

Seminario de Lenguajes 2021

Opción Python

Trabajo final - Primera Parte

Este enunciado corresponde a la **primera entrega** del trabajo integrador. Si bien se describen la gran mayoría de las funcionalidades a incluir, en la [sección Entrega](#) se detalla lo pedido en esta oportunidad.

Vamos a desarrollar una aplicación para entrenamiento mental. En particular vamos a trabajar la memoria.

La aplicación a desarrollar permitirá jugar al tradicional juego de la memoria pero con algunas consideraciones tanto para la configuración previa del juego como para el posterior análisis de las jugadas realizadas.

Para este tipo de juegos se proponen varias estrategias. A continuación les presentamos la más tradicional. Si desean implementar alguna otra, deben consultar con la cátedra. En todos los casos, se trabajará con un tablero de fichas, lo que podría modificarse es la estrategia del juego.

MemPy: entrenando la memoria con Python

El juego de la memoria consiste en encontrar pares coincidentes de elementos en un tablero. Los elementos que serán parte del juego se seleccionarán en función de criterios que detallaremos más adelante. Podrán ser textos o imágenes.

El tablero contará con una cierta cantidad de casillas (configurable desde una sección de MemPy), las cuales contendrán los elementos a encontrar en forma “oculta”. En cada turno el jugador hace clic sobre una casilla, ésta “se da vuelta” mostrando el elemento correspondiente, y luego puede hacer clic en otra para encontrar su par. Si es correcta, quedarán ambas dadas vueltas y generará los puntos correspondientes¹. En cambio si no es la correcta, se deberán dar vuelta automáticamente ambas casillas ocultando ambos elementos.

El juego contará con un tiempo máximo de juego de cierta cantidad de minutos (también configurable desde MemPy).

El programa deberá contar con las siguientes pantallas:

- (1) Menú de inicio de la aplicación.
- (2) Pantalla de configuración del juego.

¹ El sistema de puntajes queda a criterio de cada grupo y deberá ser especificado claramente en las ayudas de la aplicación.

- (3) Pantalla con la sala del juego.
- (4) Pantalla de puntajes.
- (5) Pantalla de estadísticas.

(1) Menú de inicio de la aplicación

En esta pantalla se presenta el menú principal de MemPy que nos lleve a cada una de las secciones de la aplicación y se incluirá también una opción que permita ingresar los datos del usuario. Se solicitará el **nick** (nombre) de cada usuario (que debe ser único) y, en caso de ser la primera vez que use la aplicación, se le solicitará, además los siguientes datos: **edad** y **género autopercebido**. Cada usuario además, tendrá asociada su **configuración de juego**². En esta sección también se incluirá una ayuda que mostrará las funcionalidades de MemPy.

(2) Opciones de configuración

En esta sección se permitirá configurar algunos parámetros de cada nivel con los que se trabajará. Al menos deben contemplar los siguientes:

1. Los textos de mensajes que se muestran al usuario, como por ejemplo: “¡Ganaste!”, “¡Perdiste!”, “Quedan x segundos...”, etc.
2. La cantidad de casillas del tablero para cada nivel del juego. Por ejemplo, si se proponen 3 niveles, podría ser: nivel 1: 8x8, nivel 2: 16x16, nivel 3, 20x20.
3. La cantidad de coincidencias por elemento. Por defecto serían dos.
4. El tipo de elementos de las casillas: esto es solo palabras o imágenes y palabras. A modo de ejemplo, la palabra casa podría representarse de estas dos formas:



5. El tipo de ayuda a incluir durante la jugada. Esto podría ser simplemente: “SIN ayudas”, o “CON ayudas”. O indicar el tipo de las mismas. Por ejemplo, se podría mostrar en el tablero la fila o columna donde se encuentra el par del elemento seleccionado o, en algún momento, mostrar durante un tiempo breve tres o cuatro (o más de acuerdo a la configuración) de pares de elementos existentes y luego volver a ocultarlos.
6. El tiempo total de la partida.
7. La paleta de colores a utilizar. Se podrá elegir entre varios estilos³.

Como mínimo deberán contemplar tres niveles de juego con los parámetros mencionados anteriormente.

² La primera vez, se asignará la configuración por defecto.

³ - <https://colorbrewer2.org/#type=sequential&scheme=YlGn&n=3>
 - <https://asada.website/cvsimulator/e/>

(3) Sala del juego

En esta sección se mostrará el tablero con la información correspondiente a la jugada. Es decir:

- las casillas con los elementos ocultos o visibles de acuerdo al estado de la partida;
- el nick (nombre) del jugador;
- el tiempo transcurrido de cada paso y el total de la partida.
- la cantidad de elementos encontrados en relación con el total.

