

JOC DEL PENJAT

Ronda: 1 / 3

```
| |
| |
| o
| /\
| /\
```

Lletres fallades:

Paraula: _____

Introdueix un caràcter:

```
| |
| |
| |
| |
| |
```

Lletres fallades: k i e t y u q w v

Paraula: ma_arr_n_

Introdueix un caràcter: b

Índex:

- Descripció
- Disseny del sistema o Implementació
- Funcionalitats principals
- Validació i joc de proves
- Conclusions

Descripció:

El propòsit d'aquest programa està purament centrat en l'entreteniment de l'usuari.

I la seva funció és la de poder jugar al joc del penjat.

El joc de penjat és un joc en el qual hi ha una paraula oculta, que tu has d'intentar desxifrar proposant lletres, si la lletra forma part de la paraula aquesta lletra serà visible dins la paraula, però si no es troba dins es contarà com un error.

L'objectiu és aconseguir desxifrar per complet la paraula sense perdre totes les vides (vides que estan representades mitjançant un ninot).

Disseny del sistema o Implementació:

Pel que fa a la forma en la qual he programat el joc:

- 1- He començat amb una salutació a l'usuari i la declaració de la variable «hp»(vida)

```
print()
print("Ara jugarem al joc del penjat!")
print()

hp = 0
```

- 2- He continuat tant declarant la variable «wordGuessed»(paraula encertada) com creant la llista «wordList»(llista de paraules), també he afegit un selector de dificultat que permet a l'usuari seleccionar una de les tres dificultats predeterminades.

```
wordGuessed = False

#Llista de possibles paraules
wordList = ["campana", "ampolla", "portatil", "quelcom", "temor", "blau", "fort", "dinar", "macarrons", "llibre"]

while True:
    difficulty = input("Dificultats:
    - Dificultat 1: 10 vides
    - Dificultat 2: 6 vides
    - Dificultat 3: 3 vides
    Introdueix la dificultat desitjada [1 / 2 / 3]: ")

    if difficulty == "1":
        hp = 10
    elif difficulty == "2":
        hp = 6
    elif difficulty == "3":
        hp = 3
    else:
        print("Dificultat no trovada")
        continue
    break
```

3- A continuació creo un bucle «for» que es repeteix 3 cops (ja que hi ha 3 rondes) i també imprimeixo per consola la ronda en la qual es troba l'usuari.

```
for i in range(3):
    print()
    print(f"Ronda: {i + 1} / 3")
    print()
```

4- Declaro dos variables, «currentWord»(paraula actual), que agafa com a valor una de les paraules de la llista seleccionada aleatòriament, i «updatedWord» (paraula actualitzada), que serà la paraula que es mostri per consola, i el seu valor és el de la longitud de la «currentWord» multiplicat per «_», el qual fa que tots els caràcters es converteixin en «_». També trec una paraula de la llista per tal que no es repeteixi en futures rondes. A part que declaro una llista buida en la qual hi haurà totes les lletres fallades («wrongChars»(caràcters incorrectes)).

```
currentWord = wordList[randint(0, 9)] #Aquí està la paraula que s'ha d'encertar
wordList.remove(currentWord)
updatedWord = "_" * len(currentWord) #Aquí està la paraula que es mostra en consola
wrongChars = [] #Llista on es guarden els caràcters fallats
```

5- Creo un bucle «while» que s'anirà repetint fins que s'acabin les vides.

A més a més, utilitzo un parell de funcions: «ninotPenjat» que mostra per consola l'estat del ninot en base les vides, i «lletresFallats» que ensenya totes les lletres fallades en la ronda actual.

Després imprimeixo l'estat de la paraula i declaro un «if» que quan la paraula per desxifrar està totalment desxifrada trenqui el bucle (la paraula ja està encertada)

```
while hp > 0:
    ninotPenjat(hp, difficulty)
    lletresFallats(wrongChars)

    print("Paraula: " + updatedWord)

    if currentWord == updatedWord:
        break
```

6- Demano la lletra mitjançant un bucle per tal d'evitar errors.

```
#Demana input per al valor de currentChar i no deix introduir valors que no siguin UNA lletra
while True:
    print()
    currentChar = input("Introdueix un caràcter: ").lower()
    print()
    if currentChar.isalpha() and len(currentChar) == 1:
        break
    else:
        print("Ha de ser UNA lletra")
```

7- Utilitzant una cadena de «if» i «elif» comprovo si la lletra indicada està en la paraula o si és una lletra ja utilitzada.
Si la paraula conté la lletra indicada, utilitza una funció per a actualitzar la paraula que es mostra en pantalla, fent que la lletra encertada es torni visible.
Si no la conté, afegeix la lletra a la llista de lletres fallades i resta 1 de vida.
I si ja l'has fet servir, s'imprimeix un missatge i continua el codi sense alteracions.

```
#Comprova si la paraula a encertar conté el caràcter introduït.  
# Si està actualitza la paraula, si ja has posat el caràcter abans no fa res  
if stringConte(currentWord, currentChar):  
    updatedWord = paraulaActual(currentWord, updatedWord, currentChar)  
elif currentChar in wrongChars or stringConte(updatedWord, currentChar):  
    print("Lletra ja utilitzada")  
else:  
    wrongChars.append(currentChar)  
    hp -= 1
```

8- Finalitza ensenyat l'estat definitiu del ninot i imprimint un missatge final.

```
ninotPenjat(hp, difficulty)  
missatgeFinal(hp, currentWord)
```

