



DOCUMENTACIÓN AHORCADO

1r DAM M03



PROGRAMACIÓ
Sergio Rey García
Joan Dalmau

Descripció

De que tracta el projecte

El projecte tracta de recrear el joc "El Ahorcado" amb el llenguatge de programació Python.

Les regles que havíem de seguir principalment eren les següents:

- La paraula secreta s'ha d'agarrar aleatòriament d'una llista.
- El jugador té 1 sola vida per resoldre les 3 paraules ocultes.
- Per cada fallada, el joc ha de mostrar un dibuix de l'estalvi.
- Hi ha d'haver tres nivells de dificultat. Els nivells de dificultat afecten els intents del jugador, reduint la possibilitat de trobar la paraula oculta.
- La paraula oculta s'ha d'anar mostrant per cada lletra que és encertada.
- No es poden fer servir funcions associades a cadenes de caràcter.

Disseny i funcionament del programa

Com està estructurat

```
import random # Importo la librería Random para escoger una palabra aleatoria
dificultad = 1 #Nivel de dificultad
#Creo la lista de palabras (ChatGPT y palabras mías)
listaPalabras = ["CIELO", "ENSALADA", "SUEÑO", "LIBRO", "MUSICA", "VIDA", "MUERTE", "LUZ", "OSCURIDAD", "ESPERANZA", "
Terminar = False
booleano = True #Creo un booleano que usaré siempre que lo necesite
victoria = 0 #Contador de Victoria
cambiarDificultad = True

while booleano:
    print('¿Tu teclado tiene la letra "Ñ"?') #Preguntamos si el teclado tiene la tecla "Ñ"
    print("1 - SI")
    print("2 - NO")
    letraEspecial = input('Respuesta: ')
    if letraEspecial == "1" or letraEspecial == "2":
        booleano = False
        print("")
    else:
        print("")
        print("Solo respuestas de 1 o 2")
    intentos = 10 #Número de intentos que tienes
```

Començo important la llibreria "random" per escollir la paraula, ajustant el nivell de dificultat i creant les variables principals com la de victòria o terminar.

Després, pregunto si el teclat té la lletra "Ñ", per si algú té un teclat anglès.

```
#Creo al muñeco que tendrá 5 de altura y 8 de ancho
altura1 = ["", "", "", "", "", "", "", ""]
altura2 = ["", "", "", "", "", "", "", ""]
altura3 = ["", "", "", "", "", "", "", ""]
altura4 = ["", "", "", "", "", "", "", ""]
altura5 = ["", "", "", "", "", "", "", ""]

#La intención es que se vea así
#  _____ 1
# |         0 2
# |      /|\ 3
# |         | 4
# |      / \ 5
# 12345678
```

Després, creo el ninot.

Jo el ninot he decidit fer-ho amb llistes. Tinc companys que ho han fet amb “print()”, però al moment de crear el codi no sem va acudir i vaig continuar fer-ho amb llistes.

```
while cambiarDificultad == True: #Pregunto a ver que opción elije
    print("Escoje tu nivel de dificultad")
    print('1 - dificultad "fácil"')
    print('2 - dificultad "normal"')
    print('3 - dificultad "difícil"')
    preguntar = input("Escoje: ")
    if preguntar == "1":
        dificultad = 1
        cambiarDificultad = False
        continue
    elif preguntar == "2":
        dificultad = 2
        cambiarDificultad = False
    elif preguntar == "3":
        dificultad = 3
        cambiarDificultad = False
    else:
        print("Solo puedes seleccionar 1, 2 o 3")
        print("")
if dificultad == 1: intentos = 10 #Número de intentos que tienes dependiendo el nivel de dificultad
elif dificultad == 2: intentos = 6 #Número de intentos que tienes dependiendo el nivel de dificultad
elif dificultad == 3: intentos = 3 #Número de intentos que tienes dependiendo el nivel de dificultad
```

En aquesta part del codi, faig un bucle per preguntar la dificultat que la persona vol. Depenent de la dificultat, el número d'intents canvia.

