



SCRABBLEAR.PY

Seminario de Lenguajes Python 2020

GRUPO 11

—Manuel Rúa—

1. Índice.....	1
2. Introducción.....	2
3. Reglas y condiciones	3
4. Marco Teórico.....	4
4.1. PySimpleGUI	4
4.2. Pattern.....	4
5. Problemas durante el desarrollo	5
6. Consideraciones éticas	7
7. Bibliografía	8
8. Anexo 1: Guía de usuario	9

Introducción

El proyecto de SrabbleAR.py es el trabajo final de la catedra de Seminario de Lenguajes Python 2020 de la Facultad de Informática, de la Universidad Nacional de La Plata. Con este fin se propone crear una reversión del clásico juego estadounidense Scrabble (wikipedia.org, s.f.) con algunas reglas propias y limitaciones para facilitar su implementación, pero manteniendo la esencia del juego original. El jugador se enfrentará a la computadora colocando letras en un tablero para formar palabras y sumar puntos, haciendo uso de las distintas casillas especiales que multiplicaran el valor de la letra, de la palabra o, penalizaran al jugador restando el valor de la ficha. El programa consta de un menú principal, una pantalla de configuración, una tabla con las mejores puntuaciones y una pantalla donde estará el juego principal. El programa debe estar realizado en Python y debe utilizar a menos las librerías *Pattern* (De Smedt & Daelemans, s.f.) y *PySimpleGUI* (The PySimpleGUI Organization, s.f.)

Reglas y condiciones

Al tratarse de un primer proyecto utilizando el lenguaje de programación Python se debieron plantear algunas limitaciones y reglas especiales para facilitar la implementación del juego. A diferencia del juego original, en `ScrabbleAr.py` no está permitido cruzar palabras y tampoco se permite cualquier tipo de palabra, en las dificultades media y difícil solo se está permitido formar verbos y adjetivos, y en la dificultad fácil también podrán ser sustantivos y pronombres.

Al comienzo del juego se reparten siete fichas a cada jugador, cada una contiene una letra y un puntaje. En su turno el jugador deberá formar una palabra colocando de a una por vez, de izquierda a derecha o de arriba hacia abajo, letras de su atril, en el tablero. Si la palabra formada es correcta, o sea, es válida y, además, es de alguno de los tipos permitidos, se procederá a realizar lo siguiente: se sumarán los puntos equivalentes a todas las letras que componen la palabra, aplicando los multiplicadores que correspondan según la posición en el tablero que ocupe cada ficha y se repondrán las fichas faltantes en el atril con letras aleatorias de la bolsa de letras.

En caso de que el jugador no pueda generar ninguna palabra podrá cambiar tantas letras como desee de su atril utilizando el botón “cambiar”. Se seleccionará las letras que desea devolver al pozo común de fichas y se le repartirá nuevas. Esta opción solo estará permitida 3 veces por partida.

El turno de juego de la computadora se inicia cuando la palabra del jugador es aceptada, cuando termina el tiempo del turno, cuando se realiza un cambio de fichas o cuando se presiona el botón “pasar”. Esta intentará generar al menos una palabra con las fichas de su atril, la colocará de forma aleatoria en el tablero de juego y sumará puntos de la misma forma que el jugador.

Cuando la partida llegue a su fin, ya sea porque se cumplió el tiempo estipulado o, porque no existen fichas suficientes para rellenar el atril del jugador o de la computadora. El juego restará, a los puntos acumulados durante la partida, las fichas que tenga cada uno en su atril y dará un ganador. Además, de forma automática el juego guardará la fecha, el puntaje obtenido y el nivel de dificultad para agregarlo a una tabla de clasificaciones.

El juego también tiene la posibilidad de guardar una partida para volver a retomarla en otra ocasión. En esta implementación solo se podrá tener una partida guardada y el juego preguntará al jugador si desea continuarla cada vez que inicia una nueva partida.

