OBJECT ORIENTED PROGRAMMING

FINAL PROJECT REPORT

”Tetris Elites”

Lecturer: JUDE JOSEPH LAMUG MARTINEZ, MCS

MADE BY:

Jeff Matthew Hadisaputro – 2602118906

BINUS UNIVERSITY INTERNATIONAL

2023

CONTENTS

PROJECT SPECIFICATON

Project Description

Tetris Elites is an iconic puzzle game that challenges players to showcase their strategic thinking, lightning-fast reflexes, and exceptional stacking abilities. With its timeless gameplay, this classic title guarantees hours of addictive entertainment.

The objective of Tetris Elites is to skillfully arrange falling tetromino shapes at the bottom of the playfield grid to form complete rows without any gaps. Whenever a row is successfully filled, it vanishes, and players are rewarded with points. The game continues until the stack of shapes reaches the top of the playfield, pushing players to constantly improve their skills and achieve higher scores.

During gameplay, tetromino shapes consisting of four squares descend from the top of the playfield towards the bottom. Players have the ability to maneuver the falling shapes using the arrow keys, allowing precise positioning to fit them into the desired locations. By utilizing the rotation control, players can adjust the orientation of the shapes, optimizing their placement within the available spaces. Once a shape reaches the bottom or lands on top of existing blocks, it solidifies in place, and a new shape appears. When an entire row is completely filled with shapes, it clears, causing any blocks above it to shift down, filling the vacated space.

Tetris Elites provides intuitive controls for smooth gameplay:

- "A" - Move Left: Shift the falling shape towards the left side of the playfield.
- "D" - Move Right: Slide the falling shape towards the right side of the playfield.
- "Q" - Rotate Anticlockwise: Change the orientation of the shape in a counterclockwise direction.
- "E" - Rotate Clockwise: Rotate the shape in a clockwise direction to adapt to the available spaces.
- "S" - Drop: Swiftly move the falling shape to the lowest possible position.
- "P" - Pause / Resume: Temporarily halt the game or resume gameplay as needed.
- "ENTER" - Start: Begin a new game and immerse yourself in the thrilling world of Tetris Elites.

The game logic in Tetris Elites is expertly managed by the Tetris class, which encompasses essential methods for initiating the game, updating the game state, and handling user input. The game board serves as the foundation for Tetris pieces (tiles) that descend from the top, requiring

players to strategically manipulate them to create complete horizontal lines. The code features variables and methods for efficiently managing the game state, including the current level, score, and the types of Tetris pieces in play. It also effectively handles user input for moving and rotating the pieces, pausing the game, and starting new games. As players progress, the game speed increases, and clearing lines contributes to their overall score. By leveraging various Java libraries such as java.awt, javax.swing, and java.util, the code seamlessly incorporates graphical components, event handling, random number generation, and timing.

User Interface

The visual interface for Tetris Elites is created by the programming using a Java package called Swing. It uses the JFrame component to build a window and the BorderLayout layout manager to set up the game board and score display. The SidePanel displays the score, level, and upcoming pieces while the BoardPanel displays the Tetris game board.

The tools required to generate buttons, windows, and other visual elements in Java programs are provided by Swing. It facilitates processing user input and screen images for the game. Swing is an addition to the Abstract Window Toolkit (AWT) and a library of Java Foundation Classes (JFC). As compared to AWT, Swing has significantly better functionality, new components, increased component features, and superior event handling with drag-and-drop support.

Without requiring significant modifications to the application code, Swing allows customization of the appearance and feel of each component in an application. Additionally, it has a pluggable look and feel capability that enables it to mimic the look and feel of native components while yet benefiting from platform independence. Swing stands apart from other native programs thanks to a specific characteristic that makes creating apps simple.

As a downloadable library, Swing is a component of Java Standard Edition 1.2 and is available for download. Internet Foundation Classes (IFC) was the first name of Netscape Communication Corporation's Java graphics library. IFC was first made available on December 16, 1996. When Sun Microsystems and Netscape Communication Corporation had the concept to combine IFC with other technologies in 1997, JFC began to take shape.

Object-Oriented Design

The process of developing a software system or application using an object-oriented paradigm is known as object-oriented design (OOD). This method enables the development of an object-based software solution. The object-oriented programming (OOP) paradigm is implemented by OOD. The system is viewed as a collection of objects (i.e., entities) according to the object-oriented design methodology. The objects share the same state, and each object is in charge of maintaining its own state information. Tasks created with a single objective cannot update or refer to data from other objects. Every object contains internal data that reflects its current state. A class is made up of related items. To put it another way, every item falls under a class. The Tetris Elites game is using object-oriented design. Using classes and objects, it arranges the various game elements.

