**Criando Robot**

***Mário Leite***

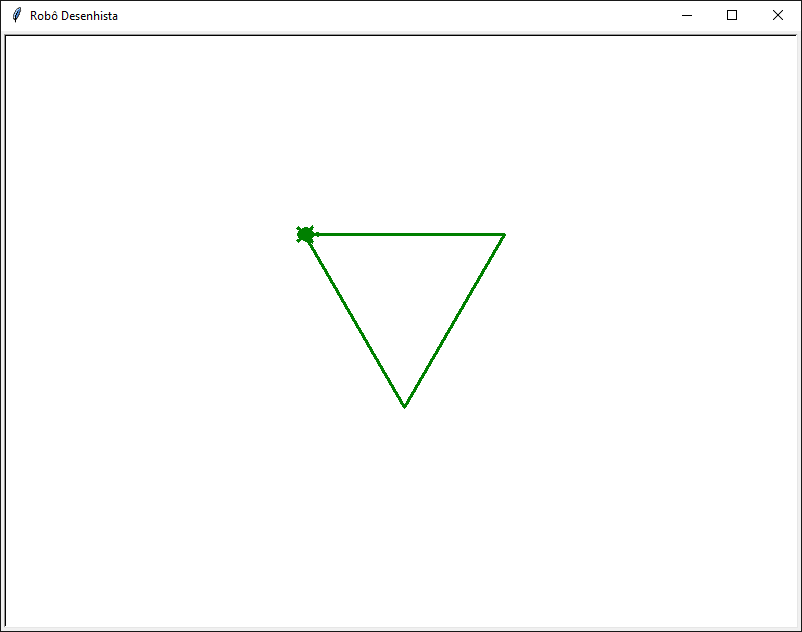
...

Segundo **Pedro Loos**, um excelente *youtuber*, estudioso das Ciências Exatas e responsável por um canal de ciências no You Tube: “*surgiu uma espécie de conspiração alertando para o fato de que cada vez mais os conteúdos gerados online parecem vir de Inteligências Artificiais ou robots*...” .

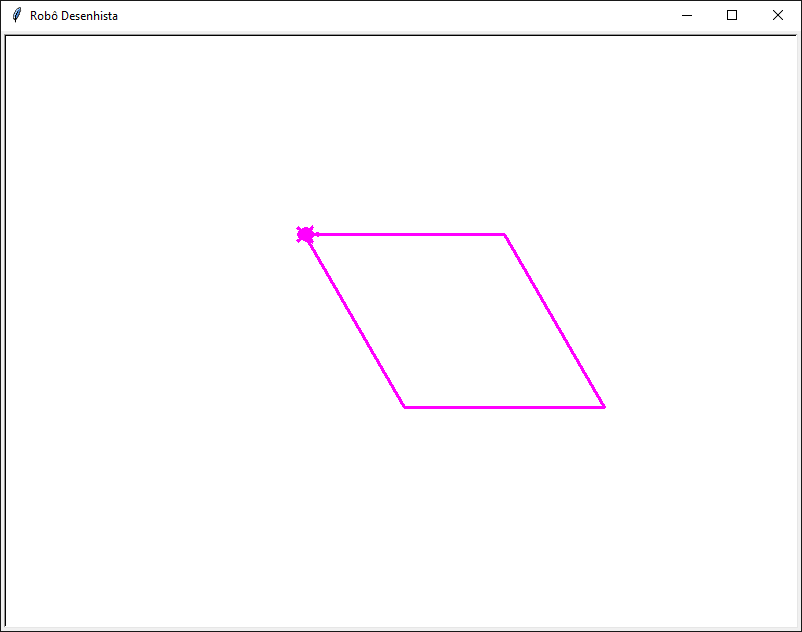
Esta dedução parece está se tornando assustadoramente verdade, pois a cada dia mais e mais “coisas estranhas” aparecem na Internet; o que ele conclui com o seguinte título da postagem: “A Teoria da Internet Morta”. Uma teoria um tanto quanto bizarra, nas suas próprias palavras, mas que ele mesmo duvida que seja tão bizarra assim... Na verdade, ele fala sobre dados bastante interessantes, tais como: cerca de duas em cada três pessoas se conectam à internet diariamente, a cada minuto são postados 500 horas de vídeo no You Tube, 60 mil fotos postadas no Instagram e 200 milhões de e-mails enviados; dados fantásticos! E continuando, diz que o mais interessante é que se pode pensar que todas essas ações de conexões e interações são feitas por seres humanos; não! Segundo ele, o resultado de pesquisadores mostra um dado interessante: 90% dessas atividades são feitas por *robots*; isto mesmo: entidades não humanas treinadas para se passarem por humanos... Isto nos leva a acreditar na tal “teoria conspiratória” de que os *robots* irão dominar o mundo com Inteligência Artificial Geral. Por outro lado, muito embora já existam *robots* que podem realizar tarefas bem sofisticadas com programas de IA, como vencer qualquer mestre em xadrez, essas entidades não humanas não conseguem realizar tarefas bem simples como por exemplo, se desviar de um obstáculo na superfície de uma sala; coisa que qualquer bebê engatinhando faz com a maior naturalidade; mas isto é assunto para outra discussão...

