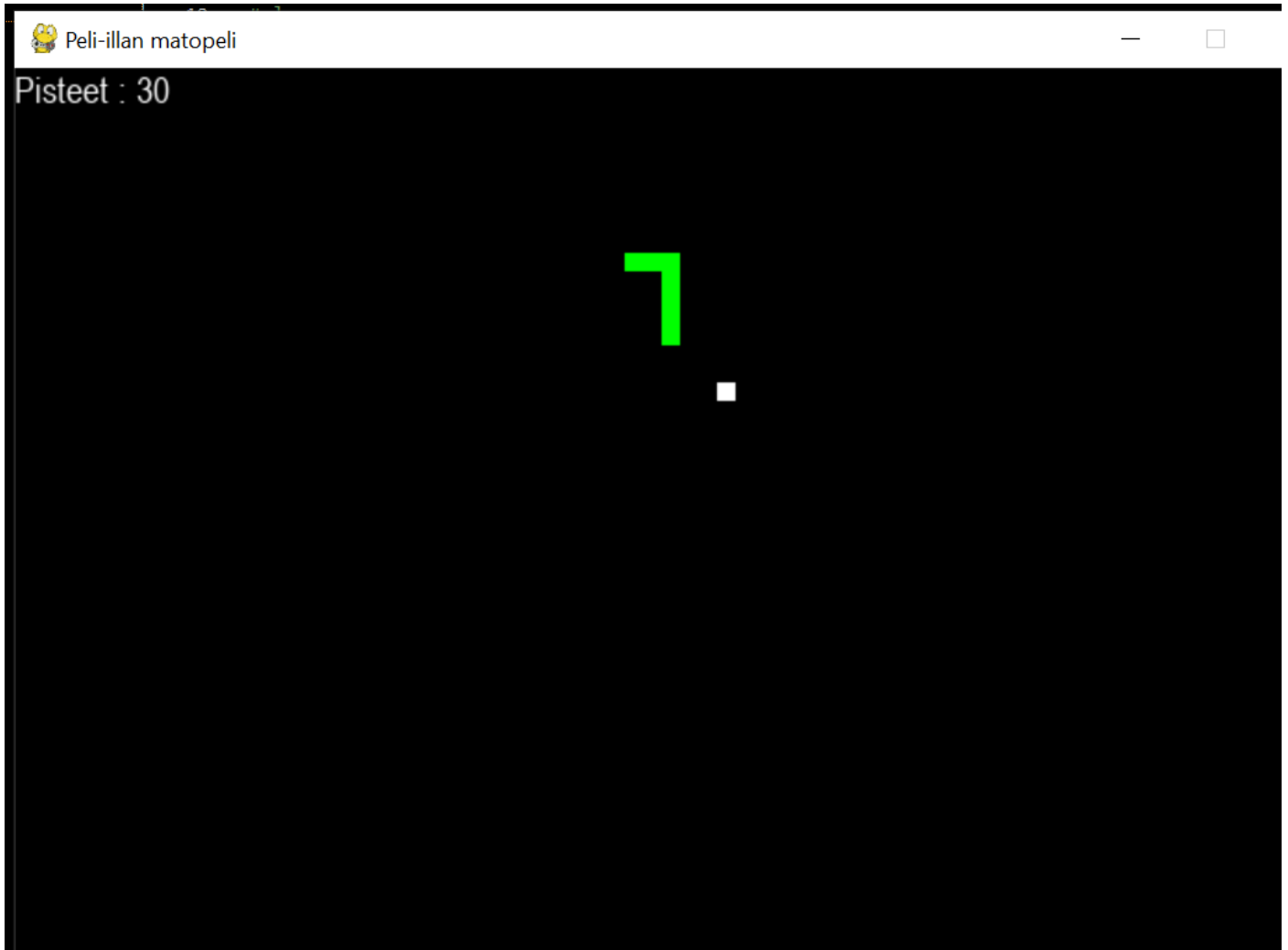


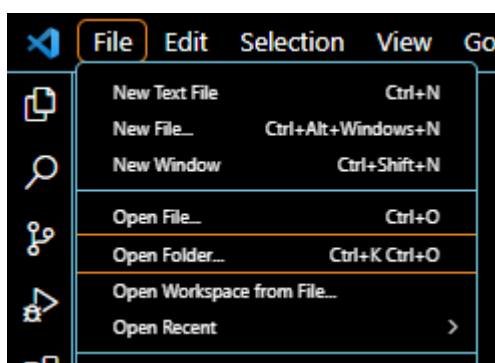
Matopelin koodaus pythonilla

Taitotalon peli-ilta 7.11.2022



Alkutoimet

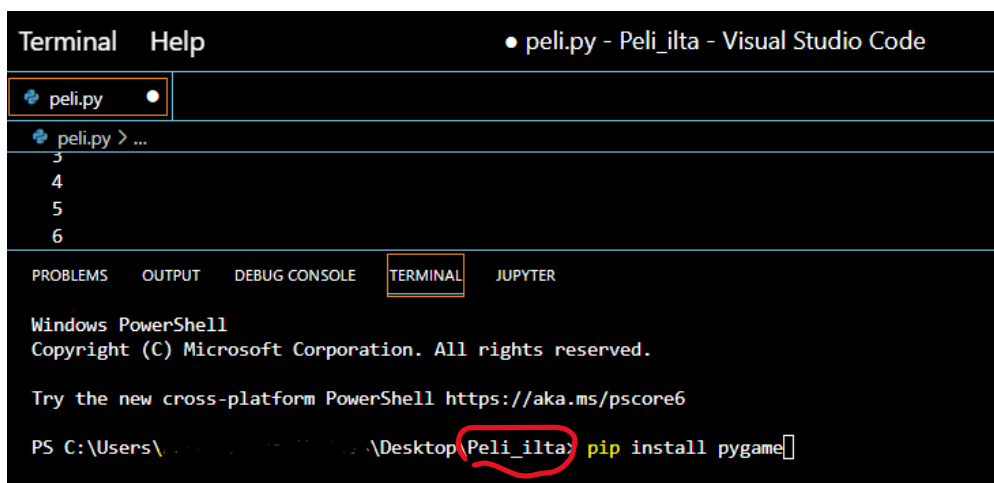
1. Luo työpöydälle uusi kansio ja anna sille nimi (hiiren oikea > Uusi > Kansio)
2. Tee ensiksi luomaasi kansioon uusi tiedosto. (hiiren oikea > Uusi > Tekstitiedosto)
3. Anna tiedostolle joku nimi ja tallenna se muotoon "jokunimi.py"
4. Avaa Visual Studio Code
5. Mene File ja Open Folder -> avaa juuri luomasi kansio



