

Trabajo Práctico en Etapas – Parte 2

Juego Pasapalabra

Objetivo

Adaptar y extender la funcionalidad del Juego logrado en la Parte 1, a los requerimientos que se detallan en cada una de las siguientes etapas.

Etapa 6 – Refactorización del Código de la Parte 1

En esta etapa deberán mejorar el código logrado en la Parte 1, introduciendo las mejoras y modificaciones indicadas por el Ayudante supervisor como devolución de lo realizado en la parte 1, y/o aquellas que ustedes consideren que quedaron pendientes.

Tengan en cuenta en la refactorización, entre otras cosas:

- Mejorar la estructuración del código y la división funcional realizada.
- Optimizar las soluciones en las que se recorran estructuras
- Revisar el PEP de la cátedra y verificar que se respeten las reglas allí indicadas.

Deberán versionar el anterior y el nuevo código refactorizado.

A continuación, podrán avanzar con las siguientes etapas, extendiendo la funcionalidad sobre el código refactorizado.

Etapa 7 – Registración de los Jugadores con Interfaz Gráfica

En esta etapa daremos la posibilidad que jueguen hasta 4 jugadores, unos contra otros.

De ahora en más, para que un jugador pueda participar de la jugada, debe estar registrado. Para ello, la aplicación deberá manejar un archivo usuarios.csv, que contenga dos campos, el nombre de usuario o jugador, y su clave.

Para lograr esto, se debe desarrollar una interfaz gráfica de ingreso de jugadores, que solicite el ingreso del nombre y la respectiva clave de cada uno de los jugadores, y que además, tenga un botón para registrarse, para cuando el jugador no se encuentre registrado.

Para que el jugador se registre se debe diseñar una nueva ventana gráfica, que permita:

1. El ingreso de un nombre de usuario, el que se debe validar que no exista, y que tenga, como mínimo, un largo de 4 caracteres y un máximo de 20, y estar formado sólo por letras, números y el guión medio.
2. Una clave, que deben solicitar que se ingrese dos veces, y compararla asegurando que los dos ingresos sean iguales.
La clave debe:

- Tener una longitud entre 6 y 12 caracteres.
- Estar formada sólo por caracteres alfanuméricos, a excepción de letras acentuadas; y por los caracteres “#” “!”
- Debe contener al menos una letra mayúscula, una letra minúscula, un número, y alguno de los siguientes caracteres: “#” “!”

Podrá considerar que el ingreso de nombres finaliza, cuando se alcanza el máximo de jugadores o cuando se presiona el botón iniciar partida.

La interfase debe tener una imagen identificatoria del juego y también del grupo.

Asignación de Turnos

Ingresados los nombres de los participantes, se deberán ordenar al azar para asignarle el turno de juego a cada uno de ellos; y se les debe informar antes del comienzo de la partida.

Etapas 8 - Archivos de Texto

Introduciremos un cambio a nuestra forma de obtener las palabras con sus definiciones.

Ahora que sabemos procesar archivos de texto, en lugar de utilizar la función **obtener_lista_definiciones()**, que nos devolvía la lista de posibles palabras a adivinar; deberemos escribir las funciones que consideremos necesarias, para procesar 2 archivos de texto, a partir de los cuales generaremos un diccionario de palabras con sus definiciones. Los archivos se llaman “palabras.txt” y “definiciones.txt”, cada palabra y su definición se relacionan según la misma línea en donde se encuentran en el archivo.

Los archivos serán provistos por la cátedra, los encontrarán en el campus, en la solapa donde se encuentra el enunciado de este TP.

Deberán tener en cuenta que, si los textos son muy grandes, será complicado leer todo el texto con una única instrucción de lectura; por ello la lectura debe ser línea por línea. A medida que lean las líneas y como parte del procesamiento, deberán extraer de lo leído, las palabras y las definiciones; considerando como palabra sólo aquellas que estén formadas por caracteres alfabéticos. A medida que se obtienen las palabras y su respectiva definición, se debe generar un diccionario con las palabras secretas

candidatas, de la longitud que se haya indicado; ya que en esta oportunidad la cantidad de letras de la palabra secreta, dependerá de lo indicado en el archivo de configuración que deberán procesar (ver Etapa 10).

Los archivos sólo pueden ser leídos una vez.

El acceso a cualquier estructura auxiliar que utilices, deberá ser eficiente y óptimo.

Como salida de la lectura de los archivos, deberán generar también el archivo diccionario.csv, que contendrá, todas las palabras secretas obtenidas, de la longitud indicada, ordenadas alfabéticamente; y en cada línea, debe estar la palabra y su definición con el siguiente formato.

Por ejemplo:

casa, 'Construcción cubierta destinada a ser habitada'
tanque, 'Vehículo automóvil de guerra fuertemente blindado y armado con cañones y ametralladoras, que se desplaza mediante dos cintas articuladas por todo tipo de terrenos'

Etapa 9 - Dinámica del Juego

Comienza el primer jugador según su turno. A partir de aquí la dinámica será, si acierta la palabra, suma los puntos correspondientes, y continúa con la siguiente palabra del roscó; así hasta que comete un desacierto, en ese caso se le restan los puntos correspondientes, y el turno de la siguiente palabra pasa al siguiente jugador.

