

Nombre del proyecto: Tetris

Profesor: Cristian Campos Agüero

Semestre: S1 del 2025

Curso: Taller de programación

Grupo: 61

Estudiante: Daniel Mendoza Morales

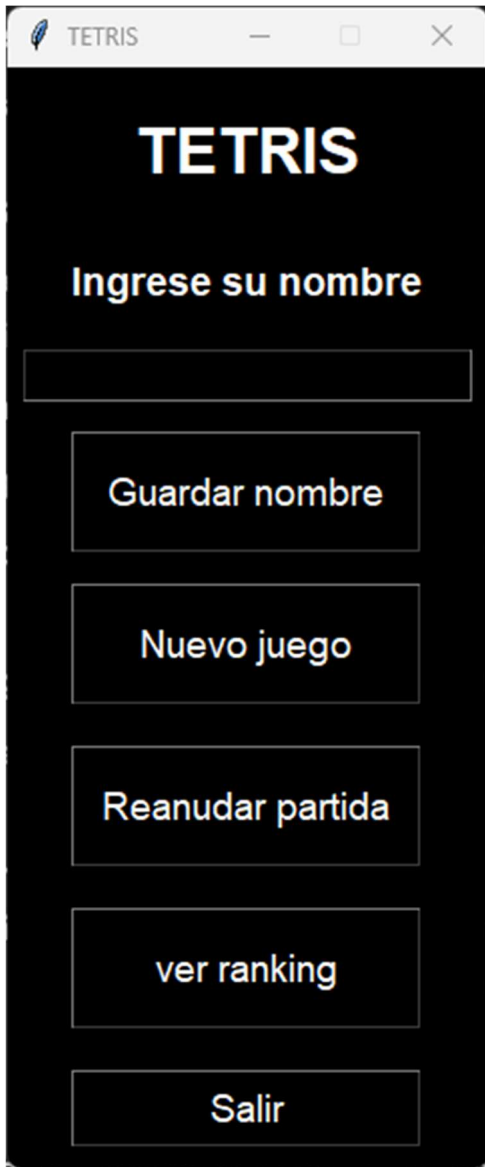
Carné: 2025110674

AÑO: 2025

2. Manual de usuario:

Este manual está diseñado para guiar a los usuarios sobre cómo usar el programa

Al iniciar el programa se le mostrara el siguiente menú



Con las siguientes opciones:

1. **Ingresar su nombre:** Antes de comenzar un nuevo juego, debes ingresar tu nombre de jugador.

2. **Nuevo juego:** Inicia una nueva partida de Tetris con el tablero limpio y el puntaje reiniciado.
3. **Reanudar partida:** Carga una partida guardada previamente, permitiéndote continuar desde donde la dejaste.
4. **Ver ranking:** Muestra los 10 mejores puntajes de los jugadores.
5. **Salir:** Cierra la aplicación del juego.

2.2. Modo de Juego

Una vez que se inicia una nueva partida o se reanuda una guardada, se mostrará el tablero de Tetris. El objetivo es organizar piezas que caen, llamadas tetrominós, para completar líneas horizontales sin espacios.

- **Tablero de Juego:** El tablero se representa como una matriz de 22 filas por 12 columnas. Las celdas de los bordes están marcadas con "+", las celdas vacías con "0" y los bloques de los tetrominós con "1".
- **Tetrominós:** Las piezas están compuestas por varios bloques, existen 8 formas en total que aparecen en el juego.
- **Completar Líneas:** Cuando una fila se llena con bloques (representados por "1"s), esta desaparece, y las filas superiores se desplazan hacia abajo. Cada línea completada otorga 100 puntos.

- **Fin del Juego:** El juego termina cuando las nuevas piezas no pueden entrar al tablero o el usuario decide salir.

2.3. Movimientos Permitidos

Los jugadores controlan los tetrominós que caen usando las siguientes teclas del teclado:

- **Flecha Izquierda:** Mueve la pieza actual una columna a la izquierda, si no hay colisión con otras piezas o el borde del tablero.
- **Flecha Derecha:** Mueve la pieza actual una columna a la derecha, si no hay colisión con otras piezas o el borde del tablero.
- **Flecha Abajo:** Mueve la pieza actual una fila hacia abajo.
- **Flecha Arriba:** Rota la pieza actual 90 grados en el sentido de las agujas del reloj. La rotación está restringida si causa una colisión o mueve la pieza fuera del tablero.

2.4. Funcionalidades Adicionales

- **Caída Automática:** Los tetrominós tienen un movimiento automático hacia abajo, cuya velocidad aumenta a medida que el jugador acumula puntos.
- **Guardado de Partida:** Los jugadores pueden guardar el estado actual del juego (incluyendo la matriz del tablero, el puntaje y la posición de la pieza) en un

archivo de texto. Esto permite a los jugadores continuar su partida más tarde.

