Introducción a la programación

Armar palabras encolumnadas

October 12, 2021

Introducción

El trabajo consiste en implementar un juego para escribir palabras con las letras que aparecen en la pantalla y que cumplen con algunos requisitos mas. Las letras aparecen en 3 columnas y van descendiendo en la pantalla hasta desaparecer. 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 con las letras que aparecen en pantalla. La palabra debe existir en el diccionario en español (lemario), debe usar las letras de la pantalla pero las columnas se deben usar en orden, es decir, si la primera letra de la palabra está en la primera columna se puede seguir usando esa columna pero si alguna letra no está en esa columna y se empieza a buscar en la segunda o tercera columna ya no se puede buscar en las anteriores columnas a las demas letras. Por ejemplo, aunque "donas" es una palabra válida en nuestro idioma y se puede formar con las letras de la pantalla en este juego es inválida. La letra "d" está en la primera columna, la "o" no está en la primera ni en la segunda columna, recién apararece en la tercera columna pero ahora ya no se pueden usar las primeras columnas para las próximas letras. Las letras "n" y "o" están en la tercera comluna pero no la "s".

Todo el tiempo (o casi) aparecen nuevas letras en pantalla que van descendiendo hasta desaparecer. Estas letras el juego las saca de un archivo, de donde elige una palabra al azar y envia las primeras letras a la primera columna, las siguientes a la segunda y las ultimas a la tercera. Cuantas letras a cada columna debe variar también al azar. Aunque las letras formaban originalmente una palabra puede el jugador sugerir otro y deberá ser considerado válido. Si el jugador ingresa la palabra correcta, la jugada es válida y se suman puntos.

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:

- variables de tipo `lista` donde se guardan las letras que están en cada una de las columnas de la pantalla.
- variables de tipo `lista` que albergan listas donde se guarda la posiciones de las letras. Por ejemplo: si listaIzq = [”h”,”o”] posicionesIzq = [[10,8],[22,60]]
- una variable de tipo `lista` donde se guardan las palabras que se leen del archivo lemario.
- en configuración tienen definido los tamaños de las letras, los colores, el ancho y el largo de la pantalla y el tiempo del juego
- en extras tienen la captura de lo que escribe el usuario y dibujar, esta última función es la que escribe en la ventana, dibuja las lineas verticales que determinan las columnas, dibuja la linea horizonal y muestra el tiempo, puntaje y lo que escribe el usuario.

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

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

La función `cargarListas(lista, listaIzq, listaMedio, listaDer, posicionesIzq, posicionesMedio, posicionesDer)` elige una palabra de la lista, carga las letras en las 3 listas y les inventa una posición para que aparezcan arriba en la columna correspondiente.

La función `bajar(lista, posiciones)`hace bajar las letras y elimina las que tocan el piso.

La función `actualizar(lista, listaIzq, listaMedio, listaDer, posicionesIzq, posicionesMedio, posicionesDer))` llama a otras funciones para bajar las letras, eliminar las que tocan el piso y cargar nuevas letras a la pantalla (esto puede no hacerse todo el tiempo para que no se llene de letras la pantalla).

La función `estaCerca(elem, lista)` es opcional, se usa para evitar solapamientos

La función `procesar(lista, candidata, listaIzq, listaMedio, listaDerecha))` chequea que candidata sea correcta en cuyo caso devuelve el puntaje y 0 si no es correcta

La función `esValida(lista, candidata, listaIzq, listaMedio, listaDerecha)` devuelve True si candidata cumple con los requisitos

La función `Puntos(nombre)` 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.

santiago montiel, [12.10.21 11:17]

2 Cómo empiezo

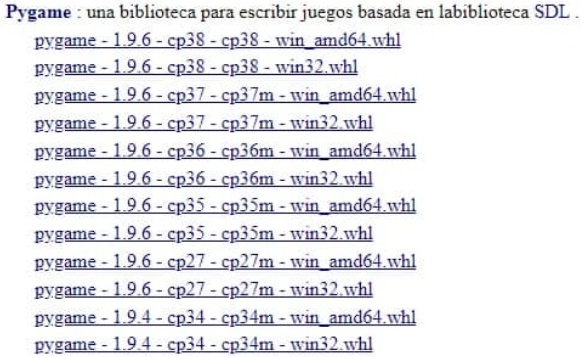
2.1 Instalar PyGame

Desde Pyscripter ir a Herramientas luego Herraminetas 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`



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 del lemario palabras de cierta cantidad mínima de caractéres. También se puede prohibir al usuario escribir palabras muy cortas.

Velocidad

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

Por color

Cambiar los colores de las letras por columna.

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 sóla 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.