Marco Teórico

PySimpleGUI:

Se trata de una librería de *Python* que permite generar una GUI, del inglés *graphical user interface*, o interfaz gráfica de usuario. Es desarrollada desde 2018, en el momento del desarrollo se utilizó la versión más actualizada (4.18.0). Es una librería muy completa que cuenta con múltiples características que facilitan la implementación de una interface gráfica, ya sean textos fijos, botones, casillas de verificación, *sliders*, iconos, imágenes, ventanas emergentes, entre muchos otras.

En el caso de *ScrabbleAr* se utilizó para en todas las pantallas, el menú principal, el tablero de juego, los atriles y todos los elementos visibles del programa.

Pattern:

Es una librería de *Python* con múltiples funciones entre las que se destacan:

- *Data Mining* (minería de datos): obtiene información de páginas webs como Google, Twitter o Wikipedia
- *Natural Language Processing* (procesamiento del lenguaje natural): analiza textos e identificar tipos de palabras y tiempos de conjugación
- *Machine Learning* (aprendizaje automático): genera modelos de espacio vectorial, agrupamientos y clasificaciones

Para el desarrollo de *ScrabbleAr* se implementó el modulo *Pattern.es* para la validación de las palabras, ya que permite comprobar si un texto es efectivamente una palabra en español e identificar qué clase de palabra es (verbo, sustantivo, adjetivo, pronombre).

Problemas durante el desarrollo

Durante el desarrollo del software surgieron una serie de dificultades que se pueden separar en tres grandes grupos, por un lado, las complicaciones de trabajar en un desarrollo totalmente remoto, los problemas ocasionados por las distintas librerías y los cambios en las consignas originales del trabajo.

Trabajo en equipo y desarrollo remoto.

El aislamiento social obligatorio a causa de la pandemia del COVID-19 obligo a desarrollar todo el proyecto de manera remota. Esto dificulto recibir una retroalimentación por parte de los profesores y ayudantes, pero la mayor complicación fue que uno de los integrantes del grupo decidiera abandonar la materia al poco tiempo de comenzar con el proyecto.

Como el trabajo estaba pensado inicialmente para ser realizado en grupos de dos o tres integrantes, ser un solo integrante generó una serie de dificultades como una mayor carga de trabajo o la falta de múltiples visiones a la hora de buscar la solución a los problemas que fueron surgiendo.

Librerías.

Para la validación de las palabras se utilizó *Pattern.es*, la versión en español de *Pattern*, al ser una versión en constante desarrollo muchas de las funciones presentaban algunas fallas. Por defecto reconocía palabras que no existían, algunas combinaciones aleatorias de letras eran consideradas como verbos o adjetivos, para evitar este tipo de problemas se optó por una triple verificación. Antes de evaluar a qué tipo pertenece cada posible palabra, ésta es buscada en dos diccionarios internos de *Pattern.es* llamados *lexicon* y *spelling*, (Pablo, s.f.). Si se encuentran en ambos, se puede considerar que es una palabra válida y luego realizar un método para comprobar su tipo de palabra.

La interface gráfica también generó una serie de problemas. Se utilizaron múltiples pantallas para que el usuario pudiese acceder a todo el contenido del programa. El usuario, por ejemplo, puede acceder al menú de configuración, después ver los mejores puntajes, después volver a la configuración y por último iniciar el juego. Esto generaba un error que detenía por completo la ejecución del programa, se detectó que *PySimpleGUI*

guardaba algunas de las variables que se utilizaban para generar la interface. Esto causaba que cuando se intentaba abrir una ventana que ya se había abierto anteriormente se sobrescribieran variables y esto generaba los errores. La solución fue reacomodar el código para que toda la interface de una ventana se generara dentro del método *main()* de cada módulo, para que siempre que se llamara a alguno de estos métodos la interface se generara nuevamente desde cero.

Cambios en las consignas originales.

A medida que se avanzó con el proyecto se tomaron algunas decisiones que alteraban las consignas originales presentadas por la cátedra para facilitar la implementación o con el fin de generar una mejor experiencia de usuario.

