

# Snake Game

## Εισαγωγή

Αρχικοποιήστε το pygame για να ξεκινήσει το πρόγραμμα.  
Φτιάξτε ένα παράθυρο με συγκεκριμένο πλάτος και ύψος.  
Βάλτε έναν τίτλο στο παράθυρο (π.χ. "Snake").  
Δημιουργήστε ένα game loop που τρέχει συνεχώς όσο το παιχνίδι είναι ανοιχτό.  
Ελέγξτε τα events, ώστε όταν ο χρήστης πατήσει το X, το παιχνίδι να κλείνει.  
Καθαρίστε την οθόνη κάθε frame (π.χ. γεμίστε την με μαύρο χρώμα).  
Ενημερώστε την οθόνη ώστε να εμφανίζεται ότι σχεδιάσατε.

## Φίδι

Ορίστε το μέγεθος ενός κελιού (π.χ. 20 pixels).  
Διαλέξτε μια αρχική θέση για το τετράγωνο στο grid (π.χ. x=10, y=10).  
Σχεδιάστε ένα πράσινο τετράγωνο στη θέση αυτή.  
Το τετράγωνο πρέπει να εμφανίζεται κάθε frame (μέσα στο loop).

```
pygame.draw.rect(screen,(0, 255, 0),(snake_x * CELL, snake_y * CELL, CELL, CELL))
```

Ζωγράφισε ένα πράσινο(0, 255, 0) τετράγωνο:

1. μεγέθους CELLxCELL
2. στη θέση (snake\_x, snake\_y)

## Κίνηση

```
keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]:
    snake_x -= 1
if keys[pygame.K_RIGHT]:
    snake_x += 1
if keys[pygame.K_UP]:
    snake_y -= 1
if keys[pygame.K_DOWN]:
    snake_y += 1
```

## Αυτόματη Κίνηση

Εισάγουμε μια νέα μεταβλητή η οποία θα καθορίζει το προς τα που θα κινεται το φίδι.

```
if direction == "LEFT":  
    snake_x -= 1  
elif direction == "RIGHT":  
    snake_x += 1  
elif direction == "UP":  
    snake_y -= 1  
elif direction == "DOWN":  
    snake_y += 1
```

Αλλάζουμε την μεταβλητή στην αρχή του loop. Προσοχή το φίδι ΔΕΝ πρέπει να μπορεί να κάνει αλλαγή 180°.

```
if event.type == pygame.KEYDOWN:  
    if event.key == pygame.K_LEFT:  
        direction = "LEFT"  
    . . .  
    . . .
```

## Φαγητό

Θα βάλουμε σε τυχαία x,y ένα τετράγωνο 10x10.

