

29/10/2019

Trabajo Práctico 4 – Fractales

Conjunto de Cantor

ALGORITMOS GENÉTICOS 2019 – GRUPO 16
ANTONELLI (44852) – RECALDE (44704) – ROHN (41355)

Índice

INTRODUCCIÓN2

 FRACTALES 2

 CONJUNTO DE CANTOR..... 2

 EJERCICIO PROPUESTO..... 2

CÓDIGO3

SALIDA POR PANTALLA.....4

EXPLICACIÓN DE FUNCIONAMIENTO5

CONCLUSIONES6

INTRODUCCIÓN

Fractales

Un fractal es un objeto geométrico cuya estructura básica, fragmentada o aparentemente irregular, se repite a diferentes escalas. El término fue propuesto por el matemático Benoît Mandelbrot en 1975 y deriva del latín “Fractus”, que significa quebrado o fracturado. Muchas estructuras naturales son de tipo fractal. La propiedad matemática clave de un objeto genuinamente fractal es que su dimensión métrica fractal es un número no entero.

Conjunto de Cantor

El conjunto de Cantor es el fractal por excelencia, y también el primero conocido. Fue ideado por Georg Cantor en 1883 como ejemplo de conjunto de longitud cero cuyos puntos se pueden identificar uno a uno con todos los puntos de una recta (que tiene longitud infinita). Para su construcción se parte de un segmento de longitud 1. Se divide en tres partes iguales y se elimina la parte central abierta (es decir, sin incluir los extremos). Cada una de las otras dos se divide en tres partes iguales y se eliminan las partes centrales (abiertas) en cada una de ellas. Se procede igual con cada uno de los cuatro segmentos que quedan. Y se repite el proceso infinitas veces.

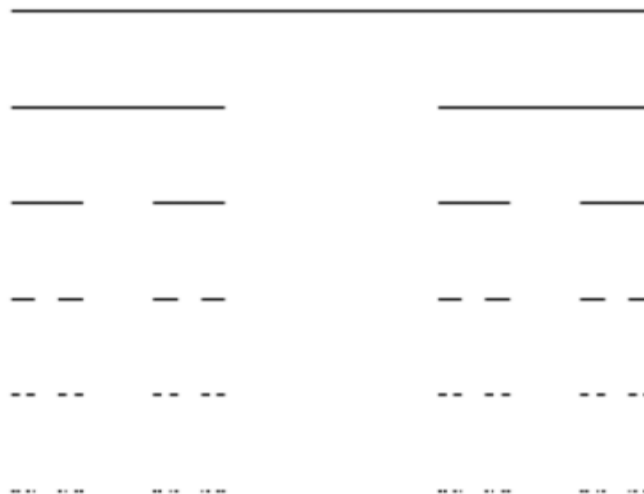


Figura: Primeros pasos de la construcción del Conjunto de Cantor.

