**TETRIS EN JAVA**

Andreina Arevalo Bencardino – 192398

Miguel Felipe Quintero Santos – 192334

Sebastián David Ropero – 192331

Luis Felipe carrascal Ortiz – 192391

DOCENTE: Jesús Eduardo Guerrero

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

FECHA : 6 Diciembre2024

**INFORME DE TETRIS EN JAVA**

**Introducción**

El código proporcionado implementa una versión básica del juego Tetris utilizando Java y la librería Swing para la interfaz gráfica. El juego permite controlar piezas que caen, realizar movimientos, detectar colisiones, eliminar filas completas y visualizar todo en un tablero de juego. Este informe explica la estructura y el funcionamiento del código, detallando los componentes y la lógica del juego.

**Descripción General**

El código contiene la implementación de un juego de Tetris usando la clase Tetris, que extiende de JPanel y maneja toda la lógica del juego y la parte gráfica. El juego se actualiza periódicamente mediante un Timer, que hace que las piezas caigan automáticamente. Las piezas pueden moverse usando las teclas de dirección del teclado y se rotan con la tecla de flecha hacia arriba.

**Componentes Principales**

1. **Constantes de Configuración**:
   * TABLERO\_ANCHO: Representa el número de columnas del tablero, con un valor de 10.
   * TABLERO\_ALTO: Representa el número de filas del tablero, con un valor de 20.
   * TAMAÑO\_BLOQUE: El tamaño de cada bloque en píxeles, con un valor de 30.
2. **Variables de Estado**:
   * forma actual: Referencia a la pieza que se encuentra en movimiento.
   * Posicion\_Actual\_X, Posicion\_Actual\_Y: Las coordenadas de la pieza actual en el tablero.
   * Tablero: Una matriz que representa el estado de cada celda del tablero. Si true, indica que la celda está ocupada.
   * Colores\_Tablero: Una matriz que guarda los colores de las piezas fijadas en el tablero.
   * gameOver: Un booleano que indica si el juego ha terminado.
   * timer: Un objeto Timer que actualiza el juego a intervalos regulares.

**Lógica del Juego**

**Inicialización**

Cuando se crea una nueva instancia de Tetris, se configura el panel con un tamaño y color de fondo. El tablero se inicializa como vacío, y se selecciona aleatoriamente la primera pieza mediante el método getRandomPiece().

**Movimiento de las Piezas**

Las piezas pueden moverse mediante las teclas de dirección:

* **Izquierda**: Mueve la pieza hacia la izquierda.
* **Derecha**: Mueve la pieza hacia la derecha.
* **Abajo**: Mueve la pieza hacia abajo.
* **Arriba**: Rota la pieza 90 grados.

**Detección de Colisiones**

El método canMove(FiguraMain piece, int x, int y) verifica si la pieza puede moverse a una nueva posición sin chocar con otras piezas ya fijas o los límites del tablero.

**Fijar la Pieza al Tablero**

Cuando la pieza ya no puede bajar más, se "fija" al tablero usando el método fixToGround(). Este método actualiza las matrices Tablero y Colores\_Tablero, marcando las posiciones ocupadas por la pieza.

**Eliminación de Filas Completas**

Cuando una fila se llena completamente de piezas, el método removeRow(int y) la elimina y las filas superiores caen hacia abajo para rellenar el espacio vacío.

**Generación de Nuevas Piezas**

Una vez que una pieza se fija en el tablero, se genera una nueva pieza usando el método spawnNewPiece(). Si no hay espacio suficiente para colocar una nueva pieza, el juego termina.

**Fin del Juego**

Cuando no se puede generar una nueva pieza debido a la falta de espacio, el juego termina. El mensaje "Has perdido" se muestra en la pantalla.

**Interfaz Gráfica**

El método paintComponent(Graphics g) es responsable de dibujar el estado gráfico del juego:

* Dibuja el tablero con líneas grises para cada celda.
* Dibuja las piezas en movimiento y las piezas que ya están fijas en el tablero.
* Si el juego ha terminado, muestra el mensaje de "Game Over".

**Funciones y Métodos Importantes**

1. **Métodos de Movimiento**:
   * moveDown(): Mueve la pieza hacia abajo. Si no es posible, fija la pieza en el tablero.
   * moveLeft(), moveRight(): Mueven la pieza hacia la izquierda o hacia la derecha, respectivamente.
   * rotatePiece(): Rota la pieza 90 grados si la nueva posición no tiene colisiones.
2. **Métodos de Gestión de Filas**:
   * isRowFull(int y): Verifica si una fila está completamente llena.
   * removeRow(int y): Elimina una fila llena y mueve las filas superiores hacia abajo.
3. **Métodos Auxiliares**:
   * getRandomPiece(): Devuelve una pieza aleatoria para iniciar una nueva jugada.
   * spawnNewPiece(): Genera una nueva pieza en la parte superior del tablero.

**Manejo de Excepciones**

El código incluye manejo básico de excepciones en las operaciones clave, como el movimiento de las piezas, la fijación al tablero y la rotación. Las excepciones son capturadas y se registra un mensaje de error para facilitar la depuración en caso de que ocurra algún problema durante la ejecución.

