FRACTALES EN EL MUSEO DEL PRADO

Y PHYTON

El cuadro es "Manzanos en Normandía"
de Carlos de Haes

El fractal se encuentra en los árboles

¿QUÉ ES UN FRACTAL?

Los fractales son objetos geométricos que tienen la particularidad de que su estructura o forma básica se repite dentro del mismo objeto en diferentes escalas. los fractales también se encuentran en la naturaleza.

Con una app para pintar hemos intentado recrear el cuadro con el arbol que hemos generado

GENERAR UN ÁRBOL CON PHYTON

La idea básica de este fractal es muy simple, se empieza con una línea que es el tronco del árbol y que se divide en dos ramas más pequeñas. Ese mismo proceso se repite para cada una de las ramas hasta llegar a una cantidad límite de divisiones. para ello vamos a utilizar lás gráficas tortugas, con esto podrremos decirle al programa que se mueva y dibuje segun lo que le indiquemos

```
CTRONCO = (100,80,0)
CTRONCOVAR = 30
CHOJAS =  (66,190,9)
CHOJASVAR = 100
CFONDO = (255,255,255)
```

Para poner color hay que importar una biblioteca RGB donde cada color tiene 3 numeros asignados. Esto se añaderá posteriormente al código final

```
f arbol(tam, prof):
  if prof==0:
      return
  else:
      turtle.forward(tam)
      turtle.left(45)
      arbol(tam*2/3, prof-1)
      turtle.right(90)
      arbol(tam*2/3, prof-1)
      turtle.left(45)
      turtle.back(tam)

def hoja(t, a):
    turtle.begin_fill()
    turtle.right(a/2)
    turtle.circle(t, a)
    turtle.left(180-a)
    turtle.circle(t, a)
    turtle.left(180-a/2)
    turtle.end_fill()
```

Aqui vemos como será la estructura del programa donde podemos ver la profundidad del árbol, el tamaño y las divisiones.

Para añadirle hojas, creamos otro código tortuga (las hojas ya no son fractales)

```
ANG = 20
RAND = 10
REL = 2/3
RANDT = 60
GROSORTRONCO = 2
TAMINIC = 150
TAMHOJA = 4
ANGHOJA = 180
PROF = 10
```

Estas son todas las propiedas que va a tener nuestro árbol, desde el grosor del tronco hasta el ángulo que forman las ramas

El codigo final quedaría así (es tan largo porque hay que repetir muchos procesos):

```
Código fuente

import turtle
import random

#Variables globales
ANG = 30        #ángulo de inclinación para las ramas
RAND = 30       #factor de aleatoriedad del ángulo de
inclinación (grados)
REL = 4/5       #relación entre la rama y las sub ramas
RANDT = 60      #factor de aleatoriedad en el tamaño de las
ramas (%)
GROSORTRONCO = 0        #pixeles que se le suman al grosor del
árbol
TAMINIC = 100           #tamaño del tronco inicial en pixeles
TAMHOJA = 5             #tamaño de la hoja
ANGHOJA = 180   #ángulo de las puntas de las hojas (180 =
circulos)
PROF = 10       #cantidad de niveles en el árbol (más de 10
puede durar mucho dibujándose)

CTRONCO = (67,128,211)  #color del tronco. tres números entre 0
y 255
CTRONCOVAR = 40         #factor de aleatoriedad en el color del
tronco
CHOJAS =  (255,98,109)  #color de las hojas
CHOJASVAR = 100         #factor de aleatoriedad en el color de
las hojas
CFONDO = (255,255,255)  #color de fondo

# Función que dibuja un árbol fractal
# Entradas:
#   t: tamaño del segmento inicial en pixeles
#   d: profundidad total del árbol
# Salidas:
#   Dibujo del árbol en pantalla
# Restricciones: no
def arbol(t, d):
    if d==0:
        turtle.forward(t)
        hoja(TAMHOJA, ANGHOJA)
        turtle.penup()
        turtle.back(t)
        turtle.pendown()
        turtle.color(CTRONCO)
        return
    else:
        angulo1 = ANG + random.randrange(-RAND, RAND+1)
        angulo2 = ANG + random.randrange(-RAND, RAND+1)
        tamano = t + t*random.randrange(-RANDT, RANDT+1)/100
```

```
        turtle.pensize(d+GROSORTRONCO)
        turtle.forward(tamano)
        turtle.left(angulo1)
        arbol(t*REL, d-1)
        turtle.right(angulo1+angulo2)
        arbol(t*REL, d-1)
        turtle.color(colortronco)
        turtle.left(angulo2)
        turtle.penup()
        turtle.back(tamano)
        turtle.pendown()


# Función que dibuja una hoja
# Entradas:
#   t: tamaño de la hoja
#   a: ángulo de las puntas de las hojas
# Salidas:
#   Dibujo de una hoja en la posición actual de la tortuga
# Restricciones: no
def hoja(t, a):
    turtle.color(variacioncolor(CHOJAS, CHOJASVAR))
    turtle.begin_fill()
    turtle.right(a/2)
    turtle.circle(t, a)
    turtle.left(180-a)
    turtle.circle(t, a)
    turtle.left(180-a/2)
    turtle.end_fill()


# Función que genera una variación de un color en RGB
# Entradas:
#   color: tupla con tres valores enteros entre 0 y 255
#   var: cantidad máxima de variación permitida en los valores
RGB
# Salidas:
#   Tupla de tres valores enteros entre 0 y 255 que es una
variación
#   del color original.
# Restricciones: no
def variacioncolor(color, var):
    Rd = random.randrange(-var, var+1)
    Gd = random.randrange(-var, var+1)
    Bd = random.randrange(-var, var+1)
    R, G, B = color
    R += Rd
    G += Gd
    B += Bd
    if R > 255:
        R = 255
    elif R < 0:
        R = 0
    if G > 255:
```

```
#   del color original.
# Restricciones: no
def variacioncolor(color, var):
    Rd = random.randrange(-var, var+1)
    Gd = random.randrange(-var, var+1)
    Bd = random.randrange(-var, var+1)
    R, G, B = color
    R += Rd
    G += Gd
    B += Bd
    if R > 255:
        R = 255
    elif R < 0:
        R = 0
    if G > 255:
        G = 255
    elif G < 0:
        G = 0
    if B > 255:
        B = 255
    elif B < 0:
        B = 0
    return R, G, B

# Función que inicializa la posición de la tortuga
la función
# de dibujar árbol fractal.
def init():
    turtle.speed(0)
    turtle.colormode(255)
    turtle.clear()
    turtle.penup()
    turtle.home()
    turtle.left(90)
    turtle.back(200)
    turtle.pendown()
    turtle.hideturtle()
    turtle.color(CTRONCO)
    turtle.bgcolor(CFONDO)
    arbol(TAMINIC, PROF)
    turtle.done()

init()
```