

Facultad de Ingeniería Universidad de San Carlos de Guatemala Introducción
a la Programación y Computación 1 - Sección A
Catedrático. Ing. Neftali Calderón
Tutor académico Moises Conde



PRÁCTICA UNICA

Juego de Tetris con Matrices

OBJETIVOS

1. Generales

- Familiarizar al estudiante con el lenguaje de programación JAVA mediante la implementación de un juego clásico utilizando estructuras de datos básicas.

2. Específicos

- Utilizar el lenguaje de programación Java como herramienta de desarrollo de software interactivo.
- Construcción de aplicaciones de consola con interacción en tiempo real.
- Implementación de sentencias de control, ciclos y arreglos bidimensionales (matrices).
- Aplicación de conceptos de manipulación de matrices para representación gráfica y lógica de juego.
- Manejo de eventos de teclado y actualización dinámica de la consola.

DESCRIPCIÓN

La Facultad de Ingeniería requiere una aplicación educativa que permita a los estudiantes comprender la manipulación de matrices mediante un juego interactivo. Se debe desarrollar una versión funcional del clásico juego **Tetris** en consola, donde el tablero de juego y las piezas serán representados mediante arreglos bidimensionales.

El juego debe implementar la mecánica completa de Tetris, incluyendo movimiento de piezas, rotación, detección de colisiones, eliminación de líneas completas y sistema de puntuación.

ESPECIFICACIONES DEL JUEGO

MENÚ PRINCIPAL

Al iniciar la aplicación de consola, se debe mostrar el menú principal con las siguientes opciones:

1. **Jugar Partida Nueva**
2. **Ver Mejores Puntajes**
3. **Ver Estadísticas Generales**
4. **Salir**

Nota: En todo momento la aplicación debe mostrar su número de carné en la parte superior de la interfaz.

CONFIGURACIÓN DEL TABLERO

El tablero de juego debe cumplir con las siguientes especificaciones:

- **Dimensiones:** 20 filas x 10 columnas (matriz de 20x10)
- **Representación:** Cada celda del tablero se representa con caracteres ASCII
 - Celda vacía: · (punto medio)
 - Celda ocupada: ■ (bloque sólido)
 - Pieza activa: ░ (bloque medio)
- **Área de vista previa:** Mostrar la siguiente pieza que aparecerá (matriz de 4x4)

PIEZAS DEL JUEGO (TETROMINOS)

Se deben implementar las 7 piezas clásicas de Tetris, cada una representada como una matriz:

Pieza I (Cyan):

```
[0][1][0][0]
[0][1][0][0]
[0][1][0][0]
[0][1][0][0]
```



Pieza O (Amarillo):

```
[1][1][0][0]
[1][1][0][0]
[0][0][0][0]
[0][0][0][0]
```

Pieza T (Morado):

```
[0][1][0][0]
[1][1][1][0]
[0][0][0][0]
[0][0][0][0]
```

Pieza S (Verde):

```
[0][1][1][0]
[1][1][0][0]
[0][0][0][0]
[0][0][0][0]
```

Pieza Z (Rojo):

```
[1][1][0][0]
[0][1][1][0]
[0][0][0][0]
[0][0][0][0]
```

Pieza J (Azul):

```
[1][0][0][0]
[1][1][1][0]
[0][0][0][0]
[0][0][0][0]
```

Pieza L (Naranja):

```
[0][0][1][0]
[1][1][1][0]
[0][0][0][0]
[0][0][0][0]
```

Cada pieza debe de almacenarse en una matriz de 4X4 donde 1 representa parte de la pieza y 0 representa espacio vacío.

1. Movimiento de Piezas

Las piezas deben poder moverse utilizando las siguientes teclas:

- **A o ←**: Mover pieza a la izquierda
- **D o →**: Mover pieza a la derecha
- **S o ↓**: Acelerar caída de la pieza
- **W o ↑**: Rotar pieza 90° en sentido horario
- **ESPACIO**: Caída instantánea (Hard Drop)
- **P**: Pausar/Reanudar juego
- **ESC**: Salir al menú principal

2. Rotación de Piezas

La rotación debe implementarse mediante transposición y reflexión de la matriz que representa la pieza:

- La matriz de la pieza se transpone (intercambiar filas por columnas)
- Se invierte el orden de las columnas para lograr rotación horaria
- Se debe validar que la rotación no cause colisión con el borde o piezas existentes
- Si hay colisión, la rotación no se ejecuta (Wall Kick básico opcional)

3. Detección de Colisiones

Se debe implementar un sistema de detección de colisiones que verifique:

- Colisión con los bordes del tablero (izquierda, derecha, inferior)
- Colisión con piezas ya colocadas en el tablero
- La detección debe realizarse antes de ejecutar cualquier movimiento

4. Colocación de Piezas

Cuando una pieza no puede descender más:

- Los valores de la matriz de la pieza se transfieren al tablero principal
- Se genera una nueva pieza aleatoria
- Se actualiza el área de vista previa con la siguiente pieza

5. Eliminación de Líneas

Después de colocar cada pieza, se debe:

- Recorrer cada fila del tablero
- Identificar filas completamente llenas
- Eliminar las filas completas
- Desplazar hacia abajo todas las filas superiores
- Actualizar la puntuación según líneas eliminadas

6. Sistema de Puntuación

El sistema de puntos debe seguir las siguientes reglas:

- **1 línea eliminada**: 100 puntos × nivel actual
- **2 líneas eliminadas simultáneas**: 300 puntos × nivel actual
- **3 líneas eliminadas simultáneas**: 500 puntos × nivel actual
- **4 líneas eliminadas simultáneas (Tetris)**: 800 puntos × nivel actual

