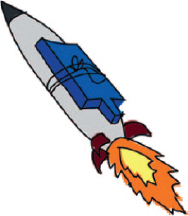
Desconocido

### Cómo mover la paleta

Para hacer que la paleta se mueva a izquierda y derecha, utilizaremos eventos para vincular las teclas de flecha *izquierda* y *derecha* a nuevas funciones de la clase Paddle . Cuando el jugador pulse la tecla de flecha izquierda, la variable x se establecerá en -2 (para moverse a la izquierda). Al pulsar la tecla de flecha derecha, la variable x se establece en 2 (para moverse a la derecha).



El primer paso es añadir la variable de objeto x a la función \_\_init\_\_ de nuestra clase Paddle , así como una variable para la anchura del lienzo, como hicimos con la clase Ball :

def \_\_init\_\_(self, canvas, color):  
 self.canvas = canvas  
 self.id = canvas.create\_rectangle(0, 0, 100, 10, fill=color)   
 self.canvas.move(self.id, 200, 300)  
 ➊ self.x = 0  
 ➋ self.canvas\_width = self.canvas.winfo\_width()

Consulta ➊ y ➋ para ver los cambios. Ahora añadiremos las funciones para cambiar la dirección entre izquierda ( turn\_left ) y derecha ( turn\_right ) justo después de la función draw :

def turn\_left(self, evt):  
 self.x = -2  
  
 def turn\_right(self, evt):  
 self.x = 2

Podemos vincular estas funciones a la tecla correcta en la función \_\_init\_\_ de la clase con estas dos líneas. En "Hacer que un objeto reaccione a algo", en [la página 162](ch10.xhtml#ch10lev1sec13) , utilizamos la vinculación para hacer que Python llame a una función cuando se pulsa una tecla. En este caso, vinculamos la función turn\_left de nuestra clase Paddle a la tecla de la flecha izquierda, utilizando el nombre de evento <KeyPress-Left> . A continuación, vinculamos la función turn\_right a la tecla de flecha derecha, utilizando el nombre de evento <KeyPress-Right> . Nuestra función \_\_init\_\_ tiene ahora este aspecto:

def \_\_init\_\_(self, canvas, color):  
 self.canvas = canvas  
 self.id = canvas.create\_rectangle(0, 0, 100, 10, fill=color)  
 self.canvas.move(self.id, 200, 300)  
 self.x = 0  
 self.canvas\_width = self.canvas.winfo\_width()  
 ➊ self.canvas.bind\_all('<KeyPress-Left>', self.turn\_left)  
 ➋ self.canvas.bind\_all('<KeyPress-Right>', self.turn\_right)

Consulta ➊ y ➋ para ver los cambios. La función draw para la clase Paddle es similar a la de la clase Ball :

def draw(self):  
 self.canvas.move(self.id, self.x, 0)  
 pos = self.canvas.coords(self.id)  
 if pos[0] <= 0 or pos[2] >= self.canvas\_width:  
 self.x = 0

Utilizamos la función move de canvas para mover la paleta en la dirección de la variable x con self.canvas.move(self.id, self.x, 0) . A continuación, obtenemos las coordenadas de la paleta para ver si ha golpeado el lado izquierdo o derecho de la pantalla, utilizando el valor de pos . En lugar de rebotar como la pelota, la pala debe dejar de moverse. Así, cuando la *coordenada x* izquierda ( pos[0] ) es menor o igual que 0 ( <= 0 ), ponemos la variable x a 0 con self.x = 0 . Del mismo modo, cuando la *coordenada* x derecha ( pos[2] ) sea mayor o igual que la anchura del lienzo  ( >= self.canvas\_width ), también ponemos la variable x a 0.

Nota

*Si ejecutas el programa ahora, es posible que tengas que hacer clic en el lienzo antes de que el juego reconozca las acciones de las teclas de flecha izquierda y derecha.* Hacer clic en el lienzo le*da foco, lo que significa que sabe que debe hacerse cargo cuando alguien pulsa una tecla del teclado.*

[anterior](ch12_2.html)[Subtema 3 de 7: (Ver todo)](ch12.html)[siguiente](ch12_4.html)