### Visualización del nivel

Para mostrar el nivel del jugador en el juego, primero necesitamos un atributo en GameStats que represente el nivel actual. Para restablecer el nivel al comienzo de cada nueva partida, inicialízalo en reset\_stats():

**game\_stats.py**

def reset\_stats(self):  
 """Initialize statistics that can change during the game."""  
 self.ships\_left = self.settings.ship\_limit  
 self.score = 0  
 self.level = 1

Para que Scoreboard muestre el nivel actual, llamamos a un nuevo método, prep\_level(), desde \_\_init\_\_():

**scoreboard.py**

def \_\_init\_\_(self, ai\_game):  
 --snip--  
 self.prep\_high\_score()  
 self.prep\_level()

Aquí tienes prep\_level():

**scoreboard.py**

def prep\_level(self):  
 """Turn the level into a rendered image."""  
 level\_str = str(self.stats.level)  
❶ self.level\_image = self.font.render(level\_str, True,  
 self.text\_color, self.settings.bg\_color)  
  
 # Position the level below the score.  
 self.level\_rect = self.level\_image.get\_rect()  
❷ self.level\_rect.right = self.score\_rect.right  
❸ self.level\_rect.top = self.score\_rect.bottom + 10

El método prep\_level() crea una imagen a partir del valor almacenado en stats.level ❶ y establece el atributo right de la imagen para que coincida con el atributo right de la puntuación ❷. A continuación, establece el atributo top 10 píxeles por debajo de la parte inferior de la imagen de la puntuación para dejar espacio entre la puntuación y el nivel ❸.

También tenemos que actualizar show\_score():

**scoreboard.py**

def show\_score(self):  
 """Draw scores and level to the screen."""  
 self.screen.blit(self.score\_image, self.score\_rect)  
 self.screen.blit(self.high\_score\_image, self.high\_score\_rect)  
 self.screen.blit(self.level\_image, self.level\_rect)

Esta nueva línea dibuja la imagen del nivel en la pantalla.

Incrementaremos stats.level y actualizaremos la imagen de nivel en \_check\_bullet\_alien\_collisions():

**alien\_invasion.py**

def \_check\_bullet\_alien\_collisions(self):  
 --snip--  
 if not self.aliens:  
 # Destroy existing bullets and create new fleet.  
 self.bullets.empty()  
 self.\_create\_fleet()  
 self.settings.increase\_speed()  
  
 # Increase level.  
 self.stats.level += 1  
 self.sb.prep\_level()

Si se destruye una flota, incrementamos el valor de stats.level y llamamos a prep\_level() para asegurarnos de que el nuevo nivel se muestra correctamente.

Para asegurarnos de que la imagen del nivel se actualiza correctamente al inicio de una nueva partida, también llamamos a prep\_level() cuando el jugador pulsa el botón Jugar:

**alien\_invasion.py**

def \_check\_play\_button(self, mouse\_pos):  
 --snip--  
 if button\_clicked and not self.game\_active:  
 --snip--  
 self.sb.prep\_score()  
 self.sb.prep\_level()  
 --snip--

Llamamos a prep\_level() justo después de llamar a prep\_score().

Ahora verás cuántos niveles has completado, como se muestra en la [Figura 14-5](#figure14-5).

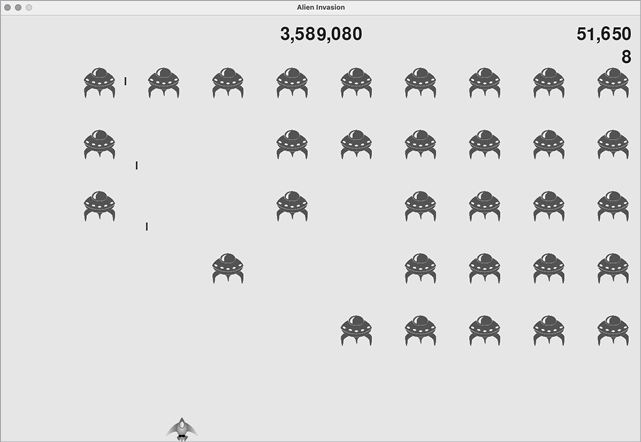


Figura14-5: El nivel actual aparece justo debajo de la puntuación actual.

[anterior](c14_23.html)[Subtema 24 de 29: (Ver todo)](c14.html)[siguiente](c14_25.html)