- **Soft Drop (tecla S):** 1 punto por celda descendida
- **Hard Drop (tecla ESPACIO):** 2 puntos por celda descendida

7. Sistema de Niveles

El juego debe incrementar la dificultad progresivamente:

- **Nivel inicial:** 1
- **Velocidad de caída:** Disminuye 100ms por cada nivel (máximo velocidad: 100ms)
- **Cambio de nivel:** Cada 10 líneas eliminadas
- La velocidad base en nivel 1 debe ser de 1000ms entre cada descenso automático

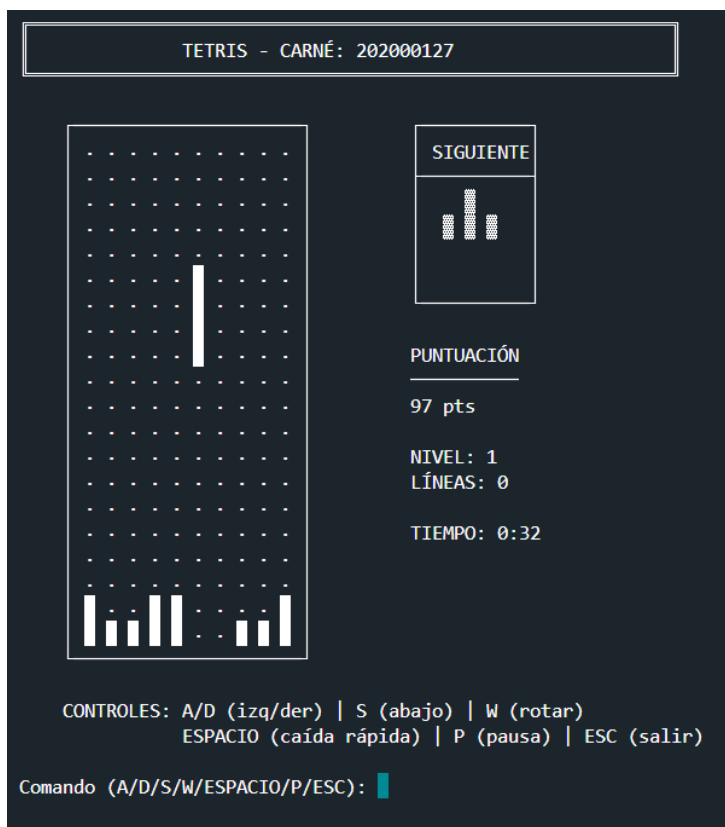
8. Condición de Fin de Juego

El juego termina cuando:

- Una nueva pieza no puede colocarse en su posición inicial (columnas 3-6, fila 0)
- Se debe mostrar mensaje de "GAME OVER"
- Se debe solicitar al jugador que ingrese su nombre (máximo 15 caracteres)
- Se debe guardar el puntaje con el nombre del jugador

INTERFAZ DE JUEGO EN CONSOLA

Ejemplo de la interfaz durante el juego debe mostrar:



PERSISTENCIA DE DATOS

Archivo de Mejores Puntajes

Se debe mantener un archivo de texto (mejores_puntajes.txt) con el top 10 de puntuaciones:

```
mejores_puntajes.txt
1  Moises,394,1,1,2025-12-02 15:24,1:02
2
```

Archivo de Estadísticas Generales

Se debe mantener un archivo (estadisticas.txt) con datos agregados:

```
estadisticas.txt
1  PARTIDAS_JUGADAS: 1
2  TOTAL_LINEAS: 1
3  TOTAL_PUNTOS: 394
4  PROMEDIO_PUNTOS: 394
5  PROMEDIO_NIVEL: 1
6  MEJOR_PUNTAJE: 394
7  TETRIS_REALIZADOS: 0
8
```

ENTREGABLES

1. Manual de Usuario: Documento que explique cómo ejecutar y jugar el juego.
2. Diagrama de Flujo: Diagramas que muestren:
 - a. Flujo general del juego
 - b. Lógica de rotación de piezas
 - c. Lógica de detección de colisiones
 - d. Lógica de eliminación de líneas
3. Manual Técnico: Documento que explique:
 - a. Estructura del código
 - b. Clases y métodos principales
 - c. Estructura de datos utilizada
 - d. Algoritmos implementados
4. Código fuente de la aplicación: Proyecto completo de Java con todos los archivos .java
5. Ejecutable de la aplicación: Archivo .jar ejecutable

RESTRICCIONES

1. El programa debe ser desarrollado en lenguaje JAVA.
2. El programa debe ser de consola (no se permite interfaz gráfica Swing/JavaFX).
3. No se permite el uso de estructuras que implemente Java (ArrayList, LinkedList, etc.). Solo se permiten arreglos primitivos.
4. No se permite utilizar código copiado o bajado de internet.
5. Para limpiar la consola y crear la animación del juego, puede utilizar códigos ANSI escape o la librería JLine (debe ser aprobada por el tutor académico).
6. Las copias parciales o totales obtendrán nota 0 y reporte a la Escuela de Ciencias y Sistemas.
7. El estudiante no tendrá derecho a calificación si no envía el código fuente de la aplicación.
8. No se aceptan entregas tardías.
9. Totalmente prohibido el uso de IA para el desarrollo de esta práctica

ENTREGA

Fecha de entrega: Se debe enviar el enlace del repositorio por medio de UEDI, antes de las **9:00 del 08/12/2025**

El repositorio debe de tener el siguiente nombre **#carne-ipc1-Dic2025** y debe contener una carpeta con el nombre **Practica** con lo siguiente:

- Carpeta /src con el código fuente
- Carpeta /docs con los manuales y diagramas
- Archivo .jar ejecutable en la raíz