Documentație proiect Python

,,Memory Game”

Pentru a rezolva tema am utilizat Python 3.12.7 si Visual Studio Code

* **Prerequisites & run**

**Există mai multe moduri pentru a rula un fișier Python, în funcție de sistemul de operare folosit și mediul de lucru. Câteva dintre acestea ar fi:**

* Din linia de comandă(Terminal sau Command Prompt)

Dacă folosim **Windows**, **Mac** sau **Linux**, trebuie să rulăm comenzi într-o aplicație numită Terminal (Command Prompt pe Windows).

1. Deschidem Command Prompt(Windows)
2. Scriem comanda: ***python –version***
3. Dacă primim un răspuns de tipul Python 3.x.x, atunci Python este instalat.  
   Dacă primim eroare, descarcăm și instalăm Python de pe [python.org](https://www.python.org/).

(Eu folosesc Python 3.12.7)

1. Deschidem folder-ul în care am salvat fișierul Python:

De exemplu, unul dintre fișierele mele se numește tema1.py si este salvat în Desktop. Cu ajutorul comenzii ***cd Desktop***, pot intra in folder-ul respectiv

( cd=change directory).

1. Rulăm fișierul Python: ***python program.py***

* Dintr-un editor de cod(IDE)

1. În IDLE( editorul implicit instalat cu Python)
2. Deschidem **IDLE**.
3. Click pe **File → Open** și selectăm fișierul program.py.
4. Apasăm **F5** sau mergem la **Run → Run Module** pentru a rula fișierul.
5. **În VS Code**:
6. Descărcam VS Code
7. Instalăm Python de pe [python.org](https://www.python.org/" \t "_new).
8. Accesăm Extensions din VS Code și căutăm extensia Python(dezvoltată de Microsoft) și apăsăm pe INSTALL
9. Deschidem fișierul Python pe care vrem să-l rulăm (File-Open File și selectăm fișierul)
10. Configurăm interpretatorul Python (Ctrl+ Shift+P pt a deschide paleta de comenzi, selectăm **Python: Select Interpreter** și alegem versiunea Python pe care o avem instalată.)
11. Pentru a rula fișierul

* Din terminalul integrat, folosind ***comanda python nume\_fisier.py***
* Folosind butonul de rulare (Run Python File)
* Explicatie cod
* import pygame, os, random, time
* pygame.init()

Importă bibliotecile necesare:

* + pygame pentru crearea jocului.
  + os pentru manipularea fișierelor și directoarelor.
  + random pentru amestecarea imaginilor.
  + time pentru măsurarea timpului.
  + pygame.init() inițializează modulele pygame.

 Setează dimensiunile ferestrei jocului, dimensiunile imaginilor și marginile.

lungime = 840

inaltime = 680

marime\_poze = 128

coloane = 5

randuri = 4

distanta\_poze = 10

margine\_stanga = (lungime - ((marime\_poze + distanta\_poze) \* coloane)) // 2

margine\_sus = (inaltime - ((marime\_poze + distanta\_poze) \* randuri)) // 2

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

fundal\_final=(150,200,100)

 coloane și randuri definesc grila jocului.

 margine\_stanga și margine\_sus poziționează grila de imagini la mijlocul ecranului.

 Definirea unor culori (alb, negru, verde pentru fundal final).

selection1 = None

selection2 = None

timp\_intoarcere\_poze= None

timp\_inceput = time.time()

incercari = 0

* selection1 și selection2 rețin indexurile imaginilor selectate.
* timp\_intoarcere\_poze stabilește momentul când imaginile greșit potrivite se vor întoarce.
* timp\_inceput măsoară timpul de la începutul jocului.
* incercari numără încercările jucătorului.

ecran = pygame.display.set\_mode((lungime, inaltime))

pygame.display.set\_caption('Memory Game')

iconita = pygame.image.load(r'C:\Users\pc\Desktop\proiect python\images\1.jpg')

pygame.display.set\_icon(iconita)

 Creează fereastra jocului cu dimensiunile specificate.

 Setează titlul ferestrei și o pictogramă pentru aplicație.

fundal = pygame.image.load(r'C:\Users\pc\Desktop\proiect python\Background.jpg')

fundal = pygame.transform.scale(fundal, (lungime, inaltime))

fundalRect = fundal.get\_rect()

Încarcă imaginea de fundal și o redimensionează pentru a se potrivi cu dimensiunile ecranului.

