



PAC-MAN

Sergio Rey García



DAM 1
M03 PROGRAMACIÓ
Joan Dalmau

INDEX

INDEX.....	1
DESCRIPCIÓ.....	1
DISSENY DEL SISTEMA I DIFICULTATS.....	1
FUNCIONALITATS PRINCIPALS.....	2
JOC DE PROVES.....	7
CONCLUSIONS	9

Descripció

Aquest programa de Python implementa un joc simple basat en el concepte del joc Pac-Man. Es una prova molt simple, ja que no conté obstacles ni fantasmes, només punts per recol·lectar.

Disseny del sistema i dificultats

Aquet programa em va semblar bastant fàcil, però hi va haver una part del codi que se'm va complicar. Va ser aquesta part del programa:

```
# Premi
def premi():
    global matrix
    while True:
        a = 0
        num1 = []
        num2 = []
        for i in range(len(matrix)):
            for j in range(len(matrix[i])):
                if i != 0 and j != 0 and i != altura+1 and j != anchura+1:
                    num1.append(i)
                    num2.append(j)
                    a += 1
        if a >= 5:
            break

    premi1 = []
    premi2 = []
    for i in range(5):
        premiNum = random.choice(num1)
        premi1.append(premiNum)
        num1.remove(premiNum)

        premiNum = random.choice(num2)
        premi2.append(premiNum)
        num2.remove(premiNum)
        matrix[premi1[i]][premi2[i]] = premiSimbol
```

Aquesta part del codi em va resultar difícil ja que tenia que implementar un sistema aleatori de punts, i els punts moltes vegades em sortien menys de 5 o damunt del bordejat del mapa, però ho vaig poder solucionar.

Funcionalitats principals

Funció taula

Demana a l'usuari l'alçada i l'amplada del tauler, assegurant-se que ingressin nombres enters.

Estableix una mida mínima per al tauler i ajusta la quantitat total de caselles en conseqüència.

```
# Pregunto al usuari l'altura i anchura que vol a la taula, corroborant que siguin números i no lletres
def taula():
    global altura
    global anchura
    global tauler
    num = []
    for i in range(10):
        num.append(str(i))
    while True:
        while True:
            boleano = True
            altura = input("Altura del tauler: ")
            for i in altura:
                if i not in num:
                    print("ERROR")
                    print("Has de posar números enters")
                    boleano = False
                    break
            if boleano == True:
                altura = int(altura)
                break
        while True:
            boleano = True
            anchura = input("Anchura del tauler: ")
            for i in anchura:
                if i not in num:
                    print("ERROR")
                    print("Has de posar números enters")
                    boleano = False
                    break
            if boleano == True:
                anchura = int(anchura)
                break
        tauler = (altura * anchura)
        if altura > 2 and anchura > 2:
            tauler -= 5
            break
        else:
            system("cls")
            print("ERROR")
            print("El tablero no puede tener menos de 6 cuadros vacíos")
```

Funció omplirMatriu

Omple la matriu del tauler amb un caràcter especificat (en aquest cas, "*").

Truca a la funció dibuixarBordejats per dibuixar les vores del tauler.

```
# Omple la matriu
def omplirMatriu(a):
    global matrix
    matrix = [[a for i in range(anchura + 2)] for i in range(altura + 2)]
    dibuixarBordejats("*")
    matrix[coco[0]][coco[1]] = "@"
```

Funció dibuixarBordejats

Dibuixa les vores del tauler amb un caràcter especificat.

Truca a la funció premi per col·locar premis al tauler.

```
# Dibuixa els bordejats
def dibuixarBordejats(a):
    global matrix
    for i in range(len(matrix)):
        for j in range(len(matrix[i])):
            if i == 0 or j == 0 or i == altura+1 or j == anchura+1:
                matrix[i][j] = a
    premi()
```

Funció premi

Col·loca aleatòriament premis al tauler, representats pel caràcter \$.

Garanteix que hi hagi 5 premis al tauler.

```

# Premi
def premi():
    global matrix
    while True:
        a = 0
        num1 = []
        num2 = []
        for i in range(len(matrix)):
            for j in range(len(matrix[i])):
                if i != 0 and j != 0 and i != altura+1 and j != anchura+1:
                    num1.append(i)
                    num2.append(j)
                    a += 1
        if a >= 5:
            break

    premi1 = []
    premi2 = []
    for i in range(5):
        premiNum = random.choice(num1)
        premi1.append(premiNum)
        num1.remove(premiNum)

        premiNum = random.choice(num2)
        premi2.append(premiNum)
        num2.remove(premiNum)
        matrix[premi1[i]][premi2[i]] = premiSimbol

```

Funció mostrarMatriu

Mostra informació rellevant sobre el joc, com ara punts, caselles i posició del personatge.