6. Asenna pygame (Terminal > New Terminal) tarkista, että olet oikeassa kansiossa

pip install pygame

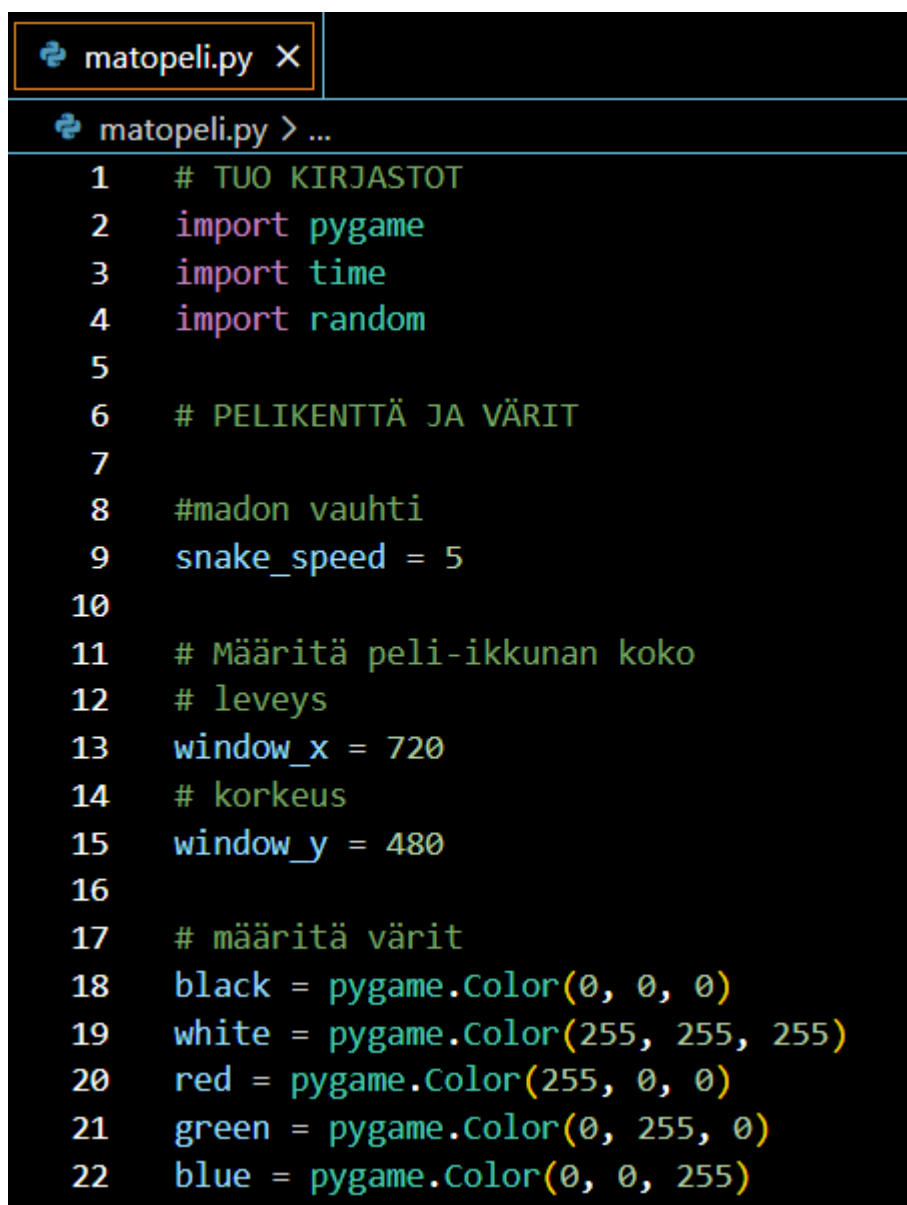
paina enter



7. Klikkaa vasemmasta alakulmasta Select Interpreter, valitse Recommended
8. Voit alkaa kirjoittaa luomaasi tiedostoon

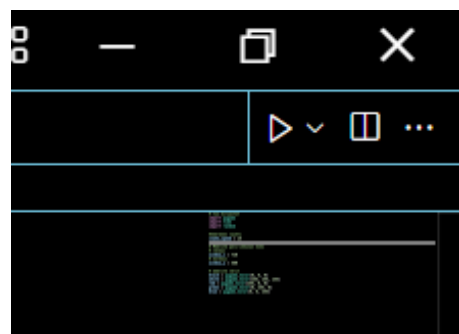
Luodaan pelin puitteet

1. Tuo tarvittavat kirjastot import-toiminnolla
2. Määritä madolle vauhti, peli-ikkunan koko ja pelissä käytettävät värit.
3. Arvoja voi muokata myöhemmin.
4. HUOM! Risuaidalla (#) merkityt kommentit voi jättää kirjoittamatta



```
matopeli.py X
matopeli.py > ...
1  # TUO KIRJASTOT
2  import pygame
3  import time
4  import random
5
6  # PELIKENTTÄ JA VÄRIT
7
8  #madon vauhti
9  snake_speed = 5
10
11 # Määritä peli-ikkunan koko
12 # leveys
13 window_x = 720
14 # korkeus
15 window_y = 480
16
17 # määritä värit
18 black = pygame.Color(0, 0, 0)
19 white = pygame.Color(255, 255, 255)
20 red = pygame.Color(255, 0, 0)
21 green = pygame.Color(0, 255, 0)
22 blue = pygame.Color(0, 0, 255)
```

5. Tarkista, että koodi toimii ajamalla koodi eli painamalla kolmionäppäintä ruudun oikeassa yläkulmassa



Alustetaan peli

```
# alusta peli
pygame.init()

# alusta peli-ikkuna
pygame.display.set_caption('Peli-illan matopeli')
game_window = pygame.display.set_mode((window_x, window_y))