The game window is represented by the primary class, Tetris. It keeps track of the score, level, and existing and upcoming Tetris pieces, among other game-related information. Along with handling user input, updating the game's logic, rendering the game, and managing progress, it also controls rendering. BoardPanel and SidePanel are two additional classes in the Tetris class that represent the game board and the side panel displaying the score and other information, respectively. Tetris and these classes work together to display the game window.

The Tetris class uses KeyListener to watch for keyboard events and react to user interaction. It associates particular keys with particular actions, such as moving and rotating the pieces or starting a new game. To control the game's time, the Tetris class additionally makes use of a Clock class. It regulates the game's speed and makes sure that updates occur at a regular pace. The BoardPanel class uses the TileType enum to manage the state of the game board and to represent the various Tetris component kinds. It offers ways to validate piece placements, check for finished lines, and add pieces to the board.

Overall, this object-oriented design separates different responsibilities into classes, making the code modular and easier to understand. Each class has its own specific purpose, and they work together to create Tetris Elites.

Testing and Debugging

This game runs smoothly without any bugs.

Documentation (User Manual)

Game Objective

The objective of Tetris Elites is to score as many points as possible by clearing lines of blocks. The game ends when the blocks stack up and reach the top of the game board.

Controls

- S: Drop the current block down faster.
- A: Move the current block to the left.
- D: Move the current block to the right.
- Q: Rotate the current block counter clockwise.
- E: Rotate the current block clockwise.
- P: Pause/Unpause the game.
- ENTER: Start a new game or restart the current game.

Game Screen

The game screen consists of a game board on the left side and a side panel on the right side.

Game Board

The game board is where the blocks fall and stack up. The objective is to create complete horizontal lines of blocks to clear them. As the game progresses, the blocks will fall faster.

Side Panel

The side panel displays the following information:

- Level: Shows the current level of the game.
- Score: Displays the player's current score.
- Next Piece: Shows the shape of the next block to appear.

Gameplay

- When the game starts, the current block will appear at the top center of the game board, and the next block preview will be shown in the side panel.
- Use WASD keys to move the block left or right, and use the Q and E keys to rotate the block.
- The block will automatically descend one row at a time. You can press the S key to make it drop faster.
- Try to position the blocks to create complete horizontal lines. When a line is complete, it will be cleared, and you will earn points.
- As you clear more lines, the game level will increase, and the blocks will fall faster, making the game more challenging.
- If the blocks stack up to the top of the game board, the game ends. You can start a new game by pressing the ENTER key.

