**DR BR AMBEDKAR NATIONAL INSTITUTE OF**

**TECHNOLOGY, JALANDHAR**



**PROJECT REPORT : TETRIS GAME**

# Submitted to: Submitted by:

Dr Naveen Kumar Gupta Harman Singh Badgaal

Department of IT 21124038

Gurtej Singh Hud

21124036

Karanbir Singh

21124048

Navtej Singh Hud

21124070

**Introduction:**

The Tetris game is a popular puzzle game that challenges the player to move and rotate tetromino shapes to create a solid horizontal line without any gaps. Our Tetris game is a single player game where the computer randomly generates tetromino blocks (in the shapes of T, J, L, Z, S, I) that the user can rotate using their game controller. Tetrominoes can be stacked to create lines to be cleared by the computer and be counted as points that will be tracked. Once a tetromino passes the boundary of the screen the user will lose.

The Java game logic generates random tetromino blocks to drop, translate key inputs to rotation of blocks, detect and clear lines, keeps track of the score, and determine when the game has ended.

The game has been recreated on various platforms and languages, including Java, which is an object-oriented programming language. This project aims to create a Tetris game in Java using Swing and AWT libraries.

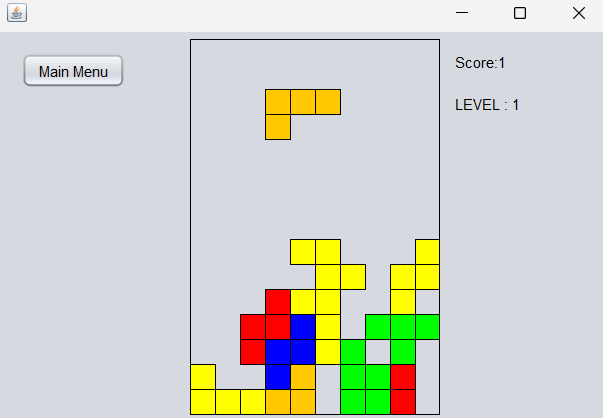


Fig 1: Screenshot of implementation of our Tetris Game

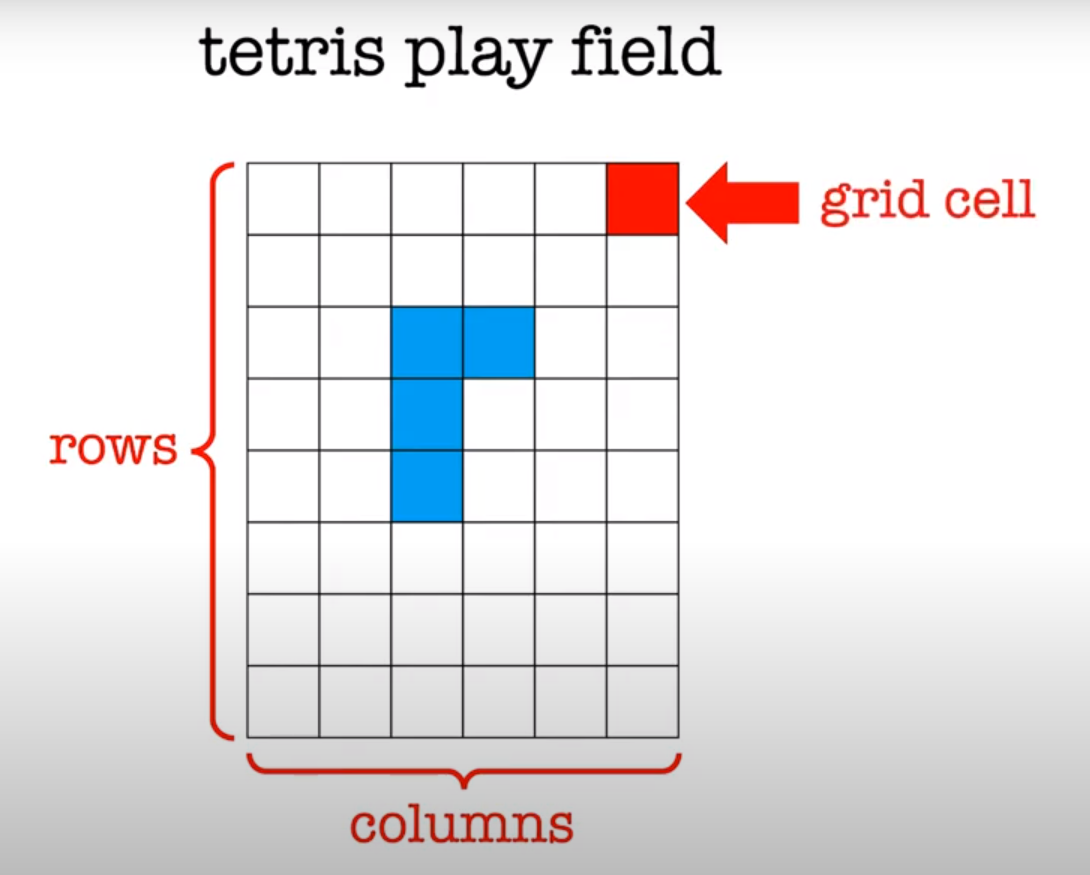
**Project Structure:**

The project consists of two packages, "tetris" and "tetrisblocks." The "tetris" package has four classes, "GameArea," "GameThread," "GameForm," and "Tetris block." The "tetrisblocks" package has six classes, "LShape," "SShape," "TShape," "IShape," "ZShape," and "JShape." These classes represent the different tetromino shapes used in the game.

**Class Descriptions:**

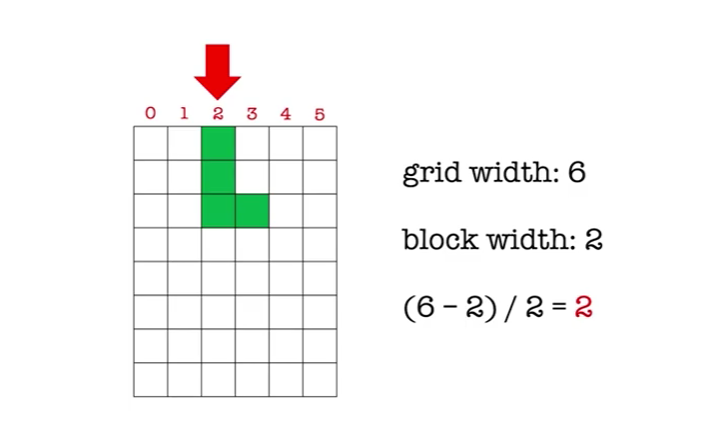
1. **GameArea:**