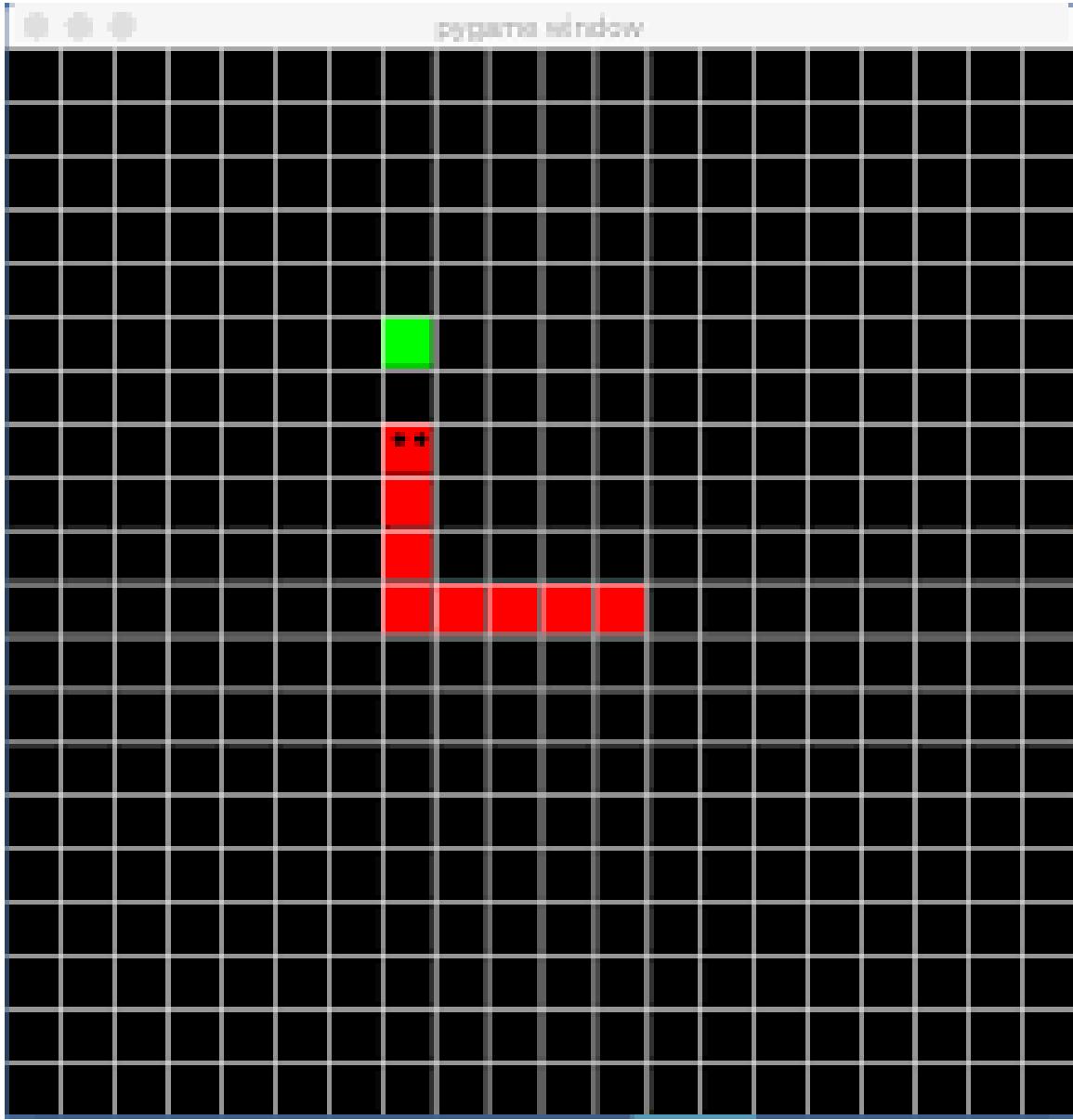
```
food_x = random.randint(0, (WIDTH // CELL) - 1)  
food_y = random.randint(0, (HEIGHT // CELL) - 1)
```

Πότε το φίδι «τρώει» το φαγητό?

όταν τα x,y του food και snake είναι ακριβως ίδια.

```
if snake_x == food_x and snake_y == food_y:  
    food_x = random.randint(0, (WIDTH // CELL) - 1)  
    food_y = random.randint(0, (HEIGHT // CELL) - 1)
```

Φίδι με σώμα, όχι μόνο κεφάλι



Το φίδι μας θα είναι ένα σύνολο από τετράγωνα 10x10 pixels σε συγκεκριμένες θέσεις x,y

```
snake = [  
    [10, 10], # kefali  
    [9, 10], # swma 1  
    [8, 10] # oura  
]
```

Πως το ζωγραφίζουμε;

```
for (x, y) in snake:  
    pygame.draw.rect(screen, (0, 255, 0), (snake_x * CELL, snake_y * CELL, CELL, CELL))
```

ετσι όμως θα είναι ένα ακίνητο

```
new_head = [snake_x, snake_y]  
snake.insert(0, new_head)  
snake.pop()
```

Πλέον δεν κουνάμε το κουτάκι αλλά σχεδιάζουμε ένα νέο κουτί ανάλογα την κατεύθυνση.

