

```
def seminario_de_lenjuajes(Python):
```

```
    title =
```

```
        ' INFORME / n  
        FINAL '
```

```
    docentes = ['CLAUDIA BANCHOFF',  
                'VIVIANA HARARI']
```

```
    alumnos = ['Juan Carlos Schachner',  
               'Martín Vallejo']
```

```
    año = 2020
```

```
    UNLP['Facultad de Informática']
```

índice

Introducción	1
Objetivos	2
Reglas de ScrabbleAr	3
La partida	3
Configuración	4
Marco teórico	5
PySimpleGUI	5
Pattern	5
Desarrollo de la aplicación	7
Consideraciones del equipo	7
Metodologías	8
Problemas y soluciones	9
Conclusiones y reflexiones finales	10
Dominio de Python y paquetes	10
Aplicación ScrabbleAR	10
Código abierto y documentación	11
Metodologías	11
Referencias	12
Anexo I: Guía de usuario	13

introducción

El presente informe detalla el proceso de desarrollo de la aplicación ScrabbleAR en el contexto de la materia Seminario de Lenguajes - opción Python, de la Facultad de Informática de la UNLP.

ScrabbleAR es un juego con interfaz gráfica para un jugador basado en el Scrabble, aunque con algunas variaciones respecto a sus reglas. El programa fue desarrollado en el lenguaje Python, haciendo uso de sus módulos de base y las librerías de código abierto PySimpleGUI y Pattern.

En este texto se elaborará sobre el funcionamiento de la aplicación, algunas características de las librerías anteriormente mencionadas y se hará foco sobre aquellos aspectos del desarrollo que implicaron dificultades u oportunidades de aprendizaje identificables para los integrantes del grupo.

objetivos

Al concluir este trabajo se espera:

- Tener una comprensión más acabada del lenguaje de programación Python.
- Aprender el uso e implementación de las funcionalidades de los paquetes PySimpleGUI y Pattern.
- Utilizar estos conocimientos para desarrollar de forma satisfactoria una aplicación con base en los lineamientos preestablecidos por la cátedra.

Para este último punto resulta relevante definir qué entiende el equipo por “satisfactorio” en el contexto de este trabajo. Principalmente se refiere a lograr la implementación -sin errores graves, que detengan el proceso de uso- de las funcionalidades solicitadas. Adicionalmente, aunque en menor grado, implica producir un código homogéneo en estilo, que refleje buenas prácticas de programación, y obtener una aplicación de fácil uso y comprensión por parte de terceros.

reglas de ScrabbleAR

la partida

Es el componente principal del juego. En cada una de ellas participan dos jugadores; un usuario humano contra la computadora. Cada uno de estos tiene a su disposición un atril con siete fichas, cada una de las cuales tiene una letra y un valor asignado. El objetivo del juego es obtener el mayor puntaje posible formando palabras en el tablero con las fichas disponibles en el atril.

Componentes elementales

Fichas: son la representación en el juego de una letra con el respectivo puntaje asignado según el modo de juego o la configuración del usuario.

Bolsa: es un repositorio de todas las fichas aún no distribuidas a los jugadores.

Tablero: una matriz rectangular de casillas en las cuales se irán colocando las fichas con las que los jugadores intentarán formar sus respectivas palabras. Cada casilla puede modificar el puntaje final de la palabra, habiendo 4 tipos de casilla de bonificación, un tipo de casilla de descuento y casillas sin modificador. Las casillas que agregan puntos pueden duplicar o triplicar el valor de la ficha que se coloque encima, o duplicar e incluso triplicar el valor total de la palabra. Las casillas que descuentan puntos son de un sólo tipo y restan 3 puntos al valor de la ficha que se coloque encima.

Instancias de juego

Inicio: el tablero y los atriles comienzan vacíos, y el reloj pausado. El juego no comienza oficialmente hasta que el usuario hace clic en el botón "comenzar"; momento en el cual se repartirán 7 fichas a cada jugador, se elegirá al azar cuál de los dos comienza su turno, y se iniciará el cronómetro.

Turno: si es el turno del usuario, este intenta armar una palabra en el tablero con las fichas que dispone en su atril. Las palabras se pueden escribir tanto en forma vertical como horizontal, estas serán interpretadas de izquierda a derecha o de arriba hacia abajo respectivamente y todas sus fichas deben ubicarse en casillas adyacentes. Si está convencido de la existencia de su palabra, presiona en el botón "jugar" y el programa evaluará si ésta realmente existe en el diccionario. Si existe, se calculará el puntaje de la palabra teniendo en cuenta el valor de cada ficha en conjunción con los posibles modificadores de cada casilla ocupada. Este puntaje se agrega al del jugador y, de ser posible, se colocarán en su atril tantas fichas de la bolsa como fueran utilizadas para conformar la palabra, dando por finalizado el turno.

En cualquier momento de su turno, el jugador puede decidir cambiar fichas, presionando el botón "cambiar fichas". Esto le permitirá seleccionar cuántas fichas desee y, presionando "cambiar", entregarlas a la bolsa, para que ésta le devuelva tantas como haya recibido. Si se concreta el cambio de fichas, el turno se da por finalizado. El usuario tiene a su disposición 5 cambios sin costo, a partir del 6to, perderá 2 puntos por cada nuevo cambio de fichas, aunque nunca puede quedar por debajo de cero.

