

Centro Académico de Limón
ESCUELA DE INGENIERIA EN COMPUTACION

Taller de Programacion
Proyecto #2

Semestre 1

Profesor: Cristian Paz Campos Aguero

Elaborado por:

David Fernández Torres
Cédula:1-2003-0842
Grupo: 61

TEC, 31 de mayo de 2025

Manual de Usuario – Tetris Python

Requisitos

- Python 3.x instalado.
- Librerías: `tkinter`, `PIL` (Pillow).
- Archivos del juego en la carpeta correspondiente.

Menú Principal

- JUGAR: Inicia una nueva partida. Se te pedirá tu nombre.
- CARGAR: Muestra partidas guardadas para continuar una anterior.
- ESTADÍSTICAS: Muestra el historial de partidas y mejores puntuaciones.
- SALIR: Cierra el juego.

Controles del Juego

- Flecha Izquierda: Mover pieza a la izquierda.
- Flecha Derecha: Mover pieza a la derecha.
- Flecha Abajo: Bajar la pieza más rápido.
- Barra Espaciadora: Rotar la pieza.
- P: Pausar el juego y mostrar menú de pausa.

Durante la Partida

- El objetivo es acomodar las piezas que caen para formar líneas horizontales completas.
- Las líneas completas desaparecen y suman puntos.
- El juego termina si las piezas llegan a la parte superior del tablero.

Menú de Pausa

- CONTINUAR: Reanuda la partida.
- GUARDAR: Guarda el estado actual de la partida.
- SALIR: Sale de la partida (puedes guardar antes de salir).

Partidas Guardadas

- Puedes cargar partidas guardadas desde el menú principal.
- Puedes eliminar partidas guardadas desde la ventana de carga.

Estadísticas

- Consulta el historial de partidas, puntuaciones y el ranking de mejores jugadores.

Obstáculos

- El tablero puede tener obstáculos fijos (marcados con “+” en el archivo de matriz).

Consejos

- Rota y mueve las piezas para evitar dejar huecos.
- Aprovecha la vista previa de la siguiente pieza para planificar tus movimientos

Link Youtube

<https://youtu.be/92SymIwq-Vs>

Problema

El problema que se solucionó fue el desarrollo de un juego de Tetris con interfaz gráfica en Python, que permite a los usuarios jugar partidas, guardar y cargar su progreso, visualizar estadísticas y personalizar el tablero con obstáculos definidos en un archivo externo.

Diseño del programa

Decisiones de desarrollo:

Se utilizó tkinter para la interfaz gráfica por su facilidad de uso y compatibilidad con Python estándar.

El tablero y las piezas se gestionan como matrices (listas de listas), facilitando la manipulación y comprobación de colisiones.

Se permite la personalización del tablero mediante un archivo de texto (matriz.txt) donde se pueden definir obstáculos fijos con el carácter +.

El juego soporta guardar y cargar partidas, así como mostrar estadísticas y ranking de jugadores.

Algoritmos usados:

Colisión: Se verifica si la pieza puede moverse o rotar comprobando si alguna de sus celdas colisiona con los bordes o con piezas fijas.

Limpieza de líneas: Se detectan filas completas y se eliminan, desplazando las filas superiores hacia abajo.

Rotación: Las piezas se rotan usando diferentes matrices predefinidas para cada tipo y orientación.

Caída automática: Se usa un temporizador (after) para mover la pieza hacia abajo cada segundo.

Gestión de archivos: Se usan archivos de texto para guardar el estado del juego y las estadísticas.

Librerías usadas

tkinter: Para la interfaz gráfica y gestión de ventanas, botones, etiquetas y cuadros de texto.

PIL (Pillow): Para cargar y mostrar imágenes (logo de Tetris).

os: Para manipulación de archivos y directorios (guardar/cargar partidas).

random: Para seleccionar piezas aleatoriamente.

time: Para registrar la fecha y hora de las partidas guardadas.

messagebox: Para mostrar mensajes emergentes al usuario.

Análisis de resultados

Objetivos alcanzados:

Juego funcional de Tetris con interfaz gráfica amigable.

Personalización del tablero mediante archivo externo.

Soporte para guardar y cargar partidas.

Estadísticas y ranking de jugadores.

Obstáculos fijos en el tablero.

Controles intuitivos y menú de pausa.

Objetivos no alcanzados:

No se implementó música ni efectos de sonido.

No se incluyó un sistema avanzado de niveles o dificultad progresiva.

Razones:

Limitaciones de tiempo y enfoque en la funcionalidad principal.

Priorización de la estabilidad y facilidad de uso sobre características adicionales.

Conclusión

El programa desarrollado cumple con los objetivos principales de un Tetris clásico, añadiendo opciones de personalización y gestión de partidas. La estructura modular y el uso de archivos externos facilitan la extensión y el mantenimiento del código. El uso de tkinter permitió una interfaz clara y funcional. Considero que el proyecto es exitoso en cuanto a jugabilidad y experiencia de usuario, aunque existen oportunidades para agregar más características en el futuro, como sonidos o niveles avanzados.