



COMP 204

Programming Studio

Project - 2

TETRIS 2048 REPORT

Instructor: Prof. Dr. Muhittin GÖKMEN

BERK ŞEN - 041905011

İDİL MERGAN - 042101007

NESİBE AYŞE ALATAŞ - 042101129

NİL DİNÇER - 042201171

Abstract

Tetris and 2048 are both popular puzzle video games. Tetris is a classic tile-matching game where the player manipulates falling blocks called Tetrominos to create horizontal lines without gaps. 2048 is a relatively newer puzzle game. The player must slide numbered tiles around to combine them and create a tile with the number 2048. Hence, we combined these two games called “Tetris 2048” in Java using the stdDraw library. This report aims to demonstrate the methods utilized to create the game, explain them in detail, and Show the accomplishments that this Project has provided.

Table of Contents

1. Description of Project

2. Description of Solution

2.1 UML Diagram

2.2 Methods

2.2.1 Tetris2048.java

2.2.2 Gamegrid.java

2.2.3 Tetromino.java

2.2.4 Tile.java

2.2.5 Point.java

3. Examples of Outputs

4. Achievements

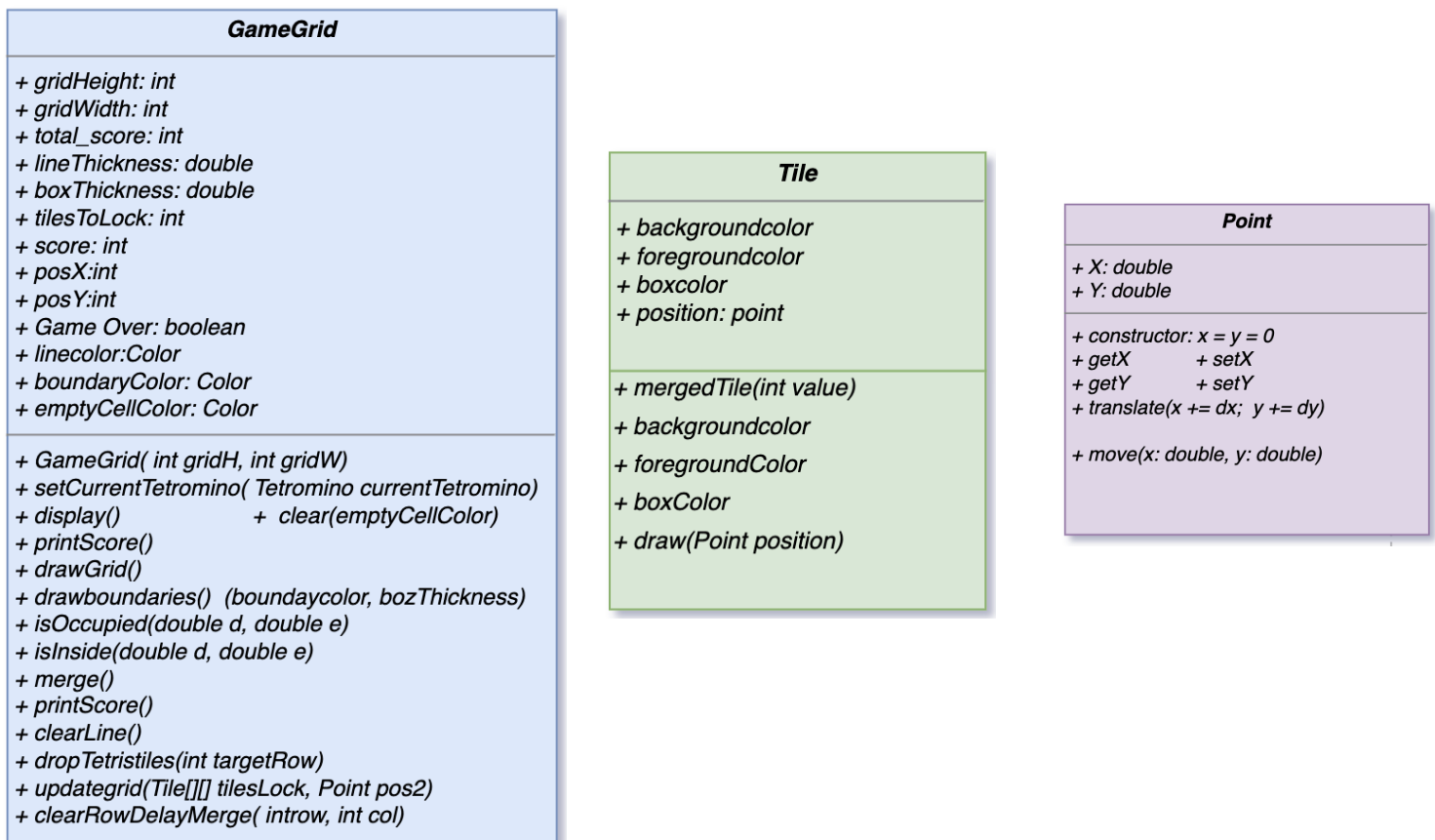
5. References

1. DESCRIPTION OF THE PROJECT

This Project aims to combine two games which are Tetris and 2048 to create a new game called “Tetris 2048”. Tetris was first released in 1984 by Alexey Pajitnov, Dimitri Pavlovsky, and Vadim Gerasimov, and has since become one of the most recognizable and beloved video games of all time. And 2048 was first released in 2014 by Gabriele Cirulli. 2048 is a challenging and addictive game but it is similar to Tetris in that the player can move and rotate 7 different shapes on the 2D playing surface. The distinction is that tetrominoes have numbers on their tiles and the number on each tile determines their color. Moreover, each approaching tetromino is made up of four tiles numbered 2 or 4 at random.

2. DESCRIPTION OF THE SOLUTION

2.1 UML Class Diagram:



<i>Tetris2048</i>
<i>+ setCanvasSize</i> <i>+ setCurrentTetromino</i> <i>+ currentTetromino = createTetromino</i> <i>+ displayGameMenu(int gridHeight,</i> <i>int gridWeight)</i>
<i>StdDraw()</i> <i>+ isKeyPressed(Left, Right, Down, z)</i> <i>+ show()</i> <i>+ pause ()</i> <i>+ isMousePressed ()</i>

<i>Tetromino</i>
