

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 ό,τι 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 CELL×CELL
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.

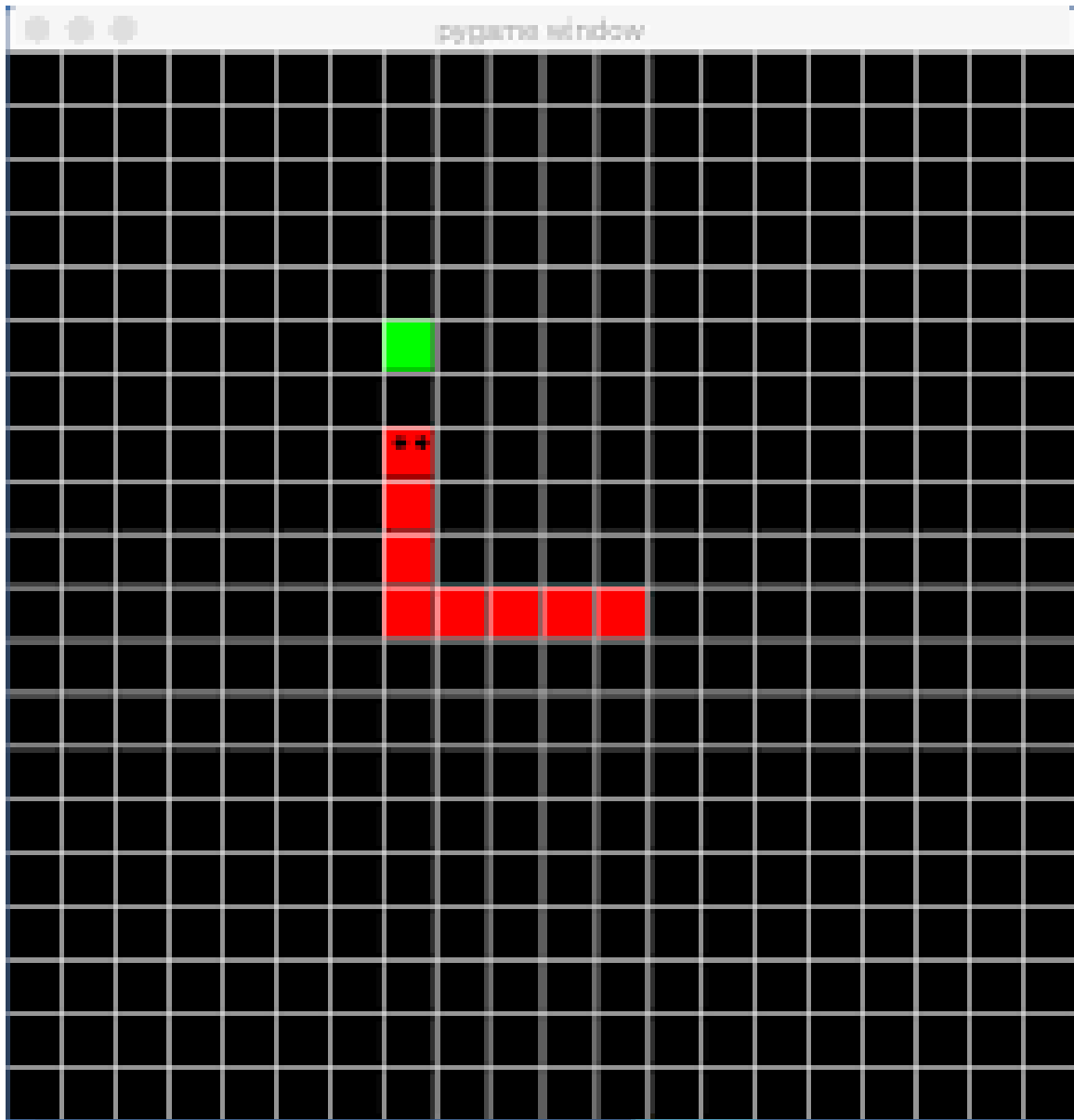
```
food_x = random.randint(0, (WIDTH // CELL) - 1)
food_y = random.randint(0, (HEIGHT // CELL) - 1)
```

Πότε το φίδι «τρώει» το φαγητο?

όταν τα x,y του food και snake είναι ακριβώς 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(f"Score: {score}", True, (255, 255, 0))
screen.blit(score_text, (WIDTH - score_text.get_width() - 20, 5))

level_text = font.render(f"Level: {level}", True, (0, 200, 255))
screen.blit(level_text, (20, 5))

highscore_text = font_ui.render(f"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 αριστερά + 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..

```

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()

```