# Ruutunopeus eli FPS (frames per second)
fps = pygame.time.Clock()
```

Määritellään madon koko, lähtöpaikka ja oletuskulkusuunta kentällä. Määritellään hedelmän paikka.

```
# määrittele madon lähtöpaikka (anna koordinaatit)
snake_position = [100, 50]

# määrittele madon koko, joka on neljä koordinaattia
snake_body = [ [100, 50],
                [90, 50],
                [80, 50],
                [70, 50]
              ]

# määrittele hedelmän paikka
fruit_position = [random.randrange(1, (window_x//10)) * 10,
                  random.randrange(1, (window_y//10)) * 10]

# hedelmä ilmiantuu uudelleen on totta
fruit_spawn = True

# määrittele madon oletussuunta oikealle
direction = 'RIGHT'
change_to = direction
```

Huom! Random-olio luo satunnaislukuja.

Alustetaan pistelasku

```
# alusta pistelasku
score = 0

# luo pistelaskufunktiot ja anna sille muuttujat: valinta, väri, fontti ja koko

def show_score(choice, color, font, size):
    # luo score_font - olio, anna sille muuttujiksi fontti ja koko
    score_font = pygame.font.SysFont(font, size)

    # luo pistelasku-olio, joka näyttää pistelaskun tulokset
    # score_surface
    score_surface = score_font.render('Pisteet : ' + str(score), True, color)

    # luo suorakulmainen olio tekstille
    score_rect = score_surface.get_rect()

    # näytä teksti suorakulmiossa
    game_window.blit(score_surface, score_rect)
```

