2022

Trabajo práctico introducción a la programación

ALUMNO:
JULIAN WAKSMANN
DNI: 42279583

Introducción a la programación

Trabajo práctico

1. Introducción:

El juego consiste en un separador de silabas, al cual le aplicamos ciertas modificaciones, llenando algunas funciones pre definidas, cambiando ciertos parámetros y armando nuevas funciones con el fin de que el programa funcione y resulte mas interesante. Nos inspiramos en el juego televisivo “los 8 escalones” para su nombre/temática.

2. Funciones Vacías:

En principio, vale aclarar que no se utilizaron todas las funciones para completar el programa. Generamos una nueva función, llamada espacioToGuion, la cual si, nos ayudo al funcionamiento del juego.

Función lectura:

```
def lectura(archivo, salida):
    seleccion= archivo.readline()

    while seleccion != "":
        salida.append(seleccion[:-1])
        seleccion = archivo.readline()
```

Básicamente lo que expresamos aquí es que mientras haya palabras, se ejecute ese ciclo while. El -1 que esta dentro de los corchetes es para que se agreguen las palabras del lemario, pero sin el ultimo carácter (el enter). Y por último expresamos que lea la siguiente línea.

Función nuevaPalabra:

```
def nuevaPalabra(lista):
    return random.choice(lista)
```

Para realizar esta función, ocupamos una función pre armada de la librería random, llamada choice, la cual selecciona aleatoriamente un elemento de una lista.

Función palabraTOsilaba:

```
def palabraTOsilaba(palabra):
    return fs.separador(palabra)
```

Esta función ya estaba hecha, por lo que decidimos importar el archivo py donde se encontraba y le agregamos un alias para que sea más simple llamarla.

Función espacioToGuion:

Antes de continuar con las demás funciones vacías, nos parece sugerente mostrar la función que armamos, debido a que hicimos de su uso en la función posterior.

Básicamente, lo que hace esta función es cambiar el carácter espacio " " por el carácter guion "-".

```
def espacioToGuion (palabraseparadaensilabas):
    nueva_separada= ""
    for caracter in palabraseparadaensilabas:
        if caracter != " ":
            nueva_separada = nueva_separada + caracter
        else:
            nueva_separada = nueva_separada + "-"
    return nueva_separada
```

Generamos una variable que acumula cada carácter recorrido por el bucle for, de manera que, si el carácter no es un espacio la acumule y que si es un espacio me acumule un guion.

Función esCorrecta:

```
def esCorrecta(palabraEnSilabasEnPantalla, palabra):
    if fce.espacioToGuion(palabra) == palabraEnSilabasEnPantalla:
        return True
    return False
```

Sabiendo que "palabraEnSilabasEnPantalla" es la palabra que esta separada correctamente en silabas, por la función palabraTOsilaba, y "palabra" es la palabra que ingresa el usuario, optamos por aplicarle la función espacioToGuion a esta última, debido a que la función palabraTOsilaba retorna la palabra separada en silabas con guiones y para poder ejecutar la comparación de manera optima utilizamos esta técnica. Entonces si son iguales nos va a retornar verdadero, por el contrario, si no lo son, los retornara Falso.

Función puntaje:

```python
def puntaje(palabra):
    if len(palabra) > 9:
        return 5
    elif len(palabra) > 5:
        return 3
    return 2
```

Para complejizar un poco el juego, nos pareció interesante generar esta función en base a la cantidad de caracteres que tiene la palabra a separar.

Cabe recalcar que tuvimos que modificar ciertos parámetros del programa principal a partir de la línea 71:

```python
if esCorrecta(palabraEnPantallaEnSilabas, palabra):
    puntos += puntaje(palabraEnPantalla)

    pygame.mixer.music.load('audio_correcto.wav')
    pygame.mixer.music.play(1)
    pygame.mixer.music.set_volume(0.3)
```

Anteriormente a esta modificación, solo se sumaba de a 5 cada vez que la palabra estaba bien separada. Con esta nueva función, ya no es constante, si la palabra tiene un len mayor a 9, te sumara 5. Si tiene un len entre 9 y 6, te sumara 3 y por último si tiene un len menor o igual a 5, sumara 2.

