**Trabajo práctico 2**

FECHA DE ENTREGA: 17/11/2023

Características:

* Los ejercicios deben ser realizados en un lenguaje interpretado.
* El trabajo deberá ser desarrollado enteramente por los alumnos del grupo, no será permitido compartir código o descargarlo de internet (será comprobado) .
* El programa debe ser legible y adoptar las buenas prácticas de programación.
* El código fuente debe estar bien comentado y acompañado con instrucciones de ejecución, parámetros de entrada etc.
* Los alumnos deben defender exitosamente el trabajo en clase mediante una presentación.
* Las consignas enuncian el funcionamiento mínimo de los programas, cualquier feature de más será tenido en cuenta a la hora de calificar el trabajo.

La falta de alguno de estos aspectos será motivo de re-entrega o desaprobación del práctico

**Ejercicio:**

* Curva de Hilbert
  + No terminales: X, Y
  + Terminales: +, −, F
  + Cadena Inicial: X
  + Reglas de producción: (X→+ YF - XFX - FY +), (Y →- XF + YFY + FX -)
  + Angulo: 90º
* Interpretación:
  + F: Dibujar Segmento (adelante)
  + +: Girar a la Izquierda
  + -: Girar a la Derecha
  + X, Y: No se hace nada.

**Investigación**

También conocida como la curva que recubre el plano de Hilbert, es una curva fractal continua que recubre el plano descrita inicialmente por el matemático alemán David Hilbert en 1891.

Se construye mediante el procedimiento siguiente:

* Se parte de un cuadrado
* Se hallan los puntos medios de los lados y se forman cuatro cuadrados iguales. Unimos los puntos medios de estos cuadrados mediante tres segmentos (Figura 1)
* Se repite el proceso en cada unos de los cuadrados anteriores uniendo además mediante segmentos adicionales (en azul) las terminaciones de las líneas poligonales en cada cuadrado (en verde) (Figura 2)
* Se repite este proceso indefinidamente.

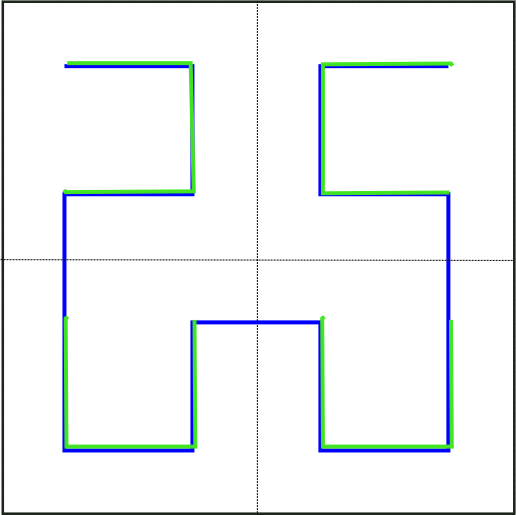
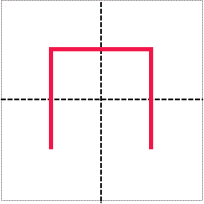
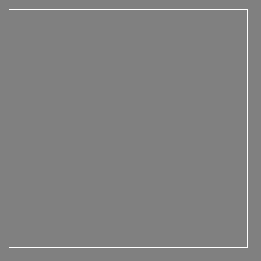


Figura 3

Figura 1

Figura 2

**Modulo Turtle en Python**

Importar el modulo:

* from turtle import \*

Nuestra ventana

Mostrar la ventana: mainloop()

Titulo de la ventana: title(‘…’)

Color del fondo: bgcolor(‘…’) ejemplo (‘black’)

Métodos Utiles

* speed(): le damos una velocidad en la que dibuja INVESTIGAR
* circle(): crea un circulo de los pixeles que le demos. Ejemplo circle(60)
* Cuadrados, lo hacemos con los métodos, forward(), right() y left()
* pencolor(‘…’): cambiar los colores de las líneas. Ejemplo Pencolor(‘green’)
* pensize(): cambiar el tamaño de la línea. Ejemplo: pensize(3)
* penup(): saltarnos una líneas, es decir que no nos grafique la línea, va a saltar la siguiente línea que aparezca en el código. Ejemplo penup()
* pendown(): Para que aparezca de nuevo la línea. Ejemplo pendown()

Métodos de Turtle

* forward(): se utiliza para mover la Turtle hacia adelante una distancia determinada en la dirección de la Turtle.
  + Avanza cierto tamaño de pixeles a donde apunta la flecha Ejemplo forward(100)
* backward() : Se usa para mover la Turtle hacia atrás una distancia determinada en la dirección de la Turtle.
  + Lo que hace es regresar ciertos pixeles hacia atrás, no borra nada. Ejemplo backward(50)
* left() : Se usa para rotar la Turtle en la dirección izquierda en un ángulo específico.
  + Hace que gire cierto grado hacia la izquierda. Ejemplo: left(90)
* right() : Se usa para rotar la Turtle en la dirección correcta en un ángulo específico.
  + Hace que gire cierto grado hacia la derecha. Ejemplo: right(90)
* goto() : Se usa para mover la Turtle a la ubicación especificada (coordenadas (x, y)).
  + Dibujar en otra ubicación de la ventana, tener en cuenta que va a dibujar la linea cuando hace el movimiento.
* penup() : Se utiliza para especificar que no se realizará ningún dibujo mientras se mueve.
* pendown() : Se usa para especificar que ese dibujo se hará mientras se mueve.
* done() : se utiliza para especificar que el trabajo de la Turtle se ha completado.

**Uso de For y While con Turtle**

Creamos un cuadrado mediante un bucle For

fillcolor(): rellenamos con color al cuadrado

fillcolor(‘red’)

begin\_fill()

for i in range(4):

forward(100)

right(90)

end\_fill()

begin\_fill() y end\_fill(): para iniciar y terminar relleno

Creamos un grafico mediante un bucle While

Importando import random, vamos a poder crear numero random, y asi distintos colores

x=1

while x<400

r=random.randint(0,255)

g=random.randint(0,255)

b=random.randint(0,255)

colormode(255)

pencolor(r,g,b)

forward(100)

right(91)

x+=1

colormode(): para que elija cualquiera color de los 255

pencolor(r,g,b): la grafica nos va a salir bien rgb

Características del puntero

bob = Turtle() //Obtenemos nuestro puntero

bob.color(‘red’) //Cambiamos el color a nuestro puntero

bob.shape(‘turtle’) //Cambiamos la figura de nuestro puntero, ejemplo una tortuga

bob.width() //Cambiamos el tamaño de nuestro puntero