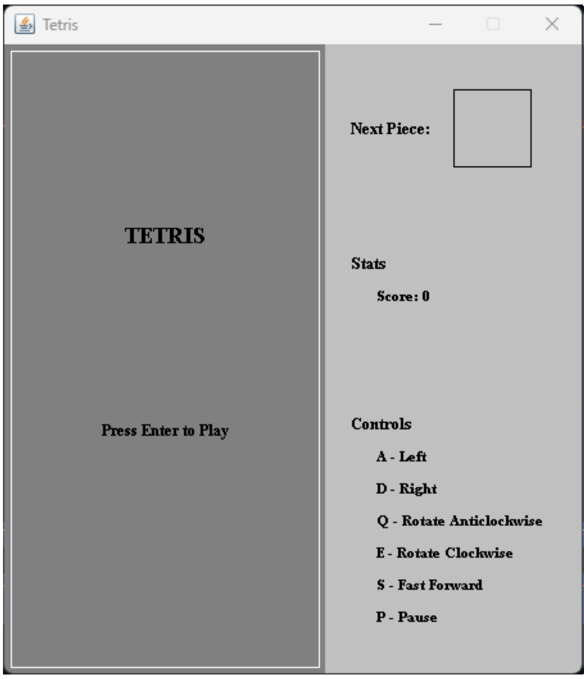
Scoring

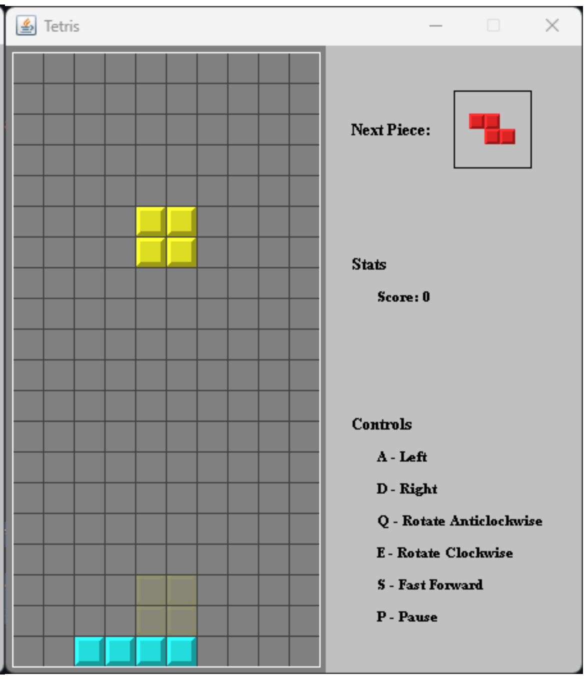
For every line cleared, you get 100 points.

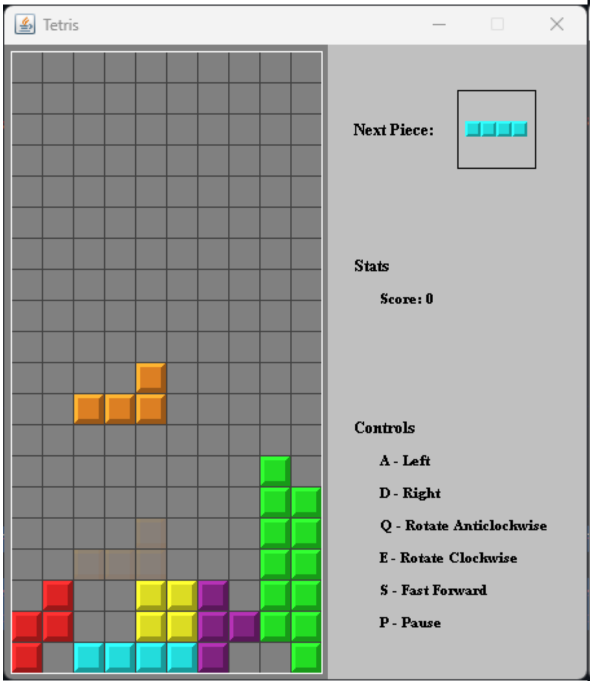
Pause and Game Over

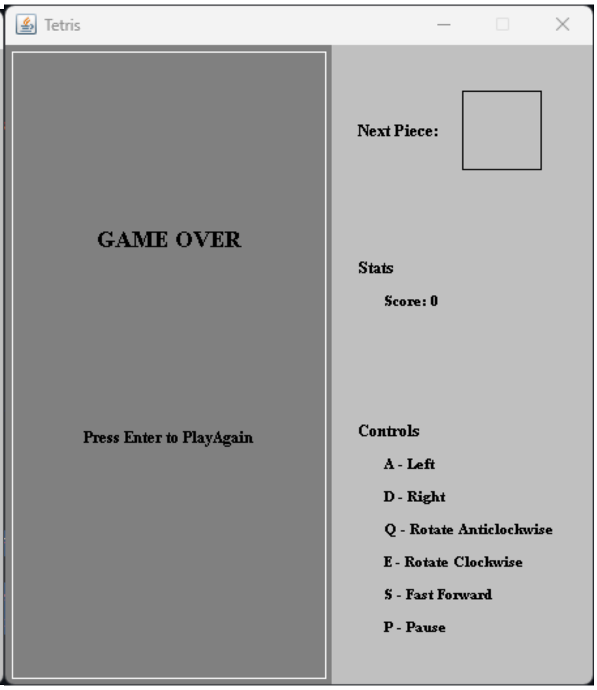
- Press the P key to pause or unpause the game. While paused, the game timer will stop, and you can resume playing by pressing P again.
- If the blocks stack up to the top of the game board, it's game over. You can start a new game by pressing the ENTER key.

