

# ΑΣΚΗΣΗ PYGAME 2

## Mini Arkanoid - Μπάλα και Ρακέτα

<b>Μάθημα:</b>	Προγραμματισμός Εφαρμογών Python - PyGame
<b>Διάρκεια:</b>	45 λεπτά (1 διδακτική ώρα)
<b>Επίπεδο:</b>	Αρχάριο - Μεσαίο
<b>Απαιτούμενα:</b>	Python 3.x, PyGame, IDLE ή Thonny
<b>Προαπαιτούμενα:</b>	Ολοκλήρωση Άσκησης 1 (Bouncing Ball)

### Μαθησιακοί Στόχοι

Μετά την ολοκλήρωση αυτής της άσκησης θα μπορείτε να:

- Δημιουργείτε αντικείμενα που ελέγχονται από τον χρήστη (ρακέτα)
- Υλοποιείτε ανίχνευση σύγκρουσης (collision detection) μεταξύ αντικειμένων
- Διαχειρίζεστε σκορ και κατάσταση παιχνιδιού (game state)
- Εμφανίζετε κείμενο στην οθόνη με τη χρήση fonts
- Υλοποιείτε συνθήκη τερματισμού παιχνιδιού (game over)

### Εκφώνηση Άσκησης

Δημιουργήστε ένα απλοποιημένο παιχνίδι τύπου Arkanoid. Μια μπάλα κινείται στην οθόνη και αναπηδά στα πάνω, αριστερά και δεξιά όρια. Στο κάτω μέρος της οθόνης υπάρχει μια ρακέτα που ελέγχει ο χρήστης με τα βέλη (αριστερά/δεξιά). Αν η μπάλα χτυπήσει τη ρακέτα, αναπηδά προς τα πάνω και ο παίκτης κερδίζει 5 πόντους. Αν η μπάλα περάσει κάτω από τη ρακέτα, το παιχνίδι τελειώνει.

### Προδιαγραφές

Στοιχείο	Τιμή
Μέγεθος παραθύρου	700 x 500 pixels
Χρώμα φόντου	Μαύρο (0, 0, 0)
Χρώμα μπάλας	Λευκό (255, 255, 255)
Ακτίνα μπάλας	10 pixels
Αρχική θέση μπάλας	x=350, y=250 (κέντρο)
Ταχύτητα μπάλας	4 pixels/frame
Χρώμα ρακέτας	Μπλε (0, 0, 255)
Μέγεθος ρακέτας	100 x 15 pixels
Θέση ρακέτας (Y)	480 pixels (κοντά στο κάτω όριο)
Ταχύτητα ρακέτας	7 pixels/frame
Πόντοι ανά χτύπημα	5
FPS	60 καρτέ ανά δευτερόλεπτο

## Βήματα Υλοποίησης

### Βήμα 1: Βασική Δομή και Αρχικοποίηση

Δημιουργήστε ένα νέο αρχείο `mini_arkanoid.py` με τη βασική δομή PyGame και τις μεταβλητές:

```
import pygame

# Αρχικοποίηση PyGame
pygame.init()

# Ορισμός χρωμάτων
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)

# Δημιουργία παραθύρου
size = (700, 500)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Mini Arkanoid")

# Μεταβλητές ελέγχου
done = False
clock = pygame.time.Clock()

# Αρχικοποίηση font για το σκορ
font = pygame.font.SysFont('Arial', 30)
```

### Βήμα 2: Μεταβλητές Μπάλας και Ρακέτας

Προσθέστε τις μεταβλητές για τη μπάλα, τη ρακέτα και το σκορ:

```
# --- Μεταβλητές μπάλας ---
ball_x = 350
ball_y = 250
ball_radius = 10
ball_speed_x = 4
ball_speed_y = 4

# --- Μεταβλητές ρακέτας ---
paddle_width = 100
paddle_height = 15
paddle_x = 300      # Αρχική θέση X (κέντρο)
paddle_y = 480     # Θέση Y (κοντά στο κάτω μέρος)
paddle_speed = 7

# --- Μεταβλητές παιχνιδιού ---
score = 0
game_over = False
```

**Συμβουλή:** Η ρακέτα είναι ένα ορθογώνιο, οπότε χρειαζόμαστε πλάτος και ύψος αντί για ακτίνα.

### Βήμα 3: Έλεγχος Ρακέτας με Πληκτρολόγιο

Μέσα στην κύρια επανάληψη, προσθέστε τον έλεγχο κίνησης της ρακέτας:

```
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True

    if not game_over:
        # --- Έλεγχος πληκτρολογίου για τη ρακέτα ---
        keys = pygame.key.get_pressed()
        if keys[pygame.K_LEFT] and paddle_x > 0:
            paddle_x -= paddle_speed
        if keys[pygame.K_RIGHT] and paddle_x < 700 - paddle_width:
            paddle_x += paddle_speed
```

**Προσοχή:** Ελέγχουμε τα όρια ( $paddle\_x > 0$  και  $paddle\_x < 700 - paddle\_width$ ) ώστε η ρακέτα να μην βγαίνει εκτός οθόνης!

