

# Pygame Tutorial

## Άσκηση 1

### 1. Import και αρχικοποίηση

```
import pygame
pygame.init()
```

Η `pygame.init()` ενεργοποιεί όλα τα υποσυστήματα της βιβλιοθήκης (γραφικά, ήχος, είσοδος, κ.λπ.). Χρειάζεται **πάντα** στην αρχή κάθε προγράμματος Pygame.

### 2. Δημιουργία παραθύρου

```
gameWindow = pygame.display.set_mode((800, 300))
pygame.display.set_caption("Detect All Mouse Motion")
```

φτιάχνει ένα παράθυρο 800x300 pixels και ορίζει τον τίτλο του παραθύρου.

```
gameWindow = pygame.display.set_mode((800, 600), pygame.FULLSCREEN)
```

<code>pygame.FULLSCREEN</code>	Ανοίγει το παιχνίδι σε πλήρη οθόνη.
<code>pygame.RESIZABLE</code>	Επιτρέπει στο χρήστη να αλλάζει το μέγεθος του παραθύρου.
<code>pygame.NOFRAME</code>	Δημιουργεί παράθυρο χωρίς περίγραμμα ή κουμπιά (X, minimize, maximize).

### 3. Κεντρικός βρόχος παιχνιδιού

```
exit_game = False

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

Ο βρόχος αυτός:

- τρέχει συνέχεια όσο το παιχνίδι είναι ενεργό,
- «πιάνει» όλα τα **γεγονότα (events)**, όπως πατήματα πλήκτρων ή κινήσεις ποντικιού, quit (x) κλπ.
- Αν ο χρήστης πατήσει το “X” του παραθύρου, το παιχνίδι τερματίζεται.

#### 4. Ανίχνευση κίνησης του ποντικιού

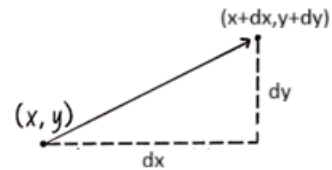
```
elif event.type == pygame.MOUSEMOTION:
    # event.pos gives the current position (x, y)
    # event.rel gives the change since the last motion (dx, dy)
    print(f"Mouse moved to {event.pos}, relative movement:
{event.rel}")

elif event.type == pygame.MOUSEBUTTONDOWN:
    print(f"Mouse button {event.button} pressed at {event.pos}")

elif event.type == pygame.MOUSEBUTTONUP:
    print(f"Mouse button {event.button} released at {event.pos}")
```

event.pos → δίνει τη **θέση**  $(x, y)$  του ποντικιού.

event.rel → δίνει τη **μεταβολή** από την προηγούμενη θέση  $(dx, dy)$ .



Event.types. Με το παρακάτω πρόγραμμα μπορείτε να δείτε όλα τα events που συμβαίνουν.

```
import pygame
pygame.init()

win = pygame.display.set_mode((400, 200))
pygame.display.set_caption("Event Viewer")

running = True
while running:
    for event in pygame.event.get():
        print(event.type, pygame.event.event_name(event.type))
        if event.type == pygame.QUIT:
            running = False

pygame.quit()
```

## Άσκηση 2

```
clock = pygame.time.Clock()
```

Αντιγράψτε το παρακάτω πρόγραμμα.

```
import pygame
pygame.init()

win = pygame.display.set_mode((800, 300))
pygame.display.set_caption("Διαφορά με και χωρίς clock")

clock = pygame.time.Clock()

x = 50
speed = 5
running = True

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

    x += speed
    if x > 800 or x < 0:
        speed = -speed

    win.fill((0, 0, 0))
    pygame.draw.circle(win, (0, 200, 255), (x, 150), 30)
    pygame.display.update()

    clock.tick(60) #Αλλάζει το πόσο γρήγορα θα τρέχει το πρόγραμμα.
    Δοκιμάστε να το αλλάξετε.