Por ejemplo, si se intenta mover una pieza fuera de los límites del tablero, la excepción es capturada y el error es reportado, lo que permite que el programa continúe funcionando sin interrupciones graves.

**Roles**

**Andreina:** Documentación.

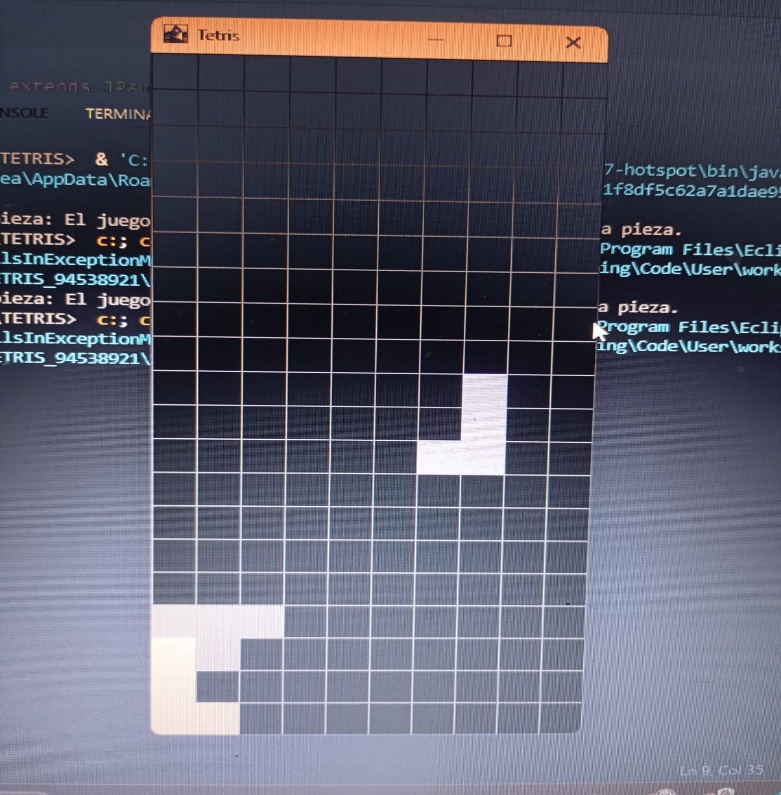
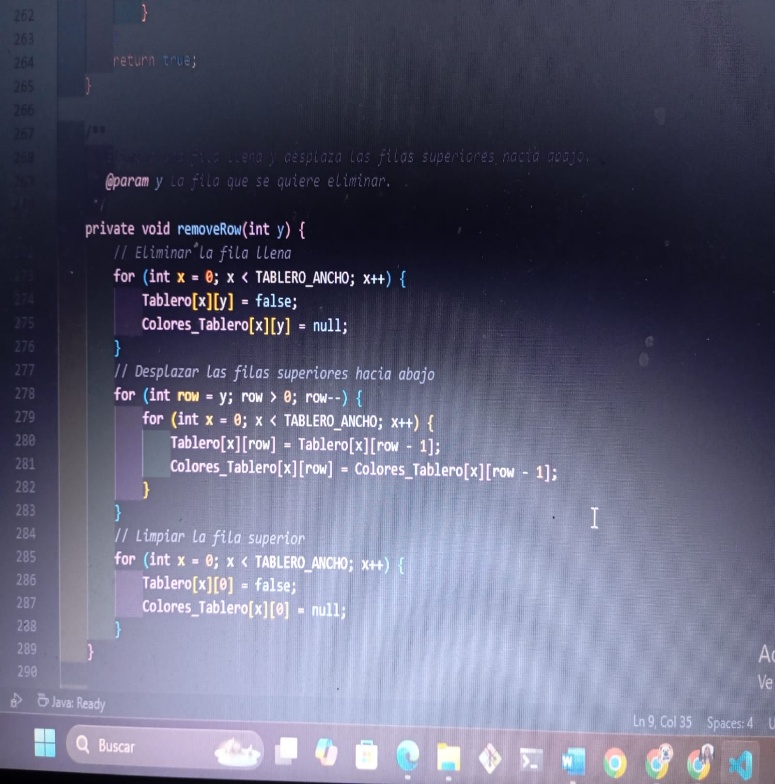
**Miguel Felipe:** Desarrollador.

**Sebastián David:** Diseño de la interfaz.

**Luis Felipe:** Debugger.

**Conclusión**

Nuestro código implementa de manera eficiente un juego básico de Tetris en Java utilizando la librería Swing para la interfaz gráfica. La funcionalidad básica del juego está completa, permitiendo el movimiento, rotación y eliminación de piezas, además de la detección de colisiones y el fin del juego. Está diseñado de forma modular, donde las clases separan la lógica del juego (movimientos de las piezas, detección de filas llenas, etc.), la representación gráfica del tablero y las piezas, y la gestión de las interacciones con el usuario a través del teclado.

****