



Informe ScrabbleAR

Autores: Arrue Tomas - Di Lorenzo Lucas - Paz Ayala Danilo
Grupo Nro. 32

ÍNDICE

ÍNDICE.....	2
INTRODUCCIÓN.....	3
REGLAS DEL JUEGO.....	3
TEMAS INVESTIGADOS.....	5
PROBLEMAS Y SOLUCIONES.....	5
CONCLUSIONES	7
REFERENCIAS.....	7

INTRODUCCIÓN

A continuación, se hará una breve introducción sobre el proyecto propuesto por la cátedra de seminario de Lenguajes (Python).

El proyecto consiste en utilizar el lenguaje aprendido durante el semestre para realizar una implementación funcional y didáctica del conocido juego Scrabble o Palabras cruzadas.

En Pos de lograr dicho objetivo se analizó las librerías "PySimpleGUI" y "Pattern" y los módulos como "OS", "Sys", "Random", "Json", ambas herramientas proporcionados por la cátedra correspondiente.

El programa está estructurado de la siguiente manera:

- **Ventana principal:** Contiene opciones para acceder al juego, cargar una partida anterior y/o opciones para realizar configuraciones adicionales. Además se visualiza un tablero en blanco a modo de presentación y la opción para el usuario cargue su nombre.
- **Configuración:** Desde él se modifican las configuraciones particulares para la ejecución del juego, tales como: la cantidad de letras totales para cada letra y su valor correspondiente, la dificultad en la que el usuario comenzara el juego, la duración de la partida.
 - Posee una configuración por defecto, que se carga al iniciar el programa si no se realizó algún tipo de modificación.
- **Juego:** Desde él se genera la matriz para el propio juego basándose en las configuraciones establecidas (o la configuración por defecto). La interfaz fue desarrollada teniendo en cuenta que sea lo más intuitiva posible, esto implica que el usuario tenga a la vista las opciones más comunes como "borrar", "evaluar", "repartir" y consultar un "ranking de jugadores"; Así también como la correcta visualización de su puntaje y su atril con letras.

REGLAS DEL JUEGO

- El juego contará con una cierta cantidad de fichas de cada letra en su bolsa de fichas (letras) para repartir entre los jugadores.
- Cuando comienza, se "reparten" siete (7) fichas a cada participante de forma aleatoria. El jugador tendrá asignado un "atril o estante" para almacenar sus fichas.
- El jugador debe formar una palabra usando dos (2) o más letras, colocándolas horizontalmente (las letras ubicadas de izquierda a derecha) o verticalmente (en orden descendente) sobre el tablero.
- Para simplificar la estrategia del juego, las palabras no deben cruzarse, pero si pueden quedar "pegadas" tanto horizontal como verticalmente.

- Para colocar una letra en el tablero primero el jugador deberá seleccionar una casilla libre dentro del tablero y posteriormente deberá seleccionar la letra a colocar dentro de su atril o estante.
- Una vez ingresada la palabra se debería confirmar la misma en el tablero y automáticamente se chequeará si la palabra corresponde a la clasificación que se está usando en el juego. Se recuerda que las únicas palabras admitidas en el tablero serán adjetivos, sustantivos y verbos, de acuerdo a las opciones de configuración establecidas previamente. En caso de no corresponder, las fichas serán devueltas al jugador para que vuelva a intentar.
- Una vez que el jugador haya ingresado y confirmado correctamente la palabra, se le repondrá la cantidad de fichas que utilizó sin contar las preexistentes en el tablero. De esta manera el jugador nunca tendrá más de siete (7) fichas en su poder.
- En cualquier momento del juego, el jugador puede decidir usar un turno para cambiar todas sus fichas, devolviéndolas a la bolsa de fichas del juego y reemplazándolas por la misma cantidad; al final, siempre debe tener siete (7).
- En este mismo turno, el jugador no podrá colocar ninguna palabra sobre el tablero. El jugador sólo podrá usarla como máximo tres veces durante el juego.
- En todo momento estarán visibles los puntajes del jugador y del rival. El mismo se calculará de la siguiente forma: se sumará el puntaje de cada letra y si la palabra cae en algún casillero con premio o descuento se sumará o restará el puntaje adicional.
- Se podrá acceder a visualizar un top 10 con los puntajes obtenidos del tipo: *fecha*, *puntaje*, *nivel*.
- El juego termina cuando:
 - ▶ el jugador en turno no puede completar sus siete (7) fichas luego de una jugada, dado que no hay más fichas en la bolsa de fichas del juego;
 - ▶ se acabó el tiempo de la partida;
 - ▶ se presionó el botón "TERMINAR" partida.

TEMAS INVESTIGADOS

- **PySimpleGUI:**

- El primer módulo en ser analizado. Se discutió la mejor distribución de las secciones principales (juego, menú y configuración) y la comunicación entre ellas. Luego, la transición entre cada ventana y Popups si se diera el caso de algún conflicto por falta de archivos y/o otras situaciones, por último, los diseños a utilizar en cada dificultad.

