



Introducción a la Programación – COM 01

Trabajo Practico

Primer Semestre - 2022

Integrantes:

- Acuña Rodrigo
- Agüero Enzo

Docentes:

- Santiago Montiel
- Nora Martínez
- Jorgelina Rial



Explicación del proyecto

El trabajo consiste en realizar un juego, en el cual la funcionalidad principal es que el jugador separe en sílabas palabras otorgadas al azar durante un periodo de tiempo. Si lo que escribe el jugador coincide con la correcta separación de la palabra, este obtendrá puntos, en cambio, si la respuesta ingresada es errónea se le restará puntaje.

Desarrollo del videojuego

En primer lugar, descargamos los archivos base que se nos otorgan. Tenemos un **lemario**, y distintas funciones que nos ayudan a tener una idea de cómo armar los primeros pasos de nuestro juego.

Una vez leído y analizado el código básico, comenzamos a trabajar en la funcionalidad.

- La primera función a completar fue la de lectura, la cual recibe como parámetros **archivo** (Nuestro **archivo.txt** que contiene las palabras) y **salida** (Una lista que guardará las palabras).

Con el ciclo **for** recorreremos toda la lista de palabras en el archivo y las guardamos en otra lista que nos servirá para analizar la palabra en las próximas funciones.

```
def lectura (archivo, salida):
```

```
    for word in archivo:
```

```
        salida.append(word.rstrip('\n'))
```

- La siguiente función **nuevaPalabra** recibe como parámetro la lista de palabras que creamos con la primer función, luego en un variable guardamos una palabra elegida al azar de toda la lista.

```
def nuevaPalabra(lista):
```

```
    word= random.choice(lista)
```

```
    return word
```



- Esta función recibe como parámetro silaba (Palabra que ingresa el jugador desde el programa principal) y lo que hace es transformar los espacios en guiones. Lo que resulta muy conveniente para comprobar fácilmente si la palabra es correcta o no.

```
def silabasTOpalabra(silaba):
```

```
    silaba= silaba.replace(' ','-')
```

```
    return silaba
```

- Esta función junto con la anterior son las fundamentales para que nuestra funcionalidad. Lo que hace es recibir la palabra seleccionada aleatoriamente por la función nuevaPalabra y la separa en silabas utilizando la función separador.
- La función **separador**, se encarga de separar una palabra en silabas poniendo un guion (-) entre las separaciones.

```
def palabraTOsilaba(palabra):
```

```
    palabra = separador(palabra)
```

```
    return palabra
```



- Por último, estas dos funciones se encargan de chequear si la respuesta dada por el jugador es correcta, en caso de serlo se recibirá el puntaje asignado.

Para comparar si la respuesta es válida, la palabra ingresada por el usuario que puede estar separada por espacios o guiones (Si la palabra está separada con espacios estos serán reemplazados con guiones), tiene que coincidir con la palabra que se está mostrando en pantalla.

Recordemos que la palabra en pantalla está guardada en una lista separada en silabas por guiones, y la respuesta del jugador tiene que coincidir con ella.

```
def esCorrecta(palabraEnSilabasEnPantalla, palabra):  
    if (palabraEnSilabasEnPantalla== silabasTOpalabra(palabra)):  
        return True  
  
def puntaje(palabra):  
    if len(palabra) < 10:  
        return 5  
    return 10
```

Implementación de funciones

Una vez terminada la lógica de nuestro juego, comenzamos a trabajar en la parte visual.

Utilizando la librería PyGame y creando varias funciones, logramos mejorar la estética de nuestras ventanas.

Agregamos escenas, SFX, Botones, colores, y distintas fuentes de texto.



Ejemplos:

Tenemos esta función que nos sirve como una transición entre escenas, recibe como parámetros las dimensiones de nuestra pantalla y va haciendo la va pintando de negro.

```
def fade(ANCHO, ALTO):  
    fade = pygame.Surface((ANCHO, ALTO))  
    dt = gameClock.tick(60)/1000  
    fade.fill((COLOR_FONDO))  
    for alpha in range(0,300):  
        fade.set_alpha(alpha)  
        screen.blit(fade, (0,0))  
        pygame.display.update()  
        pygame.time.delay(5)
```

Esta otra funciona para hacer cambiar de color a un texto, toma como parámetro el color, y la dirección en la que se realizara el cambio

```
def color_change(color,direction):  
    for i in range(3):  
        color[i]+= color_speed * direction[i]  
        if color[i]>=maximum or color[i]<= minimum:  
            direction[i] *= -1
```



Conclusiones finales

Al principio presentamos algunas dificultades a la hora de la implementación de nuestras funciones, pero una vez que ya logramos entender nos resultó más sencillo el trabajar.

Aprendimos mucho durante el desarrollo del trabajo, nos dimos cuenta de la importancia del uso de funciones, variables globales y comentarios para que todo sea más simple de manejar y leer.

Además, logramos comprender algo de la lógica utilizada en los videojuegos. Esperamos poder realizar más trabajos de este tipo e ir ganando más conocimiento a lo largo de nuestro aprendizaje.