

TEC Centro Académico de limón

Proyecto 2: Tetris

Taller programación

Grupo 61

Profesor: Cristian Paz Campos Agüero

Estudiante: Cesar Andrés Espinoza Herrera

1/6/2025

Manual de usuario:

Este proyecto consiste en la implementación de interfaz gráfica tkinter, pero en especial del Juego Tetris, en este se practica funciones básicas de interfaz gráfica como la creación de una canvas, uso de coordenadas y demás, este juego le permite al usuario mover y rotar una pieza (trenomino) que cae automáticamente, dentro de un tablero lo que se busca es ir completando líneas para subir la puntuación.

El juego incluye características adicionales como:

- Guardado y carga de partidas.
- Estadísticas y ranking de jugadores.
- Interfaz gráfica amigable.
- Control del jugador mediante el teclado.

Requisitos para jugar:

Un sistema operativo compatible con el lenguaje de programación Python, y usar la librería tkinter además de la biblioteca de random, las cuales se llaman de la siguiente manera:

- `from tkinter import *`
- `Import random`
- `from tkinter import messagebox`
- `from tkinter import simpledialog`

Esas serian todas las bibliotecas necesarias, las ultimas 2 librerias son para mostrar y pedir información por medio de ventanas emergentes.

El flujo del código seria el siguiente, si se presiona el botón “iniciar\_juego” te pide nombre\_jugador”, ya después de eso llama a la función nuevo\_juego() donde esta función también llama a la función cargar\_tablero() donde esta función carga la información que se le paso a tablero, la información se obtiene de un archivo llamado “base.txt” donde este contienen la información de como se va a dibujar el tablero, contienen el total de filas(12) y el total de columnas(22), donde los bordes y obstáculos se representan por “+”, y los espacios vacíos, después de obtener la información del tablero se llama a la función, dibujar\_tablero(canvas, tablero), aquí obtiene el canvas (donde se va a dibujar), y el contenido del tablero, aquí lee posición por posición para darle color a las celas y así poder dibujar correctamente el tablero, también en esta función muestro el porcentaje obtenido a un lado de la cuadrícula de juego, después de esto llamo a la función generar\_pieza() que con el uso de la librería random se selecciona al azar una de las piezas definidas en todas las piezas.

Nota = arriba en el código se definieron los trenominos, estos consisten en matrices, pero después todas las piezas las junto en una lista, para así poder llamarla más fácilmente, cada pieza tiene un valor numero diferente, esto hace que cada pieza tenga un color diferente.

Después de esto se llama a la función dibujar\_pieza(), esta función lo que hace es que dibuja la pieza en el tablero, la dibuja por coordenadas, en x= 4, y= 0, donde x es la esquina superior izquierda, y dependiendo del aumento del valor en x, se movería hacia la derecha, y con y es igual la posición de y = 0 representa la esquina superior izquierda, y cada que ese valor

aumenta se movería hacia abajo, es como ver el la esquina inferior derecha de un plano cartesiano, después de esto llamo a la función `mover_pieza_hacia_abajo()`, que esta función hace que la pieza caiga automáticamente, y dependiendo del valor de puntuación la velocidad de caída es mayor, aquí llamamos contantemente a `dibujar_pieza()`, para hacer piezas nuevas, también están las funciones, `mover_derecha()`, `mover_izquierda()`, `mover_abajo()`, que estas funciones lo que hacen que por medio de presionar las teclas, izquierda, derecha y abajo en el teclado, la pieza se moverá en esa dirección, pero aquí llamamos a la función `hay_colision()`, que aquí detecta si el movimiento se puede hacer o no, para que las piezas no se traspasen entre si, y también verificar los limites del tablero, y si una pieza colisiona, con otra pieza o obstáculo que este debajo de la pieza, entonces en ese momento se llama a la función `fijar_pieza_en_tablero()`, esto lo que hace es guardar la información de la pieza en el tablero, para de esta forma cuando se llame a la función `eliminar_lineas_completas()`, se pueda verificar si una línea es diferente a "0" y que no haya "+" en esa fila, si estas condiciones se cumplen y la fila esta llena, esta se eliminaría, y se sustituiría por la fila de arriba, también se puede rotar la pieza con la tecla arriba, esto lo que hace es modificar la matriz de la pieza, cambiando las columnas por filas, y las filas por columnas.

También se verifica que la pieza si pueda rotar para así evitar problemas con que la pieza supere los limites o cosas similares, esto se hace con la función `puede_rotar()`, en todo este proceso se utiliza una función `redibujar()`, donde también se crean las piezas, el juego se pierde cuando las piezas no encuentran campos vacíos para descender, entonces en ese momento se llama a la función, `mensaje_game_over()`, para detener el juego y así guardar los datos en estadística, antes de pasar a estadística, en el código hay 4 botones, 1 = `iniciar_juego`, 2 = `guardar_juego`, 3 = `cargar_juego` y 4 estadísticas.