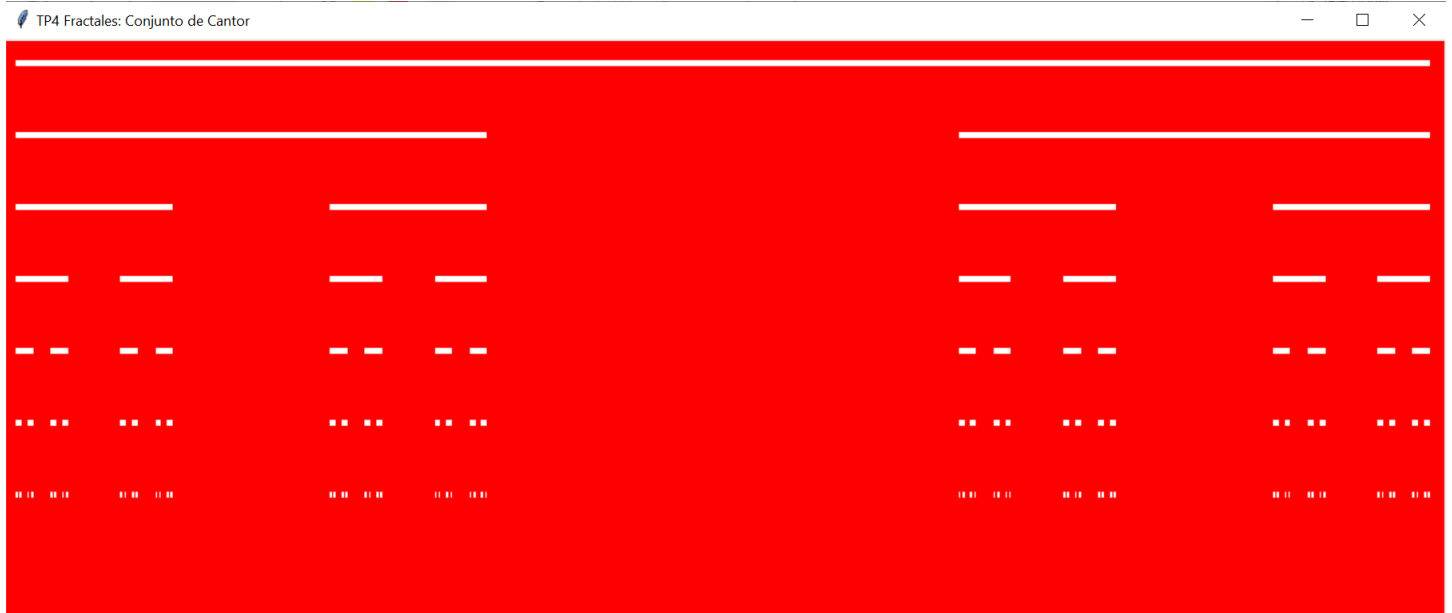
Ejercicio Propuesto

Programar utilizando la técnica de recursividad el modelo fractal del conjunto de Cantor.

cantor.py (Lenguaje: Python)

```
1.  #!/usr/bin/env python
2.  # -*- coding: utf-8 -*-
3.  """
4.  ALGORITMOS GENÉTICOS 2019 - TP4 FRACTALES
5.  Programar utilizando la técnica de recursividad el modelo fractal del CONJUNTO DE CANTOR.
6.
7.  -FECHA DE ENTREGA: 29/10/2019
8.  -AUTORES: Antonelli, Nicolás - Recalde, Alejandro - Rohn, Alex
9.  """
10.
11. from tkinter import Tk, Canvas
12. import time
13.
14. # Define Principal Variables
15. iterations = 7
16. window_title = "TP4 Fractales: Conjunto de Cantor"
17. width_max = 1200
18. height_max = width_max * (2/5)
19. x_start = 10
20. y_start = 30
21. y_step = 60
22. color_bkg = "red"
23. color_line = "white"
24. width_line = 5
25. delay = 0.2
26.
27. # Line Drawing Function
28. def draw_line(x1, x2, y):
29.     canvas.create_line(x1, y, x2, y, fill=color_line, width=width_line)
30.     canvas.update() # Draws the Lines Step by Step
31.
32. # Recursive Function for Fractal Drawing
33. def cantor(x1, x2, y, iteration):
34.     if iteration > 0:
35.         time.sleep(delay)
36.         draw_line(x1, x2, y)
37.         point1 = x1 + (x2-x1) * (1/3)
38.         point2 = x1 + (x2-x1) * (2/3)
39.         y2 = y + y_step
40.         cantor(x1, point1, y2, iteration-1)
41.         cantor(point2, x2, y2, iteration-1)
42.
43. # Main Function
44. if __name__ == "__main__":
45.     # Main Window
46.     window = Tk()
47.     window.title(window_title)
48.
49.     # Define Canvas
50.     canvas = Canvas(window, width=width_max, height=height_max, background=color_bkg)
51.     canvas.grid()
52.
53.     # Call Recursive Function and Display Canvas
54.     cantor(x_start, width_max - x_start, y_start, iterations)
55.
56.     # Don't Close the Canvas at the End
57.     canvas.mainloop()
```

