Desconocido

### El juego final

El listado completo de nuestro juego tiene ahora algo más de 200 líneas de código. A continuación se muestra el código completo del juego. Si tienes problemas para que tu juego funcione, compara cada función (y cada clase) con este listado:

from tkinter import \*  
import random  
import time  
  
class Coords:  
 def \_\_init\_\_(self, x1=0, y1=0, x2=0, y2=0):  
 self.x1 = x1  
 self.y1 = y1  
 self.x2 = x2  
 self.y2 = y2  
  
def within\_x(co1, co2):  
 if (co1.x1 > co2.x1 and co1.x1 < co2.x2) \  
 or (co1.x2 > co2.x1 and co1.x2 < co2.x2) \  
 or (co2.x1 > co1.x1 and co2.x1 < co1.x2) \  
 or (co2.x2 > co1.x1 and co2.x2 < co1.x2):  
 return True  
 else:  
 return False  
  
def within\_y(co1, co2):  
 if (co1.y1 > co2.y1 and co1.y1 < co2.y2) \  
 or (co1.y2 > co2.y1 and co1.y2 < co2.y2) \  
 or (co2.y1 > co1.y1 and co2.y1 < co1.y2) \  
 or (co2.y2 > co1.y1 and co2.y2 < co1.y2):  
 return True  
 else:  
 return False  
  
def collided\_left(co1, co2):  
 if within\_y(co1, co2):  
 if co1.x1 >= co2.x1 and co1.x1 <= co2.x2:  
 return True  
 return False  
  
def collided\_right(co1, co2):  
 if within\_y(co1, co2):  
 if co1.x2 >= co2.x1 and co1.x2 <= co2.x2:  
 return True  
 return False  
  
def collided\_top(co1, co2):  
 if within\_x(co1, co2):  
 if co1.y1 >= co2.y1 and co1.y1 <= co2.y2:  
 return True  
 return False  
  
def collided\_bottom(y, co1, co2):  
 if within\_x(co1, co2):  
 y\_calc = co1.y2 + y  
 if y\_calc >= co2.y1 and y\_calc <= co2.y2:  
 return True  
 return False  
  
class Sprite:  
 def \_\_init\_\_(self, game):  
 self.game = game  
 self.endgame = False  
 self.coordinates = None  
 def move(self):  
 pass  
 def coords(self):  
 return self.coordinates  
  
class PlatformSprite(Sprite):  
 def \_\_init\_\_(self, game, photo\_image, x, y, width, height):  
 Sprite.\_\_init\_\_(self, game)  
 self.photo\_image = photo\_image  
 self.image = game.canvas.create\_image(x, y,   
 image=self.photo\_image, anchor='nw')  
 self.coordinates = Coords(x, y, x + width, y + height)  
  
class StickFigureSprite(Sprite):  
 def \_\_init\_\_(self, game):  
 Sprite.\_\_init\_\_(self, game)  
 self.images\_left = [  
 PhotoImage(file='figure-L1.gif'),  
 PhotoImage(file='figure-L2.gif'),  
 PhotoImage(file='figure-L3.gif')  
 ]  
 self.images\_right = [  
 PhotoImage(file='figure-R1.gif'),  
 PhotoImage(file='figure-R2.gif'),  
 PhotoImage(file='figure-R3.gif')  
 ]  
 self.image = game.canvas.create\_image(200, 470,   
 image=self.images\_left[0], anchor='nw')  
 self.x = -2  
 self.y = 0  
 self.current\_image = 0  
 self.current\_image\_add = 1  
 self.jump\_count = 0  
 self.last\_time = time.time()  
 self.coordinates = Coords()  
 game.canvas.bind\_all('<KeyPress-Left>', self.turn\_left)  
 game.canvas.bind\_all('<KeyPress-Right>', self.turn\_right)  
 game.canvas.bind\_all('<space>', self.jump)  
  
 def turn\_left(self, evt):  
 if self.y == 0:  
 self.x = -2  
  
 def turn\_right(self, evt):  
 if self.y == 0:  
 self.x = 2  
  
 def jump(self, evt):  
 if self.y == 0:  
 self.y = -4  
 self.jump\_count = 0  
  
 def animate(self):  
 if self.x != 0 and self.y == 0:  
 if time.time() - self.last\_time > 0.1:  
 self.last\_time = time.time()  
 self.current\_image += self.current\_image\_add  
 if self.current\_image >= 2:  
 self.current\_image\_add = -1  
 if self.current\_image <= 0:  
 self.current\_image\_add = 1  
 if self.x < 0:  
 if self.y != 0:  
 self.game.canvas.itemconfig(self.image,   
 image=self.images\_left[2])  
 else:  
 self.game.canvas.itemconfig(self.image,   
 image=self.images\_left[self.current\_image])  
 elif self.x > 0:  
 if self.y != 0:  
 self.game.canvas.itemconfig(self.image,   
 image=self.images\_right[2])  
 else:  
 self.game.canvas.itemconfig(self.image,   
 image=self.images\_right[self.current\_image])  
  
 def coords(self):  
 xy = self.game.canvas.coords(self.image)  
 self.coordinates.x1 = xy[0]  
 self.coordinates.y1 = xy[1]  
 self.coordinates.x2 = xy[0] + 27  
 self.coordinates.y2 = xy[1] + 30  
 return self.coordinates  
  
