



Algoritmos Genéticos

Trabajo Práctico N°4 – Título: "Fractales, conjunto de Cantor"

Botello, Andrés Julián: andresjbotello@gmail.com

Ciriaci, Santiago: santiagociriaci@gmail.com

Corsetti, Ornella: orcorsetti@gmail.com

GRUPO 2

CICLO LECTIVO 2018

16/10/18

Índice

Enunciado.....2

 Solución.....2

 Código.....3

 Salida del código.....4

 Conclusión.....4

Enunciado

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

Solución

Para poder generar el conjunto de Cantor, hemos decidido usar una librería del lenguaje de programación Python llamada "Graphics"¹ que permite, dentro de los cientos de funciones que alberga, dibujar formas geométricas básicas.

"The library is designed to make it very easy for novice programmers to experiment with computer graphics in an object oriented fashion. It is written by John Zelle for use with the book "Python Programming: An Introduction to Computer Science" (Franklin, Beedle & Associates)."

Nuestra solución está estructurada en la definición de dos funciones:

1. Función "inicializar()": donde cargamos todos los parámetros de entrada (datos de la ventana donde se muestra el gráfico resultante, largo inicial de la línea y su ubicación, separación entre ellas)
2. Función "conjunto_Cantor(Win,x,y,h,Len,tope)": recibe los parámetros enviados en un principio por la función anterior y dibuja las líneas haciendo uso de la técnica de recursividad.

¹Véase documentación de la librería Graphics en <http://mcsp.wartburg.edu/zelle/python/graphics.py>

Código

```
from graphics import *

def conjunto_Cantor(win,x,y,h,Len,tope):

    if Len < tope: return

    line = Line(Point(x, y), Point(x+Len, y)) #la línea usa dos argumentos, que son los vértices

    line.setWidth(5) #esta función define el ancho de la línea

    line.draw(win)

    conjunto_Cantor(win,x,y+h,h,Len//3) #la doble barra realiza un truncamiento

    conjunto_Cantor(win,x+Len*2//3,y+h,h,Len//3)


def inicializar():

    Len=750 #Largo de línea

    tope=Len

    for i in range(5):

        tope=tope//3

    win = GraphWin("Conjunto de Cantor", Len+300, 600) #dibujo las medidas y nombre de la ventana que
muestra la salida

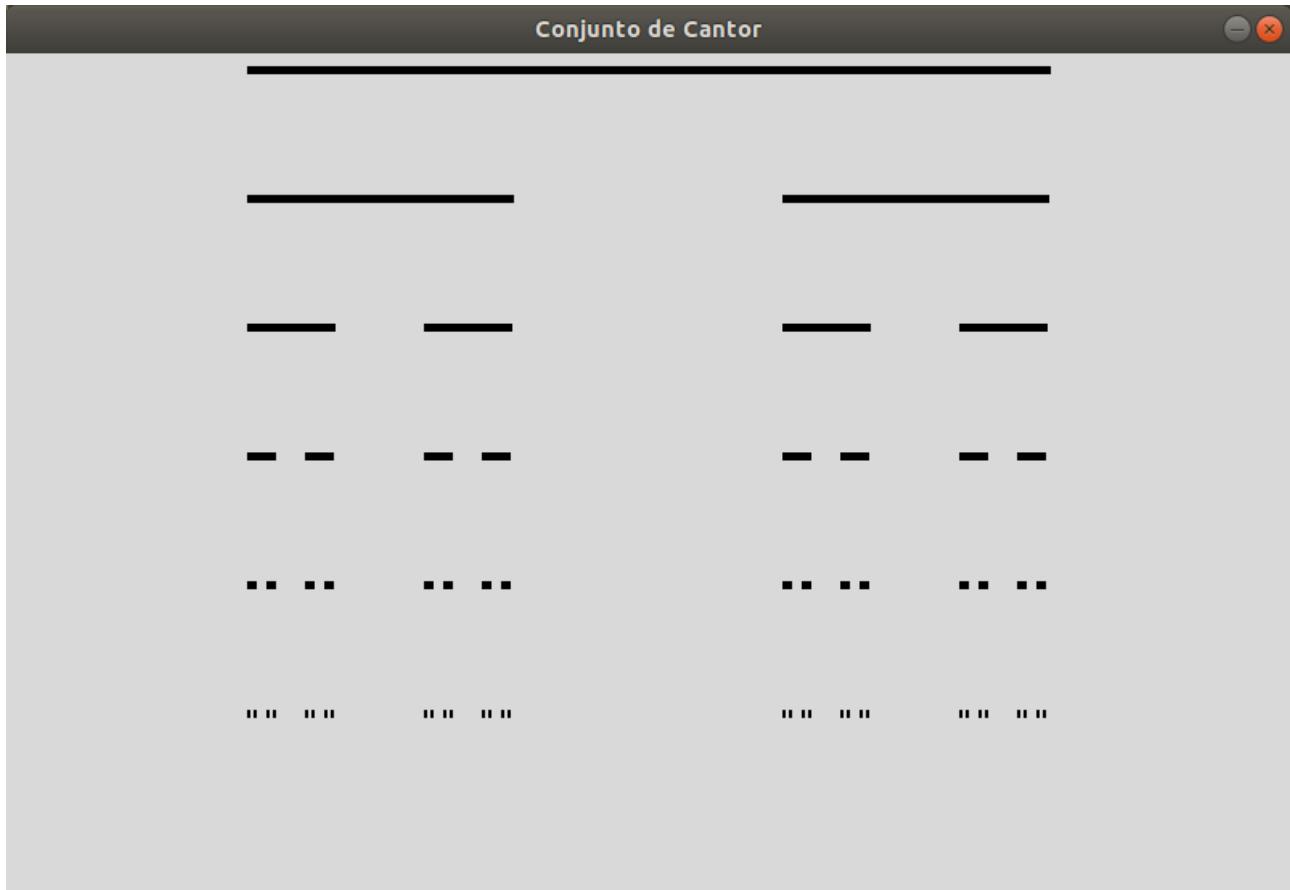
    conjunto_Cantor(win,150,10,80,Len,tope) #Parámetros: Datos de ventana + Sep. Izquierda + Sep. Superior +
Sep. "Renglones" + Largo Inicial de Línea + Condición Final

    win.getKey() #llamo a esta función para que la ventana no se cierre automáticamente

    win.close()

inicializar()
```

Salida del código



Conclusión

Podemos concluir que es válido hacer uso de la técnica de recursividad ya que los fractales se basan en la autosimilitud de sus partes facilitando la codificación. Siendo ésta la manera más óptima de resolver este problema.