The GameArea class creates the game area where the tetromino shapes will be displayed. The class extends JPanel and implements ActionListener to handle events. The class also contains an array to hold the current state of the game.



**Game Area Class Functions:**

SpawnBlock(): This function is responsible for spawning a new tetromino block at the top of the game area. It chooses a random shape from the TetrisBlock classes and initializes it with a starting position at the top of the game area.



isBlockOutOfBounds(): This function checks if the current tetromino block is out of bounds of the game area. If the block is outside the boundaries, it returns true, indicating that the game is over.

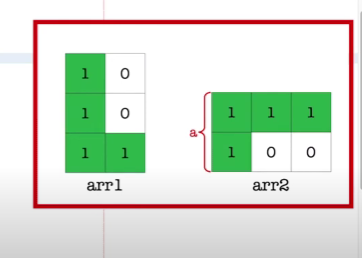
moveBlockDown(): This function moves the current tetromino block one cell down. It checks if the block is colliding with any other blocks or if it has reached the bottom of the game area. If the block cannot move down any further, it calls the SpawnBlock() function to spawn a new block.

moveBlockRight(): This function moves the current tetromino block one cell to the right. It checks if the block is colliding with any other blocks or if it has reached the right edge of the game area.

moveBlockLeft(): This function moves the current tetromino block one cell to the left. It checks if the block is colliding with any other blocks or if it has reached the left edge of the game area.

dropBlock(): This function drops the current tetromino block to the bottom of the game area. It checks if the block is colliding with any other blocks or if it has reached the bottom of the game area. If the block cannot move down any further, it calls the SpawnBlock() function to spawn a new block.

rotateBlock(): This function rotates the current tetromino block clockwise by 90 degrees. It checks if the rotated block is colliding with any other blocks or if it has reached the boundaries of the game area.