Des del meu punt de vista la part més complicada del codi és en la que s'ha d'agafar la lletra introduïda i substituir-la en la paraula que es mostra per pantalla.
Jo ho he fet amb una funció, en la qual es declara una nova variable, es recorre la cadena, i si la lletra no és la indicada, s'afegeix una «_» a la variable, però si ho és s'afegeix la lletra, i al final es retorna aquesta variable nova.

```
def paraulaActual(word, updatedWord, char):  
    newWord = ""  
    for i in range(len(word)):  
        if word[i] != char and updatedWord[i] == "_":  
            newWord += "_"  
        else:  
            newWord += word[i]  
    return newWord
```

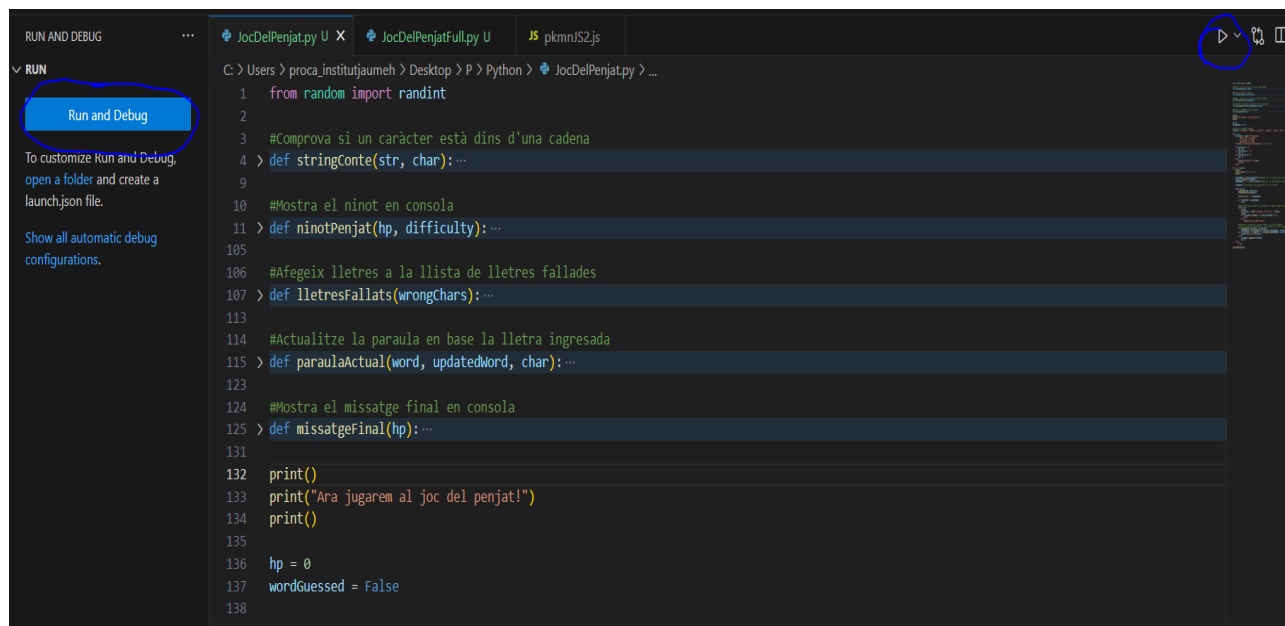
Funcionalitats principals:

Com a funció principal es troba evidentment la de poder jugar al joc del penjat, i dins d'aquesta funció principal hi ha alguna altra funció com la de poder seleccionar la dificultat.

Validació i joc de proves:

La forma base d'utilitzar aquest programa es basaria en:

- 1 - Obrir el programa o donar a compilar el codi.



Un cop amb el codi obert es pot pitjar:

- Damunt del triangle
- Damunt de «Run and Debug»
- F5 en el teclat

Això compilarà el codi.

- 2 - Seleccionar la dificultat.

```
Ara jugarem al joc del penjat!

Difícultats:
- Dificultat 1: 10 vides
- Dificultat 2: 6 vides
- Dificultat 3: 3 vides
Introdueix la dificultat desitjada [1 / 2 / 3]:
```

Es selecciona la dificultat indicant en consola una de les possibles opcions

- 3 – Comença a jugar!

```
Ronda: 1 / 3

|  |
|  o
| /|\
| /\

Lletres fallades:
Paraula: _____

Introdueix un caràcter:
```

S'introdueix un caràcter tractant d'encertar un caràcter de la paraula oculta.

A mesura que es va jugant es fallaran caràcters, que es mostraran en una llista. Però també s'encertaran, i aquests encertats es col·locaran en la paraula oculta.

```
| |  
| |  
| o  
| /|  
|  
  
Lletres fallades: s c v  
Paraula: __a_
```

Es guanya si s'encerten les paraules de totes les rondes sense que el ninot desaparegui.

Pers si el ninot desapareix abans que s'encertin totes les paraules es perd.

```
| |  
| |  
| |  
  
La paraula era: portatil  
Has perdut, t'has quedat sense vides!
```

Conclusions:

Aquest programa té una funció molt bàsica i opino que actualment la fa de sobres, ja que et permet jugar al joc de forma prou còmoda i sense cap error. Però això no impedeix que hi ha unes possibles millores com:

- Creació d'una interfície gràfica.
- Optimització i neteja en el codi.
- Millora en la claredat de la informació rebuda via consola.