En aquesta part del codi i en moltes altres mes, no faria falta un bucle per el funcionament del codi, però jo volia fer un “codi sense errors”, es a dir, un codi que no es pugues petar de ninguna forma.

```

while Terminar == False:
    #IMPORTANTE: Todas las variables que estoy creando aquí dentro es porque cada vez que se reinicie el juego, se det
    palabraAcertada = False #Creo una variable para saber si ha acertado la palabra y terminar el juego
    repetirPalabra = True #Comprueba el tamaño de la palabra respecto la dificultad del juego
    muestraPalabra = [] #Para que se muestre los _ en el juego
    letrasUsadas = [] #Para mostrar las letras usadas que estaban erróneas
    accionMenu = "" #Para escoger la acción del menú
    preguntaFin = False #Pregunta al final del juego que quieres hacer

    #Para que se compruebe a ver si tiene "Ñ"
    while repetirPalabra == True:
        palabra = (random.choice(listaPalabras)) #Escojo la palabra aleatoriamente
        rangoPalabra = len(palabra) #Para saber el rango de caracteres de la palabra
        if letraEspecial == "2":
            repetirPalabra = False
            for i in palabra:
                if i == "Ñ":
                    repetirPalabra = True
            elif letraEspecial == "1":
                repetirPalabra = False
            if repetirPalabra == False:
                for i in palabra:
                    muestraPalabra += ["_"] #Para añadir los '_' de la palabra

```

En aquesta part del codi, escolló la paraula aleatòria i corroboro si té la lletra “Ñ” per si la persona no té aquesta lletra.

Després, poso la llargada de la paraula en una llista ple de “_” per ocultar la paraula i que es mostri oculta.

```

while palabraAcertada == False:
    booleano = True #Para que siempre que se reinicie el juego, se active como True
    letraAcertada = 0 #Creo una variable que cuenta las letras acertadas en una misma palabra. Ejemplo: Letra que
    vuelta = -1 #Creo una variable para contar las iteraciones que hace al comprobar la palabra y saber la posición
    listaVuelta = [] #Creo la lista para saber las posiciones
    if intentos == 1: #Creo un if para que el texto esté bien y coincida con el número
        print(f"Tienes {intentos} intento")
    elif intentos == 0:
        print("Intentos acabados, has perdido")
        break
    else:
        print(f"Tienes {intentos} intentos")
    #Recorro la lista y elimino los corchetes, comillas, etc
    print("Palabra secreta: ", end="")
    for i in muestraPalabra:
        if i != "[" and i != "]" and i != "'":
            print(i, end=" ") #Hago que se muestre la palabra final
    print("") #Espacio en blanco para saltar el último end que hay
    print("") #Espacio en blanco para un salto de línea completo
    print("Letras erróneas: ", end="")
    for i in letrasUsadas:
        if i != "[" and i != "]" and i != "'":
            print(i, end=" ") #Hago que se muestre la palabra final
    print("") #Espacio en blanco para saltar el último end que hay

```

Aquí comença el lloc.

El primer que faig es mostrar els intents que tens, seguidament mostro la paraula secreta eliminant el “[”, “]” i “’”

```
#Pregunto la letra y corroboro que sea una letra y no otro caracter
while boleano: #Mientras sea True
    letra = input("Ingresa un caracter: ")[0:1] #Ingresamos el caracter y obligo al programa a que escoja solo
    if letra == "": #Si el usuario envia la letra vacia, que la vuelva a pedir
        print("No se permiten parametros vacios")
        continue
    corroborarLetra = int(ord(letra)) #Calculo si lo que ha ingresado es una letra
    if (corroborarLetra >= 97 and corroborarLetra <= 122) or letra == "ñ": #Si es minuscula o la letra ñ, la
        letra = letra.upper()
        boleano = False
    elif (corroborarLetra >= 65 and corroborarLetra <= 90) or letra == "Ñ": #Si es mayuscula o la letra Ñ, que
        boleano = False
    else: #Si el usuario no me hace caso, que vuelva a pedir los datos
        print("Solo letras sin tildes")
        continue
```

