### Detección de colisiones entre naves alienígenas

Empezaremos comprobando si hay colisiones entre los alienígenas y la nave, para poder responder adecuadamente cuando un alienígena choque contra ella. Comprobaremos las colisiones alienígena-nave inmediatamente después de actualizar la posición de cada alienígena en AlienInvasion:

**alien\_invasion.py**

def \_update\_aliens(self):  
 --snip--  
 self.aliens.update()  
  
 # Look for alien-ship collisions.  
❶ if pygame.sprite.spritecollideany(self.ship, self.aliens):  
❷ print("Ship hit!!!")

La función spritecollideany() toma dos argumentos: un sprite y un grupo. La función busca cualquier miembro del grupo que haya colisionado con el sprite y deja de recorrer el grupo en cuanto encuentra un miembro que haya colisionado con el sprite. En este caso, recorre el grupo aliens y devuelve el primer extraterrestre que encuentre que haya colisionado con ship.

Si no se produce ninguna colisión, spritecollideany() devuelve None y el bloque if no se ejecuta ❶. Si encuentra un alienígena que ha colisionado con la nave, devuelve ese alienígena y el bloque if se ejecuta: imprime Ship hit!!! ❷. Cuando un alienígena choque con la nave, tendremos que realizar una serie de tareas: eliminar todos los alienígenas y balas restantes, recentrar la nave y crear una nueva flota. Antes de escribir código para hacer todo esto, queremos saber si nuestro enfoque para detectar colisiones alienígena-nave funciona correctamente. Escribir una llamada a print() es una forma sencilla de asegurarnos de que detectamos estas colisiones correctamente.

Ahora, cuando ejecutes *Alien Invasion,*, debería aparecer el mensaje *Ship hit!!!* en el terminal cada vez que un alienígena choque contra la nave. Cuando estés probando esta función, ajusta fleet\_drop\_speed a un valor más alto, como 50 o 100, para que los alienígenas alcancen tu nave más rápidamente.

[anterior](c13_25.html)[Subtema 26 de 33: (Ver todo)](c13.html)[siguiente](c13_27.html)