

| | | | |
|---------------------------|------------------------|----------------------------------|-------------------------|
| Materia: | Programación I | | |
| Nivel: | 1º Cuatrimestre | | |
| Tipo de Examen: | Segundo Parcial | | |
| Apellido ⁽¹⁾ : | | Fecha: | 29 nov 2024 |
| Nombre/s ⁽¹⁾ : | | Docente a cargo ⁽²⁾ : | Cardozo / Pavlov |
| División ⁽²⁾ : | 311 | Nota ⁽²⁾ : | |
| DNI ⁽¹⁾ : | | Firma ⁽²⁾ : | |

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.

(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Desarrollar en Python utilizando el módulo PyGame: Videojuego Sudoku Clásico.

Reglas: cada uno de los números desde 1 hasta el 9 inclusive debe aparecer una vez por cada columna, una vez por cada fila y una vez por cada región o submatriz de 3x3.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 6 | 5 | 1 | | | | 8 |
| 7 | | 3 | 8 | | | 6 | 9 | 1 |
| 2 | | | | 3 | | 5 | 4 | 7 |
| | | 1 | 7 | | | | 8 | |
| 6 | | | 3 | | 1 | 7 | 5 | |
| 8 | | 7 | 4 | 5 | | 3 | 1 | |
| | 8 | 5 | 6 | | | 9 | | |
| 9 | | | | | 5 | | 7 | |
| | 7 | 4 | | | | | | |

- Desarrollar una función que genere una matriz de 9 filas por 9 columnas con números enteros aleatorios entre 1 y 9 (inclusive), que respete las reglas del sudoku clásico.
- Desarrollar una función que oculte aleatoriamente algunas de las celdas del tablero en función del nivel de dificultad seleccionado, utilizando un porcentaje específico pasado por parámetro.

1) El juego debe incluir al menos 3 pantallas principales:

- Pantalla de inicio o menú principal: debe contar con los botones “Jugar”, “Puntajes” y “Salir”. En esta pantalla se debe seleccionar la dificultad del juego.
- Pantalla principal del juego: debe mostrar el tablero de sudoku, un temporizador (en minutos y segundos), un contador de errores y un botón de reinicio que genere un nuevo tablero y reinicie el temporizador y el contador de errores. Además debe contener un botón “Volver” para volver al menú principal (queda a criterio del desarrollador si al volver la partida se pausa o finaliza).
- Pantalla de puntajes: debe contener un top 5 de los mejores puntajes con sus respectivos nombres (usernames o nicknames), ordenados de mayor a menor. Debe contar con un botón que permita volver al menú principal.

2) El juego debe contar con 3 niveles de dificultad (queda a elección del desarrollador la forma de selección):

- Fácil: se oculta aproximadamente el 20% del total de números del tablero.
- Intermedio: se oculta aproximadamente el 40% del total de números del tablero.
- Difícil: se oculta aproximadamente el 60% del total de números del tablero.

3) Las pantallas de inicio y de puntajes deben tener una alguna imagen de fondo relacionada al juego o abstracta que cubra la totalidad de la pantalla, así como música de fondo.

4) En el menú principal o al finalizar la partida, el jugador debe poder ingresar su nombre a través de una caja de texto en la pantalla del juego (NO mediante la terminal).

5) Durante la partida, al hacer clic en una celda del tablero, esta debe cambiar de color para indicar ha sido seleccionada; luego, el usuario puede ingresar un número entre 1 y 9 (inclusive) o borrar el número con la tecla “Supr”. Al hacer clic fuera del tablero se desmarca la celda seleccionada.

- Los números predeterminados del tablero deben tener un color distinto al de los números ingresados por el usuario y no deben poder modificarse.
- Si el jugador ingresa un número incorrecto, este debe marcarse en rojo oscuro y contar un (1) error.

6) Los puntajes se calculan en función de la dificultad seleccionada, la cantidad de errores y el tiempo que el jugador tarda en terminar la partida. Estos puntajes deben guardarse en un archivo junto con los nombres de los jugadores. Si el nombre del jugador ya existe, se debe conservar el puntaje más alto. Al ganar la partida debe mostrarse un aviso acorde (puede incluir el puntaje logrado y el nombre del jugador).

