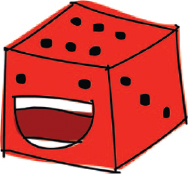
Desconocido

### Añadir un elemento de azar

Ahora es el momento de convertir nuestro programa en un juego y no sólo en una pelota que rebota y una paleta. Los juegos necesitan un elemento *de* azar, o una forma de que el jugador pierda. En nuestro juego actual, la pelota rebotará eternamente, así que no hay nada que perder.



Terminaremos nuestro juego añadiendo código que diga que el juego termina si la pelota toca el fondo del lienzo (en otras palabras, cuando toque el suelo).

En primer lugar, añadimos la variable de objeto hit\_bottom al final de la función \_\_init\_\_ de la clase Ball :

self.canvas\_height = self.canvas.winfo\_height()  
 self.canvas\_width = self.canvas.winfo\_width()  
 self.hit\_bottom = False

A continuación, cambiamos el bucle principal en la parte inferior del programa, de la siguiente manera:

while True:  
 ➊ if ball.hit\_bottom == False:  
 ball.draw()  
 paddle.draw()  
 tk.update\_idletasks()  
 tk.update()  
 time.sleep(0.01)

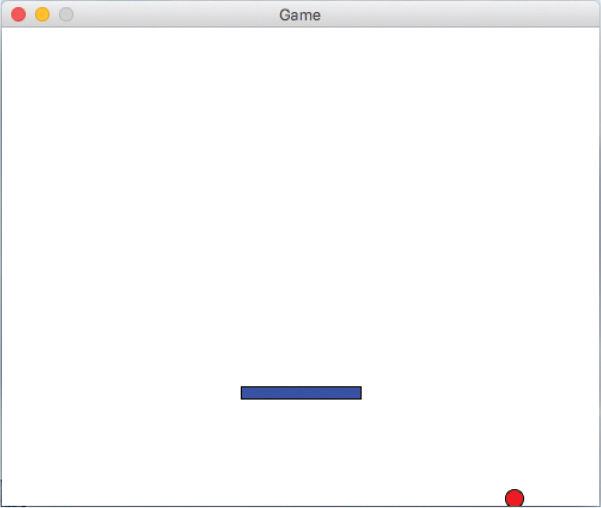
Ahora el bucle sigue comprobando hit\_bottom ➊ para ver si efectivamente la pelota ha tocado el fondo de la pantalla. El código debe seguir moviendo la pelota y la pala sólo si la pelota no ha tocado el fondo, como puedes ver en la declaración if anterior. El juego termina cuando la pelota y la paleta dejan de moverse. (Ya no los animamos.)

El último cambio es en la función draw de la clase Ball :

def draw(self):  
 self.canvas.move(self.id, self.x, self.y)  
 pos = self.canvas.coords(self.id)  
 if pos[1] <= 0:  
 self.y = self.y \* -1  
 ➊ if pos[3] >= self.canvas\_height:  
 self.hit\_bottom = True  
 if self.hit\_paddle(pos) == True:  
 self.y = self.y \* -1  
 if pos[0] <= 0 or pos[2] >= self.canvas\_width:  
 self.y = self.y \* -1

Modificamos la sentencia if para ver si la pelota ha tocado el fondo de la pantalla (es decir, si la posición de la pelota es mayor o igual que canvas\_height ) ➊ . Si es así, en la línea siguiente, establecemos hit\_bottom en True , en lugar de cambiar el valor de la variable y , porque no hay necesidad de hacer rebotar la pelota una vez que toca el fondo de la pantalla.

Cuando ejecutes el juego ahora y falles al golpear la pelota con la pala, todo el movimiento de la pantalla debería detenerse. El juego debería terminar una vez que la pelota toque el fondo del lienzo, como se muestra en la [Figura 12-4 .](ch12.xhtml#ch12fig04)



*Figura 12-4:* La pelota toca*el* fondo de la pantalla

Tu programa debería parecerse ahora al siguiente código. Si tienes problemas para que tu juego funcione, comprueba lo que has introducido con este código:

from tkinter import \*  
import random  
import time  
  
class Ball:  
 def \_\_init\_\_(self, canvas, paddle, color):  
 self.canvas = canvas  
 self.paddle = paddle  
 self.id = canvas.create\_oval(10, 10, 25, 25, fill=color)  
 self.canvas.move(self.id, 245, 100)  
 starts = [-3, -2, -1, 1, 2, 3]  
 self.x = random.choice(starts)  
 self.y = -3  
 self.canvas\_height = self.canvas.winfo\_height()  
 self.canvas\_width = self.canvas.winfo\_width()  
 self.hit\_bottom = False  
   
 def hit\_paddle(self, pos):  
 paddle\_pos = self.canvas.coords(self.paddle.id)  
 if pos[2] >= paddle\_pos[0] and pos[0] <= paddle\_pos[2]:  
 if pos[3] >= paddle\_pos[1] and pos[3] <= paddle\_pos[3]:  
 return True  
 return False  
   
 def draw(self):  
 self.canvas.move(self.id, self.x, self.y)  
 pos = self.canvas.coords(self.id)  
 if pos[1] <= 0:  
 self.y = self.y \* -1  
 if pos[3] >= self.canvas\_height:  
 self.hit\_bottom = True  
 if self.hit\_paddle(pos) == True:  
 self.y = self.y \* -1  
 if pos[0] <= 0 or pos[2] >= self.canvas\_width:  
 self.y = self.y \* -1  
  
class Paddle:  
 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()  
 self.canvas.bind\_all('<KeyPress-Left>', self.turn\_left)  
 self.canvas.bind\_all('<KeyPress-Right>', self.turn\_right)  
  
 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  
   
 def turn\_left(self, evt):  
 self.x = -2  
  
 def turn\_right(self, evt):  
 self.x = 2  
  
tk = Tk()  
tk.title('Bounce Game')  
tk.resizable(0, 0)  
tk.wm\_attributes('-topmost', 1)  
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)  
canvas.pack()  
tk.update()  
  
paddle = Paddle(canvas, 'blue')  
ball = Ball(canvas, paddle, 'red')  
  
while True:  
 if ball.hit\_bottom == False:  
 ball.draw()  
 paddle.draw()  
 tk.update\_idletasks()  
 tk.update()  
 time.sleep(0.01)

[anterior](ch12_4.html)[Subtema 5 de 7: (Ver todo)](ch12.html)[siguiente](ch12_6.html)