Προσθέτουμε έλεγχο για το πότε έφαγε:

```
if (snake_x == food_x and snake_y == food_y):  
    food_x = random.randint(0, (WIDTH // CELL) - 1)  
    food_y = random.randint(0, (HEIGHT // CELL) - 1)  
else:  
    snake.pop()
```

## Έλεγχος ορίων

```
if snake_x < 0 or snake_x >= GRID or snake_y < 0 or snake_y >= GRID:  
    show_game_over()  
    running = False  
    continue
```

```
def show_game_over():  
    screen.fill((0, 0, 0))  
    text = font.render("GAME OVER", True, (255, 255, 0))  
    rect = text.get_rect(center=(WIDTH//2, HEIGHT//2))  
    screen.blit(text, rect)  
    pygame.display.flip()  
    pygame.time.wait(2000)
```

## Έλεγχος σύγκρουσης κεφαλιού/σώματος

Θα πρέπει να ελέγξουμε αν το νέο μπλοκ που βάζουμε για κεφάλι, υπάρχει ήδη στον πίνακα του σώματος.

```
snake = [  
    [x0, y0], # κεφαλή  
    [x1, y1], # σώμα  
    [x2, y2], # σώμα
```

```
    ...
]
```

Δηλαδή αν οι συντεταγμένες [x0,y0] υπάρχουν στον υπόλοιπο πίνακα. Στην python μπορούμε να διαλέξουμε μέρος του πίνακα με τις εξής εντολές.

```
snake[0] → το πρώτο στοιχείο του πίνακα  
snake[1:] → όλος ο πίνακας χωρίς το πρώτο στοιχείο
```

Άρα αυτό που πρέπει να κάνουμε είναι:

Αν

```
new_head in snake[1:]  
call GameOver, running False
```

## Προσθήκη Score High Score και Level

Θα βάλουμε τρία κείμενα στην κορυφή.

```
score_text = font.render("Score: {score}", True, (255, 255, 0))  
screen.blit(score_text, (WIDTH - score_text.get_width() - 20, 5))  
  
level_text = font.render("Level: {level}", True, (0, 200, 255))  
screen.blit(level_text, (20, 5))  
  
highscore_text = font.render("High Score: {highscore}", True, (255, 255, 255))  
screen.blit(highscore_text, ((WIDTH - highscore_text.get_width()) // 2, 5))
```

Θέσεις τοποθέτησης

```
screen.blit(score_text, (WIDTH - score_text.get_width() - 20, 5))  
screen.blit(level_text, (20, 5))  
screen.blit(highscore_text, ((WIDTH - highscore_text.get_width()) // 2,  
5))
```

WIDTH - score\_text.get\_width() - 20 → Τέρμα δεξιά - μήκος κειμένου - 20px  
20, 5 → Τέρμα αριστέρα + 20

```
WIDTH - highscore_text.get_width()) // 2, 5 → στο μέσο της οθονης
```

Βρείτε έναν τρόπο να αυξάνονται τα:

1. score: +1 για κάθε φαγητό
2. level: +1 κάθε 5 φαγητά
3. FPS: +1 κάθε 1 level

## Restart the Game!

Το παιχνίδι τερματίζει σε 2 σημεία.

Θα πρέπει αντί να κάνουμε quit να φτιάξουμε μια συνάρτηση ώστε να περιμένει ο χρήστης να πατήσει Space και να συνεχίζει.

```
def handle_game_over():
    show_game_over()
    waiting = True
    while waiting:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
                reset_game()
                return
```

το return θα βγει από την loop και το παιχνίδι θα συνεχιστεί! Θα πρέπει όμως να αρχικοποιηθεί! Φτιάξτε την συνάρτηση reset\_game έτσι ώστε να αρχικοποιεί όλες τις μεταβλητές του παιχνιδιού..

## Press Space to start screen

Θέλουμε το παιχνίδι όταν πατάμε run να μην ξεκινάει αμέσως αλλά να βλέπουμε μια οθόνη με prompt Press Space to Play.

