

## Práctico Unidad 4 – Aplicaciones de Escritorio

Python – Tkinter

Juego de memoria Py-SimonGame

### a) Primera iteración



La empresa de juegos educativos “JugateYa” lo contrata a usted para desarrollar un juego que incentive el uso de la memoria. El juego está basado en el dispositivo “Simon”, que muestra un panel con 4 botones de 4 colores distintos. Cuando se inicia el juego, se prende o cambia de color, por unos segundos, uno de los botones, el botón que se prende es seleccionado en forma aleatoria. El jugador debe “apretar” el botón que cambió de color, si es el botón correcto, se cuenta un punto por el acierto, caso contrario, se muestra el mensaje “GAME OVER”. El juego continúa, generando una secuencia aleatoria de botones que se prenden, siempre agregando un botón la secuencia anterior. El jugador debe “apretar” la secuencia de botones que cambiaron de color. El objetivo del juego es generar la secuencia más larga posible. Cuando se equivoque, muestra el mensaje “GAME OVER”, y un mensaje en una ventana que diga el puntaje obtenido.

### Ejemplo de secuencia

#### Inicia el juego

El sistema prende botón verde (cambia de color el botón verde, por unos segundos).

Jugador aprieta botón verde, cambia de color el por unos segundos el botón presionado.

El sistema cuenta como un acierto y lo muestra.

El sistema genera aleatoriamente un nuevo botón a encender, por ejemplo el botón rojo, enciende botón verde, botón rojo.

Jugador aprieta, botón verde, luego botón rojo.

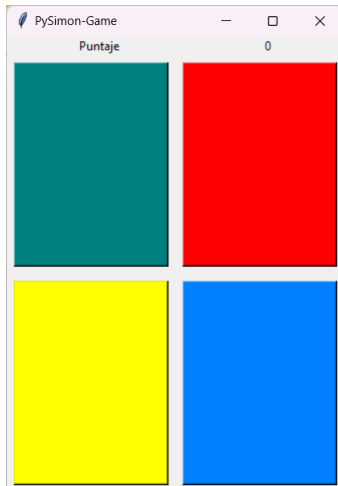
El sistema cuenta uno por cada acierto, y lo va mostrando a medida que acierta.

El sistema, genera aleatoriamente un nuevo botón a encender, por ejemplo el botón rojo, enciende botón verde, botón rojo, botón rojo.

Jugador aprieta, botón verde, botón rojo y finalmente botón rojo.

El sistema cuenta uno por cada acierto, y lo va mostrando a medida que acierta.

**La interface del juego es la siguiente:**



### Regla de negocio

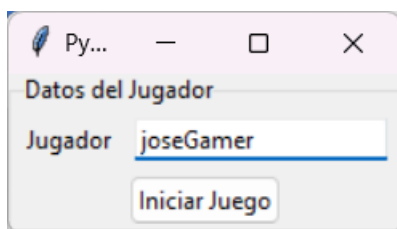
Para cada botón se deben utilizar objetos de la clase `tk.Canvas`, que admite el uso de colores en formato RGB (Red, Green, Blue), en la propiedad `background` del `Canvas`, el color rojo se especifica de la siguiente manera: `#ff0000`.

El cambio de color del botón que se activa, debe ser percibido por el ojo humano, para lo que deberá probar con distintos valores en los temporizadores. El cambio del botón que presiona el jugador, también debe ser percibido por ojo humano, se puede utilizar el cambio en el relieve del botón seleccionado (propiedad **relief**, valores posibles: `flat`, `raised`, `sunken`, `groove`, `ridge`).

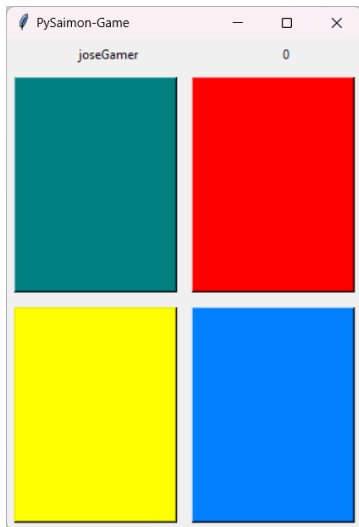
### b) Iteración 2

El programa solicita nombre del jugador antes de comenzar el juego, luego en la interfaz, muestra el nombre del jugador con el puntaje obtenido.

Interfaz que solicita datos del jugador



## Nueva interfaz



En esta iteración, el jefe de sistemas le pide que genere un archivo .json, con los datos del jugador, fecha, hora, y puntaje, el archivo se denomina “**pysimonpuntajes.json**”.

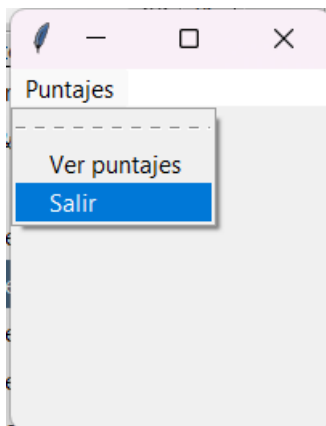
Al salir del juego, debe almacenarse el puntaje en dicho archivo.

### c) Iteración 3

Cree la clase Jugador, con el operador `__gt__` sobrecargado, para ordenar los jugadores por puntaje.

Cree la clase Gestor Jugadores, que recupere y almacene los resultados de los jugadores que hagan uso del juego. Implemente dicho gestor con una lista Python.

Agregue a la interfaz un menú de opciones, que tenga la siguiente interfaz:



Al seleccionar **Salir**, se sale del programa.

Al seleccionar **Ver puntajes**, debe abrir el archivo .json, denominado “**pysimonpuntajes.json**”, donde se almacenan los puntajes obtenidos por los jugadores que jugaron el juego, cargarlos en el Gestor de Jugadores, con el estado de todas las instancias de la clase Jugador almacenadas en el archivo, ordenarlo de mayor a menor por puntaje de los jugadores, abrir un cuadro de diálogo que muestre el nombre del jugador, la fecha, la hora y el puntaje obtenido.

### Diseño de la interfaz cuadro de diálogo de puntajes

Galería de Puntajes			
Jugador	Fecha	Hora	Puntaje
joseGamer	29/05/2024	13:59	7
lucho77	10/05/2024	11:04	5

Cerrar

#### d) Iteración 4

Agregar una selección de niveles:

**Principiante**, el juego se juega tal como se ha planteado hasta ahora.

**Experto**, se reduce el tiempo entre el que se expone el cambio de color de los botones, además se agrega un timer general, cada vez que el jugador selecciona el botón que corresponda, se reinicia, cuando se acaba el tiempo, sin que el jugador presione el siguiente botón, el sistema muestra el mensaje GAME OVER. Cada vez que el sistema agrega un nuevo botón, se reinicia el timer general.

**Super Experto**, además de lo solicitado para el nivel Experto, en lugar de cambiar de color de a un botón, debe cambiar de color de a dos botones.

## Regla de negocio

El juego comienza en nivel **Principiante**.

Agregar en el archivo .json y en la visualización de los puntajes de los jugadores, el nivel en el que obtuvo el puntaje.

## Interfaz final del juego

