Introducción a la programación

TRABAJO PRACTICO

**Integrantes:**

* Pellegrini Karin
* Escobar Yanina

**Profesoras:**

* Bottino Flavia
* Varela Viviana

**Comisión 08**

UNIVERSIDAD NACIONAL DE GENERAL SARMIENTO

Introducción

Pygame es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla.

En el siguiente informe explicaremos la creación de un videojuego utilizando Pygame. En la carpeta del juego contaremos con algunas funciones básicas ya definidas y otras por definir. La combinación de todas estas funciones da como resultado final el videojuego deseado.

El juego inicialmente cuenta con tres archivos dados y uno vacío en el cual tenemos que crear las funciones para que el juego funcione correctamente.

Objetivo

Completar el archivo "funcionesVACIAS" con las funciones que faltan definir y lograr el correcto funcionamiento del juego.

De manera opcional, ingresar las modificaciones necesarias para la creación de un video juego más completo.

**En Sí-La-Bas…**

En este juego vamos tener 60 segundo para armar palabras con las silabas que vayan cayendo por la pantalla.

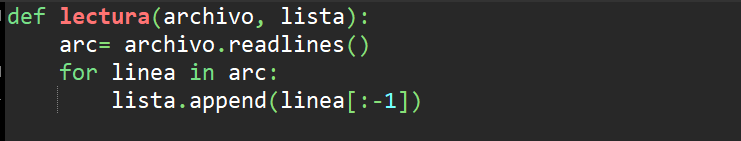
|  |  |
| --- | --- |
|  |  |

La puntuación va a variar dependiendo de las letras que tengan las palabras que escribamos. Por ejemplo, si nuestra palabra tiene vocales, cada vocal nos sumara 1 punto, si tiene consonantes 2 puntos y si tiene letras más difíciles (j,k,q,ñ,w,x,y,z) 5 puntos

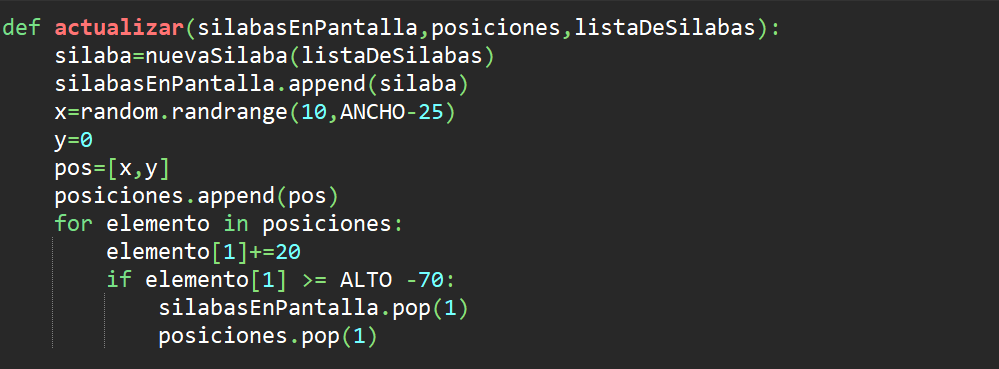
Al finalizar, el juego nos dará la opción de guardar nuestro puntaje y nos mostrará los mejores 10 puntajes del mayor al menor con los respectivos nombres.

|  |  |
| --- | --- |
|  |  |

A continuación, vamos explicar cómo están diseñadas de las funciones que le dan vida a nuestro juego.



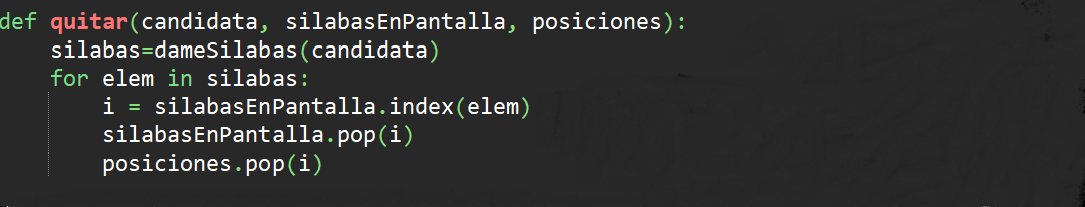
La función **lectura** se encarga de abrir los archivos que vienen desde el principal y guardarlos en una lista