O que esta postagem quer focalizar é que, mesmo o melhor *robot* já criado por humano e dotado da melhor IA atualmente disponível, segue um padrão bem antigo de computação: *entrada de dados*, *processamento das ordens* e *saída de informação*; sempre se baseando, rigorosamente, nas linhas de código mesmo com inferências e lógicas que analisam um gigante banco de dados embutido e atualizado constantemente. O programa “CriandoRobot”, codificado em Python, embora seja um exemplo bem simples, utiliza um robozinho (a famosa *turtle*) para desenhar alguns tipos de figuras planas na cor desejada pelo usuário. Alguns podem pensar: “*mas eu queria um programa que criasse um robot “de verdade” tal como aparece na TV e na Internet*, *nos filmes de ficção científica*”. Na verdade, o que se quer mostrar é que com uma programação bem sólida e com recursos de uma boa linguagem de programação é possível criar *robots* bem “inteligentes” e independentemente da estrutura física deles, o que pode ser resolvido com uma boa engenharia mecatrônica; mas o fundamental é sempre a programação...

-----------------------------------------------------------------------------------------------------------------------------------



**Figura 1a - Triângulo com lado 200 e na cor verde**



**Figura 1b - Paralelogramo com lado 200 e na cor magenta**

'''

**CriandoRobot.py**

---------------------------------------------------------------------------------------

Simula a criação de um robô com a tartaruga (turtle) para desenhar figuras planas.

---------------------------------------------------------------------------------------

'''

**import** turtle

*#Mapeamento de cores em português para inglês*

LstCores = {

"azul": "blue",

"magenta": "magenta",

"verde": "green",

"vermelho": "red",

"preto": "black"

}

*#Lista de cores*

ACCEPTED\_COLORS = list(LstCores.keys())

*#Tamanho máximo da tela*

SCREEN\_WIDTH = 800

SCREEN\_HEIGHT = 600

MARGIN = 50 *#margem para evitar que as figuras toquem as bordas*

*#-----------------------------------------------------------------------------------------------*

*#Função para verificar se o tamanho da figura cabe na tela*

**def** **VerificarFigura**(largura, altura):

**return** largura + MARGIN <= SCREEN\_WIDTH and altura + MARGIN <= SCREEN\_HEIGHT

*#-----------------------------------------------------------------------------------------------*

*#Função para mover o robô em um triângulo*

**def** **DesenharTriangulo**(comprLado):

**print**(f"Desenhando um triângulo com lado **{comprLado}**")

**for** \_ **in** **range**(3):

robot.forward(comprLado)

robot.right(120)

*#-----------------------------------------------------------------------------------------------*

*#Função para mover o robô em um quadrado*

**def** **DesenharQuadrado**(comprLado):

**print**(f"Desenhando um quadrado com lado **{comprLado}**")

**for** \_ **in** **range**(4):

robot.forward(comprLado)

robot.right(90)

*#-----------------------------------------------------------------------------------------------*

*#Função para mover o robô em um retângulo*

**def** **DesenharRetangulo**(largura, altura):

**print**(f"Desenhando um retângulo com largura **{largura}** e altura **{altura}**")

**for** \_ **in** **range**(2):

robot.forward(largura)

robot.right(90)

robot.forward(altura)

robot.right(90)

*#-----------------------------------------------------------------------------------------------*

*#Função para mover o robô em um paralelogramo em pé*

**def** **DesenharParalelogramo**(comprLado):

**print**(f"Desenhando um paralelogramo com lado **{comprLado}**")

**for** \_ **in** **range**(2):

robot.forward(comprLado)

robot.right(60) *#ângulo para formar o topo e a base do paralelogramo*

robot.forward(comprLado)

robot.right(120) *#completa os lados do paralelogramo*

*#-----------------------------------------------------------------------------------------------*