En cada turno, se debe mostrar el nombre de quien está participando, así como la lista de participantes según su orden de turno.

Ejemplo de turno:

```
[A][C][D][G][I][L][M][P][S][V]
[1][2][2][2][3][ ][ ][ ][ ][ ]
[e][a][a][e][a][ ][ ][ ][ ][ ]

Jugadores:
1. Pedro - Aciertos: 0 - Errores: 1
2. Juan - Aciertos: 2 - Errores: 1
3. Maria - Aciertos: 1 - Errores: 0

Turno Jugador 3 Maria - letra L - Palabra de 8 letras
Definición: 1. f. Máquina para lijar
Ingrese palabra:
```

Una vez completado el rosco, termina la partida y se mostrarán los puntajes parciales de los jugadores, así como los puntajes totales y el resumen de la partida. También se preguntará si se desea comenzar una partida nueva en el caso que no se haya alcanzado el máximo de partidas jugadas.

Resultado de partida:

```
Turno letra A - Jugador 1 Pedro - Palabra de 7 letras - arrenal - error - Palabra Correcta: arsenal
Turno letra C - Jugador 2 Juan - Palabra de 6 letras - cabeza - acierto
Turno letra D - Jugador 2 Juan - Palabra de 5 letras - dedal - acierto
Turno letra G - Jugador 2 Juan - Palabra de 6 letras - ganada - error - Palabra Correcta: ganado
Turno letra I - Jugador 3 Maria - Palabra de 7 letras - iceberg - acierto
...
Turno letra V - Jugador 2 Juan - Palabra de 5 letras - .....
```

Puntaje de la partida:

```
1. Pedro - 20 puntos
2. Juan - 45 puntos
3. Maria - 35 puntos
```

Puntaje parcial:

```
1. Pedro - 60 puntos
2. Juan - 105 puntos
3. Maria - 95 puntos
```

Una vez finalizado el juego, se mostrarán los resultados totales finales.

Reporte final del juego:

```
Reporte Final:
Partidas jugadas: 5
```

```
Puntaje Final:
1. Pedro - 150 puntos
2. Juan - 250 puntos
3. Maria - 260 puntos
```

Etapa 10 - Archivo de Configuración

Deberán agregar un archivo de configuración, que se llamará `configuracion.csv`, del cual leerán los valores iniciales, que deben tomar para parametrizar el juego.

Dicho archivo debe contener los siguientes elementos:

`LONGITUD_PALABRA_MINIMA,4`

`CANTIDAD_LETRAS_ROSCO,10`

`MAXIMO_PARTIDAS,5`

`PUNTAJE_ACIERTO,10`

`PUNTAJE_DESACIERTO,3`

Si por algún motivo, uno o más valores no pueden ser recuperados del archivo de configuración, deben ser establecidos con valores por defecto. Una vez establecidos los valores, al inicio del juego, mostrar por pantalla, el valor efectivamente asignado a cada elemento y si fue dado por omisión o por configuración.

A partir de ahora, estos serán los valores que rijan todas las partidas de nuestro juego. Podrán ser modificados antes de iniciar un nuevo juego de cero, y se deberá ver reflejado dicho cambio en el juego, por ejemplo, que las palabras secretas, en lugar de ser de longitud mínima 4, sean de longitud mínima 6.

Condiciones de Entrega

Las siguientes condiciones deben ser respetadas en su totalidad para que la entrega sea considerada válida:

1. Cada función que forma parte del código debe tener debajo de su firma, una descripción corta de cuál es su objetivo y quien es el autor o responsable de dicha función.
2. Se deben incluir pruebas unitarias en **al menos 10 funciones**.
3. El código correspondiente a la Parte 2, debe ser subido al campus. El nombre a dar al archivo será `TP2_NombreGrupo.py`. Deberán reemplazar *NombreGrupo*, por el nombre dado a su grupo. Si la entrega está compuesta por más de un archivo `.py`, generar un `.zip` con todos los archivos `.py`, y nombrarlo de igual modo, pero con extensión `zip`.

4. Deben llevar adelante el desarrollo del código fuente mediante el uso de git mediante la plataforma de github, el mismo será revisado durante la corrección del trabajo práctico.
5. Se deberá incluir un archivo README.md en donde se incluya el nombre del grupo, sus integrantes y un link al repositorio de github (debe ser público).
6. Deberán grabar 2 videos y subirlos a un canal de Youtube, ó a Google Drive.
El primer video, **cada integrante del equipo**, deberá contar mostrando el código, qué parte estuvo bajo su responsabilidad y los puntos de solución dados, que considere más relevantes. El video total no debe superar los 10 minutos.
Comenzar cada uno de los relatos, diciendo el nombre y apellido.
Las exposiciones se deben entender y ver claramente; y deben intentar que sean homogéneas.
7. Deberán grabar un segundo video, en el que se muestre al menos una jugada completa, y que contemple distintos casos que muestran que la aplicación responde según lo esperado. Deberán ir relatando los eventos de la jugada. En este caso el video puede estar realizado por 1 único integrante y no debe superar los 10 minutos.