<i>+ gridWidth</i> <i>+ gridHeight</i> <i>+ occupiedTiles (I,O, Z, S, T, L,J)</i> <i>+ getCellPosition(int row, int col)</i> <i>+ createMinBoundedTileMatrix ()</i> <i>+ draw()</i> <i>+ move(String direction, GameGrid gameGrid)</i> <i>+ method(): Type</i>
<i>+ canBeMoved(String dir, GameGrid gameGrid)</i> <i>+ rotatetetromino (GameGrid gameGrid, boolean direction)</i> <i>+ canBerotated(GameGrid gameGrid)</i>

2.2 Modules and Methods:

2.2.1 Tetris2048.java

Tetris 2048 is the main class to run the Tetris 2048 game.

- **restartGame(int gridH, int gridW):** A method to restart the game when it's over by pressing R (It is not working).
- **createTetromino():** A method to create a random shaped tetromino to enter the game grid.
- **displayGameMenu(int gridH, int gridW):** A method to display a simple menu before starting the game.

2.2.2 GameGrid.java

GameGrid is a class for modeling the game grid.

- **setCurrentTetromino(Tetromino currentTetromino):** A setter method for the currentTetromino data field.
- **display():** A method to display the game grid.
- **drawGrid():** A method to draw the cells and the lines of the game grid.

- **drawBoundaries():** A method to draw the boundaries around the game grid.
- **isOccupied(double d, double e):** A method to check whether the grid cell with given row and column indexes is occupied by a tile or empty.
- **isInside(double d, double e):** A method to check whether the cell with given row and column indexes is inside the game grid or not.
- **merge():** A method to merge the tiles with the same values as in 2048.
- **printScore():** A method to print the score on the game screen.
- **printNextTetromino(Tetromino tet):** A method to draw the next tetromino on the blank side of the game screen.(It is not working).
- **clearLine():** A method to clear full lines as in Tetris.
- **dropTetrisTiles(int targetRow):** A method to drop the hanging tiles after clearing a row as in Tetris.
- **updateGrid(Tile[][] tilesToLock, Point pos):** A method to lock the tiles of the landed tetromino on the game grid while checking if the game is over or not.
- **clearRowDelayMerge(int row, int col):** A method to delay the merge event when a full line is cleared.