Imprimeix la matriu del tauler a la consola.

```

# Mostro la matriu
def mostrarMatriu():
    global tauler
    global Punts
    print(f"Punts: {Punts}")
    print(f"Casilles: {tauler}")
    print(f"Posició: {coco}")
    for i in range(len(matrix)):
        for j in range(len(matrix[i])):
            print(matrix[i][j], end=" ")
        print()

```

Funció terminar

Comprova si el joc s'ha acabat, disminuint el comptador de premis al tauler.

Mostra la matriu actualitzada.

```
# Comprobo que el joc ha terminat
def terminar(a):
    global Terminar
    global tauler
    if anterior == a:
        tauler -= 1
    if PuntsPremi == 0:
        Terminar = False
    mostrarMatriu()
```

Funcions per moure amunt, avall, esquerra i dreta (moureAmunt, moureAvall, moureEsquerra, moureDreta)

Mouen el personatge principal (@) a la direcció corresponent.

Manegen col·lisions amb vores i premis, actualitzant punts i l'estat del joc.

```

# Funció de moure amunt, avall, a l'esquerra i la dreta
def moureAmunt():
    global anterior
    global PuntsPremi
    global Punts
    anterior = matrix[coco[0]-1][coco[1]]
    if matrix[coco[0]-1][coco[1]] == "*":
        print("NO POTS PASSAR")
        time.sleep(0.3)
    else:
        if matrix[coco[0]-1][coco[1]] == premiSimbol:
            PuntsPremi -= 1
            Punts += 1000
        matrix[coco[0]][coco[1]] = " "
        coco[0]-=1
        matrix[coco[0]][coco[1]] = "@"

def moureAvall():
    global anterior
    global PuntsPremi
    global Punts
    anterior = matrix[coco[0]+1][coco[1]]
    if matrix[coco[0]+1][coco[1]] == "*":
        print("NO POTS PASSAR")
        time.sleep(0.3)
    else:
        if matrix[coco[0]+1][coco[1]] == premiSimbol:
            PuntsPremi -= 1
            Punts += 1000
        matrix[coco[0]][coco[1]] = " "
        coco[0]+=1
        matrix[coco[0]][coco[1]] = "@"

def moureEsquerra():
    global anterior
    global PuntsPremi
    global Punts
    anterior = matrix[coco[0]][coco[1]-1]
    if matrix[coco[0]][coco[1]-1] == "*":
        print("NO POTS PASSAR")
        time.sleep(0.3)
    else:
        if matrix[coco[0]][coco[1]-1] == premiSimbol:
            PuntsPremi -= 1
            Punts += 1000
        matrix[coco[0]][coco[1]] = " "
        coco[1]-=1
        matrix[coco[0]][coco[1]] = "@"

def moureDreta():
    global anterior
    global PuntsPremi
    global Punts
    anterior = matrix[coco[0]][coco[1]+1]
    if matrix[coco[0]][coco[1]+1] == "*":
        print("NO POTS PASSAR")
        time.sleep(0.3)
    else:
        if matrix[coco[0]][coco[1]+1] == premiSimbol:
            PuntsPremi -= 1
            Punts += 1000
        matrix[coco[0]][coco[1]] = " "
        coco[1]+=1
        matrix[coco[0]][coco[1]] = "@"

```

Inicialització i bucle principal

S'inicialitzen variables com ara la posició inicial del personatge, el símbol del premi, els punts del premi i els punts totals.

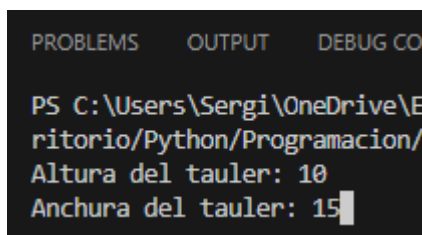
S'utilitza la biblioteca keyboard per detectar les tecles pressionades i fer accions corresponents.