pygame.quit()
```

1. Περιγράψτε τι θα κάνει πριν το τρέξετε και τι θα γίνει αν αλλάξουμε το `clock.tick()`.
2. Μπορείτε να κάνετε τον κύκλο να πηγαίνει πάνω κάτω αντί για δεξιά αριστερά;
3. Μπορείτε να κάνετε τον κύκλο να πηγαίνει προς κάθε κατεύθυνση;

### Άσκηση 3 - Κείμενο στην οθόνη

```
import pygame
pygame.init()

# Δημιουργία παραθύρου
win = pygame.display.set_mode((600, 200))
pygame.display.set_caption("Παράδειγμα blit() με αναμονή")

# Γραμματοσειρά
font = pygame.font.Font(None, 60)

# Δημιουργία δύο μηνυμάτων (surface)
text1 = font.render("Καλημέρα!", True, (255, 255, 0))
text2 = font.render("Καλώς ήρθες στο Pygame!", True, (0, 255, 0))

# Βήμα 1 - Εμφάνιση πρώτου μηνύματος
win.fill((0, 0, 100))          # Μπλε φόντο
win.blit(text1, (180, 70))     # Σχεδίαση του πρώτου κειμένου
pygame.display.update()        # Ενημέρωση οθόνης

pygame.time.wait(2000)         # Αναμονή 2 δευτερόλεπτα

# Βήμα 2 - Εμφάνιση δεύτερου μηνύματος
win.fill((0, 0, 100))          # Καθαρίζουμε ξανά
win.blit(text2, (70, 70))      # Νέο μήνυμα
pygame.display.update()

pygame.time.wait(3000)         # Αναμονή 3 δευτερόλεπτα

pygame.quit()
```

κάντε το παραπάνω πρόγραμμα να τρέχει για πάντα με χρήση events!

Στη συνέχεια δοκιμάστε να κλείσετε το παράθυρο...

Το πρόβλημα είναι....

.....

Για να το λύσουμε θα πρέπει να μετράμε τον χρόνο και όχι να βάζουμε το πρόγραμμα για ύπνο με το `pygame.time.wait(2000)`

```
dt = clock.tick(60) # 60 FPS
timer += dt
```

### **clock.tick(60)**

Η τιμή που επιστρέφει είναι “πόσος χρόνος πέρασε από το τελευταίο frame”. Αυτό είναι το λεγόμενο delta time (dt), απλώς σε χιλιοστά του δευτερολέπτου.

Περιορίζει το frame rate. Το όρισμα (60) σημαίνει:

Περίμενε όσο χρειάζεται, ώστε να μην τρέξει το επόμενο frame πριν περάσουν 1/60 του δευτερολέπτου (~16.6 ms).

Το παρακάτω πρόγραμμα μας τυπώνει στην οθόνη τον χρόνο που πέρασε σε ms από το τελευταίο frame.

```
import pygame
pygame.init()

clock = pygame.time.Clock()
running = True

while running:
    dt = clock.tick(60) # Περιορίζει στα 60 FPS και επιστρέφει ms από το
    προηγούμενο frame
    print("Χρόνος από το προηγούμενο frame:", dt, "ms")

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

και αυτό το πρόγραμμα δείχνει στην οθόνη του pygame τον χρόνο

```
import pygame
pygame.init()

# Ρύθμιση παραθύρου
win = pygame.display.set_mode((500, 200))
pygame.display.set_caption("Εμφάνιση dt (χρόνος ανά frame)")

# Ρολόι
clock = pygame.time.Clock()

# Γραμματοσειρά
font = pygame.font.Font(None, 48)

running = True
while running:
    dt = clock.tick(60) # Περιορίζει στα 60 FPS και επιστρέφει ms από το
    προηγούμενο frame

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

    # Καθαρίζουμε το φόντο
    win.fill((0, 0, 0))

    # Δημιουργία κειμένου με το dt
    text = font.render(f"Διάρκεια frame: {dt} ms", True, (255, 255, 0))
    win.blit(text, (50, 80))

    # Εμφάνιση στην οθόνη
    pygame.display.update()

pygame.quit()
```