- **Pattern:**

- Se aprovecharon los métodos provistos por este módulo para obtener una verificación fiable sobre las palabras formadas durante el juego a través del propio diccionario del módulo Pattern.

- **Otros módulos:**

- Se utilizaron módulos provistos por la propia librería de Python 3.6.8, tales como os, random, time, entre otros.

PROBLEMAS Y SOLUCIONES

- **Sobre la Versión de Python y compatibilidad con Pattern:**

- Hay un error de Pattern que produce problemas para ejecutarlo en ciertas versiones de Python.
- **Solución:** Se aclara en el ReadMe los requisitos, y las versiones óptimas para ejecutar correctamente el programa.

- **Sobre PySimpleGUI y los widgets:**

- En algunas versiones de PySimpleGUI existen conflictos entre los widgets que se fueron añadiendo durante la vida de la librería.
- **Solución:** se recomienda utilizar la versión declarada en el ReadMe o una posterior
- Columnas: algunos botones dentro del widget columns si se marcan como invisibles para simular el efecto de transición entre el atril y el tablero pueden degenerar la interfaz al ser recuperados luego de un borrado o una verificación inválida.
- **Solución:** se optó por hacer que el botón seleccionado luego se desactive hasta la verificación correcta o incorrecta de la palabra, proceso durante el cual será reemplazado por un nuevo valor o el mismo será recuperado.

- **Sobre el Tablero de juego:**

-
- El tablero es una matriz cuadrada de botones que se genera tomando como base un número asignado en un archivo de configuración. Las palabras no deben superponerse y o comenzar desde una letra de una palabra previa:
 - **Solución:** Las palabras pueden formarse en cualquier lugar del tablero (que este disponible) teniendo en cuenta las restricciones ya mencionadas.
 - La forma de calcular los puntos puede generar conflictos entre las casillas que operan para un valor asociado a una letra y para un valor asociado a una palabra:
 - **Solución:** Se asignan distintos colores para aquellas casillas que deban realizar la operatoria sobre letras y aquellas que deban hacerlo sobre una palabra. Se utiliza una función que sabe cómo diferenciar y reconocer la forma en la que debe operar según cada caso específico.
 - El progreso de la partida puede ser descartado o salvado.
 - **Solución:** en cualquier momento el jugador tiene la opción de guardar su progreso hasta el momento. Esto implica tomar los datos necesarios desde el tablero y guardarlo en algún formato específico que permita la correcta recuperación en un momento posterior.
 - **Sobre Pattern y la forma de verificar una palabra válida:**
 - El módulo Pattern considera todas las palabras que no logra reconocer como Sustantivos. También tiene problemas para reconocer aquellas palabras con tildes o diéresis.
 - **Soluciones:**
 - Se consideran solo como palabras válidas por el momento aquellas que sean adjetivos o verbos.
 - No se tomarán en cuenta las palabras con acentos, es decir, no se podrán formar palabras que tengan tildes.

CONCLUSIONES

Python es un lenguaje de programación muy versátil, con la capacidad de utilizar multitud de librerías para cualquier necesidad.

La realización de este trabajo nos planteó un reto que nos permitió comprobar nuestros conocimientos adquiridos durante la cursada y reforzar aquellos que hicieran falta. Además, nos planteó instancias para poder investigar y encontrar soluciones propias y creativas tanto para problemas técnicos como lógicos.

Las pautas dadas por la cátedra nos permitieron diagramar un enfoque profesional al brindarnos herramientas del mundo laboral como GitHub para la correcta organización y verificación sobre el progreso que se iba teniendo día a día y semana a semana.

En lo personal cada integrante tomo una tarea específica sin perder de vista el punto de la unificación de su trabajo con el de los demás integrantes, como a su vez, los aportes subjetivos en cuanto a las soluciones planteadas para cada tipo de problema y funcionalidad que se quiso implementar siendo estas evaluadas desde distintos puntos de vista como (facilidad de uso, optimización, legibilidad, reusabilidad, mantenimiento, depuración, etc.).

Sin dudas a este programa le quedan muchas mejoras que podrían realizarse, muchas puramente estéticas, y la implementación de diversas funcionalidades que podrían lograr una experiencia más deleitable.

- Algunas actualizaciones que podrían hacerse son:
 - Guardar la Lista de Palabras en un archivo para poder verificar aquellas que no formen parte del diccionario de Pattern.
 - Mejorar la interfaz gráfica del tablero con un diseño temático y relacionar las palabras permitidas a esta temática.
 - Incluir una opción para jugar contra una o varias personas y no siempre la máquina.

REFERENCIAS:

- **PySimpleGUI:**
 - <https://pysimplegui.readthedocs.io/en/latest/>
- **Pattern:**
 - <https://github.com/clips/pattern>
- **Documentación oficial de Python:**
 - <https://docs.python.org/3/>
- **Repositorio del juego::**
 - https://github.com/TomasArrue/python_facu