

Instituto Tecnológico de Costa Rica

IC-1803 Taller de Programación

Proyecto:

Tetris con obstáculos

Estudiante:

Maykel Poveda Paniagua

Profesor:

Cristian Paz Campos Aguero

I Semestre 2025

Manual de Usuario

¿Qué es Tetris?

Este proyecto es una versión moderna del clásico juego Tetris. Fue creado en Python usando Tkinter para la interfaz gráfica. A diferencia del Tetris tradicional, esta versión te permite agregar obstáculos en el tablero, y guardar tu puntaje en un ranking. Todo está hecho con amor para ofrecer una experiencia divertida.

¿Qué necesito para jugar?

Para ejecutar este juego necesitas tener Python instalado. También necesitas instalar la librería Pillow si no la tienes:

```
pip install pillow
```

¿Qué archivos vienen con el proyecto?

- Proyecto Tetris: una carpeta que incluye:
 - fondo1.png, ra.ico: imágenes para mejorar la apariencia
 - Ranking.txt: donde se guarda tu puntaje
 - Matriz.txt: define el estado del tablero
 - Codigo.py: el archivo principal con todo el juego

¿Cómo empiezo?

1. Abrí una terminal en la carpeta donde está el juego.
2. Ejecuta el archivo
3. Ingresa tu nombre. No puede estar vacío ni tener más de 11 letras.
4. Después puedes hacer clic en el tablero para poner obstáculos.
5. Cuando estés listo, hace clic en Jugar.

Controles

- Flecha Izquierda: mover a la izquierda
- Flecha Derecha: mover a la derecha
- Flecha Arriba: rotar la pieza

- Flecha Abajo: bajar más rápido la pieza

¿Qué más tiene?

Vas a ver un panel a la derecha con tu nombre, puntos, líneas y piezas jugadas. También hay un botón de Ranking donde puedes ver tu posición entre los mejores puntajes.

¿Cuándo termina?

Cuando ya no hay espacio para una nueva pieza, el juego se termina. Te va a salir un mensaje y tu puntaje se guarda automáticamente. Podes volver a intentarlo añadiendo obstáculos cuando quieras.

Enlace del video de YouTube

#

Descripción General del Proyecto

Este proyecto implementa una versión del juego Tetris utilizando Python y la biblioteca gráfica Tkinter. El jugador puede interactuar con una interfaz visual donde puede ingresar su nombre, colocar obstáculos personalizados antes de comenzar, y jugar una partida donde se le muestran estadísticas como líneas completadas, puntaje acumulado y número de piezas usadas. Además, el juego cuenta con un sistema de ranking que registra los mejores puntajes alcanzados.

Diseño del Programa

En un principio las creaciones de piezas se iban a realizar todas por medio de bibliotecas, pero después de analizar cómo se iba hacer deduje que sería mejor hacerlo por medio de listas de listas, donde su color iba a estar en la misma posición que su figura, y así acceder a ella por medio de un índice.

El tablero en un principio se iba a realizar por medio de imágenes, pero sentí que iba a ser muy complicado después de pensar en una solución llegué a la conclusión que mejor sería trabajarlo con un canvas, luego dibujar un rectángulo dentro de él, lo cual sentí mucho más cómodo, así como también fácil la implementación del tablero.

Justificación de Uso de “global” y “min”

“global”: se usó para modificar variables definidas fuera de una función. Esto fue necesario para actualizar valores globales como la pieza activa, tablero, el puntaje o el estado del juego.

“min”: se utilizó al mostrar los mejores puntajes para limitar la cantidad máxima a 10, aún si hay menos jugadores registrados. Esto asegura que la tabla sea compacta y clara para el usuario.

Análisis de Resultados

Objetivos Logrados

- Se logra una versión funcional y jugable del clásico Tetris.
- Permitir que el jugador coloque obstáculos antes de iniciar la partida.
- Validar correctamente el nombre del jugador antes de comenzar.
- Registrar y mostrar los mejores puntajes al finalizar el juego.

Objetivos No Logrados

No se logró implementar el guardado y la carga de partida. En su momento, consideré que sería muy difícil porque no sabía cómo guardar el estado del tablero ni cómo volver a cargarlo respetando los colores y posiciones de cada pieza. Por ende, se dejó para último, si todavía se tenía tiempo para entregar el proyecto se haría a lo último, pero no fue el caso, por ende, no se realizó.

Funciones que dieron problemas

- `eliminar_lineas()`: Al eliminar una línea y bajar las líneas hacia abajo se bajaban los obstáculos o se eliminaban pero al final se logra que los obstáculos queden fijos. Al eliminar la línea no se borra si hay dos líneas completas si no una por una.
- `rotar_pieza()`: En un principio no se sabía cómo implementarse pero se decidió hacer invirtiendo las columnas por las filas y viceversa, esa solución me dio muchos

problemas ya que al rotar no se rotaba por la celda del medio al final se soluciona parcialmente ya que la figura (1) rota de la posición (0,0).

- `fijar_pieza()`: Se tuvo problemas a la hora de fijar por que se quería poner los colores al fijar pero estando todas las celtas con (1) no se sabía exactamente qué color era, así que después de pensar, hice que esa celda tuviera el valor del color de la figura así a la hora de extraer el archivo y actualizar el tablero se actualizara dependiendo el valor de la celda.
- `mover()`: No se sabía cómo implementar, luego de pensar deduje que la mejor opción sería hacer una biblioteca tipo sombra, así dependiendo el valor de las coordenadas dadas al hacer click en las flechas de mover analiza si en esa posición colisionaría, si no colisiona entonces el movimiento es válido.
- organizar la función de jugar y su bucle: La estructura del ciclo principal del juego me dio problema ya que no sabía cómo llamar a cada función. Manejar los eventos del teclado y refrescar el canvas constantemente fueron parte del problema a hora de llamar.

A pesar de esas dificultades, se logró un juego totalmente funcional, estable y visualmente agradable.

Librerías Utilizadas

- `tkinter`: para la interfaz gráfica del juego.
- `random`: para seleccionar piezas aleatoriamente.
- `re`: para validar el nombre del jugador con una expresión regular.
- `time`: para generar pausas breves durante mensajes.
- `PIL (Pillow)`: para cargar imágenes PNG dentro de la ventana del juego.
- `tkinter.messagebox`: para mostrar alertas e información al jugador.

Archivos del Proyecto

- `Codigo.py`: archivo principal que contiene toda la lógica del juego.
- `Proyecto Tetris/Ranking.txt`: archivo donde se almacenan los puntajes más altos.
- `Proyecto Tetris/Matriz.txt`: archivo de referencia para el tablero inicial.
- `Proyecto Tetris/fondo1.png` y `ra.ico`: archivos gráficos que mejoran la presentación visual.

Conclusión

Este proyecto me permitió desarrollar un poco más la lógica de programación, manejo de eventos, uso de archivos, e implementación de interfaces gráficas en Python. Aunque todo el proyecto presentó desafíos se logró la implementación de la mayoría de los requerimientos del proyecto.