### Βήμα 4: Κίνηση Μπάλας και Αναπήδηση στα Όρια

Προσθέστε τη λογική κίνησης της μπάλας και αναπήδησης στα πάνω, αριστερά και δεξιά όρια:

```
# --- Κίνηση μπάλας ---
ball_x += ball_speed_x
ball_y += ball_speed_y

# Αναπήδηση στο πάνω όριο
if ball_y <= ball_radius:
    ball_speed_y *= -1

# Αναπήδηση στα πλαϊνά όρια
if ball_x <= ball_radius or ball_x >= 700 - ball_radius:
    ball_speed_x *= -1
```

## Βήμα 5: Ανίχνευση Σύγκρουσης με τη Ρακέτα

Προσθέστε τον έλεγχο σύγκρουσης μεταξύ μπάλας και ρακέτας:

```
# --- Έλεγχος σύγκρουσης με τη ρακέτα ---
if ball_y >= paddle_y - ball_radius:
    if paddle_x <= ball_x <= paddle_x + paddle_width:
        ball_speed_y *= -1
        ball_y = paddle_y - ball_radius # Αποφυγή κόλλησης
        score += 5
```

**Εξήγηση:** Ελέγχουμε αν η μπάλα έφτασε στο ύψος της ρακέτας ΚΑΙ αν βρίσκεται οριζόντια μέσα στο πλάτος της ρακέτας.

## Βήμα 6: Έλεγχος Game Over

Προσθέστε τον έλεγχο για τερματισμό του παιχνιδιού όταν η μπάλα φύγει από κάτω:

```
# --- Έλεγχος αν η μπάλα έφυγε κάτω (Game Over) ---
if ball_y > 500:
    game_over = True
```

## Βήμα 7: Σχεδίαση Αντικειμένων και Σκορ

Προσθέστε τη σχεδίαση της μπάλας, της ρακέτας και του σκορ:

```
# --- Σχεδίαση ---
screen.fill(BLACK)

# Σχεδίαση μπάλας
pygame.draw.circle(screen, WHITE, [int(ball_x), int(ball_y)], ball_radius)

# Σχεδίαση ρακέτας
pygame.draw.rect(screen, BLUE, [paddle_x, paddle_y, paddle_width, paddle_height])

# Εμφάνιση σκορ
score_text = font.render("Score: " + str(score), True, WHITE)
screen.blit(score_text, [10, 10])

# Εμφάνιση Game Over
if game_over:
    game_over_text = font.render("GAME OVER! Score: " + str(score), True, WHITE)
    screen.blit(game_over_text, [220, 250])

pygame.display.flip()
clock.tick(60)
```

```
pygame.quit()
```

## Ολοκληρωμένος Κώδικας

Παρακάτω βλέπετε τον ολοκληρωμένο κώδικα της άσκησης:

```
import pygame

pygame.init()

# Χρώματα
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)

# Παράθυρο
size = (700, 500)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Mini Arkanoid")

done = False
clock = pygame.time.Clock()
font = pygame.font.SysFont('Arial', 30)

# Μπάλα
ball_x = 350
ball_y = 250
ball_radius = 10
ball_speed_x = 4
ball_speed_y = 4

# Ρακέτα
paddle_width = 100
paddle_height = 15
paddle_x = 300
paddle_y = 480
paddle_speed = 7

# Παιχνίδι
score = 0
game_over = False

while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True

    if not game_over:
        # Κίνηση ρακέτας
        keys = pygame.key.get_pressed()
        if keys[pygame.K_LEFT] and paddle_x > 0:
            paddle_x -= paddle_speed
        if keys[pygame.K_RIGHT] and paddle_x < 700 - paddle_width:
            paddle_x += paddle_speed

        # Κίνηση μπάλας
        ball_x += ball_speed_x
```

```

ball_y += ball_speed_y

# Αναπήδηση πάνω
if ball_y <= ball_radius:
    ball_speed_y *= -1

# Αναπήδηση πλαϊνά
if ball_x <= ball_radius or ball_x >= 700 - ball_radius:
    ball_speed_x *= -1

# Σύγκρουση με ρακέτα
if ball_y >= paddle_y - ball_radius:
    if paddle_x <= ball_x <= paddle_x + paddle_width:
        ball_speed_y *= -1
        ball_y = paddle_y - ball_radius
        score += 5

# Game Over
if ball_y > 500:
    game_over = True

# Σχεδίαση
screen.fill(BLACK)
pygame.draw.circle(screen, WHITE, [int(ball_x), int(ball_y)], ball_radius)
pygame.draw.rect(screen, BLUE, [paddle_x, paddle_y, paddle_width, paddle_height])

score_text = font.render("Score: " + str(score), True, WHITE)
screen.blit(score_text, [10, 10])

if game_over:
    game_over_text = font.render("GAME OVER! Score: " + str(score), True, WHITE)
    screen.blit(game_over_text, [220, 250])

pygame.display.flip()
clock.tick(60)

pygame.quit()

```

## Προτάσεις Επέκτασης (Προαιρετικές)

Αν ολοκληρώσετε νωρίτερα, δοκιμάστε τις παρακάτω επεκτάσεις:

1. **Αύξηση δυσκολίας:** Αυξήστε την ταχύτητα της μπάλας μετά από κάθε επιστροφή της με τη ρακέτα κατά 1.
2. **Restart:** Προσθέστε δυνατότητα επανεκκίνησης με το πλήκτρο SPACE.