Luodaan pelin lopetusfunktio

```
def game_over():  
  
    # luo my_font - olio  
    my_font = pygame.font.SysFont('Arial', 50)  
  
    # luo tekstiolio  
    game_over_surface = my_font.render('Pisteesi ovat : ' + str(score), True, red)  
  
    #luo suorakulmainen olio tekstin ympärille  
    game_over_rect = game_over_surface.get_rect()  
  
    # aseta teksti paikka  
    game_over_rect.midtop = (window_x/2, window_y/4)  
  
    # blit() piirtää tekstin pelin  
    game_window.blit(game_over_surface, game_over_rect)  
    pygame.display.flip()  
  
    # kahden sekunnin jälkeen ohjelma suljetaan  
    time.sleep(2)  
  
    # lopeta pygame  
    pygame.quit()  
  
    # lopeta ohjelma  
    quit()
```

Pääohjelma

```
# Pääohjelma
# peli jatkuu kunnes "while True"

while True:

    # pygame-kirjaston tapahtumankäsittelijä
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                change_to = 'UP'
            if event.key == pygame.K_DOWN:
                change_to = 'DOWN'
            if event.key == pygame.K_LEFT:
                change_to = 'LEFT'
            if event.key == pygame.K_RIGHT:
                change_to = 'RIGHT'

    # jos kaksi näppäintä on painettuna samaan aikaan,
    # mato kulkee kuitenkin vain yhteen suuntaan
    if change_to == 'UP' and direction != 'DOWN':
        direction = 'UP'
    if change_to == 'DOWN' and direction != 'UP':
        direction = 'DOWN'
    if change_to == 'LEFT' and direction != 'RIGHT':
        direction = 'LEFT'
    if change_to == 'RIGHT' and direction != 'LEFT':
        direction = 'RIGHT'

    # ohjataan matoa aina kymmenen pykälää suuntaansa
    if direction == 'UP':
        snake_position[1] -= 10
    if direction == 'DOWN':
        snake_position[1] += 10
    if direction == 'LEFT':
        snake_position[0] -= 10
    if direction == 'RIGHT':
        snake_position[0] += 10
```

pääohjelma jatkuu:

```
# jos mato ja hedelmä törmäävät, niin
# pisteet kasvavat kymmenellä pisteellä
snake_body.insert(0, list(snake_position))
if snake_position[0] == fruit_position[0] and snake_position[1] == fruit_position[1]:
    score += 10
    fruit_spawn = False
else:
    snake_body.pop()

# törmäyksen jälkeen asetetaan hedelmälle uusi satunnainen paikka
if not fruit_spawn:
    fruit_position = [random.randrange(1, (window_x//10)) * 10,
                     random.randrange(1, (window_y//10)) * 10]

fruit_spawn = True
game_window.fill(black)
```


pääohjelma jatkuu:

```
# madon ja hedelmän värit
for pos in snake_body:
    pygame.draw.rect(game_window, green, pygame.Rect(
        pos[0], pos[1], 10, 10))

pygame.draw.rect(game_window, white, pygame.Rect(
    fruit_position[0], fruit_position[1], 10, 10))

# jos mato menee kentän yli, peli päättyy
if snake_position[0] < 0 or snake_position[0] > window_x-10:
    game_over()
if snake_position[1] < 0 or snake_position[1] > window_y-10:
    game_over()

# Jos mato koskee itseään, peli päättyy
for block in snake_body[1:]:
    if snake_position[0] == block[0] and snake_position[1] == block[1]:
        game_over()

# näytä tulokset
show_score(1, white, 'arial', 20)

# päivitä ruutu
pygame.display.update()

# Frame per second ruutunopeus/madon nopeus
# päivittyy eli nopeutuu tai hidastuu näppäilemällä
fps.tick(snake_speed)
```

Nyt voitkin ajaa ohjelman oikealla yläkulmassa olevasta nuolesta ja alkaa pelata!



EXTRA

Voit vaihtaa esimerkiksi

- **värejä**
 - lisää pygame Color-olion käytöstä:
<https://www.pygame.org/docs/ref/color.html>
 - voit kokeilla eri värejä osoitteessa:
https://www.w3schools.com/css/css_colors_rgb.asp
- **madon nopeutta**
- **tekstiä ja pistelaskua, esim. fontti, koko**

KILPAILU

Parhaimmat pisteet saanut kilpailija palkitaan extrapalkinnolla. Kilpailuun osallistuaksesi

1. Määritä madon alkunopeudeksi 15.
2. Määritä nopeus kasvamaan 1,1 kertaiseksi joka kerta kun mato syö hedelmän.
3. Vaihda madon väri esimerkiksi siniseksi.
4. Vaihda lopputekstin "Pisteesi ovat" fontin kooksi 80.

Kun nämä muutokset on tehty, saat kolme yritystä. Osallistu parhaalla pistemäärällä kilpailuun.

Pelin alkuperäinen koodi

<https://www.geeksforgeeks.org/snake-game-in-python-using-pygame-module/?ref=gcse>