Άρα το `timer += dt` επιστρέφει το άθροισμα του χρόνου καθώς περνάνε τα frames. Εδώ θα πρέπει να αντικαταστήσουμε το `pygame.time.wait(2000)` με έναν έλεγχο για το timer. Αν είναι `>2000` κάνε αυτό και μηδένισε το, αν είναι `>3000` κάνε το άλλο και μηδένισέ το.

```

font = pygame.font.Font(None, 60)
text1 = font.render("Καλημέρα!", True, (255, 255, 0))
text2 = font.render("Καλώς ήρθες στο Pygame!", True, (0, 255, 0))

# Αλλάζουμε μήνυμα κάθε 2 δευτερόλεπτα
if show_text1 and timer > 2000:
    show_text1 = False
    timer = 0
elif not show_text1 and timer > 3000:
    show_text1 = True
    timer = 0

win.fill((0, 0, 100))
if show_text1:
    win.blit(text1, (180, 70))
else:
    win.blit(text2, (20, 70))

```

Με βάση όλα τα παραπάνω φτιάξτε ένα πρόγραμμα που θα δείχνει πάνω δεξιά την θέση του ποντικιού στην οθόνη!

### Οδηγίες βήμα-βήμα

1. Δημιούργησε ένα παράθυρο με `pygame.display.set_mode((πλάτος, ύψος))`.
2. Φτιάξε μια γραμματοσειρά με `pygame.font.Font(None, μέγεθος)`.
3. Μέσα στο `while` loop:

- ο Πάρε τη θέση του ποντικιού με

```
mouse_x, mouse_y = pygame.mouse.get_pos()
```

- ο Φτιάξε το κείμενο με `font.render()`.

```
text = font.render(f"Θέση ποντικιού: ({x}, {y})", True, (255, 255, 0))
```

- ο Ζωγράφισε το κείμενο στην οθόνη με `blit()`.

4. Χρησιμοποίησε `pygame.display.update()` για να φαίνεται η αλλαγή.
5. Χρησιμοποίησε `clock.tick(60)` για σταθερή ανανέωση (60 FPS).

## BONUS

```
import pygame
pygame.init()

# Παράθυρο
WIDTH, HEIGHT = 800, 300
win = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Απλό Mouse Trail")

clock = pygame.time.Clock()
running = True

# Επιφάνεια για το εφέ ξεθωριάσματος
trail_surface = pygame.Surface((WIDTH, HEIGHT))
trail_surface.set_alpha(20)      # μικρότερο = πιο μακρύ ίχνος
trail_surface.fill((0, 0, 0))   # μαύρο "πέπλο" για fade

prev_pos = None

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

    win.blit(trail_surface, (0, 0))
    pos = pygame.mouse.get_pos()
    pygame.draw.circle(win, (255, 255, 255), pos, 4) # λευκός κύκλος
    pygame.display.update()
    clock.tick(120)

pygame.quit()
```



with color

```
import pygame
import random
import colorsys
pygame.init()

WIDTH, HEIGHT = 800, 300
win = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Smooth Colorful Mouse Trail")

clock = pygame.time.Clock()
running = True

# Surface for fade effect
trail_surface = pygame.Surface((WIDTH, HEIGHT))
trail_surface.set_alpha(25)      # smaller = longer trail
trail_surface.fill((0, 0, 0))   # black fade

hue = 0
# Slightly bigger brush radius
RADIUS = 16

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

    # Gradually fade old circles
    win.blit(trail_surface, (0, 0))

    # Draw a colorful circle at mouse position
    pos = pygame.mouse.get_pos()
    rgb = colorsys.hsv_to_rgb(hue, 1, 1)
    color = tuple(int(c * 255) for c in rgb)
    pygame.draw.circle(win, color, pos, RADIUS)
    hue = (hue + 0.005) % 1 # smooth color cycling

    pygame.display.update()
    clock.tick(120)

pygame.quit()
```

