

Tectris

Desarrollado por: Geyer Chirino

Carné: 2024165693

Fecha: 01/06/2025

Lenguaje: Python

Librerías principales: tkinter, PIL

Taller de Programación

Profesor: Cristian Campos

2. Manual de Usuario

Requisitos del Sistema

- Python 3.x instalado
- Librerías adicionales: Pillow (para PIL)

Instalación

1. Instalar Python desde python.org
2. Instalar las librerías necesarias con pip:

bash

Copy

Download

pip install pillow

Controles

- **Flecha Izquierda:** Mover pieza a la izquierda
- **Flecha Derecha:** Mover pieza a la derecha
- **Flecha Abajo:** Acelerar caída
- **Flecha Arriba:** Rotar pieza
- **Botones en pantalla:** Guardar, Cargar, Pausa, Salir

Funcionalidades

- Modo normal de Tetris
- Sistema de puntuación
- Guardado y carga de partidas
- Menú de pausa
- Efectos visuales

3. Video Demostrativo

Aun no lo e hecho

4. Descripción del Problema

El proyecto consiste en implementar un clon del clásico juego Tetris con funcionalidades adicionales:

- Sistema de guardado de partidas
- Interfaz gráfica amigable
- Mecánicas extendidas (piezas especiales, rotaciones)
- Manejo de puntuaciones

5. Diseño del Programa

Estructura Principal

- **ventanaInicio()**: Pantalla inicial en modo pantalla completa
- **ventanaPrincipal()**: Menú con opciones de juego
- **matriz()**: Configuración del tablero de juego

Algoritmos Clave

1. **Generación de piezas**: Selección aleatoria entre 8 formas diferentes
2. **Rotación**: Transformación matemática de coordenadas alrededor de un pivote
3. **Detección de colisiones**: Verificación de posiciones válidas
4. **Eliminación de líneas**: Escaneo de filas completas y actualización de la matriz

Decisiones de Diseño

- Uso de diccionarios para almacenar formas y valores del tablero

- Implementación de efectos visuales para mejor experiencia de usuario (más o menos)
- Sistema de guardado en archivos de texto plano

6. Librerías Usadas

- **tkinter**: Interfaz gráfica principal
- **PIL (Pillow)**: Manejo de imágenes para los gráficos
- **random**: Generación de piezas aleatorias
- **os**: Manejo de rutas para guardado de archivos

7. Análisis de Resultados

Objetivos Alcanzados

- El juego utiliza una matriz de 22 filas y 12 columnas
- Bordes representados con "+"
- Celdas vacías con 0
- Bloques con identificadores de pieza ("O", "I", "L", etc.)
- Interfaz gráfica completa
- Sistema de guardado y carga
- Mecánica de rotación de piezas
- Movimiento con Teclas Direccionales
 - Flecha izquierda: mover izquierda
 - Flecha derecha: mover derecha
 - Flecha abajo: acelerar caída
 - **Nota**: Flecha arriba se usa para rotar
- Piezas que Aparecen Una a Una

- Piezas Aleatorias con Colores
- Eliminación de Líneas Completas
- Fin del Juego
- Puntaje Obtenido
- Guardar Estado
- Cargar Partida
- Caída Automática
-

Objetivos No Alcanzados

- **No implementado:** El código actual no muestra obstáculos internos en el tablero
- Nombre del Jugador
- Ranking/Top 10
- Niveles con Dificultad Progresiva

Razones: Limitaciones de tiempo y complejidad adicional requerida para estas características, principalmente las hitboxes de las piezas, tuve que reestructurar el juego unas cuantas veces porque el enfoque que seleccione de inicio me dio muchos problemas, aparte que empecé a hacer funciones sin un orden concreto y termine incluso haciendo funciones que borre del producto final ya que no pude implementarlas correctamente.

8. Conclusión

El proyecto Tectris cumple con los objetivos básicos de un clon funcional de Tetris, con características adicionales como el sistema de guardado que añaden valor. La implementación demuestra un buen manejo de:

- Manipulación de interfaces gráficas

- Lógica de juego

Las principales áreas de mejora serían la optimización del código y la adición de más características avanzadas.