

Proyecto Programado #2 - Tetris

1. Historia

Tetris es un videojuego de tipo rompecabezas en el que el objetivo principal es organizar correctamente una serie de piezas geométricas que caen desde la parte superior de la pantalla. Estas piezas, llamadas *tetrominós*, están compuestas por cuatro bloques cuadrados unidos en diferentes formas.

El juego fue creado en 1984 por el ingeniero informático **Alexéi Pázhitnov** en la Unión Soviética, mientras trabajaba en la Academia de Ciencias de Moscú. Pázhitnov se inspiró en un juego de lógica con fichas llamado *Pentominós*, pero lo simplificó para hacerlo más ágil y jugable en computadoras de la época.



A medida que las piezas caen, el jugador debe moverlas lateralmente y rotarlas para encajarlas en la parte inferior de la pantalla. Cuando una fila se completa sin espacios vacíos, desaparece y otorga puntos al jugador. El reto consiste en evitar que las piezas se acumulen hasta alcanzar la parte superior del área de juego, lo que finalizaría la partida.

Tetris se convirtió en uno de los videojuegos más populares e influyentes de todos los tiempos, y ha sido adaptado a múltiples plataformas a lo largo de las décadas.

2. ¿Qué se busca con este proyecto?

El objetivo general de este proyecto es facilitar un acercamiento con el juego de tetris desde la perspectiva de la programación, de manera que las partes del juego y las reglas que lo enmarcan sean diseccionadas minuciosamente para desarrollar un diseño que sea programado mediante el lenguaje Python.

1. Practicar las habilidades en el desarrollo de aplicaciones de software
2. Ejercitar la toma de decisiones sobre el dominio del problema y de la solución
3. Aplicar los conceptos de programación y manejo de matrices.
4. Manejo de componentes para el uso de interfaz gráfica
5. Investigar funciones específicas como el manejo de teclado para controlar las piezas

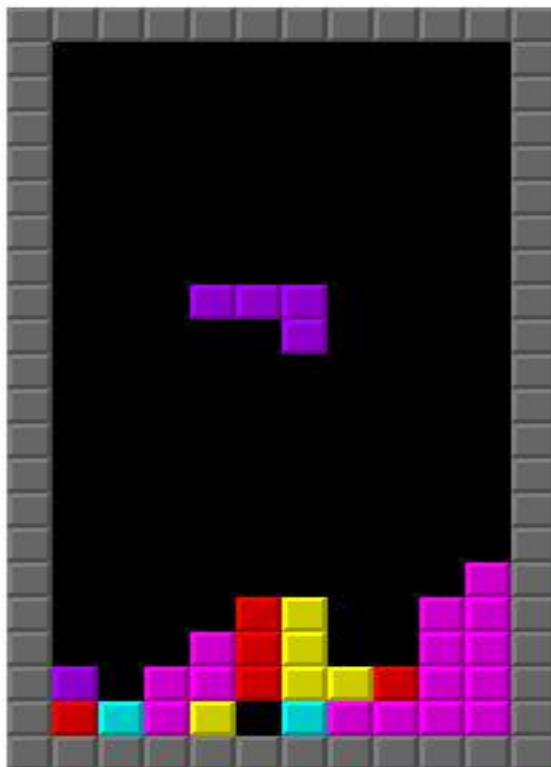
3. Proyecto por desarrollar

Este proyecto, aunque no requiere de un conocimiento profundo en cuanto a estrategias, sí requiere de un conocimiento detallado del juego de tetris (<https://play.tetris.com/>). Antes de iniciar el trabajo debe comprender por completo el funcionamiento del juego, ya que lo que se desea es reproducir el comportamiento de este.

Asegúrese de conocer a la perfección:

1. El modo de juego
2. Tamaño del tablero
3. Movimientos de las piezas
4. Piezas del Juego
5. Ejecución correcta del juego



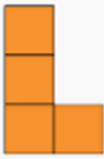

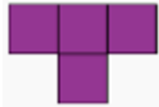


3.1. Estructura del Tablero y Matriz




+	+	+	+	+	+	+	+	+	+	+	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	+	+	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	1	1	1	0	0	0	+
+	0	0	0	0	0	1	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	0	+
+	0	0	0	0	0	0	0	0	0	1	+
+	0	0	0	0	1	1	0	0	1	1	+
+	0	0	0	1	1	1	0	0	1	1	+
+	1	0	1	1	1	1	1	1	1	1	+
+	1	1	1	1	0	1	1	1	1	1	+
+	+	+	+	+	+	+	+	+	+	+	+

Ilustración 1 Representación Gráfica y Matriz

3.2. Piezas del juego (Tetrominós)

Tetrominó	Pieza (Forma)	Descripción
O		Cuatro bloques en un cuadrado de 2x2
I		Cuatro bloques en línea recta
L		Una columna de 3 bloques con uno agregado debajo del lado derecho
J		Una columna de 3 bloques con uno agregado debajo del lado izquierdo
T		Una fila de tres bloques con uno agregado abajo en el centro
Z		Dos filas horizontales con un desplazamiento de la superior a la izquierda
U		Una fila de tres bloques con un agregado arriba de cada extremo

+		Una fila de tres bloques con un agregado abajo en el centro y otro arriba en el centro
---	---	--