SALIDA POR PANTALLA



EXPLICACIÓN DE FUNCIONAMIENTO

Primero definimos las variables principales que necesitaremos:

- ✓ Cantidad de “primeros niveles” o iteraciones (ejemplo: 7) de los infinitos que conforman el Fractal del Conjunto de Cantor
- ✓ Título y Tamaño de la Ventana
- ✓ Coordenada Inicial de la primera Línea
- ✓ Distancia en Y entre Línea y Línea
- ✓ Color de Fondo de la Ventana
- ✓ Color y Tamaño de Línea
- ✓ Demora al graficar entre Línea y Línea (ejemplo: 0.2 segundos)

Se inicializa la Ventana dentro de un Canvas (significa “Lienzo de Dibujo”) con los parámetros necesarios a partir de las variables de arriba, y se llama a una función recursiva llamada “Cantor” pasándole las coordenadas iniciales y finales y la posición en Y donde queremos que dibuje la primera línea.

La función Cantor lo que hace es, primero esperar un tiempo expresado en la variable Delay para que la creación del Fractal sea menos instantánea y se dibuje paso a paso; luego llamar a otra función “Draw_Line” que lo que hace es graficar en el Canvas la línea básicamente, y actualizar la ventana para visualizarla cuando esta se dibuja.

Después de esto, aún en la función Cantor, se divide la línea dibujada en 3: se guarda en variables locales la posición en X de $1/3$ y de $2/3$ de la línea, y se guarda una nueva posición en Y tomando la actual hasta ahora sumándole la variable de “distancia entre línea y línea” o Step. Teniendo ahora estos nuevos valores en X e Y, se pasan estas coordenadas recursivamente en otros llamados a la función Cantor (uno desde el inicio de la línea “base” en ese nivel hasta $1/3$ de la longitud, y otro llamado desde $2/3$ hasta el final de la línea base; ambos en la altura de la nueva Y). Esto se repite para cada línea en cada “nivel” a dibujar, hasta que se alcancen la cantidad de iteraciones y estén dibujadas todas las líneas de cada nivel. En cada iteración se hacen 2 llamados recursivos, por lo que en total habrá $2^N - 1$ llamados a función Cantor y el mismo número de Líneas, donde N es la cantidad de “niveles” que queremos representar del Fractal.

Finalmente, se llama a un método que hace que la ventana no se cierre luego de graficar todas las Líneas deseadas del Fractal.

CONCLUSIONES

La realización de un Fractal posee un algoritmo muy poco complejo, pues solo se necesitó una función para graficar, y una función recursiva con información de coordenadas sencilla.

El Conjunto de Cantor podría decirse que es uno de los más sencillos de implementar, pero aun así en su simpleza está bueno para practicar y entender cómo funciona este tipo de geometría.

La función recursiva como se calculó antes, será llamada $2^N - 1$ veces a lo largo del programa y dibujará la misma cantidad de líneas (donde N es la cantidad de iteraciones o “niveles” que queremos representar del Fractal) por lo que en un N de valor entre 1 y 8, y un Delay entre línea y línea de 0.2 segundos, se completa en menos de un minuto. Este tiempo se duplicaría con un N = 9 y así sucesivamente por cada unidad agregada a N.

Al estar la ventana a dibujar dentro de un monitor compuesto de Píxeles, hay un límite físico que nos impide seguir altas iteraciones del Conjunto de Cantor si no comenzamos a simular el “Zoom”; hasta 8 iteraciones en una matriz amplia de píxeles (Display Full HD) puede realizarse perfectamente; luego es probable que los números de las posiciones de la línea comiencen a dar inevitablemente no enteros y el tamaño de la línea total sea menor que el tamaño de un único píxel, por lo que se necesitaría una pantalla más grande, o como dijimos antes, simular un Zoom a través de los niveles del Fractal. Sin embargo, hacer esa simulación de Zoom en la ventana, y pensándolo con movimiento suave y no de golpe, si sería un trabajo bastante más complejo para realizar.