Por último, el jugador tiene la opción de pasar su turno sin realizar ninguna acción, haciendo click en el botón "pasar turno", esta acción no tiene límites ni costo alguno.

El comportamiento de la computadora es similar al del jugador, ésta buscará armar una palabra en el tablero; si lo logra, se procederá de la misma forma que en el caso

del usuario, agregando los puntos correspondientes, recibiendo fichas de reemplazo y finalizando el turno. De no poder armar una palabra, la computadora cambiará dos fichas al azar de su atril con la bolsa, finalizando así su turno. Los cambios de fichas para la IA nunca implican un costo adicional.

Fin: los turnos se seguirán intercalando hasta que se acabe el tiempo, hasta que el usuario decida finalizar la partida (haciendo click en el botón "terminar"), o hasta el momento en que, una vez jugada una palabra, la bolsa no disponga de suficientes fichas como para llenar ese atril. En cualquiera de estas instancias, se proclamará un vencedor de acuerdo a los puntajes obtenidos y, si el puntaje del jugador es suficiente para entrar en el top 10 de su respectiva dificultad de juego, se le anunciará este hecho, para finalmente regresar al menú principal

Posponer: en todo momento, mientras la partida esté activa, el usuario tiene la posibilidad de guardarla para continuar posteriormente. Para ello debe hacer click en el botón "posponer", lo cual lo dirigirá nuevamente al menú principal. Para retomarla, sólo debe hacer click en el botón "continuar partida" del menú principal. Se permite el guardado de una sólo partida en todo momento, por lo que sucesivos "posponer" generarán las correspondientes sobreescrituras, y sólo se podrá acceder a la última de estas. Cabe aclarar que si se inicia un nuevo juego, la partida archivada será eliminada.

configuración

Desde el menú principal, y en la ventana previa al inicio de cada partida, el usuario tiene acceso a la modificación de múltiples variables de juego, a fin de tener una experiencia personalizada. Existen tres modos de juego pre-configurados, cuyas características no pueden modificarse a excepción del tiempo, y un único modo "personalizado". Las variables susceptibles de alteración a las que tiene acceso el usuario en este último caso son:

Tiempo: el único parámetro que puede modificarse en cualquiera de los modos de juego. Determina la duración máxima de una partida antes de dar por concluido el juego.

Letras: tenemos una lista desplegable de letras, al seleccionar una, se actualizan las barras deslizadoras con las cuales podemos seleccionar la cantidad de veces que aparece esa letra en la bolsa al igual que el puntaje que otorga. Se incluye una lista desplegable en la que podremos elegir la dificultad del tablero así como la de la inteligencia artificial. Cada vez que se realicen cambios en una letra, se deberá guardarlos o estos cambios serán descartados

Tipo de tablero: hay tres variantes de tablero, cada una con sus dimensiones particulares y diversa distribución de casillas comunes, de bonificación y de descuento.

Dificultad de la IA: existen tres dificultades de contrincante; una por cada dificultad predeterminada: Fácil, Medio, y Difícil, cada una de estas seleccionará, de todas las palabras que pueda construir con las fichas de su atril, la que menor puntaje dé en el caso Fácil, la que mayor puntaje dé en el caso Difícil, y una de puntaje intermedio en el caso Medio. Todas las versiones de IA seleccionan casillas al azar donde colocar la palabra, independientemente de los modificadores que éstas contengan.

marco teórico

PySimpleGUI

La evidente complejidad del proyecto hace casi inevitable la incorporación de elementos de interfaz gráfica de usuario que faciliten su comprensión y empleo, sin pasar por alto el incremento en atractivo visual que su aplicación implica. Si bien el contexto de este trabajo apunta al empleo de una librería específica, es importante en primera instancia formar algunas definiciones generales.

Se entiende por interfaz gráfica de usuario (GUI por sus siglas en inglés) al conjunto de signos a través de los cuales una persona puede interactuar con un dispositivo electrónico determinado. Como su nombre indica, se distingue de otros tipos de interfaz digital (interfaz de línea de comando) por su énfasis en el empleo de elementos gráficos (widgets) que son representaciones simbólicas de las acciones disponibles para el usuario (íconos de carpetas en sistemas de archivos, un tubo de teléfono como indicador de acceso a la realización de llamadas en las interfaces de telefonía móvil), o que indican su accionabilidad por su similitud con objetos del mundo real (efectos de profundidad en botones).

La presencia de una GUI implica casi necesariamente una mayor complejidad en el código de un programa, ya que requiere agregar una capa adicional de instrucciones que identifiquen los diferentes elementos, y la forma en la que éstos se relacionan entre sí y con el funcionamiento principal (las acciones que el usuario desea llevar a cabo y cómo éstas se ejecutan en el programa).

La librería estándar de Python para el desarrollo de interfaces es Tkinter, una adaptación del lenguaje Tcl/tk (siendo tk específicamente la librería de widgets gráficos). PySimpleGUI, por otro lado, es un módulo que adapta tkinter¹ para su uso a través de una sintaxis más sencilla, y con algunas funcionalidades creadas en forma nativa.