Pregunto la lletra que vol ingressar i nomes em quedo amb la primera lletra ingressada, per si l'usuari ha premut dues tecles, nomes emmagatzemar la primera. Corroboro que sigui una lletra i no un caràcter amb l'exercici que vam fer del codi ascii i continuo si està tot bé.

```
#Compruebo si la letra está en la palabra
for i in palabra:
    vuelta += 1 #Sumo la iteración para saber en que posición de la palabra me encuentro
    if letra == i: #Si la letra introducida coincide con una letra de la palabra
        letraAcertada += 1 #Sumo la letra acertada para saber cuantas letras bien tengo
        listaVuelta += [vuelta] #Apunto las posiciones donde está la letra en la palabra
    else:
        continue
if letraAcertada == 0 and letra not in letrasUsadas: #Corroboro que sea una letra que no haya sido acertada ni
    letrasUsadas += [letra] #Apunto la letra usada que no estaba en la palabra (si es que hay)
    intentos -= 1
```

Comprovo si la lletra la lletra hi es en la paraula amb el bucle, recorrent la paraula lletra a lletra i apuntant la posició on hi es.

Si la lletra no hi es en la paraula i no hi es en la llista de lletres fallades, que s'apunti a la llista i et redueixi un intent per fallar.

```
#Ahora dibujo al monigote:
if dificultad == 1:
    if intentos == 9:
        altura4.pop(0), altura4.insert(0,"|"), altura5.pop(0), altura5.insert(0,"|")
    elif intentos == 8:
        altura2.pop(0), altura2.insert(0,"|"), altura3.pop(0), altura3.insert(0,"|")
    elif intentos == 7:
        altura1.pop(0), altura1.insert(0,"_"), altura1.pop(1), altura1.insert(0,"_"), altura1.pop(2), altura1.insert(2,"_")
    elif intentos == 6:
        altura2.pop(5), altura2.insert(5,"0")
    elif intentos == 5:
        altura3.pop(5), altura3.insert(5,"|")
    elif intentos == 4:
        altura4.pop(5), altura4.insert(5,"|")
    elif intentos == 3:
        altura3.pop(5), altura3.insert(5,""), altura3.pop(4), altura3.insert(4,"|"), altura3.insert(4,"/")
    elif intentos == 2:
        altura3.pop(5), altura3.insert(5,"\\")
    elif intentos == 1:
        altura5.pop(4), altura5.insert(4,"/")
    elif intentos == 0:
        altura5.pop(5), altura5.insert(5,"\\")
```

Després dibuixo el ninot afegint i eliminant posicions a la llista que he creat abans de començar el joc. El ninot es construeix mes ràpid o mes lent depenent del nivell de dificultat.

```

for i in altura1:
    if i != "[" and i != "]" and i != "'":
        print(i,end=" ")
print("")
for i in altura2:
    if i != "[" and i != "]" and i != "'":
        print(i,end=" ")
print("")
for i in altura3:
    if i != "[" and i != "]" and i != "'":
        print(i,end=" ")
print("")
for i in altura4:
    if i != "[" and i != "]" and i != "'":
        print(i,end=" ")
print("")
for i in altura5:
    if i != "[" and i != "]" and i != "'":
        print(i,end=" ")
print("")

