticos baseados nas leis da física do mundo real, este método automatiza os processos ambientais, por meio de simulações destes, de modo que o resultado será variável todas as vezes que a simulação ocorrer, variando esta mesma simulação com interações naturais e entre objetos exponencialmente de acordo com a quantidade de parâmetros estabelecidos.

Em contrapartida, alterações ou adição de parâmetros podem levar ao aparecimento de caracteres indesejados, uma vez que, obedecendo as leis da física,

¹⁵ Criação de uma modelagem tridimensional criando do zero todas as faces que pertencerão ao modelo.

qualquer alteração no processo de desenvolvimento, do ambiente, do clima, ou intervenções externas geram alterações locais nos objetos desenvolvidos, o que resulta em imprecisões nos resultados obtidos.

CÁLCULO FRACTAL

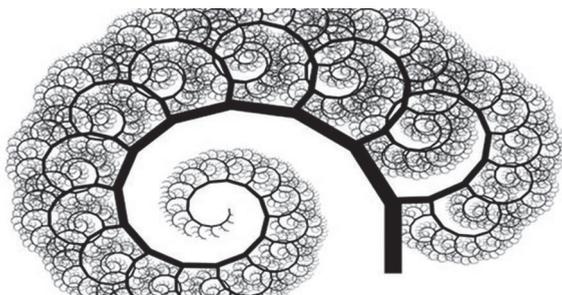


Figura 3.12. Exemplo de fractal.

Um fractal é um objeto geométrico não euclidiano que pode ser dividido em partes sendo que cada parte é semelhante ao objeto original. Geralmente por seguirem um padrão repetido. O termo fractal foi criado por Benoit Mandelbrot (1924-2010), matemático nascido na Polônia ao descobrir a geometria em meados da década de 1970. Quando se fala sobre cálculo fractal nos referimos a área da matemática que estuda e relativiza as propriedades e comportamentos de fractais. Esse estudo serve para explicar coisas que a geometria clássica não consegue explicar facilmente.

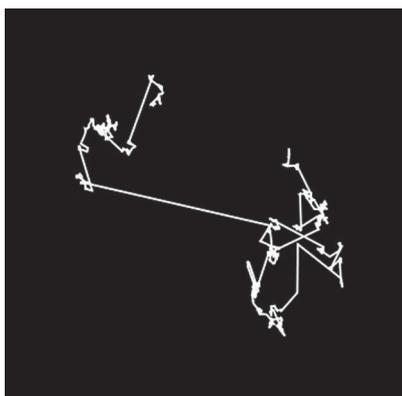


Figura 3.13. Fractal aleatório, voo de Lévy.

Os fractais podem ser divididos em três categorias: um sistema de funções iteradas, ou seja, possuem regra fixa de alteração geométrica; os chamados de fractais de fuga temporal, ou seja, são definidos por terem recorrências em pontos no espaço; fractais aleatórios que ocorrem sem um estado inicial fixo.

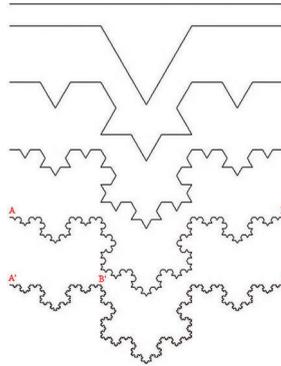


Figura 3.14. Sistema de funções iteradas, curva do floco de neve.

Com isso em mente, é possível perceber porque cálculos fractais são utilizados em modelagem procedural, devido a sua capacidade de gerar informação seguindo regras simples se torna possível desde simular cristais de neve até mesmo o comportamento natural das plantas. Ao observarmos uma planta detectamos padrões pseudo-fractais (detém uma estrutura similar ao longo de um intervalo com fim definido).

A propriedade de repetição é visível ao se reparar em uma folha ou ramo de folhas, cada parte das suas ramificações são semelhantes, embora não idênticas, permitindo uma alternância e uma singularidade para cada objeto criado a partir de seu cálculo fractal.

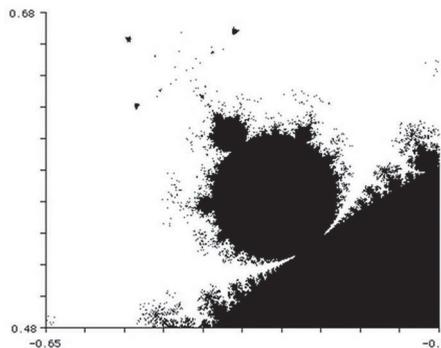
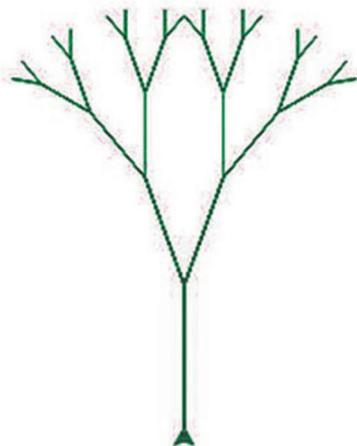


Figura 3.15. Exemplo de fractais de fuga do tempo, conjunto de Mandelbrot.

Um exemplo simples é o seguinte código na linguagem *Python* com seu módulo gráfico *Turtle* funcionando em um computador pessoal¹⁶.

```
import turtle
def tree(branchLen,t):
    if branchLen > 5:
        t.forward(branchLen)
        t.right(20)
        tree(branchLen-15,t)
        t.left(40)
        tree(branchLen-15,t)
        t.right(20)
        t.backward(branchLen)
def main():
    t = turtle.Turtle()
    myWin = turtle.Screen()
    t.left(90)
    t.up()
    t.backward(100)
    t.down()
    t.color("green")
    tree(75,t)
    myWin.exitonclick()
main()
```



Este exemplo gera uma forma semelhante a estrutura de uma árvore obedecendo pequenos parâmetros, por exemplo, a linha *t.right(20)* determina a angulação na qual o vetor se dirigirá quando for se mover para o lado direito, o *t.left(40)* para o lado esquerdo rotacionará os 20° para anular para o lado direito anterior mais 20° para o lado esquerdo para gerar a ramificação, quando chega no valor de 5 ramificações ele volta uma etapa de ramificação e, assim por diante.

Como vimos neste capítulo, a utilização de modelagem procedural e cálculos fractais estão intrínsecos para a construção de modelos tridimensionais de vegetação para sua utilização em jogos. Com isso em mente, no próximo capítulo abordaremos um exemplo da utilização dessa forma de produção de conteúdo no mercado com o jogo de *The Witcher III* que utilizou do programa *SpeedTree* que gera vegetação de forma procedural.

¹⁶ Link para instruções e criação interativa: <https://interactivepython.org/runestone/static/python-nds/Recursion/python-ndsintro-VisualizingRecursion.html>

