Alumnos: Ezequiel Encinas, Gabriel Martinez, Gustavo Pintos
Introducción a la Programación - 1er Semestre 2023

Informe Trabajo Práctico Final

En el siguiente escrito desarrollaremos los procesos que llevamos a cabo para la realización de un videojuego, el cual consiste en: formar en sesenta segundos distintas palabras con siete letras que aparecen en pantalla, para sumar puntos.

Para empezar, partiremos de la base brindada por los profesores:cinco archivos tipo Python File (configuración, extras, funcionesVACIAS, lemario, principal). Nuestra tarea principal fue la de modificar las funciones principales dentro de “funcionesVACIAS.py” para que el juego funciones de acuerdo a los criterios presentados anteriormente.

En primer lugar, antes de empezar con las funciones principales, creimos necesario implementar las siguentes para usarlas mas de una vez:

- def eliminarSalto(cadena), devuelve la cadena sin saltos de linea (“\n”).
- def esta(cad,letra), para saber si la letra esta en la cadena, o elemento en lista.
- def cadenaALista(cadena), devuelve una lista que contiene c/u de las letras de cadena.

Al implementar las funciones principales modificadas, en un nuevo archivo llamado “funcionesCOMPLETAS.py”, tuvimos los siguientes inconvenientes y soluciones:

- Dos errores en def lectura(diccionario)
    - Primero al leer el lemario completo, en el intérprete saltaba:
      UnicodeDecodeError: 'utf8' codec can't decode byte 0xf1 in position 233: invalid continuation byte. La solución luego de consultarlo, fue de incorporar **encoding="ISO-8859-1”.** Ahora podía leer el lemario completo.
    - Segundo al cargar las letras a la lista “diccionario” la pantalla del juego quedaba en negro y no respondia. Luego de consultarlo, la función retornaba la misma lista diccionario. En ese caso de manejo de archivos **no hacia falta retornar** algo , porque la función lectura **solo agregaba elementos(de lemario.txt) a la lista vacía “diccionario”**. La solución fue eliminar la nueva lista que se creaba en def lectura.

- Error en def dameAlgunasCorrectas: la pantalla del juego quedaba en negro y no respondia. La solución luego de consultarlo fue la de modificar el código para que compruebe primero **si esta la LetraPrincipal en la palabra del lemario recorrida**.

- Error en def procesar: si se ingresaba una palabra correcta, no sumaba los respectivos puntos, y en el intérprete saltaba: TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'.La solución fue agregar un **return** para devolver def Puntos(), ya que en un principio no lo poseía.

En segundo lugar, ya con el juego funcionando correctamente, decidimos agregar las siguientes funciones extras:

- **def correctaRepetida(palabrasAcertadas,candidata)**, vimos necesario definirlo para que el participante no repita una palabra correcta y sume muchos puntos, la función devuelve un True en caso de que la palabra candidata ya estaba en la lista de palabras acertadas.
- **def reproducir_musica(musica)**, toma una serie de funciones de la librería de pygame, para permitir cargar y reproducir archivos de sonido.
- Se le hicieron los cambios a **def dibujar(screen, letraPrincipal, letrasEnPantalla, candidata, puntos, segundos):** se le agrego una imagen de fondo por medio de una función de la librería Pygame y se agrego la siguiente cadena a la hora de ingresar candidata: "Escribe una Palabra: ".
- **def dibujarPalabrasAcertadas(screen, palabrasAcertadas)**, muestra una pantalla final que muestra las palabras acertadas que el usuario fue ingresando. Se tomó como referencia def dibujar para su creación.
- Se personalizo el siguiente valor en configuraciones.py: COLOR FONDO = (139,0,0).