Desconocido

### Cómo hacer que un objeto reaccione ante algo

Podemos hacer que el triángulo reaccione cuando alguien pulse una tecla utilizando *eventos* *.* *Los eventos* son cosas que ocurren mientras se ejecuta un programa, como que alguien mueva el ratón, pulse una tecla o cierre una ventana. Puedes decirle a tkinter que esté atento a estos eventos y que haga algo en respuesta.

Para empezar a *manejar* eventos (hacer que Python haga algo cuando se produce un evento), primero creamos una función. La parte de la vinculación se produce cuando le decimos a tkinter que una función concreta está vinculada (o asociada) a un evento específico. En otras palabras, tkinter la llamará automáticamente para gestionar ese evento.

Por ejemplo, para hacer que el triángulo se mueva cuando pulsamos ENTER, podemos definir esta función:

def movetriangle(event):  
 canvas.move(1, 5, 0)

La función toma un único parámetro ( event ), que tkinter utiliza para enviar información a la función sobre el evento. Le decimos a tkinter que esta función debe utilizarse para un evento concreto utilizando la función bind\_all en el lienzo. El código completo tiene ahora este aspecto: introdúcelo en un nuevo archivo en IDLE y guárdalo como *moviendotriángulo.py* antes de ejecutarlo:

from tkinter import \*  
tk = Tk()  
canvas = Canvas(tk, width=400, height=400)  
canvas.pack()  
canvas.create\_polygon(10, 10, 10, 60, 50, 35)  
def movetriangle(event):  
 canvas.move(1, 5, 0)  
canvas.bind\_all('<KeyPress-Return>', movetriangle)

El primer parámetro de esta función describe el evento que queremos que tkinter vigile. En este caso, se llama <KeyPress-Return> , que es una pulsación de la tecla ENTER o RETURN. Le decimos a tkinter que la función movetriangle debe ser llamada siempre que se produzca este evento KeyPress . Ejecuta este código, haz clic en el lienzo con el ratón, y luego prueba a pulsar INTRO con el teclado.



Probemos a cambiar la dirección del triángulo en función de la pulsación de distintas teclas, como las teclas de dirección. Primero tenemos que cambiar la función movetriangle por la siguiente:

def movetriangle(event):  
 if event.keysym == 'Up':  
 canvas.move(1, 0, -3)  
 elif event.keysym == 'Down':  
 canvas.move(1, 0, 3)  
 elif event.keysym == 'Left':  
 canvas.move(1, -3, 0)  
 else:  
 canvas.move(1, 3, 0)

El objeto evento pasado a movetriangle contiene varias variables. Una de estas variables, keysym (por *símbolo de tecla* ), es una cadena que contiene el valor de la tecla real pulsada. La línea if event .keysym == ’Up’ dice que si la variable keysym contiene la cadena ’Up’ , debemos llamar a canvas.move con los parámetros (1, 0, -3), como hacemos en la línea siguiente. Si keysym contiene ’Down’ , como en elif event.keysym == ’Down’ , la llamamos con los parámetros (1, 0, 3), y así sucesivamente.

Recuerda: El primer parámetro es el número identificativo de la forma dibujada en el lienzo, el segundo es el valor a añadir a la coordenada *x (* horizontal), y el tercero es el valor a añadir a la coordenada *y (* vertical).

A continuación, le decimos a tkinter que la función movetriangle se utilizará para manejar eventos de cuatro teclas diferentes (arriba, abajo, izquierda y derecha). A continuación se muestra el aspecto que debería tener ahora el código de *movingtriangle.py* :

from tkinter import \*  
tk = Tk()  
canvas = Canvas(tk, width=400, height=400)  
canvas.pack()  
canvas.create\_polygon(10, 10, 10, 60, 50, 35)  
def movetriangle(event):  
 ➊ if event.keysym == 'Up':  
 ➋ canvas.move(1, 0, -3)  
 ➌ elif event.keysym == 'Down':  
 ➍ canvas.move(1, 0, 3)  
 elif event.keysym == 'Left':  
 canvas.move(1, -3, 0)  
 ➎ else:  
 ➏ canvas.move(1, 3, 0)  
canvas.bind\_all('<KeyPress-Up>', movetriangle)  
canvas.bind\_all('<KeyPress-Down>', movetriangle)  
canvas.bind\_all('<KeyPress-Left>', movetriangle)  
canvas.bind\_all('<KeyPress-Right>', movetriangle)

En la primera línea de la función movetriangle , comprobamos si la variable keysym contiene ’Up’ ➊ . Si es así, movemos el triángulo hacia arriba utilizando la función move con los parámetros 1 , 0 , {3 ➋ . El primer parámetro es el identificador del triángulo, el segundo es la cantidad a desplazar hacia la derecha (no queremos desplazarnos horizontalmente, por lo que el valor es 0), y el tercero es la cantidad a desplazar hacia abajo (-3 píxeles).

A continuación, comprobamos si keysym contiene ’Down’ ➌ ; si es así, movemos el triángulo hacia abajo (3 píxeles) ➍ . La última comprobación es si el valor es ’Left’ ; si es así, movemos el triángulo hacia la izquierda (-3 píxeles). Si no coincide ninguno de los valores, el último else ➎ desplaza el triángulo hacia la derecha ➏ .

Ahora el triángulo debería moverse en la dirección de la tecla de flecha pulsada.

[anterior](ch10_13.html)[Subtema 14 de 17: (Ver todo)](ch10.html)[siguiente](ch10_15.html)