En un comienzo el nivel de dificultad “fácil” aceptaba solo sustantivos, verbos y adjetivos, el nivel “medio” solo verbos y adjetivos, y el nivel “difícil” iba a ser una combinación aleatoria de tipos. Esto generó muchos problemas en todos los grupos por las fallas que presenta *Pattern.es* para reconocer los tipos de algunas palabras. Esto se solucionó modificando los tipos de palabras aceptados en cada nivel, el nivel “fácil” ahora aceptaba cualquier tipo de palabra y los niveles “medio” y “difícil” solo adjetivos y verbos.

Otra de las consignas dadas por la cátedra consistía que el usuario podría modificar cada ficha que había en juego y el puntaje de cada una. Esto causaba que el juego en ocasiones se volviera imposible de jugar: como el usuario podía modificar la cantidad de fichas en juego existía la posibilidad de bajar demasiado la cantidad de algunas letras y con esto hacer que fuese imposible armar palabras. Para evitar este tipo de problemas se optó por eliminar por completo la posibilidad de que el usuario modificara estos valores, se creó un archivo que contuviera distintas cantidades de letras y distintos puntajes para cada nivel de dificultad (tabla 1). Es posible modificar este archivo en un futuro, si fuese necesario.

Consideraciones Éticas

Desde un comienzo se buscó que el programa fuera lo más accesible posible. Para ello, se utilizaron librerías e imágenes que son de uso libre. Además, desde el comienzo, se pensó el software para que funcione en los dos de los sistemas operativos más utilizados, Windows 10 y GNU/Linux (Ramirez, 2019).

En el repositorio se indica que el código cuenta con licencia *GNU General Public License v3.0* (Free Software Foundation, 2017), esto garantiza que cualquier persona u organización tenga la libertad de usar, estudiar, compartir, copiar y modificar el software, y para facilitarlo se documentaron las distintas funciones para que futuros desarrolladores puedan estudiarlas, copiarlas o modificarlas de una manera más sencilla.

Bibliografía

- De Smedt , T., & Daelemans, W. (s.f.). Pattern. Obtenido de [www.GitHub.com: https://github.com/clips/pattern](https://github.com/clips/pattern)
- Free Software Foundation. (29 de junio de 2017). GNU. Obtenido de <https://www.gnu.org/licenses/gpl-3.0.html>
- GIT. (s.f.). GIT. Obtenido de <https://git-scm.com/>
- Pablo, J. (s.f.). [www.github.com/pibytes](https://github.com/pibytes). Obtenido de https://github.com/pibytes/notas_python/blob/master/patternES.md
- Ramirez, P. (22 de diciembre de 2019). itsoftware. Obtenido de <https://itsoftware.com.co/content/sistemas-operativos-mas-usados/>
- The PySimpleGUI Organization. (s.f.). PySimpleGUI. Obtenido de [www.readthedocs.org: https://pysimplegui.readthedocs.io/en/latest/](https://pysimplegui.readthedocs.io/en/latest/)
- wikipedia.org. (s.f.). wikipedia.org. Obtenido de <https://es.wikipedia.org/wiki/Scrabble>

Anexo: Guía de Usuario

Requisitos:

- Python 3.6.8
- PySimpleGUI 4.18.0
- Pattern 3.6
- Windows 10 o Ubuntu 20.04 LTS



Imagen 1: Menú principal

La primera pantalla que ve el usuario cuando ejecuta ScrabbleAr.py es el menú principal del juego (Imagen 1). En este menú podremos acceder a las distintas funcionalidades del software: se podrá iniciar el juego, entrar a la sección configuración, ver los mejores puntajes guardados y acceso rápido al repositorio del proyecto.



Imagen 2: Pantalla de configuración

En esta pantalla se podrá configurar la duración de cada turno (en segundos), la duración total de la partida (en minutos), y la dificultad de la partida (Imagen 2).