Se trata de un proyecto abierto, aunque principalmente mantenido por un equipo reducido, con extensa documentación y atención de problemas. Su principal ventaja desde el punto de vista de este trabajo es la de proveer una interfaz relativamente sencilla para el desarrollo de GUI. A mayor escala, sin embargo, otro de sus beneficios es el de permitir cambiar el framework de base sin necesidad de modificar más que la sentencia import, pero manteniendo el mismo resultado visual.

Pattern

De la naturaleza de este proyecto deriva la evidente necesidad de, de alguna forma poder reconocer la validez de las palabras que ponen los jugadores en el tablero y, en un nivel más invisible al usuario común, contar con una herramienta relativamente eficiente que permita a la computadora construir palabras válidas con las fichas que disponga en su atril. Siendo el procesamiento de lenguaje natural algo que excede al alcance de este proyecto (y de la asignatura en general), se recurrió a una librería preexistente -Pattern- que carga con la mayor parte de la responsabilidad por llevar a buen término las tareas anteriormente mencionadas.

¹ En realidad diferentes variantes de PySimpleGUI funcionan como wrappers de diferentes librerías. La variante que se instala de forma predeterminada, y fue empleada para el desarrollo de este trabajo, es la que utiliza tkinter, que por otro lado es la más completa.

De acuerdo con sus creadores, "Pattern es un paquete para Python 2.4+ con funcionalidades de minería web, procesamiento de lenguaje natural, aprendizaje de máquinas y análisis de redes." (De Smedt & Daelemans,). De esta descripción resulta claro que las capacidades de la herramienta superan con creces los requerimientos de este proyecto, pero no por ello deja de ser idónea para el mismo. Ante todo, una de las principales ventajas del paquete es su facilidad de uso (uno de los objetivos de sus creadores), al menos en lo que a las funciones utilizadas por el equipo se refiere.

Con todo esto, sin embargo, algo que vale la pena destacar es que el paquete, publicado en 2012, no ha sido extendido en los últimos 2 años, por lo que no ha sido adaptado oficialmente a las últimas versiones de Python, de modo que todas las implementaciones deben desarrollarse considerando la los puntos de compatibilidad con versiones anteriores de Python².

De todas las utilidades disponibles de este paquete, se utilizaron los módulos `.es` y `.web`. El primero es el módulo de procesamiento de lenguaje natural en español, que no es más que una adaptación al castellano del código original desarrollado en inglés. De este sólo se emplearon sus diccionarios de base lexicon y spelling y el módulo `parse`, que identifica las cualidades lingüísticas de una palabra o expresión -esta última se empleó específicamente para reconocer la clase de palabra a la que pertenecía un string dado-. El módulo `.web`, por otro lado, se utilizó para realizar consultas al sitio `wiktionary.es` y verificar su contenido para asegurar la validez de las palabras utilizadas.

² La incompatibilidad está dada por unas pocas líneas de código, que pueden ser modificadas manualmente, como se explica en esta publicación:

<https://gist.github.com/juanpenia/4ea3f58aedbfaefd44f97ad905992ccf>

desarrollo de la aplicación

consideraciones del equipo

El trabajo desarrollado por este grupo no se amolda completamente a las pautas del trabajo según la propuesta de la cátedra, en este apartado se discutirán aquellos aspectos en los que el programa difiere en alguna medida de lo solicitado

Top 10 en dificultad personalizada

Debido a la evidente disparidad que puede generarse en los puntajes de diversos jugadores en la dificultad personalizada (a través de la modificación de cantidades de letras y sus respectivos valores para una partida), el equipo decidió no implementar un módulo de top 10 para esta dificultad.

Filtros más estrictos para determinar la validez de una palabra

El grupo considera injusto para el usuario que un juego que considera válida a un grupo de palabras inexistente en la lengua castellana, aún más cuando este repertorio de "palabras" no es conocido por el usuario. Por lo que se empleó un filtro más estricto (provisto por el mismo módulo), que permite garantizar en mayor medida la validez de una cadena de caracteres.

Inicialmente se pudo reconocer que la mayor parte de estas cadenas que se consideran inválidas son identificadas por Pattern como sustantivos. De esta forma, cuando el jugador computadora define una nueva palabra a partir de sus fichas disponibles, se evalúa su clase con el método `parse` y, en caso de tratarse de un sustantivo, se recurre al módulo web de la librería.

Las consultas al módulo web retornan un objeto que es una lectura de la página resultante de buscar un determinado término en el sitio `wiktionary`, retornando un objeto vacío en caso de no encontrarse la palabra en dicha página web. Sin embargo, la existencia de una página con este término no es garantía de la pertenencia del mismo al léxico español, puesto que el sitio es un diccionario multilingüe. De esta forma, la única manera de reconocer si una palabra es susceptible de utilizarse, es identificar la presencia del encabezado "Español" en el objeto resultante de la búsqueda - lo que indica que esa cadena tiene una entrada en la sección español.

Utilización de módulos no incluídos en los contenidos de la asignatura

Más allá del requisito de poseer una conexión a Internet, la solución descrita anteriormente trae aparejada una traba a la buena usabilidad: la comprobación de validez de las palabras lleva mucho tiempo, durante el cual la partida se encuentra bloqueada.

