

Introducción a la programación

Dicccionar.io

October 18, 2016

Introducción

El trabajo consiste en implementar un juego para aprender definiciones de palabras. Aparecerán en pantalla definiciones sacadas de un diccionario y el jugador deber escribir la palabra que se está definiendo. Inicialmente se eligen al azar dos letras que salen en pantalla y la palabra que se busca contiene dichas letras. Gran parte del juego ya está resuelto, solamente faltan implementar las funcionalidades más importantes y resolver algunos problemas. Usted ya cuenta con un archivo con algunas palabras con sus definiciones separadas por `:`.

1 El Juego

Reglas del Juego

Se juega de a un jugador, que cuenta con 60 segundos para escribir la mayor cantidad de palabras, en caso de considerar muy difícil alguna palabra el jugador puede escribir el número 1 indicando que desea pasar. El jugador debe escribir la palabra en pantalla y si la palabra es la correcta, se deben sumar puntos al jugador y aparecerán dos nuevas letras al azar y una definición.

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.

Que usted deberá realizar para que el programa funcione correctamente.

Para la mayoría de las 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

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

La función `lectura(listaPalabras, listaDefiniciones)` debe leer el archivo y cargar en `listaPalabras` las palabras y en `listaDefiniciones` las definiciones. Ojo, `listaDefiniciones[5]` debe ser la definición de la palabra `listaPalabras[5]`

La función `eligeLaPalabra(listaPalabras, letras)` debe elegir de la lista de palabras una que contiene estas letras

La función `damePosicion(palabra, listaPalabras)` debe devolver la posición de un elemento en una lista de elementos

La función `dameDefinicion(palabra, listaPalabras, listaDefiniciones)` debe devolver la definición de la palabra.

La función `azar(listaPalabras)` debe elegir 2 letras al azar.

La función `puntuar(candidata, palabra)` debe recibir una palabra y la candidata y si es correcta 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 `esCorrecta(candidata, palabra)` recibe la candidata y verifica si es correcta (devuelve un valor booleano).

La función `esta(palabra, listaPalabras)` recibe un elemento y una lista y devuelve True si encuentra al elemento en la lista y False en caso contrario.

Además usted debe resolver dos problemas:

- Evitar que se repita la palabra elegida del diccionario.
- No elegir letras al azar que no estén en palabras del diccionario.

2 Cómo empiezo

2.1 Instalar PyGame

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>. Al momento de instalar PyGame, el programa de instalación intentará reconocer la versión de Python instalada en la computadora. De no detectarla correctamente, se sugiere no continuar con la instalación, dado que PyGame no funcionará. En ese caso chequear que la versión de PyGame sea la indicada para la versión de Python. En nuestro caso es la 3.1.

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 PAREJAS, se pueden usar TODAS las funciones auxiliares que desee. Para aprobar el trabajo, se deberá cumplir los siguientes items:

- El juego debe funcionar correctamente.
- Debe resolver estos problemas:
 - Evitar que se repita la palabra elegida del diccionario.
 - No elegir letras al azar que no esten en palabars del diccionario.

- 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.

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 palabras solo aquellos que tengan más de cierta cantidad de caracteres.

Ayuda

A medida que pasa el tiempo ayudar al jugador, indicando donde se encuentran la letras que salieron al azar. Mostrando la primer letra, etc.

Castigos

Castigar errores con quita de puntos, o quita de tiempo. Y castigar por presionar 1 para saltar alguna palabra.

Premios y Castigos en tiempo

Usar como premio y castigo el tiempo y no el puntaje. Es decir, al acertar agregar segundos y al equivocarse disminuir el tiempo.

Evitar recorte de palabras

Evitar que al mostrar las definiciones se corten algunas palabras.

Temático

Conseguir o armar un diccionario con algun tema específico como nombre de trago, ingredientes.

Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes: ingresa una palabra, ingresa una palabra con mucho puntaje, 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, por ejemplo:

- el usuario indica la longitud mínima (o máxima) de letras que deben tener las palabras,
- el usuario indica los puntos asignados a las letras difíciles (baja, media, alta).
- va cambiando la cantidad de letras al azar o se asegura al menos una vocal, etc
- el nivel fácil es para sinónimos o antónimos, etc .

6 Fecha de entrega

El trabajo debe ser presentado y defendido por el grupo la última semana de Noviembre, confirmar con el docente la fecha exacta.

7 Forma de entrega

El TP se considera correctamente entregado y apto para ser corregido si se cumple con la entrega presencial, la impresa y la digital.

- La entrega presencial 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 precisamente todo el trabajo realizado por todo el grupo.

- La entrega impresa consiste de un informe impreso 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.