- **Estadísticas del Juego:** Al finalizar cada partida se guarda el nombre del jugador y la puntuación

3. Descripción del Problema

El problema consistió en desarrollar un juego de Tetris funcional utilizando Python, incorporando las mecánicas clave del juego, elementos de interfaz gráfica y características como guardar/cargar partidas y mostrar puntajes altos. Este proyecto tuvo como objetivo desglosar las reglas y componentes del juego.

4. Diseño del Programa: Decisiones de Desarrollo y Algoritmos Usados

El juego fue desarrollado utilizando Python y la librería Tkinter para la interfaz gráfica de usuario.

- **Representación del Tablero:** El tablero de juego se representa como una lista (matriz) 2D donde diferentes caracteres representan bordes, espacios vacíos y bloques activos/asentados.
- **Movimiento y Rotación de Piezas:**
 - El movimiento (izquierda, derecha, abajo) implica verificar colisiones con bloques existentes o límites del tablero antes de actualizar la posición de la pieza.

- La rotación se implementa transformando la representación matricial de la pieza y validando la nueva posición para colisiones. La pieza cuadrada está excluida de la rotación.
- **Detección de Colisiones:** Se han implementado funciones para verificar colisiones en todas las direcciones (hacia abajo y laterales) para asegurar movimientos válidos y una correcta colocación de la pieza.
- **Eliminación de Líneas:** Cuando una fila está llena de bloques, se elimina y las filas superiores se desplazan hacia abajo para llenar el espacio vacío. Esto implica iterar a través del tablero, identificar las filas completas y reconstruir el tablero.
- **Puntaje y Niveles:** Se otorgan puntos por las líneas eliminadas, y la velocidad del juego (tasa de caída automática) aumenta en umbrales de puntaje predefinidos, implementando un sistema de niveles.

5. Librerías Usadas

Las librerías principales utilizadas para el desarrollo de la interfaz gráfica son:

- **tkinter:** La librería estándar de Python para la interfaz gráfica. Se utiliza para crear la ventana del juego, el canvas para dibujar el tablero y las piezas, botones, etiquetas y campos de entrada para la interacción del usuario.

- **random:** Utilizada para seleccionar aleatoriamente las formas y colores de los tetrominós para las nuevas piezas.
- **tkinter.messagebox:** Utilizada para mostrar mensajes informativos, errores y alertas de fin de juego al usuario

6. Análisis de Resultados

6.1. Objetivos Alcanzados

- **Diseño de Interfaz Gráfica:** Se implementó con éxito una interfaz gráfica utilizando Tkinter, permitiendo la representación visual del tablero y las piezas del juego.
- **Mecánicas Centrales del Juego:**
 - Las piezas pueden moverse a la izquierda, derecha y abajo.
 - Las piezas pueden rotarse.
 - Las líneas completadas se eliminan correctamente, y las filas superiores se desplazan hacia abajo.
 - Las validaciones de movimiento están implementadas para evitar que las piezas salgan de los límites o colisionen con otros bloques.
- **Gestión del Estado del Juego:** La capacidad de guardar y restaurar el progreso del juego utilizando archivos de texto está implementada.

- **Estadísticas del Juego:** El juego rastrea y muestra con éxito el puntaje del jugador y un ranking del Top 10.
- **Caída Automática y Niveles:** La caída automática de los tetrominós está implementada, y la velocidad de caída aumenta con el puntaje del jugador, creando niveles de dificultad.
- **Interacción del Usuario:** Se utiliza la entrada del teclado para controlar los movimientos de las piezas. Se proporcionan mensajes claros al usuario.

6.2. Objetivos No Alcanzados (y Razones)

- **Color Fijo por Tipo de Figura:** No se logró que cada tipo de figura mantuviera un color específico y único una vez que se bloqueara en el tablero. En su lugar, la pieza que cae tiene un color aleatorio, y al bloquearse, todas las piezas en el tablero se muestran en color gris.
- **Razón:** No se encontró una forma clara de implementar que cada bloque conservara su color individual una vez dibujado en el canvas de manera permanente. Se consideró la opción de que cada número en la matriz del tablero representara un color, pero parecía complejo

7. Conclusión(es)

Este proyecto de Tetris fue una buena oportunidad para aprender y profundizar en el uso de tkinter y el canvas, que al principio fue lo más complicado. Lograr implementar el tablero también fue un desafío importante.

Me siento muy satisfecho con el resultado general, considero que el juego está bastante completo en sus funciones principales. Mi único cambio que haría sería haber podido implementar colores fijos para cada tipo de figura, algo que no puede lograr.

Evalúo mi proyecto como Bueno. Sé que hay partes del código que podrían optimizarse para mejorar el rendimiento, y es algo que tendré en cuenta para futuros desarrollos. En resumen, este proyecto fue un muy interesante y bueno para adquirir conocimientos en el diseño de interfaces gráficas y la lógica de juegos.

Link del video

<https://youtu.be/NKQzdzJWWPY>

