

Introducción a la programación

En sílabas... (Aprendiendo a separar en sílabas...)

2 de mayo de 2022

Introducción

Saber dividir una palabra en las sílabas que la componen resulta de suma utilidad para los alumnos y alumnas porque los ayuda a pronunciar la palabra correctamente, aprenden a colocar el acento ortográfico (la tilde) y poder realizar correctamente la partición de una palabra al final de una línea (mediante el uso del guión). El trabajo consiste en hacer el famoso juego de las sílabas. En la pantalla aparece una palabra y se la debe escribir separada en sílabas. Las palabras se deben buscar en el leuario (diccionario sin definiciones) y el jugador o jugadora puede usar para separar las sílabas un espacio o un `—`. Luego, cuando aparece en pantalla una nueva palabra también debe aparecer la palabra anterior separada correctamente en sílabas.

Por ejemplo:

En pantalla aparece: “gato”

Usuarix: “ga-to” o “ga to” y ambas son correctas

Gran parte del juego ya está resuelto, faltan implementar las funcionalidades más importantes.

1. El Juego

Reglas del Juego

El jugador o jugadora cuenta con 60 segundos para acertar la mayor cantidad de palabras y se le suman puntos por las respuestas correctas o se le restan las incorrectas.

Lo que ya está implementado

El juego actualmente consta de un archivo con el programa principal. Este se encarga de capturar la entrada del teclado y de dibujar en la pantalla. Se brinda un archivo de texto con palabras, leuario (un diccionario sin las definiciones).

Para la mayoría de estas tareas, el programa hace uso de una biblioteca de código llamada *PyGame*. Una biblioteca de código es un conjunto de subprogramas utilizados para desarrollar software. En particular PyGame es una biblioteca especialmente diseñada para el desarrollo de juegos interactivos en Python.

La posición (0,0) de la pantalla es el vértice superior izquierdo, las x crecen hacia la derecha y las y crecen hacia abajo.

Lo que falta implementar

Lo primero que se debe hacer es leer el archivo. Luego debe hacer funcionar el juego, para ello debe completar las funciones vacías obligatorias.

Aun faltan implementar las funciones del archivo `funcionesVACIAS.py`. Estas funciones son utilizadas desde el programa principal.

La función `lectura(archivo, salida)` lee el archivo y lo carga en la lista `salida`, no hace falta que devuelva una lista sino que lo carga en la lista que recibe como segundo parámetro

La función `nuevaPalabra(lista)` debe recibir una lista de palabras y devuelve una de ellas elegida al azar.

La función `silabasT0palabra(silaba)`: recibe una palabra separada en silabas y devuelve la palabra sin "

La función `palabraT0silaba(palabra)` Es una función que separa en silabas, hasta ahora usa unas funciones realizadas por los docentes.

La función `esCorrecta(palabraEnSilabasEnPantalla, palabra)`) debe recibir la palabra de la pantalla ya separa en sílabas y controlar que sea la misma que escribió el usuario. Cuidado que el usuario puede usar espacios o "-" para separar en sílabas.

La función `puntaje(palabra)`) debe recibir una palabra y devolver el puntaje que corresponde.

2. Cómo empiezo

2.1. Instalar PyGame

Deben dirigirse a Herramientas, luego Herramientas, Python Packages with pip y ahí escribir pygame. Si tienen problemas les dejo links para descargarse las últimas versiones de todo:

<https://sourceforge.net/projects/pyscripter/>
<https://www.python.org/downloads/>
<https://www.pygame.org/download.shtml>

2.2. Descargar archivos

Descargar del moodle de la materia el archivo comprimido con todos los archivos necesarios para el TP. Descomprimir todo el contenido del archivo en una carpeta y abrir los archivos `.py` con el PyScripter. El único archivo que se espera que modifiquen al principio es `funcionesVACIAS.py` pero a la hora de ejecutar el proyecto, hay que ejecutar el archivo `principal.py`.

3. Consigna

Implementar las funciones requeridas para el correcto funcionamiento del juego.

Pensar e implementar funciones auxiliares que resuelvan tareas intermedias, de forma tal que el código sea más claro, sencillo, ordenado, legible y fácil de corregir.

Las funciones que reciben listas como parámetros deberán también chequear que dichas listas permanezcan en el estado correcto luego de utilizada la función.

Sugerimos **fuertemente** probar y corregir las funciones más sencillas antes de encarar las funciones más complicadas.

4. Requisitos de aprobación y criterio de corrección

El presente trabajo debe realizarse en grupo, deben realizar pre entregas y hacer una defensa con la presencia de todos los integrantes del grupo. Para aprobar el trabajo, se deberá cumplir los siguientes items:

- El juego debe funcionar correctamente.

- El código debe ser claro. Es decir, las variables y funciones deben tener nombres que hagan fácil de entender el código a quien lo lea, y deben haber comentarios que ayuden al fácil entendimiento de cada porción de código. Además debe hacer funciones siempre que se considere necesario, y se evaluará el buen uso de las mismas.
- El código debe ser coherente. Es decir, no deben haber variables que no se usan, funciones que tomen parámetros que no necesitan, ciclos innecesarios, etc.
- Los casos de prueba deben ser completos y debe ser posible correrlos de nuevo por cualquiera.

Nota: El correcto funcionamiento del juego no es suficiente para la aprobación del trabajo, son necesarios todos los items mencionados arriba.

5. Opcionales

Las siguientes funcionalidades del juego no son necesarias para la aprobación (con nota mínima), pero sirven para mejorar la nota del trabajo. De optar por hacerlas, se aplican las mismas reglas y criterios de corrección que para las funcionalidades básicas. Cualquier otra funcionalidad extra que se desee implementar debe ser antes consultada con los docentes.

Separar en sílabas

Hacer tu propia función para separar una palabra en sílabas o buscar una implementación en internet y adaptarla.

Longitud mínima

Toma de la lista de palabras solo aquellas que tengan más de cierta cantidad de caracteres.

Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes.

Diferentes niveles

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

- el usuario indica la longitud mínima (o máxima) de letras que deben tener las palabras,
- (mas facil)Muestra algunas sílabas con otro color.
- (mas dificil)No permitir repetir palabras.
- tener varios archivos de letreros ordenados por nivel de dificultad para separar en sílabas.

6. Fecha de entrega

El trabajo debe ser presentado, defendido en grupo de manera presencial. Además se deben ir mostrando los avances del trabajo a modo de pre-entregas. Lxs docentes comunicarán la fecha exacta de la defensa de cada grupo, se espera que sea las últimas semanas de clase.

7. 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 entrega impresa consiste de un **informe** donde haya una introducción que explique de qué se trata el trabajo (explicado para alguien que no leyó el enunciado), que incluya el código de las funciones implementadas y una breve explicación de cada una de ellas junto con las dificultades de implementación con las que se encontraron. No se deben imprimir los casos de prueba. El informe también debe incluir las decisiones que hayan tomado ante diferentes alternativas posibles, cuáles fueron sus alternativas consideradas y por qué tomaron esas decisiones. El informe **no** debe incluir párrafos para ocupar lugar, explicaciones de funcionalidades básicas de Python, PyGame o de programación en general.

- La entrega digital consiste en enviar un e-mail a sus docentes:
 1. **Con asunto:** entrega-tp <año>< nro – grupo >< Apellidos > .
 2. 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.