```

Després dibuixo el ninot amb bucles que recorren la llista i no afegeixen les claus o l'apòstrof.

```

if letraAcertada > 0: #Este if es solo para ahorrar calculos a la CPU, si lo quitase no pasaría nada
    for i in range(letraAcertada): #Elimino la posición de la letra acertada y la añado en el mismo lugar con
        muestraPalabra.pop(listaVuelta[i])
        muestraPalabra.insert(listaVuelta[i],letra)

if "_" not in muestraPalabra: #Compruebo si la palabra se ha encontrado o no
    palabraAcertada = True
    print("")
    print("Palabra secreta: ", end="")
    for i in muestraPalabra:
        if i != "[" and i != "]" and i != "'":
            print(i,end=" ") #Hago que se muestre la palabra final completa
    print("")
    print("")

```

I al final del joc elimino la posició de la llista de la paraula on hi es les "_", canviant-les per la lletra encertada. La "letraAcertada" es la llista que es guardava les posicions de la paraula encertada.

Després corroboro si la paraula s'ha encertat, fent un "if "_" not in muestraPalabra:". Si s'ha encertat, el joc continua, sinó, el bucle des de preguntar la lletra es repeteix.

```
#Acabo el juego
#
if palabraAcertada == True:
    print("¡Felicidades! Has logrado encontrar la palabra secreta ^.^")
    victoria += 1
else:
    print(f"La palabra secreta era: {palabra}")
    Terminar = True

if victoria == 3:
    Terminar = True
```

Per últim, al final del codi, comprovo si a ha encertat o si les ha fallat totes. Per això, utilitzo “palabraAcertada” que activo només quan s’encerta la paraula. També faig que si has guanyat tres vegades, que el joc termini.

Problemes al codi

Els problemes que m’he trobat

A l’hora de fer el programa, he tingut la sort de tindre pocs problemes, però el que podria dir que es el més complex va ser crear el ninot.

Com a l’hora de crear el ninot, jo ho vaig fer amb tables, vaig tindre un error de que el cos no estigues a la seva posició. Es a dir, a partir de “x” fallo, el cos del ninot es desplaçava a la dreta o a l’esquerra, i això feia que no sortís el ninot bé.

Per solucionar-ho, vaig tenir que fer prova i error, intent rere intent per veure l’evolució del ninot. Al final ho vaig solucionar eliminant la posició on hi havia l’error i posant-la a la dreta o esquerra

Exemple:

```
elif intentos == 5:
    altura3.pop(5), altura3.insert(5,"|")
elif intentos == 4:
    altura4.pop(5), altura4.insert(5,"|")
elif intentos == 3:
    altura3.pop(5), altura3.insert(5,""),
```

Al “elif intentos == 5:” elimino la posició 5 de l’altura 3 i insereixo la columna vertebral del cos del ninot, i a l’“elif intentos == 3:” elimino la mateixa posició de la mateixa taula per inserir espai. Mes endavant en el codi, insereixo la columna al ninot a la posició 4

```
altura3.pop(4), altura3.insert(4,"|"),
```

Joc de proves

Com jugar i provocar errors

```
¿Tu teclado tiene la letra "Ñ"?  
1 - SI  
2 - NO  
Respuesta: 
```

El joc comença preguntant si el teu teclat te la lletra "Ñ". Si poses alguna resposta que no deus posar, com per exemple un caràcter, altre número o un espai, succeeix això

```
Solo respuestas de 1 o 2  
¿Tu teclado tiene la letra "Ñ"?  
1 - SI  
2 - NO  
Respuesta: 
```

I et torna a preguntar, ja que està dins d'un bucle i només es permet el valor "1" o "2" que està en format string.

```
Escoje tu nivel de dificultad  
1 - dificultad "fácil"  
2 - dificultad "normal"  
3 - dificultad "difícil"  
Escoje: 
```

Després ens pregunta el nivell de dificultat. Si tornem a posar altre cosa que no es vàlida, surt aquest missatge

```
Solo puedes seleccionar 1, 2 o 3  
  
Escoje tu nivel de dificultad  
1 - dificultad "fácil"  
2 - dificultad "normal"  
3 - dificultad "difícil"  
Escoje: 
```