Creează un font implicit cu dimensiunea 36 pentru afișarea textului pe ecran

poze = []

for item in os.listdir(r'C:\Users\pc\Desktop\proiect python\images'):

    poze.append(item.split('.')[0])

pozeCopy = poze.copy()

poze.extend(pozeCopy)

pozeCopy.clear()

random.shuffle(poze)

font = pygame.font.Font(None, 36)

* Încarcă toate numele imaginilor dintr-un folder și creează o listă dublată (imaginile apar de două ori).
* Amestecă aleator lista de imagini.

poze\_joc= []

poze\_jocRect = []

poze\_ascunse = []

for item in poze:

    poza = pygame.image.load(fr'C:\Users\pc\Desktop\proiect python\images/{item}.jpg')

    poza = pygame.transform.scale(poza, (marime\_poze, marime\_poze))

    poze\_joc.append(poza)

    pozaRect = poza.get\_rect()

    poze\_jocRect.append(pozaRect)

for i in range(len(poze\_jocRect)):

    poze\_jocRect[i][0] = margine\_stanga + ((marime\_poze + distanta\_poze) \* (i % coloane))

    poze\_jocRect[i][1] = margine\_sus + ((marime\_poze + distanta\_poze) \* (i // coloane))

    poze\_ascunse.append(False)

 Încarcă imaginile și le redimensionează.

 Stochează imaginile și dreptunghiurile lor asociate (Rect) într-o listă.

 Calculează pozițiile fiecărei imagini în funcție de rânduri și coloane.

def text(text, x, y, color=BLACK):

    text\_surface = font.render(text, True, color)

    ecran.blit(text\_surface, (x, y))

Rendează textul pe ecran la coordonatele specificate cu o culoare dată.

 ecran.blit(fundal, fundalRect)

    cat\_timp\_a\_trecut = int(time.time() - timp\_inceput)

    text(f"Timp: {cat\_timp\_a\_trecut}s", 10, 10, (154, 245, 18))

    text(f"incercari: {incercari}", 10, 40, (255, 68, 187))

Desenează fundalul și afișează timpul trecut și numărul de încercări.

if timp\_intoarcere\_poze and time.time() >= timp\_intoarcere\_poze:

        poze\_ascunse[selection1] = False

        poze\_ascunse[selection2] = False

        selection1, selection2 = None, None

        timp\_intoarcere\_poze= None

Dacă timpul de întoarcere a fost setat și a trecut, resetează imaginile greșit potrivite.

for event in pygame.event.get():

        if event.type == pygame.QUIT:

            gameLoop = False

        if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:

            if not timp\_intoarcere\_poze:

                for item in poze\_jocRect:

                    if item.collidepoint(event.pos):

                        if not poze\_ascunse[poze\_jocRect.index(item)]:

                            if selection1 is not None:

                                selection2 = poze\_jocRect.index(item)

                                poze\_ascunse[selection2] = True

                                incercari += 1

                            else:

                                selection1 = poze\_jocRect.index(item)

                                poze\_ascunse[selection1] = True

 Gestionează închiderea jocului și clicurile jucătorului.

 Stabilește imagini selectate pe baza poziției clicului.

# Display images or white rectangles

    for i in range(len(poze)):

        if poze\_ascunse[i]:

            ecran.blit(poze\_joc[i], poze\_jocRect[i])

        else:

            pygame.draw.rect(ecran, WHITE, (poze\_jocRect[i][0], poze\_jocRect[i][1], marime\_poze, marime\_poze))

Afișează fie imaginea descoperită, fie un dreptunghi alb pentru imaginile ascunse.

if selection1 is not None and selection2 is not None and not timp\_intoarcere\_poze:

        if poze[selection1] == poze[selection2]:

            selection1, selection2 = None, None

        else:

            timp\_intoarcere\_poze= time.time() + 1

 Compară imaginile selectate.

 Dacă se potrivesc, le lasă descoperite. Altfel, setează un timp de întoarcere.

 if all(poze\_ascunse):

        gameLoop = False

        print(f"Bravo frate! L-ai facut in {cat\_timp\_a\_trecut}s cu doar {incercari} incercari \(°o°)/")

        ecran.fill(fundal\_final)

        text("Bravo frate!", lungime // 2 - 150, inaltime // 2 - 50, (0,0,255))

        text(f"L-ai facut in {cat\_timp\_a\_trecut}s", lungime // 2 - 120, inaltime // 2, (255,220,0))

        text(f"cu doar {incercari} incercari", lungime // 2 - 130, inaltime // 2 + 50, (255,0,0))

        pygame.display.update()

        pygame.time.wait(5000)

 Verifică dacă toate imaginile sunt descoperite.

 Afișează un mesaj de câștig și oprește jocul.

 pygame.display.update()

pygame.quit()

 Actualizează ecranul în fiecare iterație a buclei.

 Închide pygame odată ce jocul se termină.