1. Προσθήκη Loop πριν το main Running Loop
2. Προσθήκη κειμένου (\*\* πως θα αναβοσβήνει?? \*\*)
3. Αν ο χρήστης πατήσει Space μόνο τότε βγαίνει από το Loop

```
for event in pygame.event.get():
```

```
if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:  
    start=0  
    break
```

## Αποθήκευση και ανάκτηση High Score

Θέλουμε το highscore να μην είναι μια απλή μεταβλητή που θα χάνεται όταν κλείσει το πρόγραμμα. Για αυτό θα χρειαστεί να δημιουργήσουμε ένα εξωτερικό αρχείο στο οποίο θα αποθηκεύουμε και θα ανακτούμε **τον αριθμό** του highscore.

Χειρισμός Αρχείων

### 1) `open("file.txt", "w")`

Ανοίγει (ή δημιουργεί) αρχείο ΜΟΝΟ για γράψιμο.

- "w" = write
- Αν το αρχείο **δεν υπάρχει** → το δημιουργεί
- Αν το αρχείο **υπάρχει** → το **σβήνει** και ξεκινά από την αρχή (overwrite)

Παράδειγμα:

```
f = open("test.txt", "w")
```

προσοχή! Το test.txt θα ψάξει να το βρει από το path που βρισκέται το terminal οπου καλούμε το αρχείο .py να τρέξει!

### 2) `open("file.txt", "r")`

Ανοίγει αρχείο ΜΟΝΟ για διάβασμα.

- "r" = read
- Αν το αρχείο **δεν υπάρχει** → error

Παράδειγμα:

```
f = open("test.txt", "r")
```

### 3) `with open(...) as file:`

Είναι ο “ασφαλής τρόπος” να ανοίγεις αρχείο.

- ανοίγει το αρχείο
- σου δίνει ένα αντικείμενο με όνομα file (μπορεί να είναι ό,τι θέλετε)
- όταν τελειώσει το block, το αρχείο **κλείνει μόνο του**

Παράδειγμα:

```
with open("a.txt", "r") as file: data = file.read()
```

#### 4) file.write("text")

Γράφει κείμενο στο αρχείο.

- Δέχεται μόνο string
- Αν δώσεις αριθμό, πρέπει πρώτα str(...)

Παράδειγμα:

```
with open("a.txt", "w") as file:  
    file.write("Hello")
```

#### 5) file.read()

Διαβάζει όλο το αρχείο και το επιστρέφει ως string.

```
with open("a.txt", "r") as file:  
    content = file.read()  
    print(content)
```

#### 6) .strip()

Αφαιρεί κενά και αλλαγές γραμμής από αρχή/τέλος.

Παράδειγμα:

```
text = "12\n"  
print(text.strip())
```

#### 7) try: ... except: ...

Προσπαθεί να τρέξει κάτι που μπορεί να αποτύχει.

Αν αποτύχει, πάει στο except. Παράδειγμα (για αρχείο που μπορεί να μην υπάρχει):

```
try:  
    with open("highscore.txt", "r") as file:  
        highscore = int(file.read().strip())  
except:  
    highscore = 0
```

Τι πετυχαίνουμε:

- Αν το αρχείο υπάρχει → διαβάζει
- Αν δεν υπάρχει → βάζει 0, δεν κρασάρει

## Ασκήσεις για αρχεία

### Άσκηση 1

Γράψτε πρόγραμμα που:

1. Δημιουργεί αρχείο data.txt
2. Γράφει μέσα τη λέξη Python

### Άσκηση 2

Γράψτε πρόγραμμα που:

1. Δημιουργεί αρχείο data.txt
2. Γράφει μέσα τη λέξη Python
3. Διαβάζει το αρχείο
4. Εκτυπώνει το περιεχόμενο

### Άσκηση 3 – Αριθμός από αρχείο



Γράψτε πρόγραμμα που

1. Γράφει τον αριθμό 10 στο αρχείο number.txt
2. Διαβάζει τον αριθμό από το αρχείο
3. Τον μετατρέπει σε int
4. Εκτυπώνει τον αριθμό + 5

Άρα:

Στην αρχή του προγράμματος ανοίγουμε το αρχείο highscore.txt (αν υπάρχει), διαβάζουμε το περιεχόμενο και το αποθηκεύουμε στη μεταβλητή highscore. Αν δεν υπάρχει την μηδενίζουμε.

```
try:  
    with open("./highscore.txt", "r") as file:  
        print("inside")  
        highscore = int(file.read().strip())  
except:  
    highscore = 0
```