 def move(self):  
 self.animate()  
 if self.y < 0:  
 self.jump\_count += 1  
 if self.jump\_count > 20:  
 self.y = 4  
 if self.y > 0:  
 self.jump\_count -= 1  
 co = self.coords()  
 left = True  
 right = True  
 top = True  
 bottom = True  
 falling = True  
 if self.y > 0 and co.y2 >= self.game.canvas\_height:  
 self.y = 0  
 bottom = False  
 elif self.y < 0 and co.y1 <= 0:  
 self.y = 0  
 top = False  
 if self.x > 0 and co.x2 >= self.game.canvas\_width:  
 self.x = 0  
 right = False  
 elif self.x < 0 and co.x1 <= 0:  
 self.x = 0  
 left = False  
  
 for sprite in self.game.sprites:  
 if sprite == self:  
 continue  
 sprite\_co = sprite.coords()  
 if top and self.y < 0 and collided\_top(co, sprite\_co):  
 self.y = -self.y  
 top = False  
 if bottom and self.y > 0 and collided\_bottom(self.y,   
 co, sprite\_co):  
 self.y = sprite\_co.y1 - co.y2  
 if self.y < 0:  
 self.y = 0  
 bottom = False  
 top = False  
 if bottom and falling and self.y == 0 \  
 and co.y2 < self.game.canvas\_height \  
 and collided\_bottom(1, co, sprite\_co):  
 falling = False  
 if left and self.x < 0 and collided\_left(co, sprite\_co):  
 self.x = 0  
 left = False  
 if sprite.endgame:  
 self.game.running = False  
 if right and self.x > 0 and collided\_right(co, sprite\_co):  
 self.x = 0  
 right = False  
 if sprite.endgame:  
 self.game.running = False  
  
 if falling and bottom and self.y == 0 \  
 and co.y2 < self.game.canvas\_height:  
 self.y = 4  
 self.game.canvas.move(self.image, self.x, self.y)  
  
class DoorSprite(Sprite):  
 def \_\_init\_\_(self, game, photo\_image, x, y, width, height):  
 Sprite.\_\_init\_\_(self, game)  
 self.photo\_image = photo\_image  
 self.image = game.canvas.create\_image(x, y,   
 image=self.photo\_image, anchor='nw')  
 self.coordinates = Coords(x, y, x + (width / 2), y + height)  
 self.endgame = True  
  
class Game:  
 def \_\_init\_\_(self):  
 self.tk = Tk()  
 self.tk.title('Mr. Stick Man Races for the Exit')  
 self.tk.resizable(0, 0)  
 self.tk.wm\_attributes('-topmost', 1)  
 self.canvas = Canvas(self.tk, width=500, height=500,   
 highlightthickness=0)  
 self.canvas.pack()  
 self.tk.update()  
 self.canvas\_height = self.canvas.winfo\_height()  
 self.canvas\_width = self.canvas.winfo\_width()  
 self.bg = PhotoImage(file='background.gif')  
 w = self.bg.width()  
 h = self.bg.height()  
 for x in range(0, 5):  
 for y in range(0, 5):  
 self.canvas.create\_image(x \* w, y \* h,   
 image=self.bg, anchor='nw')  
 self.sprites = []  
 self.running = True  
  
 def mainloop(self):  
 while True:  
 if self.running == True:  
 for sprite in self.sprites:  
 sprite.move()  
 self.tk.update\_idletasks()  
 self.tk.update()  
 time.sleep(0.01)  
  
g = Game()  
platform1 = PlatformSprite(g, PhotoImage(file='platform1.gif'),   
 0, 480, 100, 10)  
platform2 = PlatformSprite(g, PhotoImage(file='platform1.gif'),   
 150, 440, 100, 10)  
platform3 = PlatformSprite(g, PhotoImage(file='platform1.gif'),   
 300, 400, 100, 10)  
platform4 = PlatformSprite(g, PhotoImage(file='platform1.gif'),   
 300, 160, 100, 10)  
platform5 = PlatformSprite(g, PhotoImage(file='platform2.gif'),   
 175, 350, 66, 10)  
platform6 = PlatformSprite(g, PhotoImage(file='platform2.gif'),   
 50, 300, 66, 10)  
platform7 = PlatformSprite(g, PhotoImage(file='platform2.gif'),   
 170, 120, 66, 10)  
platform8 = PlatformSprite(g, PhotoImage(file='platform2.gif'),   
 45, 60, 66, 10)  
platform9 = PlatformSprite(g, PhotoImage(file='platform3.gif'),   
 170, 250, 32, 10)  
platform10 = PlatformSprite(g, PhotoImage(file='platform3.gif'),   
 230, 200, 32, 10)  
g.sprites.append(platform1)  
g.sprites.append(platform2)  
g.sprites.append(platform3)  
g.sprites.append(platform4)  
g.sprites.append(platform5)  
g.sprites.append(platform6)  
g.sprites.append(platform7)  
g.sprites.append(platform8)  
g.sprites.append(platform9)  
g.sprites.append(platform10)  
door = DoorSprite(g, PhotoImage(file='door1.gif'), 45, 30, 40, 35)  
g.sprites.append(door)  
sf = StickFigureSprite(g)  
g.sprites.append(sf)  
g.mainloop()

[anterior](ch16_10.html)[Subtema 11 de 13: (Ver todo)](ch16.html)[siguiente](ch16_12.html)