

Trabalho 2 de Conceitos de Linguagens de Programação

Eduarda Pereira Medeiros

Fractal de Mandelbrot

É um fractal formado pelo conjunto de Mandelbrot, que consiste nos pontos c no plano complexo para o qual a sequência a seguir não tende ao infinito:

$$z_0 = 0$$
$$z_{n+1} = z_n^2 + c$$

A representação gráfica do fractal de Mandelbrot mostra uma figura central em forma de cardióide, acompanhada de “quase-círculos” que tangenciam essa região principal. Cada uma das figuras formadas têm seu próprio conjunto infinito de pequenos círculos e quanto mais a imagem é ampliada, mais réplicas do desenho são reveladas.

As imagens do conjunto são em “preto e branco” se considerarmos apenas a ausência ou a presença dos pontos. No entanto, é possível usar gradientes para tornar a representação do fractal colorida ao levar em consideração quantas iterações cada ponto leva para divergir. Por exemplo, os pontos que divergem rapidamente podem ser mais claros e os que levam mais tempo, mais escuros. Dessa forma, é possível utilizar esquemas de cores para gerar belas composições visuais detalhadas.

Implementação

Conforme determinado na especificação do trabalho, foram escolhidas duas linguagens com características bastante distintas para o desenvolvimento da aplicação: **C** e **Python**, sendo **C** utilizada para a realização dos cálculos que permitem determinar se um ponto está ou não no conjunto de Mandelbrot e, se não estiver, quantas iterações ele leva para divergir. **Python** foi utilizada para a interface gráfica que exibe o fractal gerado e permite que o usuário digite seus próprios valores e amplie a imagem.

C é uma linguagem voltada para desempenho e controle direto sobre o hardware, o que justifica seu uso para a implementação do serviço de cálculo. O programador é responsável pela gestão manual da memória, pode definir o tipo de cada variável e precisa lidar com ponteiros e endereços. Isso a torna extremamente eficiente, mas também mais complexa e sujeita a erros como vazamentos de memória. No programa em questão, a aplicação direta da fórmula evita esse tipo de problema e aproveita o bom desempenho da linguagem para otimizar o programa.

Já **Python** é uma linguagem de alto nível, interpretada e com sintaxe simples, criada para ser legível e rápida de escrever. Ela gerencia automaticamente a memória, não exige declaração explícita de tipos e oferece bibliotecas prontas para diversas aplicações. No entanto, Python é mais lento que C quando se trata da velocidade de processamento. Como a linguagem oferece uma série de bibliotecas úteis para o desenvolvimento de interfaces gráficas, como a Tkinter, a linguagem foi considerada ideal para a aplicação em questão.

Apesar do contraste entre as linguagens, foi possível juntá-las utilizando uma biblioteca compartilhada gerada a partir do código feito em C, o qual é importado pelo código em Python para calcular os valores que formam a imagem mostrada na interface gráfica. A biblioteca **ctypes** em Python conecta as duas linguagens, carregando a biblioteca, definindo os parâmetros que a função recebe e realizando a chamada dela com os argumentos passados. O **Makefile** foi utilizado para facilitar o processo de compilação com comandos simples, mas a aplicação também pode ser executada no Windows. Mais detalhes sobre como compilar e executar o programa podem ser encontrados no arquivo README.md localizado no repositório do GitHub.

No desenvolvimento da interface gráfica, foram utilizadas 3 bibliotecas Python para tornar a implementação mais simples e intuitiva:

- **Tkinter:** Utilizada para criar os elementos essenciais da interface: janelas, botões, caixas de texto e a área de desenho que exibe o fractal. É uma biblioteca padrão de Python, portanto, não costuma exigir instalação extra, facilitando sua distribuição.
- **NumPy:** Muito utilizada em aplicações que exigem cálculos, a NumPy foi utilizada na aplicação para a criação do array que armazena o valor de cada pixel. Considerando a quantidade grande de dados manipulados para a geração do fractal, o uso da biblioteca foi útil para a conversão rápida dos números de iteração para valores de cor em escala de cinza.
- **Pillow:** Cria a imagem que representa o fractal a partir dos valores dos pixels armazenados no array NumPy, então a converte para um formato que Tkinter consegue exibir. Disponibiliza ferramentas úteis para a manipulação de imagens, como a conversão para o modelo RGB.

Fontes utilizadas na consulta:

- Artigo: de Petta, José Luiz U., and Júlia Harue S. de Petta. "Fractais: obras de arte geradas por equações matemáticas." Pensamento & Realidade 3 (1998).
- Site: [A quick explanation of the Mandelbrot set | by Alonso Del Arte](#)
- Site: [The Mandelbrot set](#)