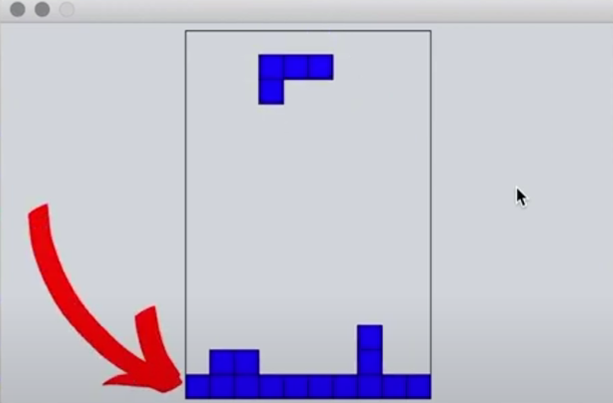


checkBottom(): This function checks if the current tetromino block has reached the bottom of the game area or if it is colliding with any other blocks.

checkLeft(): This function checks if the current tetromino block has reached the left edge of the game area or if it is colliding with any other blocks.

checkRight(): This function checks if the current tetromino block has reached the right edge of the game area or if it is colliding with any other blocks.

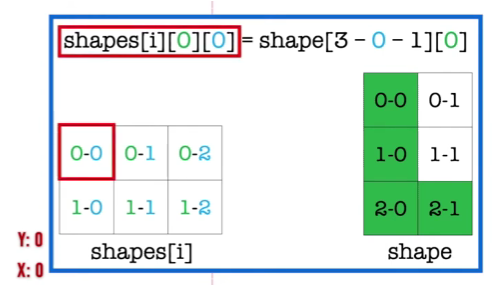
clearLines(): This function checks if any horizontal lines in the game area are complete and removes them. It then shifts the lines above the removed line down by one cell.



shiftLineDown(): This function shifts a single horizontal line down by one cell.

moveBlockToBackground(): This function moves the current tetromino block to the background of the game area. It does this by adding the block's coordinates to the game area array.

drawBlocks(): This function draws the current tetromino block on the game area.



drawBackground(): This function draws the background of the game area.

1. **GameThread:**

The GameThread class runs the game loop that updates the game state and renders the game area. The class extends Thread and implements Runnable. The class also has a method to pause the game and a method to restart the game.

**3) GameForm:**

The GameForm class creates the main form where the game area and game controls will be displayed. The class extends JFrame and implements ActionListener to handle events. The class also contains buttons to start, pause, and restart the game.

**4) Tetris block:**

The Tetris block class is an abstract class that defines the properties and behavior of tetromino shapes. The class has methods to move the shape left, right, down, and rotate. The class also has abstract methods to draw the shape and check for collision.

**TetrisBlock Class Functions:**

getShape(): This function returns the shape of the tetromino block as an array of coordinates.

getColor(): This function returns the color of the tetromino block.

getHeight(): This function returns the height of the tetromino block.

getWidth(): This function returns the width of the tetromino block.

getX(): This function returns the x-coordinate of the tetromino block.

getY(): This function returns the y-coordinate of the tetromino block.

getBottomEdge(): This function returns the y-coordinate of the bottom edge of the tetromino block.

getRightEdge(): This function returns the x-coordinate of the right edge of the tetromino block.

getLeftEdge(): (): This function returns the x-coordinate of the left edge of the tetromino block.

**5) LShape, SShape, TShape, IShape, ZShape, and JShape:**

These classes extend the Tetris block class and implement the draw and check collision methods for their specific shape.

**Authorship**

Harman and Gurtejworked together to write the game logic in Java.

Tasks like building various shapes,checking the collision and generating the bounds of a shape.

Karanbir worked on the Thread managing and synchronisation part of code.

Navtej designed the GameForm part and helped with setting up

integration between user input and output.

**Conclusion:**

In conclusion, the Tetris Game project built using Swing and AWT in Java is a fun and interactive way to engage users in a classic arcade game. The project is organized into two packages: Tetris and TetrisBlocks, containing classes with functions that handle the game's mechanics, such as spawning blocks, moving blocks, rotating blocks, checking collisions, clearing lines, and drawing the game area.

The GameArea class handles the game's mechanics by manipulating the game area's array and interacting with the TetrisBlock class to spawn, move, and draw tetromino blocks. The TetrisBlock class defines the properties of each block, such as its shape, color, height, width, and position.

Overall, this project demonstrates the use of object-oriented programming and graphical user interface (GUI) development using Swing and AWT in Java. It also provides a good foundation for further enhancements and customization, such as adding new game modes, increasing difficulty levels, and improving graphics.