Para salvar este problema, se decidió recurrir a la programación en hilos, en donde se generan flujos de ejecución a los cuales se le denomina hilo o subproceso, pudiendo cada uno de estos realizar o no una misma tarea. Haciendo uso de la librería `concurrent.futures`, el proceso de validación de la computadora forma tantos hilos como palabras a validar, generando así consultas casi en paralelo a los servidores de `wiktionary`. Como consecuencia se obtiene una notable reducción del tiempo requerido por la inteligencia artificial para la selección de la palabra a jugar. Cabe aclarar, sin embargo, que sigue existiendo un tiempo durante el cual la partida se congela, aunque este se ha visto disminuido considerablemente.

Redefinición del nivel

Las indicaciones del trabajo solicitan que, según cada dificultad, se permitan diferentes clases de palabras; sustantivos, verbos y adjetivos en fácil; verbos y sustantivos en medio, y una categoría al azar en difícil. Dadas las dificultades anteriormente mencionadas que se encuentran en el uso de algunas categorías de palabras, y considerando que las limitaciones atribuidas a cada nivel implicaría un mayor perjuicio a la capacidad del jugador humano que a la máquina, el equipo decidió redefinir el parámetro principal de la dificultad.

Según esta nueva concepción, los diferentes niveles conllevan modificaciones en cuatro aspectos esenciales de una partida:

- Cantidad de fichas disponibles por letra: a medida que aumenta la dificultad, la proporción de vocales y otras letras de uso frecuente en el español³ se ve reducida respecto al total.
- Tablero de juego: cada nivel tiene asociado un tablero específico. No se puede afirmar que cada diseño de tablero implique objetivamente una mayor o menor dificultad.
- Longitud de palabras generadas por la computadora: en la medida que incrementa la dificultad, la máquina seleccionará palabras más largas entre aquellas que pueda generar con sus fichas, por lo que el puntaje obtenido será mayor.

Inclusión de pautas visuales en el tablero durante el desarrollo de una jugada

A falta de retornos táctiles, la virtualidad ofrece la posibilidad de darle al usuario múltiples indicadores visuales que no serían posibles en una versión analógica del juego. En este sentido, el proceso de construcción de palabras en el tablero es tierra fértil para el agregado de elementos que ayuden al jugador a comprender mejor la situación y sus posibilidades. Es por esto que se decidió incluir un conjunto de comportamientos de la interfaz que, mientras el usuario coloca fichas en el tablero, marca qué otras casillas contiguas son susceptibles de ser ocupadas.

metodologías

A falta de conocimientos sobre metodologías de desarrollo de software, el proceso de construcción de la aplicación fue guiado por la intuición de los integrantes del equipo. En principio, se trabajó en las funcionalidades más elementales: un menú principal, el acceso a la partida y la visualización de los elementos básicos de esta. A partir de allí se fueron acoplando más funcionalidades, intentando generar soluciones modulares, y buscando el funcionamiento armonioso de las partes, de cuyos resultados las conclusiones dan cuenta. Sin embargo, el proceso no fue planificado u organizado a priori, sino que consistió esencialmente en resolver, en la manera que más ordenada se consideró, la lista de funcionalidades que se pretendía que tuviera el producto final.

Tampoco la división de tareas fue excesivamente planificada, si no que se fue resolviendo según el avance del proyecto: en la medida que determinada tarea o funcionalidad era completada, el integrante que la hubiera hecho asumía una nueva responsabilidad con otra tarea.

³ Para establecer estos valores de base se utilizó la información en Pratt (1939).

problemas y soluciones

Durante el desarrollo del programa, como es de esperarse, surgieron varios desafíos. De todos estos, a continuación se enumeran aquellos que el grupo considera más relevantes o complejos, junto con la estrategia que se siguió para resolverlos:

- Implementar un temporizador que se actualice constantemente sin tener que esperar que el turno de un jugador termine. Se soluciona usando un timeout en el `window.read()` del tablero del juego que actualiza el reloj cada 500 milisegundos
- Tener una función que valide la palabra correctamente, solucionando el problema de los sustantivos. Se solucionó comprobando los sustantivos con `pattern.web` utilizando `wiktionary`
- Lograr un diseño que permita tener balance en la dificultad de la Inteligencia artificial (IA). Se diseñaron tres métodos para la selección de la palabra que la computadora va a utilizar. Todos los métodos parten con la base de una lista de palabras. En el primer método se utiliza en la dificultad "fácil", se calcula los puntajes de todas las palabras y se utiliza la que menos puntaje otorga. Para el segundo método que es usado en la dificultad "medio", se calcula el puntaje de todas las palabras, se ordenan de menor a mayor, se toma una palabra que esté en el medio, es decir la parte entera de la longitud de la lista dividida entre dos. Y por último en el tercer método, que es utilizado en la dificultad "difícil" y al igual que en métodos anteriores, se calcula los puntajes de cada palabra y se toma la palabra que más puntaje otorga
- Lograr que la IA no demore significativamente en procesar su respuesta. Debido a la demora en validar si las palabras que son candidatos a elegirse son válidas, hicimos uso de la librería `concurrent.futures` que nos permite usar threads. Se generan tantos threads como palabras a validar existen. La ventaja de usar threads es que podemos validar varias palabras casi simultáneamente, lo que reduce considerablemente el tiempo de procesamiento
- Solucionar errores de `pattern.web`. El uso de threads en varias ocasiones genera que las múltiples peticiones hacia `wiktionary` causen errores y detengan la ejecución del programa. Por lo que implementamos dos `try/except` anidados en caso de que se generen dichos errores, que estén contemplados. Luego de implementar la solución no se volvió a tener esa clase de error