2.2.3

Tetromino.java

Tetromino is a class for modeling tetrominoes with 7 different shape as follows: I, O, Z, S, T, L and J.

- **getCellPosition(int row, int col):** A getter method to get the position of the cell in the tile matrix with given row and column indexes.
- **createMindBoundedTileMatrix():** A method to create a copy of tile matrix omitting empty rows and columns.
- **getMindBoundedTileMatrix():** A getter method to get the minimum bounded tile matrix.
- **getMindBoundedTileMatrixPosition():** A getter method to get the position of the minimum bounded tile matrix.
- **draw():** A method to draw the tetromino on the game grid.

- **move(String direction, Gamegrid gamegrid):** A method to move the tetromino in a given direction by 1 on the game grid.
- **canBeMoved(String direction, Gamegrid gamegrid):** A method to check if the tetromino can be moved in the given direction or not.
- **rotateTetromino(Gamegrid gamegrid, boolean direction):** A method to rotate the tetromino in the given direction.
- **canBeRotated(Gamegrid gamegrid):** A method to check if the tetromino can be rotated or not.

2.2.4 Tile.java

Tile is a class for modeling numbered tiles as in 2048.

- **mergedTile(int value):** A method to get the properties such as number(value), background, foreground and box color values of tiles from 4 to 8192.
- **draw(Point position):** A method for drawing the tile.

2.2.5 Point.java

Point is a class for modeling points as a location in 2D space.

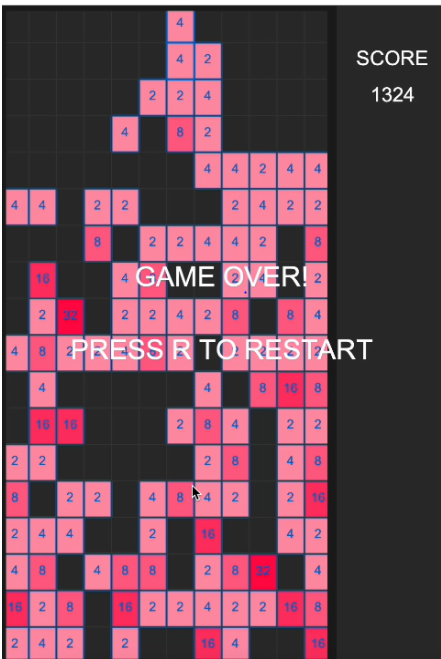
- **getX():** Getter method for the x coordinate of a point.
- **getY():** Getter method for the y coordinate of a point.
- **setX(double x):** Setter method for the x coordinate of a point.
- **setY(double y):** Setter method for the y coordinate of a point.
- **translate(double dx, double dy):** A method to move a point by dx along x axis and by dy along the y axis.
- **move(double x, double y):** A method to move a point to a given x,y coordinate.