<u>Game Pictures</u>

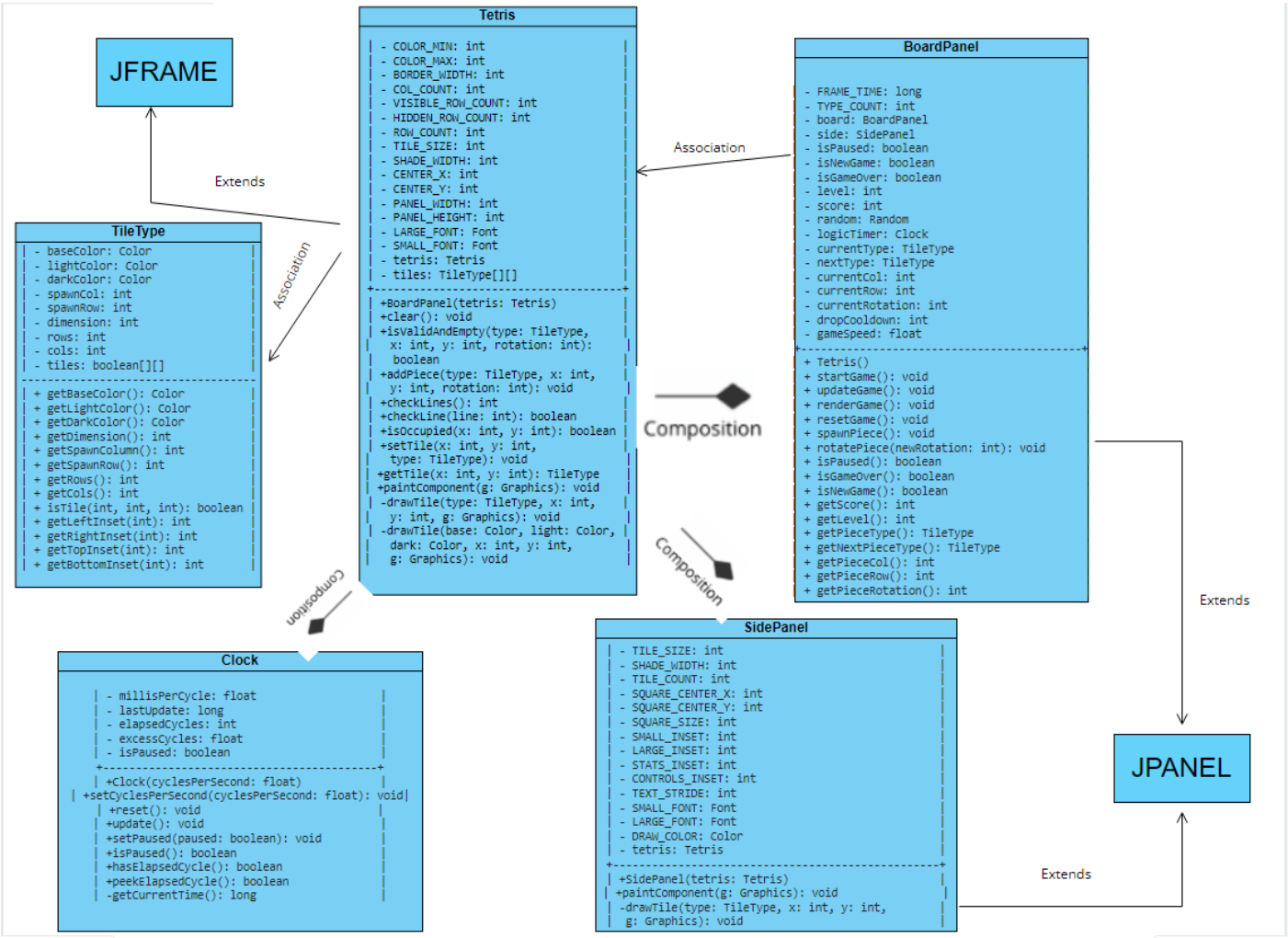








SOLUTION DIAGRAM (CLASS DIAGRAM)



- Tetris class: This class extends JFrame and represents the main game window. It contains the game logic, user input handling, and manages other components like BoardPanel and SidePanel. It includes methods for starting the game, updating the game state, rendering the game, resetting the game, and handling piece rotation. It also provides getters for various game-related information such as score, level, current piece, etc.
- BoardPanel class: This class represents the game board where the Tetris pieces are placed. It is responsible for storing and managing the state of the board, checking for line clears, and rendering the board on the screen. It is used by the Tetris class to add pieces, check their validity, and perform operations like line clearing.
- SidePanel class: This class represents the side panel of the game window. It displays information about the game, such as the next piece, score, and level. It is responsible for rendering this information on the screen.
- Clock class: This class represents a timer used for game logic. It keeps track of the elapsed time and allows setting the speed of the game by specifying cycles per second.