conclusiones y reflexiones finales

dominio de Python y paquetes

Dado el contexto de una educación de nivel universitario, es esperable que uno de los principales objetivos de un proyecto de estas características sea el aprendizaje y refuerzo de conocimientos específicos de un determinado dominio. En este sentido, el proyecto propuesto y su desarrollo han colaborado en gran medida a la adquisición de destrezas asociadas a la programación en Python, y al uso de los paquetes requeridos. Por otro lado, las complejidades involucradas en este trabajo también han colaborado a cimentar conceptos técnicos que no son aspecto central de esta materia, como lo son el desarrollo de interfaces, la optimización de algoritmos y la programación en hilos. Todas ellas disciplinas en sí mismas, que sólo han sido abordadas en forma parcial durante la producción de este trabajo, pero no por ello han resultado menos enriquecedoras del saber de los integrantes.

aplicación ScrabbleAR

En términos generales, el equipo considera que el objetivo principal sobre la aplicación fue alcanzado de manera satisfactoria. ScrabbleAR funciona, en la medida en que fue testeado, sin errores mayores. Lo único que impide dar por completamente logrado este objetivo, es el ya mencionado bloqueo de la aplicación durante el proceso de validación de palabras (más específicamente durante las consultas a wiktionary.es). No se considera un error grave en cuanto a la funcionalidad general del juego, pero atenta contra uno de los objetivos secundarios: la facilidad y agrado de uso.

En este sentido también, se da por alcanzado el objetivo, lo que nuevamente no implica la ausencia de aspectos a mejorar. Más allá del problema mencionado en el párrafo anterior, se considera que la interfaz es aún susceptible a mejoras en términos de diseño visual y de usabilidad. Para empezar, salvo por algunos de los elementos del tablero, la aplicación no cuenta con una identidad visual concreta. En cuanto a la usabilidad, si bien se incorporaron pautas visuales para facilitar el juego, se considera que algunos signos sonoros podrían ayudar también en este sentido (indicadores de "palabra jugada", "cambio de turno", "últimos 30 segundos", etc. que faciliten la comprensión de los diferentes sucesos de una partida).

De los procesos que puede llevar adelante un jugador determinado, el menos satisfactorio en términos de su fluidez resulta el del cambio de fichas. Y la necesidad de implementar una ventana de explicación del mismo es un indicador de aspectos a mejorar. Sin embargo, en el tiempo disponible para el desarrollo no pudo encontrarse una solución más sencilla que fuera compatible con el resto de las funcionalidades.

De todos los objetivos, el menor logrado fue el referido al uso de buenas prácticas. Si bien el código de la aplicación se encuentra profundamente documentado, algunos de los módulos implementados y sus relaciones han sido difíciles de mantener, lo cual es evidencia de problemas graves en la legibilidad de este.

Mención aparte merece la evaluación de la homogeneidad del código. En su mayor parte el desarrollo fue llevado adelante de forma más análoga a los conceptos del paradigma procedural. Sin embargo, la implementación de la partida incluye la definición de varias clases que hacen a las funcionalidades principales, y son estas las que más problemas traen en lo que a buenas prácticas se refiere. Algunas de estas clases cumplen tareas excesivamente complejas en un sólo método y general no se respeta a rajatabla el encapsulamiento de los objetos. De esta forma el código resultante, aunque funcional, termina siendo confuso, y los módulos de interfaz entre los objetos de base y los de la GUI se hacen, a entender del equipo, más verbosos de lo necesario.

código abierto y documentación

Resulta indiscutible que sin la existencia de los paquetes de Pattern y PySimpleGUI y su libre disponibilidad, este proyecto no podría haber sido llevado adelante con los conocimientos de los integrantes del equipo. Esto es un gran argumento a favor de la existencia en general de software de código abierto, no sólo por su mera existencia, si no porque además sus funcionalidades están extensamente documentadas, y pueden ser analizadas en profundidad a fin no sólo de comprender de qué manera implementarlas en el código propio, si no también entender sus principios de funcionamiento y, de esta forma adquirir nuevos conocimientos, un aspecto esencial dado el contexto principalmente educativo de esta experiencia.

Más allá de las licencias abiertas, una clara diferencia entre ambos paquetes utilizados, es su documentación y comunidad. Pattern es, como se ha mencionado, una librería para usos relativamente específicos y que no es mantenida con frecuencia. De esta forma, si bien hay documentación disponible y el código se encuentra comentado, no hay demasiada información disponible más allá de estas explicaciones superficiales de funcionamiento. Esto es un fuerte contraste con PySimpleGUI, que cuenta con una extensa comunidad de usuarios y se encuentra aún hoy en desarrollo. Para este último paquete, la documentación es mucho más extensa y pormenorizada, por lo que la barrera de entrada para su comprensión y uso es proporcionalmente más baja.