*#Função principal para obter entrada do usuário e desenhar a figura*

**def** **DesenharFigura**():

*#Solicita ao usuário a forma desejada*

forma = turtle.textinput("Escolha a Forma", "Digite a forma (triângulo, quadrado,

retângulo, paralelogramo):")

**if**(forma **is** **None**): *#verifica se o usuário cancelou a entrada*

**return**

forma = forma.**strip**().**lower**()

*#Solicita a cor do traço*

**while**(**True**):

cor = turtle.textinput("Escolha a Cor", "Digite a cor desejada para o

traço:\n" + ", ".**join**(ACCEPTED\_COLORS))

**if**(cor **is** **None**): *#verifica se o usuário cancelou a entrada*

**return**

cor = cor.strip().**lower**()

**if**(cor **in** ACCEPTED\_COLORS):

corFigura = LstCores[cor]

robot.color(corFigura)

**break**

**else**:

turtle.textinput("Cor Inválida", "Cor não reconhecida; escolha outra cor!)

*#Solicita os lados conforme a forma escolhida*

**if**(forma == 'triângulo'):

comprLado = **int**(

turtle.numinput("Comprimento do Lado", "Digite o comprimento do lado do

triângulo:", minval=1, maxval=500))

**if**(**VerificarFigura**(comprLado, comprLado \* (3 \*\* 0.5) / 2)):

robot.pendown()

**DesenharTriangulo**(comprLado)

**else**:

turtle.textinput("Tamanho Excedido", "A figura não cabe na tela; entre com

valores menores!")

**elif**(forma == 'quadrado'):

comprLado = **int**(

turtle.numinput("Comprimento do Lado", "Digite o comprimento do lado do

quadrado:", minval=1, maxval=500))

**if**(**VerificarFigura**(comprLado, comprLado)):

robot.pendown()

**DesenharQuadrado**(comprLado)

**else**:

turtle.textinput("Tamanho Excedido", "A figura não cabe na tela; entre com

valores menores!")

**elif**(forma == 'retângulo'):

largura = **int**(turtle.numinput("Comprimento da Largura", "Digite a largura do

retângulo:", minval=1, maxval=500))

altura = **int**(turtle.numinput("Comprimento da altura", "Digite a altura do

retângulo:", minval=1, maxval=500))

**if**(VerificarFigura(largura, altura)):

robot.pendown()

**DesenharRetangulo**(largura, altura)

**else**:

turtle.textinput("Tamanho Excedido", "A figura não cabe na tela; entre com

valores menores!")

**elif**(forma == 'paralelogramo'):

comprLado = **int**(

turtle.numinput("Comprimento do Lado", "Digite o comprimento do lado do

paralelogramo:", minval=1, maxval=500))

**if**(VerificarFigura(comprLado, comprLado)):

robot.pendown()

**DesenharParalelogramo**(comprLado)

**else**:

turtle.textinput("Tamanho Excedido", "A figura não cabe na tela; entre com

valores menores!")

**else**:

turtle.textinput("Forma Inválida",

"Forma não reconhecida. Por favor, escolha triângulo, quadrado, retângulo

ou paralelogramo!")

*#-----------------------------------------------------------------------------------------------*

*#Função principal*

**def** main():

**global** robot *#define variável pública*

*#Configura a tela*

screen = turtle.Screen()

screen.title("Robô Desenhista")

screen.setup(width=SCREEN\_WIDTH, height=SCREEN\_HEIGHT) *#tamanho fixo para a tela*

*#Cria o "robô" (tartaruga)*

robot = turtle.Turtle()

robot.shape("turtle") *#define a forma do robô*

robot.speed(1) *#define a velocidade da tartaruga para visualizar o desenho*

robot.pensize(3) *#espessura da linha*

*#Posiciona a tartaruga 100 pixels à esquerda e 100 pixels acima do centro da tela*

robot.penup()

robot.goto(-100, 100)

robot.pendown()

**DesenharFigura**() *#chama a função para desenhar a forma*

*#Espera que o usuário feche a janela para terminar*

screen.mainloop()

*#===============================================================================================*

*#Programa principal #executa a função principal*

**if**(\_\_name\_\_ == "\_\_main\_\_"):

main()

**#Fim do programa "CriandoRobot" ------------------------------------------------------**