

JUEGO ADIVINAR BANDERAS

En esta actividad, crearemos un juego de adivinar banderas. Al comenzar, el jugador tendrá 3 vidas. Cada vez que adivine una bandera, le sumamos una nueva vida y cada vez que falle, le restamos una.

El juego consiste en dibujar una bandera en la pantalla y el usuario deberá adivinar a qué país pertenece.

Parte1: Dibujar las banderas

1. Para dibujar banderas utilizaremos el módulo *turtle*. Los módulos deben importarse a nuestra secuencia de comandos colocando una línea de código al principio del programa:

```
import turtle
```

También importaremos el módulo *random*, que utilizaremos más adelante.

2. Daremos un nombre a nuestra tortuga (*turtle* en inglés). Esto resulta especialmente útil cuando tienes más de un módulo *turtle* en el programa. Yo la llamaré **Bob**.

```
bob = turtle.Turtle() #give name to a turtle
```

NOTA: Algunos comandos básicos del módulo *turtle*:

<code>fd(20)</code>	Move forward 20 turtle steps. Number of steps is defined inside parenthesis().
<code>lt(90)</code>	Change which side the turtle is facing. Turn left for 90 degrees (right angle).
<code>rt(45)</code>	Change which side the turtle is facing. Turn right for 45 degrees.
<code>circle(50)</code>	Draw a circle of radius 50.
<code>pu()</code>	Pen up.
<code>pd()</code>	Pen down.
<code>ht()</code>	Hide turtle icon

Paso2: Dibujar una bandera (p.ej. Alemania)

La bandera alemana consta de tres rectángulos, todos del mismo tamaño. Empecemos con el rectángulo negro.

- Para llenar las formas de color, utilizaremos las instrucciones *begin_fill()* y *end_fill()*.

```
def germany(): #define function which stores drawing of German flag

    bob.fillcolor("black") #fill the turtle with some color
    bob.begin_fill() #begin filling with color
```

Añadiremos *end_fill()* una vez hecho el dibujo.

- Crearemos una bandera de 150 pasos de tortuga de ancho y 90 de alto. Dado que los rectángulos superior, medio e inferior son del mismo tamaño, el tamaño de cada rectángulo puede calcularse mediante $90/3$.

Dibujemos el rectángulo negro.

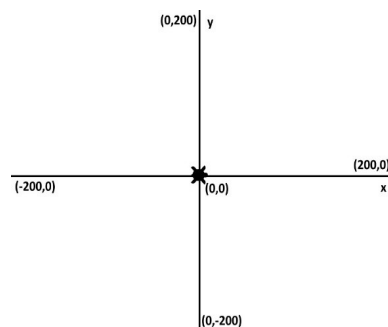
```
def germany(): #define function which stores drawing of German flag

    bob.fillcolor("black") #fill the turtle with some color
    bob.begin_fill() #begin filling with color

    for i in range(2):
        bob.fd(150)
        bob.rt(90)
        bob.fd(30)
        bob.rt(90)
    bob.end_fill() #end filling with color
```

Para dibujar un rectángulo, utilizamos el bucle *for* para repetir el bloque de código dos veces (el número de dentro del paréntesis define las veces que se repetirá el bloque de código). La variable *i* es la variable contador/iterante, que empieza con 0 y se incrementa a cada paso iterativo del bucle.

Turtle siempre empieza en el centro de la pantalla en las coordenadas (0,0).



Para dibujar un rectángulo rojo, utilizaremos *goto()* para desplazarnos hasta las coordenadas (0,-30). Pero si no queremos dibujar algo mientras nos movemos, utilizaremos *pu()* para levantar el lápiz y *pd()* para bajar el lápiz:

```

def germany(): #define function which stores drawing of German flag

    bob.fillcolor("black") #fill the turtle with some color
    bob.begin_fill() #begin filling with color

    for i in range(2):
        bob.fd(150)
        bob.rt(90)
        bob.fd(30)
        bob.rt(90)
    bob.end_fill() #end filling with color

    bob.pu()
    bob.goto(0, -30)
    bob.pd()

```

Utilizaremos el mismo código para dibujar un rectángulo rojo:

```

bob.fillcolor("red")
bob.begin_fill()
for i in range(2):
    bob.fd(150)
    bob.rt(90)
    bob.fd(30)
    bob.rt(90)
bob.end_fill()

```

Ve a las coordenadas (0,-60) para dibujar el rectángulo amarillo:

```

bob.pu()
bob.goto(0, -60)
bob.pd()

bob.fillcolor("yellow")
bob.begin_fill()
for i in range(2):
    bob.fd(150)
    bob.rt(90)
    bob.fd(30)
    bob.rt(90)
bob.end_fill()

```

Y, al final, esconde la tortuga de la pantalla (puedes probar la diferencia entre esconderlo y no esconderlo):

```

bob.ht()

```

Prueba a dibujar algunas banderas más sencillas (Italia, España, Cataluña...)

Parte2: Añadir la lógica del juego

Pistas: Colocar todas las banderas que has definido en una lista y, a continuación, generaremos pintaremos una de ellas aleatoriamente

- Para generarla aleatoriamente puedes usar:
`bandera_se_dibujara=random.choice(flags)`
- Para colocar la tortuga en el centro del eje de coordenadas puedes usar: `bob.reset()`
- Para que dibuje la bandera un poco más rápido, puedes usar: `bob.speed(200)`

Al comenzar el juego, se dibujará una de las bandera escogida aleatoriamente y se preguntará al usuario que bandera es.

Si acierta, lo felicitas y se sumas una vida.

Si falla, se lo dices y le restas una vida.

El juego termina cuando el usuario no quiera seguir jugando o cuando se le terminen las vidas.