- TileType: This represents the different types of Tetris pieces. It includes information about each piece's shape, spawn position, and rotation. It is used by the Tetris class to handle piece manipulation and spawning.

HOW THE PROGRAM WORKS

The program Tetris Elites is a simplified implementation of the classic game Tetris. These are the necessary components used by the program to work together:

Class Structure: The program consists of multiple classes, but the main class is `Tetris`, which extends `JFrame` to create the game window. It also contains the game logic and handles user input.

Window Setup: The `Tetris` constructor initializes the game window by setting its properties, such as the title, layout, and resizable option. It creates instances of `BoardPanel` and `SidePanel`, which represent the game board and side information panel, respectively. These panels are added to the window using a `BorderLayout`.

User Input: The program uses a `KeyListener` to handle user input. The `KeyAdapter` within the `Tetris` class listens for key events and performs specific actions based on the pressed key. For example, pressing the "A" key moves the current piece to the left, and pressing the "S" key increases the drop speed.

Game Loop: The `startGame` method contains the game loop, which runs continuously until the program is closed. Within each iteration of the loop, several actions are performed:

- The logic timer is updated to control the game's speed. The game logic is only updated if a certain cycle has elapsed.
- The current piece is moved down one row if possible. If it cannot move down further, it is added to the board.
- If one or more lines are cleared as a result of adding the piece, the player's score is increased.
- The game speed is adjusted, and the drop cooldown is set.
- The game state is rendered, meaning that the panels representing the game board and side information are repainted.

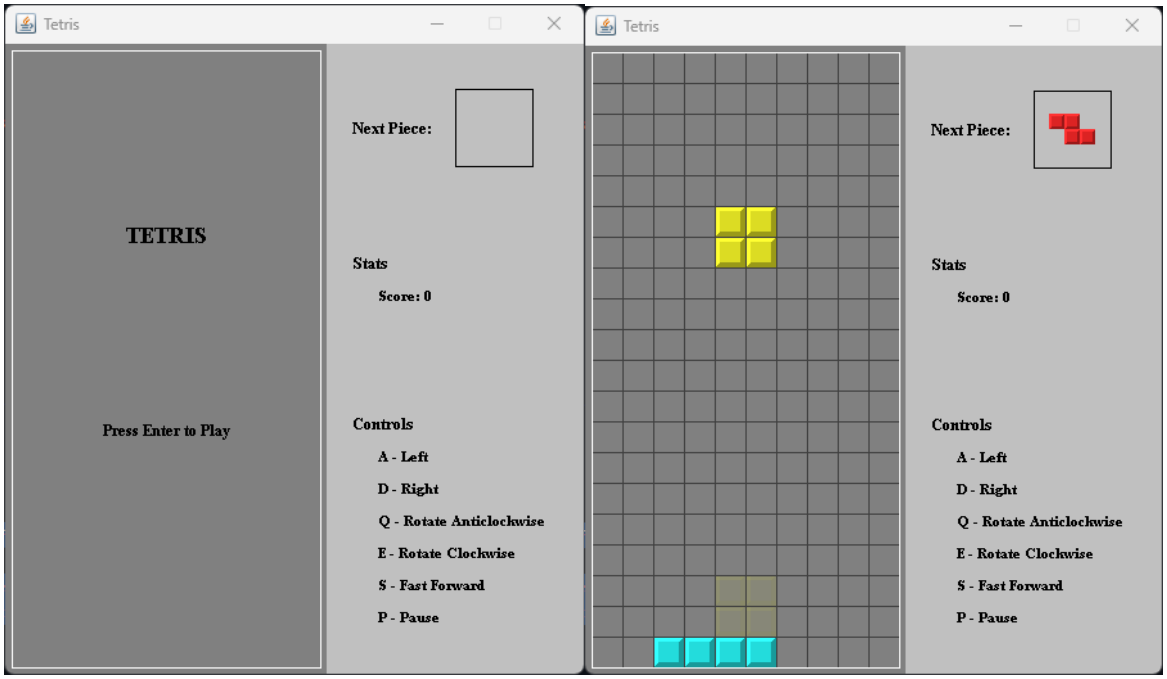
Piece Spawning and Rotation: The `spawnPiece` method is responsible for spawning a new piece onto the game board. It selects the next piece randomly and resets the position and rotation variables. If the spawn point is invalid (i.e., the piece immediately collides with existing blocks), the game is marked as over. The `rotatePiece` method handles piece rotation. It checks if the new rotation is valid and adjusts the piece's position if necessary to prevent it from going outside the game board.