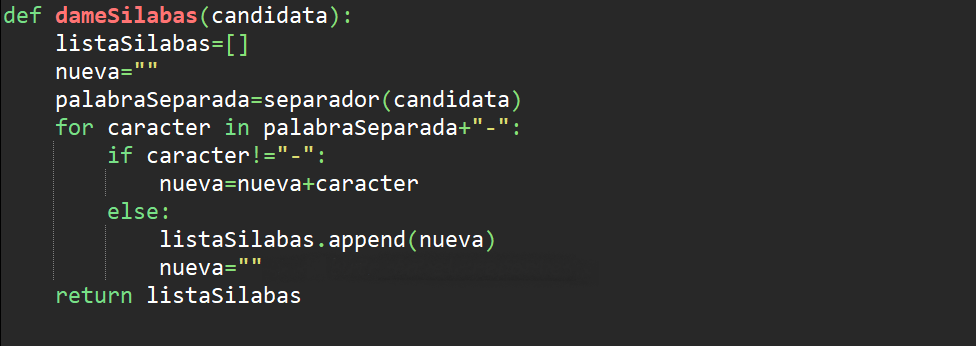
La función **actualizar** recibe las sílabas, sus posiciones y la lista de todas las sílabas. Va actualizando las posiciones de las sílabas para que vayan descendiendo y elimina las sílabas cuando empiezan a escaparse de la pantalla.



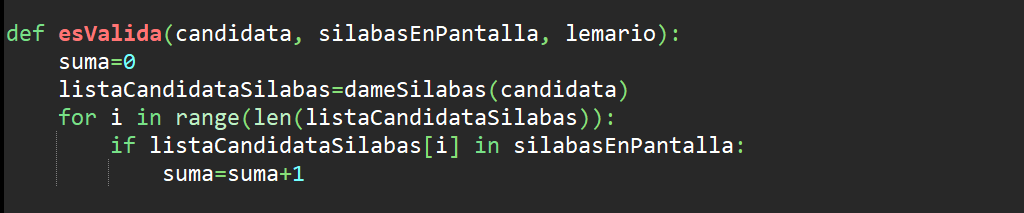
La función **nuevaSilaba** recibe la lista de sílabas y devuelve una al azar.



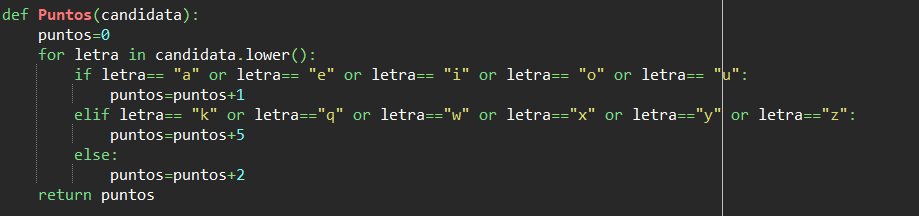
La función **quitar** recibe la palabra que escribió el usuario, la lista de sílabas en pantalla y sus posiciones, la quita de la lista y elimina su posición**.**



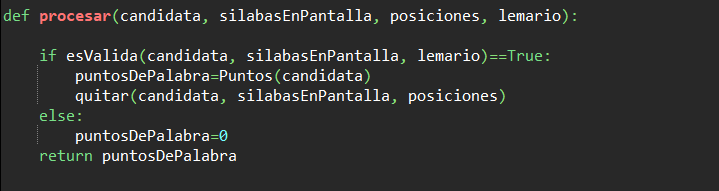
La función **dameSilabas** recibe la palabra que escribió el usuario y devuelve una lista con sus sílabas.



La función **esValida** recibe la palabra que escribió el usuario, la lista de silabas en pantalla y el lemario y controla si la palabra es correcta y se encuentra en lemario



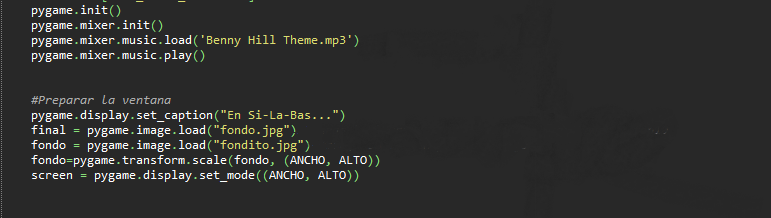
La función **Puntos** es la encargada de darnos las puntuaciones, recibe la palabra y nos da los puntos dependiendo las letras con las que esta formada nuestra palabra .Si tiene una vocal ,es un punto, si tiene una consonante ,dos puntos y si tiene una consonante difícil (j,k,q,w,x,y,z) van a ser tres puntos .



La función **procesar** recibe la palabra que escribe el usuario, la lista de silabas en pantalla, sus posiciones y el lemario y se encarga de controlar que la palabra se a válida y en caso de que lo sea, quitarla de la pantalla y sumar lun pontos correspondientes.

**Extras!**

Música y fondos



En estas líneas podemos observar que ingresamos dos imágenes, una para el fondo del juego, que decidimos acomodar para que quede acorde al tamaño de nuestra ventana y la otra para el fondo del “GAME OVER”, esta no la acomodamos porque nos gustó el resultado.

