Desconocido

### Creación de la clase Sprite

La clase padre de nuestros elementos de juego, Sprite , proporcionará dos funciones: move para mover el sprite, y coords para devolver la posición actual del sprite en la pantalla. Añadimos el código de la clase Sprite debajo de la función collided\_bottom , como sigue:

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

La función \_\_init\_\_ de la clase Sprite toma un único parámetro, game , que será el objeto del juego. Lo necesitamos para que cualquier sprite que creemos pueda acceder a la lista de otros sprites del juego. Almacenamos el parámetro del juego como una variable de objeto.

A continuación, almacenamos la variable de objeto endgame , que utilizaremos para indicar el final del juego. (En este momento, se establece en False .) La última variable de objeto, coordinates , se establece en nada ( None ).

La función move no hace nada en esta clase padre, así que utilizamos la palabra clave pass en el cuerpo de esta función. La función coords simplemente devuelve la variable de objeto coordinates .

Así que nuestra clase Sprite tiene una función move que no hace nada y una función coords que no devuelve ninguna coordenada. No parece muy útil, ¿verdad? Sin embargo, cualquier clase que tenga como padre a Sprite siempre tendrá las funciones move y coords . Así, en el bucle principal del juego, cuando recorramos una lista de sprites, llamar a la función move no provocará ningún error porque cada sprite tiene esa función.



Nota

*Las clases con funciones que no hacen gran cosa son bastante comunes en programación.* En*cierto modo, son un acuerdo que garantiza que todos los hijos de una clase proporcionan el mismo tipo de funcionalidad, aunque en algunos casos las funciones de las clases hijas no hagan nada.*

[anterior](ch14_10.html)[Subtema 11 de 16: (Ver todo)](ch14.html)[siguiente](ch14_12.html)