En ambos casos, de todas formas, incluso siendo el código de libre acceso, sin la documentación apropiada y las comunidades que la amplían y mejoran, la utilización de paquetes externos no sería posible sin conocimientos técnicos más profundos o, cuanto menos, la inversión de gran cantidad de tiempo a la lectura del propio código para comprender la funcionalidad específica de sus módulos.

metodologías

Como se hace evidente en la descripción de metodologías, el desarrollo del proyecto no estuvo guiado por ningún concepto teórico o método preestablecido que no fuera la propia intuición. Si bien el equipo concluye que los resultados han sido satisfactorios, también resulta claro que contar con un marco metodológico bajo el cual llevar adelante un proyecto de estas características, implicaría una mejora sustancial en la fluidez del desarrollo, así como en la calidad técnica del producto final.

Sin embargo, esta experiencia también ha servido para reforzar la idea de que la buena comunicación en un equipo resulta clave para llevar adelante un buen proceso, puesto que fue esta la que permitió al equipo salvar las dificultades un proceso poco organizado.

referencias

Peña, J. (2020, julio 20). *Pattern y Python 3.7+*. Sitio web:
<https://gist.github.com/juanpenia/4ea3f58aedbfaefd44f97ad905992ccf>

De Smedt, T. & Daelemans, W. (2012). *Pattern for Python*. The Journal of Machine Learning Research. 13. 2063-2067.

Pratt, F.. (1939). *Secret and Urgent - The Story of Codes and Cyphers*. Nueva York: The Bobs-Merryl Company. pp.254-255.

Sitio en Github.com del proyecto PySimpleGUI:
<https://github.com/PySimpleGUI/PySimpleGUI>

Sitio en Github.com del proyecto Pattern:
<https://github.com/clips/pattern>

anexo I: guía de usuario

menú principal

Al momento de iniciar la aplicación, esta será la primera ventana con la que cualquier usuario se encontrará, y a partir de la cual se puede acceder a todas las funcionalidades del juego:

- 1. Permite iniciar una nueva partida.
- 2. Permite continuar una partida guardada. Esta opción sólo estará habilitada en cuanto exista dicha partida.
- 3. Permite acceder a la configuración personalizada de fichas, puntajes y tablero.
- 4. Permite acceder a los rankings de mejores diez puntajes por dificultad
- 5. Permite acceder a información del juego.

nueva partida

Si se desea comenzar una nueva partida, primero se debe hacer click en el botón correspondiente del menú principal. Así se presentará una pantalla que permite seleccionar las características que el jugador desea para la misma:

- 1. Lista desplegable de selección de dificultad (Las características de cada una se detallan en otras secciones).
- 2. Lista desplegable para la selección del tiempo de duración de la partida.
- 3. Espacio para que el jugador ingrese el nombre con el que desea identificarse. Si no se modifica, se mantendrá en sucesivas partidas. En caso de entrar en un ranking, quedará registrado con este nombre.
- 4. Permite entrar a la ventana de juego.
- 5. Permite configurar la dificultad personalizada.



Fig 1. ventana de menú principal sin partida guardada

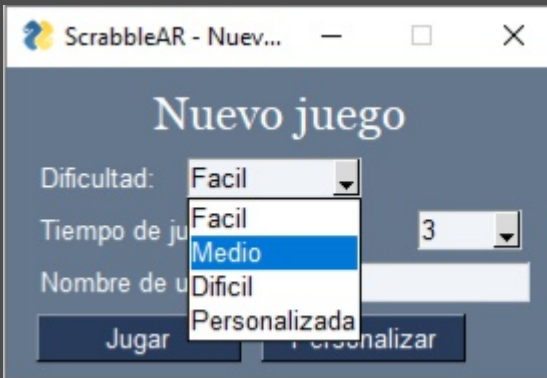
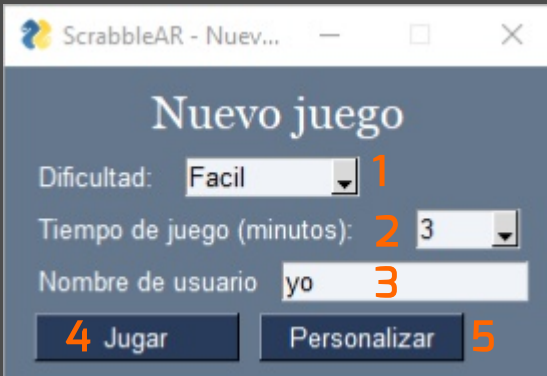


Fig 2 (sup.). ventana de configuración de partida. Fig 2.1 (inf) selección de dificultad.

desarrollo de una partida

Una vez presionado el botón "jugar" del menú de nueva partida, se abrirá una ventana nueva, como la que se muestra a continuación (puede variar el tablero según la configuración). La mayor parte de los botones iniciarán desactivados y los atriles vacíos.

