

14.11-Millora

```
import random
import pygame
from pygame.locals import QUIT

# Constantes para la inicialización de la superficie de dibujo
VENTANA_HORI = 800 # Ancho de la ventana
VENTANA_VERT = 600 # Alto de la ventana
FPS = 60 # Fotogramas por segundo
BLANCO = (255, 255, 255) # Color del fondo de la ventana (RGB)

class PelotaPong:
    def __init__(self, fichero_imagen):
        # --- Atributos de la Clase ---

        # Imagen de la Pelota
        self.imagen = pygame.image.load(fichero_imagen).convert_alpha()

        # Dimensiones de la Pelota
        self.ancho, self.alto = self.imagen.get_size()

        # Posición de la Pelota
        self.x = VENTANA_HORI / 2 - self.ancho / 2
        self.y = VENTANA_VERT / 2 - self.alto / 2

        # Dirección de movimiento de la Pelota
        self.dir_x = random.choice([-5, 5])
        self.dir_y = random.choice([-5, 5])

    def mover(self):
        self.x += self.dir_x
        self.y += self.dir_y

    def rebotar(self):
        if self.x <= -self.ancho:
            self.reiniciar()
        if self.x >= VENTANA_HORI:
            self.reiniciar()
        if self.y <= 0:
            self.dir_y = -self.dir_y
        if self.y + self.alto >= VENTANA_VERT:
            self.dir_y = -self.dir_y
```

def reiniciar(self):

```
self.x = VENTANA_HORI / 2 - self.ancho / 2
self.y = VENTANA_VERT / 2 - self.alto / 2
self.dir_x = -self.dir_x
self.dir_y = random.choice([-5, 5])
```

class RaquetaPong:**def __init__(self):**

```
self.imagen = pygame.image.load("raqueta.png").convert_alpha()
```

```
# --- Atributos de la Clase ---
```

```
# Dimensiones de la Raqueta
```

```
self.ancho, self.alto = self.imagen.get_size()
```

```
# Posición de la Raqueta
```

```
self.x = 0
```

```
self.y = VENTANA_VERT / 2 - self.alto / 2
```

```
# Dirección de movimiento de la Raqueta
```

```
self.dir_y = 0
```

def mover(self):

```
self.y += self.dir_y
```

```
if self.y <= 0:
```

```
    self.y = 0
```

```
if self.y + self.alto >= VENTANA_VERT:
```

```
    self.y = VENTANA_VERT - self.alto
```

def mover_ia(self, pelota):

```
if self.y > pelota.y:
```

```
    self.dir_y = -3
```

```
elif self.y < pelota.y:
```

```
    self.dir_y = 3
```

```
else:
```

```
    self.dir_y = 0
```

```
self.y += self.dir_y
```

def golpear(self, pelota):

```
if (
    pelota.x < self.x + self.ancho
    and pelota.x > self.x
    and pelota.y + pelota.alto > self.y
    and pelota.y < self.y + self.alto
):
    pelota.dir_x = -pelota.dir_x
    pelota.x = self.x + self.ancho
```

def golpear_ia(self, pelota):

```
if (
    pelota.x + pelota.ancho > self.x
    and pelota.x < self.x + self.ancho
    and pelota.y + pelota.alto > self.y
    and pelota.y < self.y + self.alto
):
    pelota.dir_x = -pelota.dir_x
    pelota.x = self.x - pelota.ancho
```

def main():

```
# Inicialización de Pygame
```

```
pygame.init()
```

```
# Inicialización de la superficie de dibujo (display surface)
```

```
ventana = pygame.display.set_mode((VENTANA_HORI, VENTANA_VERT))
```

```
pygame.display.set_caption("Pong 8")
```

```
pelota = PelotaPong("bola_roja.png")
```

```
raqueta_1 = RaquetaPong()
```

```
raqueta_1.x = 60
```

```
raqueta_2 = RaquetaPong()
```

```
raqueta_2.x = VENTANA_HORI - 60 - raqueta_2.ancho
```

Bucle principal

```
jugando = True
while jugando:
    pelota.mover()
    pelota.rebotar()
    raqueta_1.mover()
    raqueta_2.mover_ia(pelota)
    raqueta_1.golpear(pelota)
    raqueta_2.golpear_ia(pelota)

    ventana.fill(BLANCO)
    ventana.blit(pelota.imagen, (pelota.x, pelota.y))
    ventana.blit(raqueta_1.imagen, (raqueta_1.x, raqueta_1.y))
    ventana.blit(raqueta_2.imagen, (raqueta_2.x, raqueta_2.y))

for event in pygame.event.get():
    if event.type == QUIT:
        jugando = False

# Detecta que se ha pulsado una tecla
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_w:
        raqueta_1.dir_y = -5
    if event.key == pygame.K_s:
        raqueta_1.dir_y = 5

# Detecta que se ha soltado la tecla
if event.type == pygame.KEYUP:
    if event.key == pygame.K_w:
        raqueta_1.dir_y = 0
    if event.key == pygame.K_s:
        raqueta_1.dir_y = 0

pygame.display.flip()
pygame.time.Clock().tick(FPS)

pygame.quit()

if __name__ == "__main__":
    main()
```

Pràctica:

Dissenya un programa diferent, enunciat inclòs, que combini els exemples anteriors, prova'l, mostra'l i enganxa'l aquí.