En cada jugada se debe registrar cierta información que luego será utilizada para calcular estadísticas del juego. Al menos se deberán registrar:

Datos de la jugada:

- nick del usuario;
- día y hora de inicio de la partida;
- tiempo total de la partida;
- tiempo de cada paso;
- cantidad de intentos correctos y equivocados;
- nivel del juego;
- el criterio aplicado para la selección de los elementos (esto se explicará en la sección “Selección de elementos”).

Datos generales del juego:

- Se guardará, por cada elemento (palabra o imagen) que aparece en el tablero el tiempo transcurrido desde que fue clicado por primera vez hasta que se encontró su par (si es que se encontró), si fue emparejado o no.
- Para fines estadísticos, se registrará cuántas veces se eligió el elemento (independientemente si es en su forma de texto o imagen) para que aparezca en el juego.

(4) Puntajes

En esta sección se mostrará una tabla con los puntajes obtenidos por los usuarios en cada nivel ordenada por algún criterio.

(5) Estadística

En esta sección se mostrará distintos gráficos y resúmenes que se obtendrán de los datos guardados durante las jugadas. Esto se explicará en detalle en la segunda entrega.

Selección de elementos a incluir

Como se mencionó anteriormente los elementos que se utilizarán en cada jugada se obtendrán de acuerdo al análisis de un conjunto de datos específicos. En esta primera entrega, se

trabjará con dos tipos archivos en formato cvs y json que contienen información sobre diferentes tipos de datos

En estos casos se analizará estos datos y de acuerdo al día y hora del ingreso a la aplicación (login) se podrá elegir (a modo de ejemplo):

- Las 10 provincias según diferentes criterios.
- Los 10 memes más utilizados/rankeados.
- Los animales agrupados por un criterio específico.

Esta sección será explicada en detalle en las prácticas.

Consideraciones sobre la implementación

El programa principal se debe llamar **mem_py.py**.

El software debe correr correctamente tanto en Windows como en Linux.

Se deberá armar una estructura de directorios para organizar los archivos en carpetas y subcarpetas de manera tal que sea clara su corrección. Las mismas se subirán a un repositorio de GitLab en donde se deberá incluir además un archivo denominado README.md, que incluya el nombre de los integrantes, la forma de ejecutar cada aplicación y cualquier consideración especial para su ejecución, por ejemplo, requerimientos.

En todos los casos se deberán utilizar imágenes o sonidos, libres o propios, que puedan ser publicados con licencias libres.

Para el desarrollo de la aplicación se deberá:

- utilizar **PySimpleGUI** para las interfaces gráficas.
- definir los archivos que se requieran generar con cualquiera de los formatos vistos en la cursada;
- documentar el código usando docstrings en las funciones y clases;
- incluir en cada archivo fuente los nombres de los autores y la licencia;

Entrega

Este enunciado corresponde a la **primera entrega** del trabajo integrador.

Si bien el trabajo es grupal, la defensa es INDIVIDUAL. **Con lo cual sólo obtendrán los puntos de cada entrega aquellos que realicen satisfactoriamente la defensa de cada entrega.** Es importante también tener participación activa en las consultas prácticas para que el ayudante tenga un panorama conceptual del conocimiento de los participantes del grupo.

Funcionalidad incluida en esta primera entrega

La primera versión de MemPy a entregar en esta primera parte incluye un primer prototipo de todas las pantallas de la aplicación sin su funcionalidad salvo las mencionadas a continuación:

- todas las opciones de la pantalla de configuración guardadas en un archivo json;

- el registro y login del usuario.
- una primera selección de elementos de acuerdo al procesamiento de algunos de los con archivos csv y json mencionados en la sección “Selección de elementos a incluir”.

Fechas de entrega

Esta entrega corresponde a la actividad 2 de práctica, cuya fecha de entrega es la semana del **17 de Mayo y la defensa individual es 19 y 20 de Mayo.**

Criterios de evaluación

- Funcionalidad implementada de acuerdo al enunciado.
- Cumplimiento de las consideraciones planteadas.
- Código subido a un repositorio de GitLab.
- Se asignan ayudantes a cada grupo para tener un seguimiento del trabajo.
- En la defensa deberán demostrar los conocimientos utilizados para la realización del trabajo.

Puntos de la entrega (en la defensa individual)

- **Puntos de cursada:** 60.
- **Python plus:** 50