

Introducción a la programación

En sí - la - bas...

May 10, 2021

Introducción

El trabajo consiste en implementar un juego para escribir palabras con las sílabas que aparecen en la pantalla. Gran parte del juego ya está resuelto, solamente faltan implementar las funcionalidades más importantes.

1 El Juego

Reglas del Juego

Se juega de a un jugador, que cuenta con 60 segundos para escribir la mayor cantidad de palabras que aparecen en pantalla. El jugador debe escribir la palabra que puede armar con las sílabas de la pantalla y si la palabra es la correcta, se deben sumar puntos al jugador y las sílabas de dicha palabra desaparecen. Todo el tiempo aparecen nuevas sílabas en pantalla que van descendiendo hasta desaparecer. Estas sílabas el juego las toma de un archivo y para controlar que la palabra que escribe el usuario existe usted cuenta con un archivo con palabras del diccionario (lemario).

Lo que ya esta implementado

El juego actualmente consta de un archivo con el programa principal. Este se encarga de capturar la entrada del teclado, llevar la cuenta de los puntos y del tiempo, así como también de dibujar en la pantalla.

El programa principal cuenta también con:

- una variable de tipo `lista` que guarda las sílabas que están en pantalla.
- una variable de tipo `lista` que guarda las posiciones de cada sílabas
- Además llama a otras funciones que deberán realizar. Y cuenta con una función que separa una palabra en sílabas, es decir, le agrega un " - " entre sílabas.

El programa toma la sílabas al azar desde el archivo `silabas.txt`, le asigna una posición una sílabas. Luego espera que el jugador o jugadora escriba una palabra y chequea que sea correcta, es decir, que se pueda construir con sílabas de la pantalla y que la palabra exista, eso lo verifica controlando que la palabra se encuentre en el archivo `lemario`.

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 sub-programas 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

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

La función `def lectura(archivo, lista)` debe leer el archivo y guardarlo en la lista.

La función `actualizar(silabasEnPantalla, posiciones, listaDeSilabas)` recibe las sílabas que están en la pantalla, sus posiciones y la lista de todas las sílabas. Debe eliminar las sílabas y sus posiciones cuando empiezan a escaparse de la pantalla, también debe ir actualizando las posiciones de las sílabas para que vayan descendiendo y agregando nuevas en posiciones, siempre con el cuidado de que las sílabas no se superpongan, ni se escapen de la pantalla.

La función `nuevaSilaba(silabas)` recibe la lista de sílabas y devuelve una al azar.

La función `quitar(candidata, silabasEnPantalla, posiciones)` recibe la palabra que escribió el usuario, la lista de sílabas en pantalla y sus posiciones y debe quitarla de la lista y eliminar su posición.

La función `dameSilabas(candidata)` recibe la palabra que escribió el usuario y devuelve una lista con sus sílabas. (Su uso NO es obligatorio)

La función `esValida(candidata, silabasEnPantalla, lemario)` debe recibir la palabra que escribe el usuario, la lista de sílabas en pantalla y el `lemario`. Debe controlar si la palabra es correcta, es decir, sílabas de la candidata deben estar en la pantalla y formar una palabra en el `lemario`.

La función `Puntos(candidata)` debe recibir una palabra y retornar el puntaje total correspondiente a la palabra formada según:

- cada vocal otorga 1 punto,
- cada consonante otorga 2 puntos, salvo las difíciles (j, k, q, w, x, y, z).
- cada consonante difícil (j, k, q, w, x, y, z) otorga 5 puntos.

La función `procesar(candidata, silabasEnPantalla, posiciones, lemario)` debe recibir la palabra que escribe el usuario, la lista desílabas en pantalla y sus posiciones y el lemario. Debe controlar si la palabra es válida, y en caso afirmativo quitarla de la pantalla y sumar los puntos correspondientes. Devuelve el puntaje obtenido.

La función `separador(palabra)` recibe una palabra y devuelve una cadena con las sílabas de la palabra separadas por un guión. Esta función se usa directamente, NO hay que realizarla.

2 Cómo empiezo

2.1 Instalar PyGame

Desde Pyscripter ir a Herramientas luego Herramientas y finalmente a install packages with pip y allí escribir pygame.

Otra opción es: La versión más reciente de PyGame para Windows y Python 3.1 se descarga directamente desde <http://pygame.org/ftp/pygame-1.9.1.win32-py3.1.msi>.

para diferentes versiones, chequear <http://www.pygame.org/download.shtml>. La versión de python instalada recomendamos que sea de 32 bits.

Si tienen problemas pueden:

1. Descargar pygame y luego este archivo desde <https://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame>

Pygame : una biblioteca para escribir juegos basada en la biblioteca SDL .

[pygame - 1.9.6 - cp38 - cp38 - win_amd64.whl](#)
[pygame - 1.9.6 - cp38 - cp38 - win32.whl](#)
[pygame - 1.9.6 - cp37 - cp37m - win_amd64.whl](#)
[pygame - 1.9.6 - cp37 - cp37m - win32.whl](#)
[pygame - 1.9.6 - cp36 - cp36m - win_amd64.whl](#)
[pygame - 1.9.6 - cp36 - cp36m - win32.whl](#)
[pygame - 1.9.6 - cp35 - cp35m - win_amd64.whl](#)
[pygame - 1.9.6 - cp35 - cp35m - win32.whl](#)
[pygame - 1.9.6 - cp27 - cp27m - win_amd64.whl](#)
[pygame - 1.9.6 - cp27 - cp27m - win32.whl](#)
[pygame - 1.9.4 - cp34 - cp34m - win_amd64.whl](#)
[pygame - 1.9.4 - cp34 - cp34m - win32.whl](#)

2. Ejecutar cmd como administrador e ir a la carpeta donde hayan guardado el archivo.
3. Escribir `py -3.7 -m pip install pygame-1.9.6-cp37-cp37m-win32.whl`

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 es `funciones.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, consultar con sus docentes la cantidad mínima y máxima. 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.

Longitud mínima

Toma de la lista de sílabas solo aquellos que tengan más de cierta cantidad de caracteres.

Velocidad

A medida que pasa el tiempo las sílabas descienden con mayor velocidad.

Por color

Cada sílaba tiene un color aleatorio.

Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes: acierta una palabra, ingresa una palabra con mucho puntaje, obtuvo una seguidilla de aciertos, errores, etc.

Mejores records

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.

Diferentes niveles

Que el usuario tenga opciones de niveles donde se modifique el juego.

6 Fecha de entrega

El trabajo debe ser entregado en la fecha estipulada en el cronograma, recordar que es requisito hacer pre-entregas.

7 Forma de entrega

Se debe enviar un informe en pdf 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. 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. Además se debe enviar el trabajo para que los docentes lo puedan correr todo debe estar en una carpeta con los nombres de los estudiantes que integran el grupo.

Nota: Se acepta una sólo entrega por grupo, de modo que si un grupo entrega su trabajo más de una vez, los docentes elegirán cuál de las versiones entregadas corregir.