7) Al presionar el botón “Salir”, se cierra el juego.

OPCIONALES:

- Agregar Imágenes, sonidos, animaciones, movimientos en cualquiera o todas las pantallas del juego.
- Implementar transiciones (entre las pantallas del juego).
- Hacer que los botones cambien ligeramente de color (al apretarse y/o pasar el puntero por encima).
- Añadir sombras simuladas a los botones, tablero y/o cualquier otro lugar donde considere necesarias.
- Incluir un teclado virtual al lado del tablero en la pantalla del juego (con los números del 1 al 9 y botón “borrar”) que debe permanecer visible durante toda la partida.
- Validar la longitud (entre 3 y 15) y los caracteres (alfanuméricos) del nombre ingresado.
- Calcular y mostrar el puntaje en tiempo real durante la partida.
- Permitir al jugador agregar “notas” (uno o más números posibles en tamaño más pequeño sin indicar cuál de ellos es el correcto, y que se borren automáticamente al ingresar el número correcto en la misma fila, columna o submatriz).
- Incluir un botón y pantalla de “configuraciones” en el menú principal que permita cambiar la resolución del juego, activar el modo oscuro, modificar el volumen de los sonidos, activar o desactivar efectos de sonido, etc.
- Sombra o color tenue que marque la fila, columna y submatriz al seleccionar una celda.
- Cualquier otro extra a elección del desarrollador.

Fórmula de cálculo de puntaje: (Ejemplo)

$$\text{Puntaje} = (\text{Puntos Base} - (\text{Errores} \times \text{Penalización por Error}) - (\text{Tiempo Transcurrido en Minutos} \times \text{Penalización por Tiempo})) \times \text{Dificultad}$$

Donde:

Puntos Base: Es el puntaje inicial que se le da al jugador al comenzar la partida. Por ej 1000 puntos.

Errores: Es la cantidad de errores cometidos por el jugador durante la partida.

Penalización por error: Un valor que se resta por cada error. Por ejemplo, 50 puntos por error.

Tiempo Transcurrido en Minutos: Tiempo transcurrido entre el inicio y la finalización de la partida.

Penalización por Tiempo: Un valor que se resta por cada minuto. Por ejemplo, 10 puntos por minuto.

Dificultad: Un multiplicador en función de la dificultad seleccionada.

Ejemplo con números reales:

Puntos Base= 1000

Errores= 3

Penalización por Error = 50

Tiempo Transcurrido= 8 minutos

Penalización por Tiempo = 10

Dificultad (intermedia) = 1.5

Aplicando la fórmula

$\text{Puntaje} = (1000 - (3 \times 50) - (8 \times 10)) \times 1.5$

$\text{Puntaje} = (1000 - 150 - 80) \times 1.5$

$\text{Puntaje} = 770 \times 1.5$

Puntaje = 1155

Condiciones de aprobación:

- **Aprobación NO Directa:**
 - o Deberán estar realizados en su totalidad los ítems 1, 2, 3, 4, 5 y 7.
- **Aprobación Directa:**
 - o Deberán estar realizados en su totalidad los ítems 1 al 7. (**OPCIONALES:** solamente sumará puntos si están en condiciones de Aprobación Directa).

NOTAS:

Nota 1: Se deberá desarrollar biblioteca y funciones propias, las mismas deberán estar correctamente documentadas.

Nota 2: El set de datos debe generarse dinámica y aleatoriamente (la distribución del tablero deberá ser diferente en cada partida).

Nota 3: El formato del archivo que se debe crear para guardar los puntajes debe ser TXT, CSV o JSON.

Nota 4: Puede utilizar cualquier colección, tipo de datos, funciones propias y métodos vistos en clase donde considere necesario.

Nota 5: Los colores a utilizar (excepto indicados explícitamente) quedan a criterio del desarrollador.

Nota 6: La fórmula utilizada para calcular el puntaje es una simple sugerencia, puede utilizar su propia fórmula.