3.3. Movimientos permitidos

- **Izquierda:** Mueve la pieza (Tetrominó) una columna a la izquierda, si no hay colisión con otra pieza o el borde del tablero
- **Derecha:** Mueve la pieza (Tetrominó) una columna a la derecha, si no hay colisión con otra pieza o el borde del tablero
- **Abajo:** Mueve la pieza (Tetrominó) una fila a la derecha, si no hay colisión con otra pieza o el borde del tablero
- **Rotación:** Todas las piezas (excepto la O/cuadro) se pueden **rotar 90°** en el sentido de las agujas del reloj. Rotar implica aplicar un cambio la matriz a la figura. La rotación se limita si la nueva posición colisiona con otras piezas o sale del tablero
- **Colocación:** Cuando la pieza toca el suelo del tablero o una pieza debajo, y ya no se mueve más, se bloquea (ya no se pueden hacer movimientos) y forma parte del tablero

3.4. Funcionalidades del juego

1. El tablero debe ser representado como una matriz de 22 filas y 12 columnas. Las celdas del borde del tablero deber ser representados con el símbolo "+", las celdas vacías con un 0 y cada bloque de los tetrominós con un 1
2. El tablero debe contener **obstáculos**, representados con un bloque similar a los del borde del tablero ubicado en el centro
3. Los tetrominós pueden moverse con las teclas direccionales (izquierda, derecha, abajo), nótese que la dirección **arriba no es correcta**, dado que las piezas van saliendo de la parte superior central del tablero.
4. Las piezas van apareciendo una a la vez, es decir, hasta que la pieza actual es colocada o bloqueada puede salir la siguiente
5. Las piezas deben ir saliendo de forma **aleatorio** y cada figura debe tener un color asignado

6. Cuando una línea es completada por bloques (celdas compuestas de 1s) esta se debe eliminar y desplazar las filas superiores hacia abajo. Cada línea **separada genera 100 puntos**.
7. El juego finaliza cuando una nueva pieza no puede ingresar al tablero o el usuario oprime el botón finalizar.

3.5. Estadísticas de Juego:

Por cada juego se finalice debe indicar lo siguiente:

- a) Nombre del jugador
- b) Puntaje Obtenido
- c) Mostrar el Ranking o posición con respecto a otros jugadores (Top 10)

3.6. Guardar estado de juego:

Permite al jugador guardar el estado del juego (matriz), se debe utilizar archivos de texto para esta funcionalidad. Adicionalmente, el usuario puede continuar y regresar el juego a su último estado.

3.7. Opcionales

- Permitir que los tetrominós tengan una caída automática sin necesidad de utilizar la flecha hacia abajo.
- Implementar niveles en el juego, es decir cada vez que se llega a cierta cantidad de puntos el juego debe incrementar la velocidad en la caída de los tetrominós

4. Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación **Python** y se deben desarrollar las funcionalidades de usuario por medio de interfaz gráfica con **TKInter**.

En el desarrollo del juego sólo se permiten movimientos válidos (investigar según se indica arriba).

Y considerar lo siguiente:

- Se deben manejar mensajes claros al usuario
- Realizar validaciones de los diferentes movimientos de los tetrominós
- Toda función built-in de Python que deseen utilizar debe ser validada con el profesor.
- En el desarrollo del programa se deberá utilizar iteración.

5. Documentación

El código fuente debe tener documentación interna, con comentarios **precisos** y bien ubicados. Cada función debe tener descripción, entradas, salidas y restricciones.

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos.

Para la documentación **interna**, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas y restricciones.

La documentación **externa** deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Un **enlace** a un video en **youtube** donde muestre la **funcionalidad** de su aplicación.
Su duración no debe ser mayor a los 15 minutos.
4. Descripción del problema.
5. Diseño del programa: decisiones de desarrollo, algoritmos usados.
6. Librerías usadas: creación de archivos, etc.
7. Análisis de resultados: lista detallada de objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. **Bitácora** en GitHub con los **commit** por usuario incluyendo su descripción al momento de hacerlo.
9. Conclusión (es), sería su punto de vista sobre el programa desarrollado

6. Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación. El proyecto programado tiene un valor de **20%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

- Documentación interna 5 pts.
- Documentación externa 5 pts.
- Funcionalidad 75 pts (ver detalle en Software a Desarrollar)
- Revisión del proyecto 10 pts (**Defensa del proyecto, es presencial**).
- Hora de Entrega 5 pts.

7. Forma de trabajo

El trabajo se debe realizar de forma individual.

8. Aspectos administrativos

Crear 2 carpetas llamadas **documentación** y **programa**, en la primera deberá incluir el documento **PDF** solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de este proyecto programado.

URL del proyecto: <https://classroom.github.com/a/Et4r0lVo>

Deben modificar el archivo llamado **README.md**, este archivo debe contener la siguiente información:

- Nombre del Estudiante
- Número de carné del estudiante
- Enlace del video
- Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Con este enlace <https://docs.github.com/es/github/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax> le ayudará a dar formato al archivo README.md.

9. Rúbrica de evaluación

FUNCIONALIDAD	80%	0%
Rubro	0,75	Obtenido
Juego		
Diseño de Interfaz grafica	10	
Juego		
Permite moverse abajo, izquierda y derecha	15	
Permite rotar	15	
Eliminar línea completada*	15	
Valida movimientos	15	
Guardar y restaurar partida	10	
Estadísticas del juego	20	
Extras		
Animación	10	
Animación con tiempo	10	
TOTAL	120	0

10. Entrega

En el rubro de “Hora de Entrega” valdrá 5 puntos de la nota total del proyecto, según la siguiente escala:

- Si se entrega antes de las 11:55:55 PM del viernes 30 de mayo de 2025, 5 puntos.

- b. Si se entrega antes de las 11:55:55 AM del sábado 31 de mayo de 2025, 2.5 puntos.
- c. Si se entrega antes de las 11:55:55 PM del sábado 31 de mayo de 2025, 0 puntos.

Después de este punto, **NO SE ACEPTARÁN** más trabajos.

Los archivos fuentes pueden ser revisados en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100% original y en caso de plagio se asignará nota cero.**