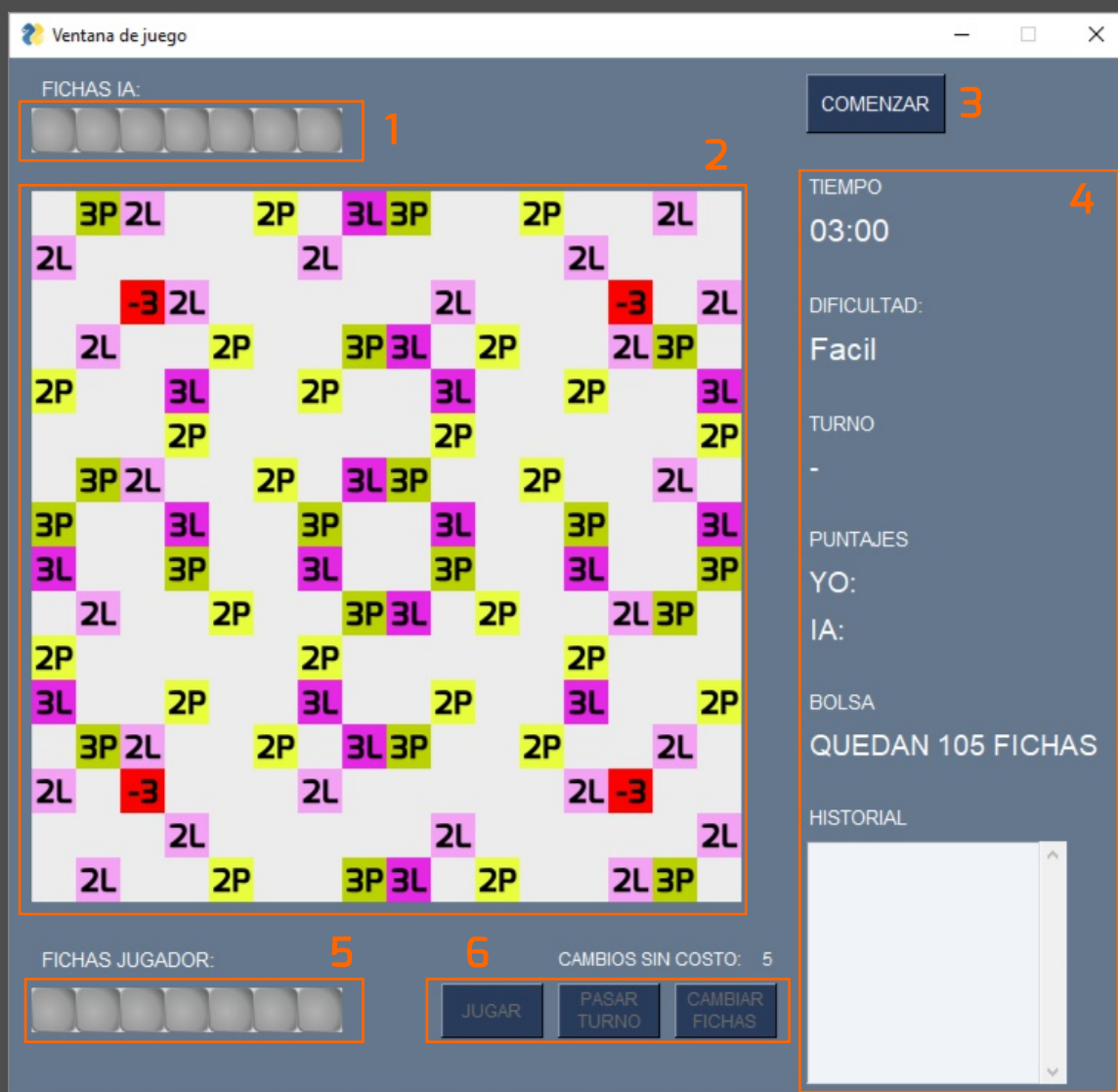


Fig 3. Partida aún no iniciada, donde pueden apreciarse los elementos esenciales de la interfaz

1. Atril de la computadora, aquí aparecerán las fichas del jugador oponente, aunque se verán siempre con el signo "?".

2. Tablero de juego, aquí se colocarán las fichas buscando construir palabras.

3. Botón de comienzo de la partida. En una partida no iniciada, es el único elemento activo.

4. Indicadores de estado de la partida: reloj que marca el tiempo restante, dificultad en la que se está jugando, a quién corresponde el turno, puntajes, cantidad de fichas restantes en la bolsa, e historial de jugadas.

2. Atril del jugador, el jugador podrá ver y usar sus fichas desde aquí.

3. Botones del jugador.

Si se hace click en el botón "comenzar" se dará inicio efectivamente a la partida. El reloj comenzará la cuenta regresiva y de distribuirán las fichas a los jugadores. Habilitándose los botones correspondientes al jugador en caso de ser su turno.



Fig 3.1. Partida recién iniciada. Aquí pueden verse los cambios respecto a la figura anterior, en particular; se han distribuido 14 fichas, por lo que en la bolsa ahora quedan 91, es el turno del jugador "yo" (usuario), por lo que tiene habilitados los botones "pasar turno" y "cambiar fichas". A su vez, puede notarse que en donde aparecía el botón "comenzar" (arriba a la derecha), ahora hay dos botones: "posponer" y "terminar". Los pormenores de estas acciones se definirán más adelante.