Un altre vegada, això són valors en strings i només es pot permetre una d'aquestes opcions.


```
Escoje: 1
Tienes 10 intentos
Palabra secreta: _ _ _ _ _
Letras erroneas:
Ingrese un caracter: 
```

Continua el lloc, si tornem a intentar petar-ho surt això

```
Ingrese un caracter: 4
Solo letras sin tildes
Ingrese un caracter:
No se permiten parametros vacíos
Ingrese un caracter: 
```

I si provem a posar una paraula sencera, passa això

```
Ingrese un caracter: abecedario

Tienes 10 intentos
Palabra secreta: _ _ _ _ _ A
Letras erroneas:
Ingrese un caracter: 
```

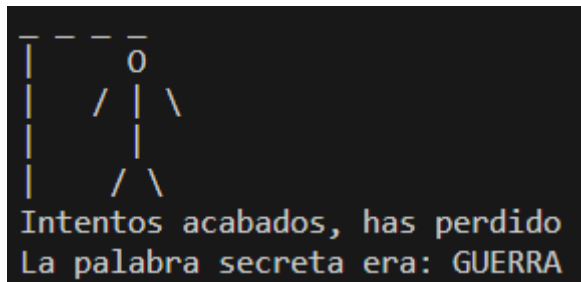
Nomes es queda amb la primera lletra de tota la paraula. En cas de posar una paraula que comenci amb un caràcter especial, em sortirà "Solo letras sin tildes"

Si fallo una lletra, passa això

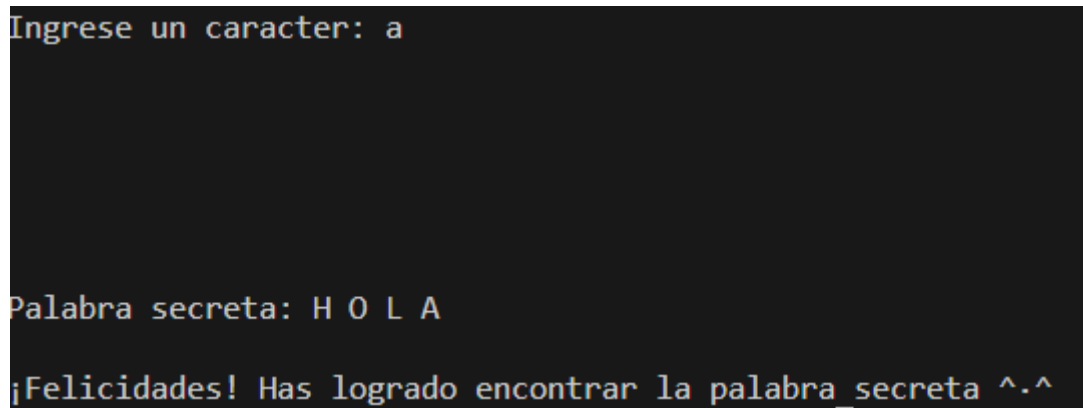
```
Ingrese un caracter: ñ

|
|
Tienes 9 intentos
Palabra secreta: _ _ _ _ _ A
Letras erroneas: Ñ
Ingrese un caracter: 
```

Així fins perdre



En cas de guanyar, passa això (he quitat l'aleatori per poder mostrar-ho)



Conclusions

Millores que posaria i que he après

El treball m’ha agradat molt i em sembla molt divertit i útil per aprendre altres formes de fer les coses, a mes a mes de estructurar el codi i aprendre a comentar les coses. Sobre tot, ha sigut útil per refrescar codi i aprendre al cent per cent com utilitzar les diferents opcions que he posat, com per exemple el “.pop” i “.insert”, o el “not in”, entre altres.

Hi han moltes coses per millorar, per exemple els comentaris del codi, la seva estructuració o la seva optimització. També podria millorar la seva interfície o el seu us.