Στη συνέχεια όποτε το highscore αλλάζει, κάνουμε ότι κάναμε KAI αποθηκεύουμε και στο αρχείο.

```
if score > highscore:  
    highscore = score  
    with open("./highscore.txt", "w") as file:  
        file.write(str(highscore))
```

## Super Food

Εμφανίζεται σε τυχαία σημεία σπάνια με διάρκεια 3s και θα δίνει +3 στο σώμα/score/lvl.  
Θα φτιαξουμε τρεις συναρτήσεις.

```
def spawn_super_food():
def update_super_food():
def draw_super_food():
```

Η spawn:

Draw θα σχεδιάσει το super food σε x,y που θα καθοριστούν από την spawn και η update καθορίζει για πόση ώρα θα φαίνεται. Τέλος θα ορίσουμε ένα τυχαίο χρονικό διάστημα (random.randint(0, 200) == 0 πχ) που θα καλείται η spawn. Η draw καλείται συνέχεια αφού θα έχει έλεγχο.

```
def spawn_super_food():
    global super_food_active, super_food_x, super_food_y, super_food_timer
    super_food_active = True
    super_food_x = random.randint(0, GRID - 1)
    super_food_y = random.randint(0, GRID - 1)
    super_food_timer = SUPER_FOOD_DURATION

def update_super_food():
    global super_food_active, super_food_timer
    if super_food_active:
        super_food_timer -= 1
        if super_food_timer <= 0:
            super_food_active = False

def draw_super_food():
    if super_food_active:
        pygame.draw.rect(
            screen,
            (0, 0, 255),
            (super_food_x * CELL, super_food_y * CELL, CELL, CELL))
```

## Random obstacles

```
import pygame
import sys
import random

def show_game_over():
    screen.fill((0, 0, 0))
    text = font.render("GAME OVER", True, (255, 255, 0))
    rect = text.get_rect(center=(WIDTH//2, HEIGHT//2))
    screen.blit(text, rect)
    pygame.display.flip()
    pygame.time.wait(2000)

pygame.init()

WIDTH, HEIGHT = 600, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Snake")
clock = pygame.time.Clock()
font = pygame.font.Font(None, 74)

FPS = 10
CELL = 20
GRID = WIDTH//CELL
snake_x = 10
snake_y = 10
direction = "RIGHT"
running = True
food_x = random.randint(0, (WIDTH // CELL) - 1)
food_y = random.randint(0, (HEIGHT // CELL) - 1)
snake = [
    [10, 10],
    [9, 10],
    [8, 10]
]
```

```
while running:
    clock.tick(FPS)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT and direction != "RIGHT":
                direction = "LEFT"
            if event.key == pygame.K_RIGHT and direction != "LEFT":
                direction = "RIGHT"
            if event.key == pygame.K_UP and direction != "DOWN":
                direction = "UP"
            if event.key == pygame.K_DOWN and direction != "UP":
                direction = "DOWN"

        if direction == "LEFT":
            snake_x -= 1
        elif direction == "RIGHT":
            snake_x += 1
        elif direction == "UP":
            snake_y -= 1
        elif direction == "DOWN":
            snake_y += 1

    new_head = [snake_x, snake_y]
    snake.insert(0, new_head)

    if snake_x < 0 or snake_x >= GRID or snake_y < 0 or snake_y >= GRID:
        show_game_over()
        running = False
        continue

    if (snake_x == food_x and snake_y == food_y):
        food_x = random.randint(0, (WIDTH // CELL) - 1)
        food_y = random.randint(0, (HEIGHT // CELL) - 1)
    else:
        snake.pop()

    screen.fill((0, 0, 0))
    pygame.draw.rect(screen,(255, 0, 0),(food_x * CELL, food_y * CELL, CELL, CELL))
    for (x, y) in snake:
        pygame.draw.rect(screen, (0, 255, 0), (x * CELL, y * CELL, CELL, CELL))
```

```
pygame.display.flip()  
  
pygame.quit()  
sys.exit()
```