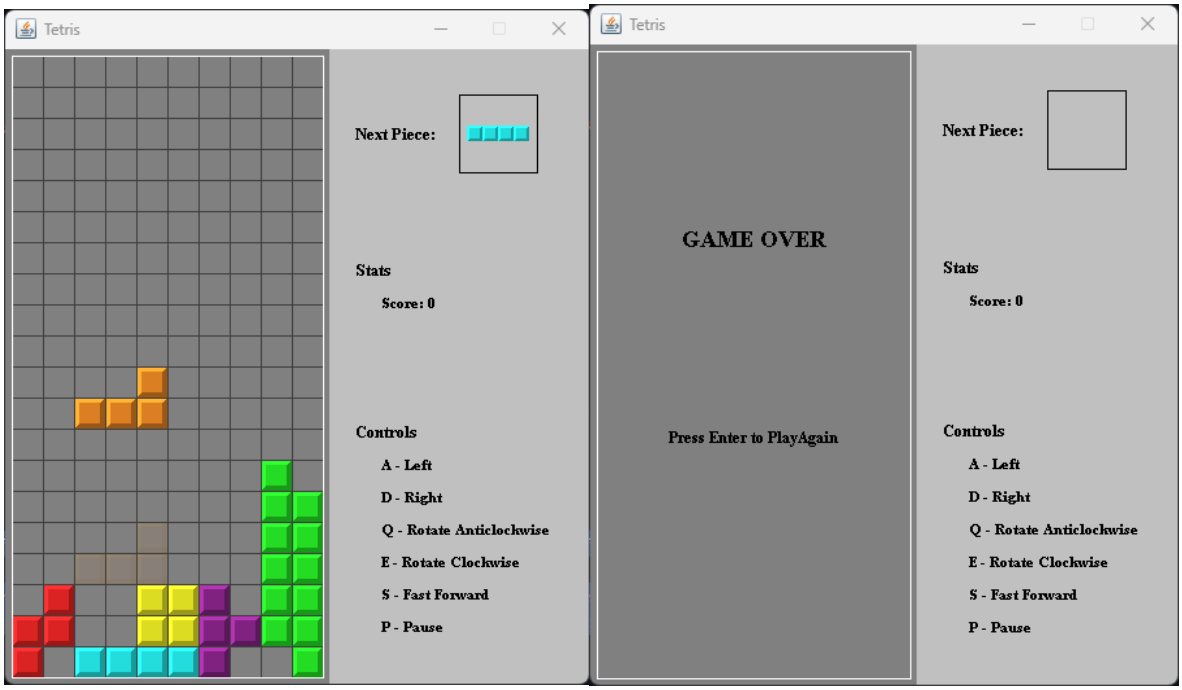
Game Over and New Game: The `resetGame` method is called when the game is over or a new game is started. It resets various variables, such as the score, game speed, and current and next piece types. The game board is cleared, and the logic timer is reset and paused. Finally, a new piece is spawned.

Other Helper Methods: The program also includes additional methods to check the game's state (e.g., paused, game over, new game), retrieve information (e.g., score, level, current and next piece types, position, rotation), and handle rendering.

Overall, the program follows a basic game loop structure, continuously updating the game state, processing user input, and producing the game. It makes use of many classes and methods to control the game logic, piece movement and rotation, and user interaction.

SCREENSHOTS OF THE GAME





GITHUB REPO LINK

Attached is the link for the Tetris Elites github repo:

https://github.com/jeffmatthew/Tetris-Elites

REFERENCES

GeeksforGeeks. (2023a). Introduction to Java Swing. *GeeksforGeeks*. https://www.geeksforgeeks.org/introduction-to-java-swing/

Kumar, S. (2022, December 16). Object-oriented Design (OOD) - Scaler Topics. *Scaler Topics*. https://www.scaler.com/topics/software-engineering/object-oriented-design/

Prisco, J. (2019, November 1). Tetris: The Soviet 'mind game' that took over the world. *CNN*. https://edition.cnn.com/style/article/tetris-video-game-history/index.html

Techopedia. (2011b, October 11). *What is Java Swing? - Definition from Techopedia*. https://www.techopedia.com/definition/26102/java-swing