El bucle principal actualitza el tauler en funció de les accions del jugador fins que es compleix una condició de finalització (Terminar).

```
while Terminar:
    time.sleep(0.05)
    if keyboard.is_pressed('w') or keyboard.is_pressed('W'):
        moureAmunt()
        system("cls") #Això es per esborrar l'anterior taula
        terminar(fons)
    if keyboard.is_pressed('a') or keyboard.is_pressed('A'):
        moureEsquerra()
        system("cls")
        terminar(fons)
    if keyboard.is_pressed('s') or keyboard.is_pressed('S'):
        moureAvall()
        system("cls")
        terminar(fons)
    if keyboard.is_pressed('d') or keyboard.is_pressed('D'):
        moureDreta()
        system("cls")
        terminar(fons)
    if keyboard.is_pressed('q') or keyboard.is_pressed('Q'):
        Terminar = True
```

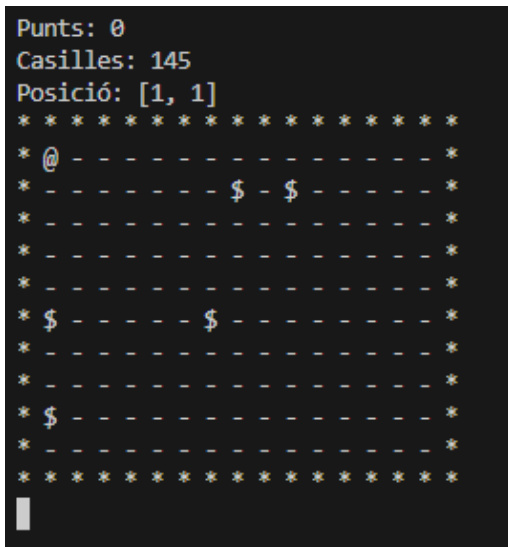
Joc de proves

Comença el joc preguntant l'altura i l'amplada del tauler, si no poses números, el programa et dona un error

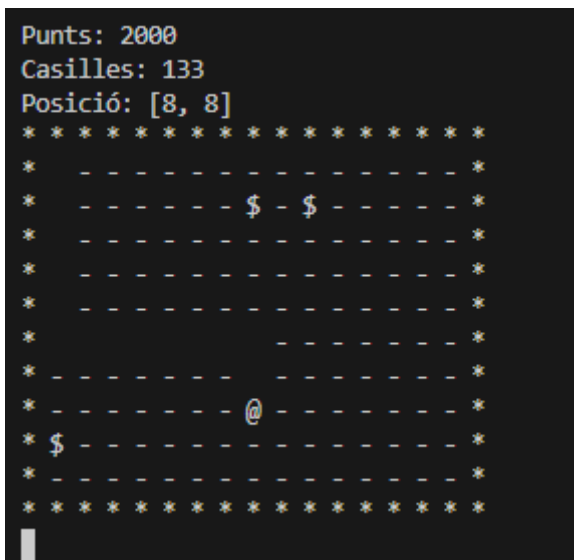


```
PROBLEMS  OUTPUT  DEBUG CO
PS C:\Users\Sergi\OneDrive\Escritorio\Python\Programacion\
Altura del tauler: 10
Anchura del tauler: 15
```

I es genera una taula amb les proporcions que has escollit



Cada vegada que et mous, el coco es mou, menjant-se el fons i els premis. Cada vegada que et menges un premi, el suma 1000 punts



Si et xoques amb una paret, et surt un missatge de “No pots passar” i el congela la pantalla 0.3 segons

```
Punts: 2000
Casilles: 124
Posició: [10, 1]
* * * * *
* - - - - -
* - - - - $ - $ - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* $ - - - - - - - - -
* @ - - - - - - - - -
* * * * *
NO POTS PASSAR
█
```

I una vegada que et menjes tots els premis, termines el joc i es mostra la posició on hi ets i el temps que has tardar

```
Punts: 5000
Casilles: 114
Posició: [2, 10]
* * * * *
* - - - - -
* - - - - @ - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* - - - - - - - - -
* * * * *
Felicitats, has terminat el joc
Temps: 219.19 segons
```

Conclusions

El joc m'ha semblat divertit i un bon repte, però al meu codi l'hi falten moltes millores, com per exemple millorar el codi per estalviar línies, implementar un fantasma, implementar més premis en cas de que el mapa sigui més gran o implementar parets que compliquin el moviment del coco. També un sistema de vides com el Pac-Man original, entre altres coses.