También, se puede observar que ingresamos una canción de fondo que se va a reproducir mientras jugamos, una vez finalizado el tiempo, la canción va a cortar y va a empezar a reproducir otra mientras estemos en la ventana de GAME OVER y los puntajes



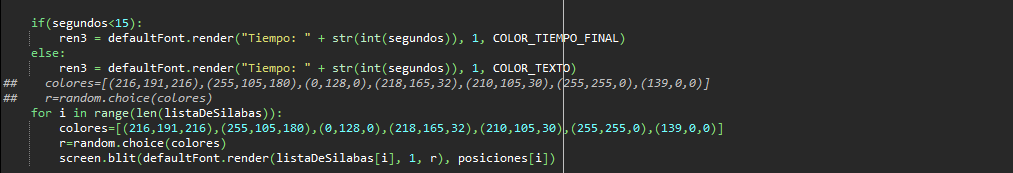
Puntaje y top10



Para el puntaje y top 10, creamos dos funciones nuevas. La función **guardarPuntaje** nos va a crear un archivo.txt donde se van a ir guardando los puntajes y sus respectivos nombres.

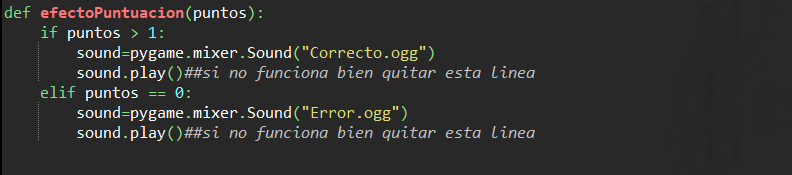
**topTen** va a tomar ese archivo.txt y nos lo va a mostrar de mayor a menor puntaje.

**Letras de colores:**



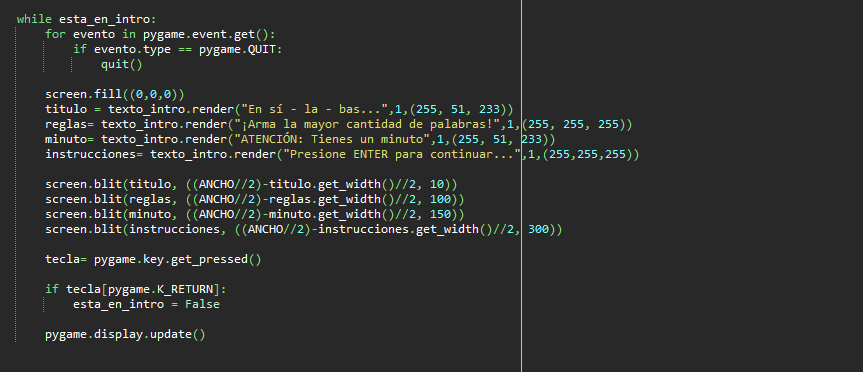
Acá agregamos una variable con varios colores y hicimos que vayan cambiando al azar con un random.choice. En un principio lo pusimos por fuera del for y eso le daba un color a todas las silabas que iba cambiando, luego lo pusimos dentro del for y eso nos dio como resultado que las silabas vayan cambiando de color continuamente, lo cual nos pareció divertido y decidimos mantenerlo.

**Efectos de puntuación:**



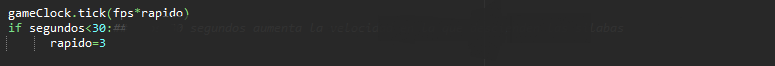
A demás de las funciones ya mencionadas, también decidimos agregar una que nos permita agregarle sonidos a los aciertos y a los errores.

**Pantalla de inicio:**



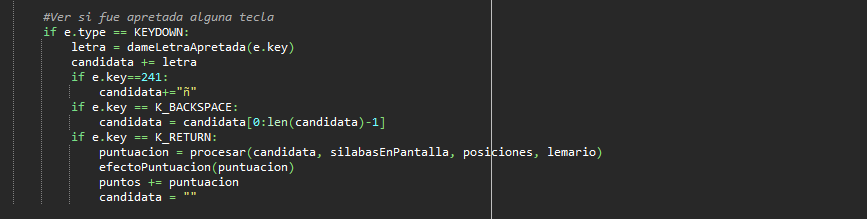
Para hacer el juego más vistoso, también se agrego una pantalla inicial donde nos dice el nombre del juego, una breve explicación de que trata y el tiempo disponible. Una vez que el usuario haya leído todo puede iniciar el juego apretando la tecla enter.

**Dificultad:**



En estas líneas, hicimos que el juego a partir de los 30 segundo vaya más rápido. Para esto, inicializamos una variable “rápido” en 1, mientras el juego este por encima de los 30 segundos, Fps se va a multiplicar por 1. Una vez que el juego se encuentre por debajo de los 30 segundos esa variable “rápido” va a pasar a tener el valor de 3, esto va a hacer que nuestras silabas caigan con mayor velocidad.

**Letra Ñ:**



También agregamos la posibilidad de agregar palabras con la letra Ñ que originalmente no se puede.