Colocar fichas en el tablero

El objetivo principal del juego es armar palabras con las fichas disponibles en el propio atril, y a partir de ello, eligiendo las posiciones más convenientes en el tablero, obtener el mayor puntaje posible. A continuación se describe el proceso por el cual se pueden colocar las fichas en el tablero, y comprobar la validez de la palabra formada.

El primer paso es observar las fichas disponibles y tratar de encontrar entre ellas una palabra de dos o más letras. Una vez hecho esto, para colocarlas en el tablero debe primero hacerse click en la ficha correspondiente del atril, y luego en la posición del tablero en la que se desea colocar.

Para deshacer esta decisión se puede hacer click nuevamente en la ficha (sea en el atril o en el tablero). También es posible, una vez seleccionada una ficha del atril, colocarla en lugar de otra ya ubicada en el tablero. Esta acciones son válidas sólo para fichas que están siendo colocadas en un mismo turno. Las palabras jugadas con anterioridad no pueden ser modificadas.



Fig 3.2. Ficha colocada en el tablero. Nótese el resaltado de las casillas susceptibles de recibir nuevas fichas para armar una palabra.

Una vez colocadas todas las fichas en el tablero, el jugador debe hacer click en el botón "jugar" a fin de corroborar la validez de la palabra. Si ésta es inválida se mostrará una advertencia, y el jugador podrá seguir intentando armar otra palabra.



Fig 3.3. Ya se ha formado una palabra de dos o más letras, por lo que se ha habilitado el botón "jugar".

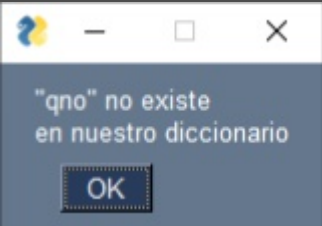


Fig 3.4. Advertencia de palabra no válida.

Otras opciones durante el turno

Si el jugador considera que no puede armar ninguna palabra con las fichas que tiene disponibles, puede elegir uno de dos caminos: Puede pasar su turno, haciendo click en el botón "pasar turno", con lo que dará por abandonada su ronda, siendo entonces el turno de la computadora. Su segunda opción es realizar un cambio de fichas, que también implica un cambio de turno, pero tiene el potencial de dejarlo con fichas más susceptibles de armar una palabra. Para esto deberá hacer cick en el botón "camiar fichas", lo cual quitará del tablero cualquier ficha que haya estado siendo jugada en ese turno, y agregará a su vez el botón "cancelar", que permite abandonar el estado de intercambio de fichas. Para realizar el cambio, el jugador deberá seleccionar en su atril tanta fichas como desee cambiar, y luego hacer click en "cambiar fichas". De esta forma, las fichas seleccionadas serán entregadas a la bolsa, y el jugador recibirá otras tantas como entregó, seleccionadas al azar. Cabe destacar que un jugador tiene disponibles sólo cinco cambios de ficha gratuitos (cinco turnos, no cinco fichas), aplicandose un descuento de dos puntos por cada intercambio por encima de ese número.



Fig 3.5. Interfaz de juego mostrando el estado de cambio de fichas, en este caso el jugador ha decidido cambiar las fichas "H", "O", y "T".

posibles finales

Hay tres maneras de dar por terminada una partida. La primera, la más natural, por haberse acabado el tiempo o las fichas disponibles en la bolsa. La segunda implica la decisión directa del jugador de terminar la partida, haciendo click en el botón "terminar". En cualquiera de estos dos casos, se realizará una contabilización de los puntos, indicando el ganador.

Independientemente del resultado, a su vez, si el jugador hizo suficientes puntos como para entrar en el top 10 de puntajes en su dificultad, también se mostrará un mensaje en este sentido, y se procederá a mostrar la ventana del ranking.

El tercer final posible, es en realidad un final de carácter temporal, y consiste en pausar la partida para retomarla en otro momento. Para hacer esto debe hacerse click en el botón "posponer" de la ventana de juego, a partir de lo cual surgirá una advertencia y, si se confirma, un mensaje de éxito indicando que la partida fue guardada sin problemas.

Para continuar con una partida guardada de esta forma, debe seleccionarse la opción "continuar partida" desde el menú principal (opción que como se mencionó anteriormente sólo estará ahabilitada en caso de haber efectivamente una partida para continuar). Luego de esto se abrirá una ventana de juego con el mismo estado en el que estaba éste al momento de posponerse, pero con el cronómetro detenido y los botones inhabilitados. Para comenzar sólo es necesario hacer click en el botón homónimo.

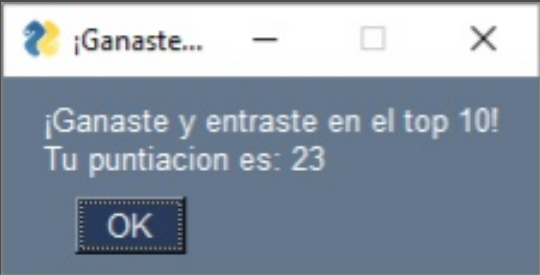


Fig 4. Mostrando el mensaje de fin de una partida en el que el jugador ganó y accedió al ranking de los 10 mejores de su dificultad.



Fig 4.5. El estado de la interfaz de una partida pospuesta luego de seleccionar "continuar partida".