**Trabajo Práctico Grupal**

**Introducción a la programación**

**COM-08**

**Docentes: Flavia Bottino y Viviana Varela**

**Coordinador: Montiel Santiago**

**Informe del Equipo 10**

Integrante:

* **Lucas Bustamante**

**Introducción:**

El trabajo consiste en completar las funciones del archivo ‘funcionesVACIAS.py’, para poder iniciar el juego y que funcione correctamente. El juego consta de adivinar una palabra, arriesgando palabras y no letra por letra (como en un ahorcado). La longitud de la palabra ya esta previamente establecida y se tiene hasta 5 intentos para adivinarla o hasta que termine el tiempo (120 segundos).

Para ayudar al usuario a adivinar la palabra, a un costado de la ventana se verán las letras del abecedario (en color blanco) que, a partir de cada intento irán cambiando de color por cada palabra utilizada por el usuario. Por ejemplo:

* si la letra fue correcta en el lugar se marcará de color verde.
* sí la letra está en la palabra, pero no en el lugar correcto su color será azul.
* Si no está en la palabra su color será rojo.

**Lo que falta implementar:**

La función **nuevaPalabra (lista):**

recibe una lista de palabras y devuelve una de ellas elegida al azar.

La función **lectura (archivo, salida, largo):**

lee el archivo que ya fue abierto en principal y carga en salida sólo las palabras cuya longitud es el indicado en el tercer parámetro (largo).

La función **revisión (palabraCorrecta, palabra, correctas, incorrectas, casi):**

chequea la palabra ingresada por el usuario, carga las letras en la lista que corresponda y devuelve True en caso de que la palabra sea la correcta y False en caso contrario.

No permitir que el usuario ingrese palabras de longitud diferente a la palabra que se desea

adivinar.

No permitir que el usuario ingrese palabras que no estén en el lemario.

Mostrar las letras incorrectas, correctas parcialmente y correctas de diferentes colores.

Opcionales para agregar:

1. Si el usuario acierta la palabra aparece una nueva. El reto es acertar muchas

palabras en un lapso de tiempo.

1. Evita que el usuario ingrese más de una vez una palabra ya utilizada.
2. Al final el juego hay un cartel de Ganó o perdió indicando cuál era la palabra correcta
3. Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más

importantes: acierta al menos una letra, hay varios aciertos o todos errores, etc

1. Hacer que el usuario luego de jugar tenga la opción de ingresar su nombre y se

muestren los 10 mejores puntajes históricos con sus nombres.

1. Que el usuario tenga opciones de niveles donde se modifique el juego, por ejemplo:

* el usuario indica la longitud que deben tener las palabras.
* Temático.
* Se juega por tiempo, por cantidad de aciertos, etc.

Forma de entrega:

El TP se considera correctamente entregado y apto para ser corregido si se cumple

con una entrega impresa y una digital.

La defensa del TP es donde se muestra el correcto funcionamiento del programa.

Deben asistir a la misma todos los integrantes del grupo y es necesario que todos

los integrantes entiendan todo el trabajo realizado por todo el grupo.

La entrega digital consiste en enviar un e-mail de la siguiente manera:

1. A la cuenta de correo electrónico de los docentes de la comisión.

2. Con asunto: entrega- tp 2022 nombre − estudiantes. < nro − grupo >.

3. Adjuntando por separado un archivo comprimido con los archivos que

conforman el código fuente de todo el programa y un archivo con el informe

(preferentemente en formato pdf).

El código debe cargar sin errores. Si el trabajo es muy pesado recomendamos subirlo a google drive, dropbox, etc y pasarnos el link.

Desarrollo de las funciones:

Función **nuevaPalabra (lista):** para comenzar cree una variable llamada ‘newpalabra’ y dentro ingrese random.choice(lista) utilizando el parámetro lista, para que me de una palabra al azar de la lista y me la devuelva con un return.

Función **lectura(archivo,salida,largo):** comencé leyendo el archivo completo en una variable (text1)

con ‘readlines()’. Con un ciclo(for) recorrí cada palabra de la lista text1, en el identado utilicé un condicional para que cada palabra pasara por la función sacarEspacio (le sacara los ‘\n’) y las que tuvieran la longitud del parámetro largo, la agregaría a ‘salida’.

Para esta función tuve que crear la función **sacarEspacios(palabra):** inicio con una variable(palabra\_sin\_espacio) que le ingresé solo comillas (‘’) para luego recorrer con un ciclo(for) las letras de la palabra tomada del parámetro, en el identado utilizo un condicional para que cada letra tomada sea agregada a la variable(palabra\_sin\_espacio) solo si es distinta del elemento ‘\n’. Saliendo del ciclo(for) devuelve la variable(palabra\_sin\_espacio).

Función **revisión(palabraCorrecta,palabra,correctas,incorrectas,casi):** Para la misma empecé con un condicional que me devolviera un True si la palabraCorrecta era la misma que palabra(palabra dada por el usuario). Si no, lo que haría es iniciar un ciclo(for) que recorrería por índice la palabra que ingreso el usuario y compararía si las letras son iguales con un condicional, si la letra de palabra con índice[i] fuera igual a la letra de palabraCorrecta con índice[i] agregaría a la lista ‘correctas’ la letra. Si no, entonces preguntaría si la letra está en palabraCorrecta y la agregaría a la lista ‘casi’ y sino agregaría la letra a incorrectas, por último, el programa saldría del identado del condicional y del ciclo(for) y devolvería False.

En el programa principal (‘principal.py’) utilice la función **controlDePalabra (),** que hice para que antes de usar la función revisión (), me devolviera true si la palabra recién ingresada por el usuario perteneciera al lemario, tuviera la misma longitud que el parámetro ‘largo’, y no la haya escrito en un intento anterior. Si no lo hacía (devolvía un false), entonces devolvería false y le pediría al usuario que vuelva a escribir otra palabra.

En extras(‘extras,py’) al final de la función dibujo() en el ultimo ciclo(for) cambie el código agregando condicionales para que cada letra que estuviera al costado de la ventana se viera con diferente color en caso de no estar en ninguna lista(color blanco), en el caso de estar en correctas(verde), en casi (azul)y en incorrectas (rojo).

Para terminar los puntos opcionales realice el 1) y el 2), los demás no los realice por cuestión de organización y un poco de tiempo ya que lo hice solo al trabajo. Mas allá de la nota que me den el trabajo fue muy divertido de hacer y bastante cargado de frustraciones (por lo que lo hice solo). lo poco que aprendí mirando tutoriales de pygame, me sirvió para seguir motivado con la programación y para seguir intentando aprobar.

**Bibliografía:**

<https://www.pygame.org/wiki/tutorials>

<https://www.youtube.com/c/ProgramaciónFacilYT>

https://stackoverflow.com/questions/55563399/how-to-solve-unicodedecodeerror-utf-8-codec-cant-decode-byte-0xff-in-positio