3. AGREGADOS

Letra ñ:

```python
elif key == 241:
    return("ñ")
```

Encontramos esta funcionalidad mediante la documentación de la librería pygame https://www.pygame.org/docs/ref/key.html .La misma nos retorna el código al que está asociado cierto carácter:

```
    print('el codigo de la letra ñ es :', pg.key.key_code('ñ'))
 ✓ 0.5s
el codigo de la letra ñ es : 241
```

Aplicamos esa lógica en el archivo de extras y esto nos permitió poder usar ñ en el juego.

Imagen de fondo:

Utilizando nuevamente la documentación y realizando una investigación, imaginamos que seria una buena idea, continuar con la temática y establecer una imagen de fondo de pantalla asociada al título del juego.

(línea 100)

```
#Fondo del juego
fondo_juego = pygame.image.load("fondo_escalon.png").convert()
screen.blit(fondo_juego, (0 ,0))
```

Mediante image.load cargamos la foto y la establecemos como una variable, para luego aplicar la función blit a la pantalla y determinar que nuestra imagen será el fondo. Cabe recalcar que el (0,0) es para centrar la imagen.

Icono del juego:

Cambiamos el icono que viene por defecto, por uno que tenga relación con la temática, decidimos poner unos escalones utilizando la misma técnica de carga que la función anterior, pero usando la función set_icon para establecer la imagen como icono.

(línea 96)

```
#Icono del juego
icono= pygame.image.load("esc.png")
pygame.display.set_icon(icono)
```

Corrección del tiempo:

En la consigna del trabajo, se establece que el tiempo de duración sea de 60 segundos. En el juego por defecto se estableció en 120. Simplemente cambiamos el valor de la variable, ubicada en configuración.py, llamada TIEMPO_MAX por 60.

Sonido de fondo:

Relacionado con la victoria o perdida, agregamos estos parámetros dentro de las condiciones en donde se especifica si era correcta o falsa la respuesta:

```
if esCorrecta(palabraEnPantallaEnSilabas, palabra):
    puntos += puntaje(palabraEnPantalla)

    pygame.mixer.music.load('audio_correcto.wav')
    pygame.mixer.music.play(1)
    pygame.mixer.music.set_volume(0.3)

else:
    puntos-=1

    pygame.mixer.music.load('audio_incorrecto.wav')
    pygame.mixer.music.play(1)
    pygame.mixer.music.set_volume(0.3)
```

Utilizamos las funciones mixer.music.load para cargar el sonido, el play para iniciarlo con el parámetro en 1 para que solo lo haga 1 vez y por ultimo en un volumen al 30%

Acomodar palabra del usuario:

Centramos la palabra que el usuario ingresa.

(Línea 89 del archivo extras.py)

```
#muestra lo que escribe el jugador
screen.blit(defaultFont.render(palabraUsuario, 1, COLOR_TEXTO), (590, 570))
```

Modificación del ancho y alto:

Realizamos una modificación en los valores de las variables ancho y alto del archivo configuración.py para que se vea más grande la pantalla.

4. Conclusión / Trabas:

Es un trabajo que nos ayudó muchísimo a poner en práctica todo lo visto a lo largo de la cursada y afianzar conocimientos. También nos sirvió para aprender a investigar, vía internet, las trabas que se nos presentan.

```
print('el codigo de la letra á es :', pg.key.key_code('á'))
print('el codigo de la letra é es :', pg.key.key_code('é'))
print('el codigo de la letra í es :', pg.key.key_code('í'))
print('el codigo de la letra ó es :', pg.key.key_code('ó'))
print('el codigo de la letra ú es :', pg.key.key_code('ú'))
```

✓ 0.4s

```
el codigo de la letra á es : 225
el codigo de la letra é es : 233
el codigo de la letra í es : 237
el codigo de la letra ó es : 243
el codigo de la letra ú es : 250
```

Como complicación, no pudimos ejecutar la misma técnica que aplicamos para poder utilizar la ñ en el programa, con las letras que llevan tilde:

```
elif key == 225:
    return("á")
elif key == 233:
    return("é")
elif key == 237:
    return("í")
elif key == 243:
    return("ó")
elif key == 250:
    return("ú")
else:
    return("")
```

Agregando estos códigos en el archivo de extras.

Pero no nos fue de utilidad, debido a que, no funciona de esta manera.

Por otro lado, como fuentes de información nos gustaría brindar las 3 que utilizamos:

1. https://www.pygame.org/news
2. https://stackoverflow.com/
3. https://www.youtube.com/playlist?list=PLVzwufPir356RMxSsOccc38jmxfxqfBdp