Que justo voy a documentar estadística, esta función es lo que hace, que cada vez que se llame a la función, `mensaje_game_over()`, se guarda `nombre_jugador` y puntuación, este se guarda en el documento "estadisticas.txt", cuando se presiona el botón, nos da a elegir 3 opciones: 1 = mostrar nombre jugador actual, 2= mostrar puntuación de jugador actual, y 3 = mostrar ranking (top 10), donde se muestra la puntuación y el nombre de las personas con mayor puntuación, se muestra de mayor a menor.

Después de eso tenemos a los botones, `guardar_juego`, que esta función lo que hace es crear un archivo nuevo donde se guarda la matriz de la partida actual, junto con otros parámetros, los cuales son, la puntuación del jugador, el nombre del jugador, la pieza actual, un mensaje para saber dónde termina la pieza actual, la posición de la pieza actual, y por ultimo el tablero, `cargar_juego()`, funciona similar que mis otros códigos, lee la matriz y recrea el escenario a partir de eso.

Y eso seria el manual, de usuario, para saber más a fondo de cada función, visitar la documentación interna.

Video YouTube:

Video = <https://youtu.be/NFyAfKL3UFc>

### Descripción del Problema

El propósito de este proyecto es implementar una versión funcional del clásico juego Tetris utilizando el lenguaje de programación Python, integrando una interfaz gráfica a través de la librería Tkinter. El principal desafío consiste en simular el comportamiento dinámico del juego:

caída automática de piezas, rotación, detección de colisiones, líneas completas, y control mediante el teclado.

Aparte de eso, se decidió implementarle funciones auxiliares como guardado y carga de partidas, estadísticas personales y ranking de puntuación, lo que requiere el uso de archivos para persistencia de datos. El objetivo es construir un sistema estable, visualmente claro y funcional, se busca que este proyecto no solo muestre el código como tal, si no que también con algunos cambios, pueden mejorar considerablemente el código, dejándolo mas extenso y que abarque más cosas.

Diseño del programa:

Se decidió utilizar un modelo sencillo sin tanta funciones propias de Python, si use ciertas funciones como len además de len también se usaron funciones globales, para sabe la longitud de ciertas cosas, pero por lo general todo se soluciona con bucles, y use un modelo simple para ahorrar tiempo y que el código no fuera tan difícil de entender

Librerías usadas:

- from tkinter import \*
- Import random
- from tkinter import messagebox
- from tkinter import simpledialog

Además de eso las variables globales son clave para que el código sea funcional, las cuales son:

```
TAM_CELDA = 30
```

```
EXTRA = 10
```

```
ruta = ""
```

```
ruta1 = "base"
```

```
pieza_actual = None
```

```
posicion_pieza = [4,0]
```

```
tablero = []
```

```
puntaje = 0
```

```
juego_terminado = False
```

```
nombre_jugador = ""
```

```
contador_juegos = 0
```

Análisis resultados:

Todas las funciones que se pidieron fueron implementadas y funcionan correctamente, no digo que estén perfectas, pero suelen funcionar correctamente, se diseñó interfaz gráfica, permitir moverse abajo, izquierda, derecha, permite rotar, elimina la línea completa (sin bajar los obstáculos del tablero), valido que los movimientos si se puedan hacer, guardo y restauro la partida, tengo las estadísticas de juego, y los extras, todo eso fue implementado correctamente en el código fuente

Bitácora:

La bitácora esta en el GitHub: <https://github.com/2025-Semestre-1/proyecto-2-Cesar6775/commits/main/>

Archivos para el correcto funcionamiento del código:

Base.txt = aquí es donde se guarda el tablero original

Estadísticas.txt = aquí es donde se guardan los datos de estadísticas

Proyecto2.py = que es el código fuente con todas las funciones

Conclusión:

La creación de este proyecto, me ayudo a asentar las bases del uso de tkinter, creación de canvas, uso de coordenadas y demás, además de eso también se volvió a manipular archivos, y matrices, implementación de comandos por tecla.

Se lograron las funciones que se pedían en el archivo guía, aunque mi código no sea el mas ordenado, las funciones que se usan las unas con las otras están bastante cerca entonces es fácil encontrar una función para modificarla, y la mayoría usa variables entonces hace que la modificación no sea tan difícil, en conclusión fue un buen reto, que se pudo lograr pero aun así se puede mejorar.