La dificultad modifica tanto el tablero de juego (Imagen 4) como la cantidad y el puntaje de cada letra (Tabla 1)

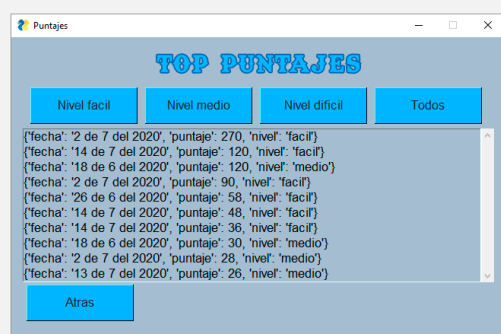
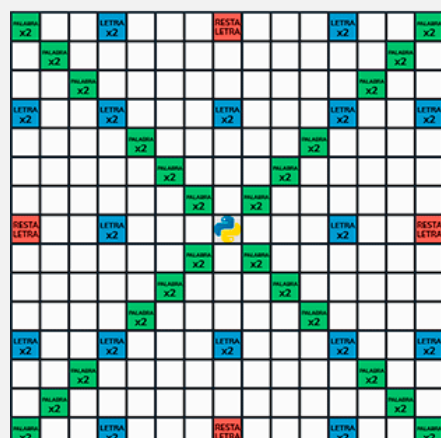


Imagen 3: Mejores puntajes

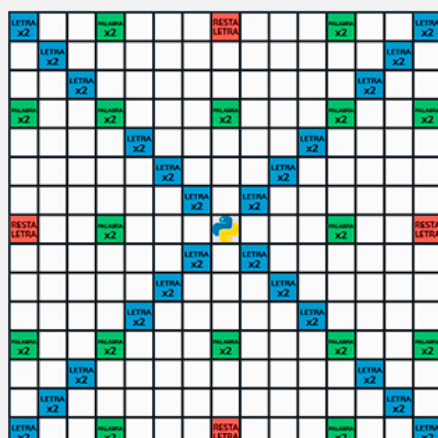
Al finalizar cada partida se almacena el puntaje total obtenido junto con la fecha y el nivel de dificultad jugado (Imagen 3).

Cada botón muestra los mejores diez puntajes de cada dificultad. El botón “todos” muestra los mejores diez puntajes sin importar la dificultad.

FACIL



MEDIO



DIFICIL

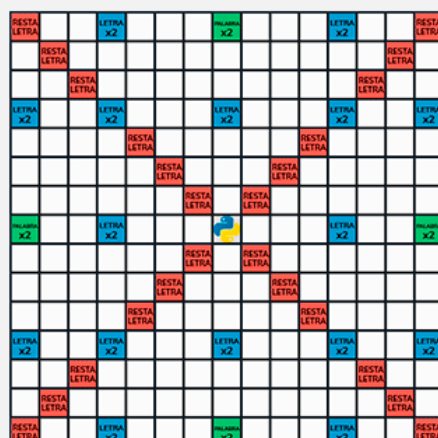


Imagen 4: Tableros de juego

	FACIL		MEDIO		DIFICIL	
	Puntaje	Cantidad	Puntaje	Cantidad	Puntaje	Cantidad
A	2	22	1	11	1	11
B	6	6	3	3	2	3
C	4	8	2	4	1	4
D	4	8	2	4	1	4
E	2	22	1	11	1	11
F	8	4	4	2	2	2
G	4	4	2	2	1	2
H	8	4	4	2	2	2
I	2	12	1	6	1	6
J	12	4	6	2	3	2
K	16	2	8	1	4	1
L	2	8	1	4	1	4
M	6	6	3	3	2	3
N	2	10	1	5	1	5
O	2	16	1	8	1	8
P	6	4	3	2	2	2
Q	16	2	8	1	4	1
R	2	8	1	4	1	4
S	2	14	1	7	1	7
T	2	8	1	4	1	4
U	2	12	1	6	1	6
V	8	4	4	2	2	2
W	16	4	8	2	4	2
X	16	2	8	1	4	1
Y	8	2	4	1	2	1
Z	20	2	10	1	5	1

Tabla 1: Fichas de en juego