3. EXAMPLES OF OUTPUTS

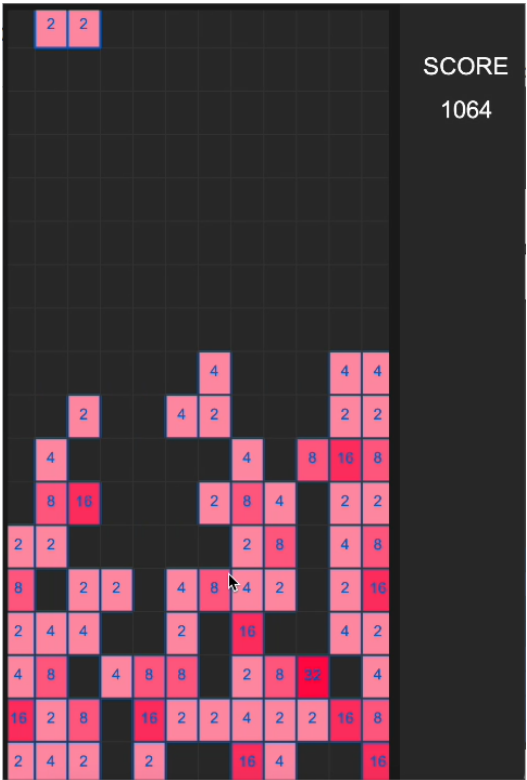
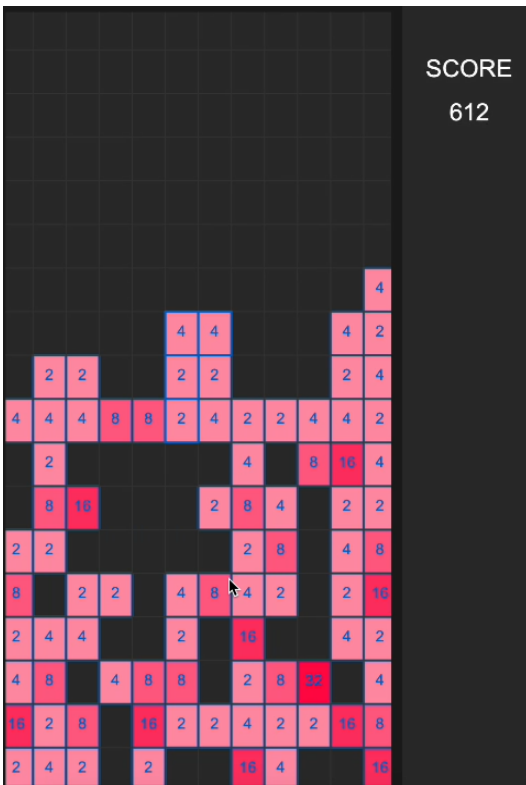
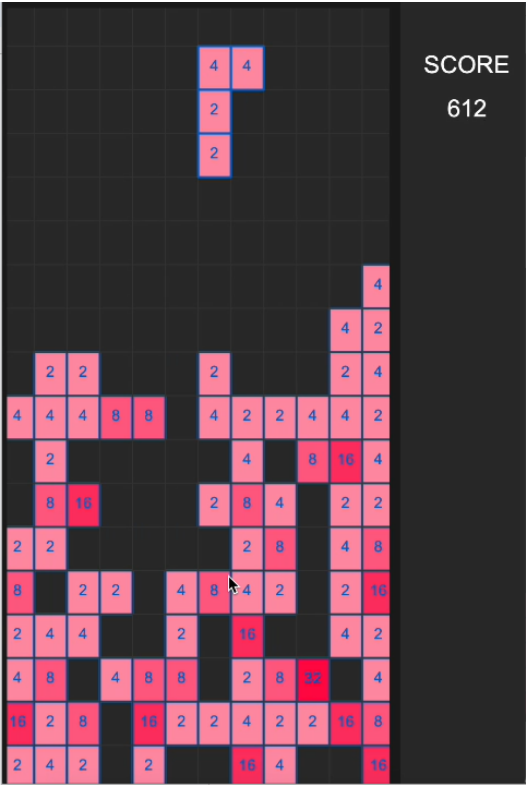
a. Game Menu



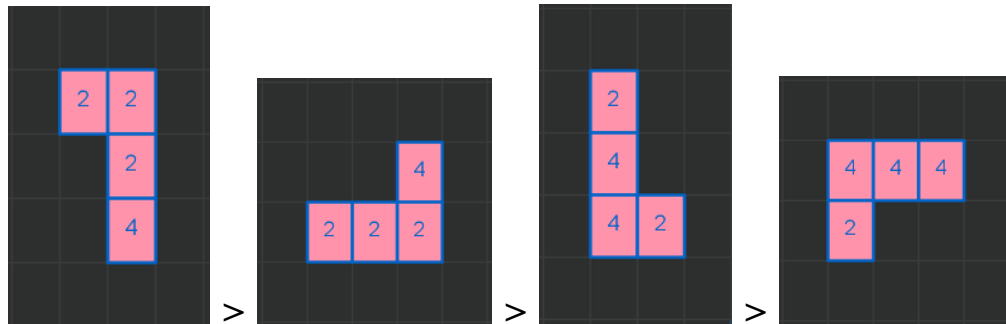
b. Game Over



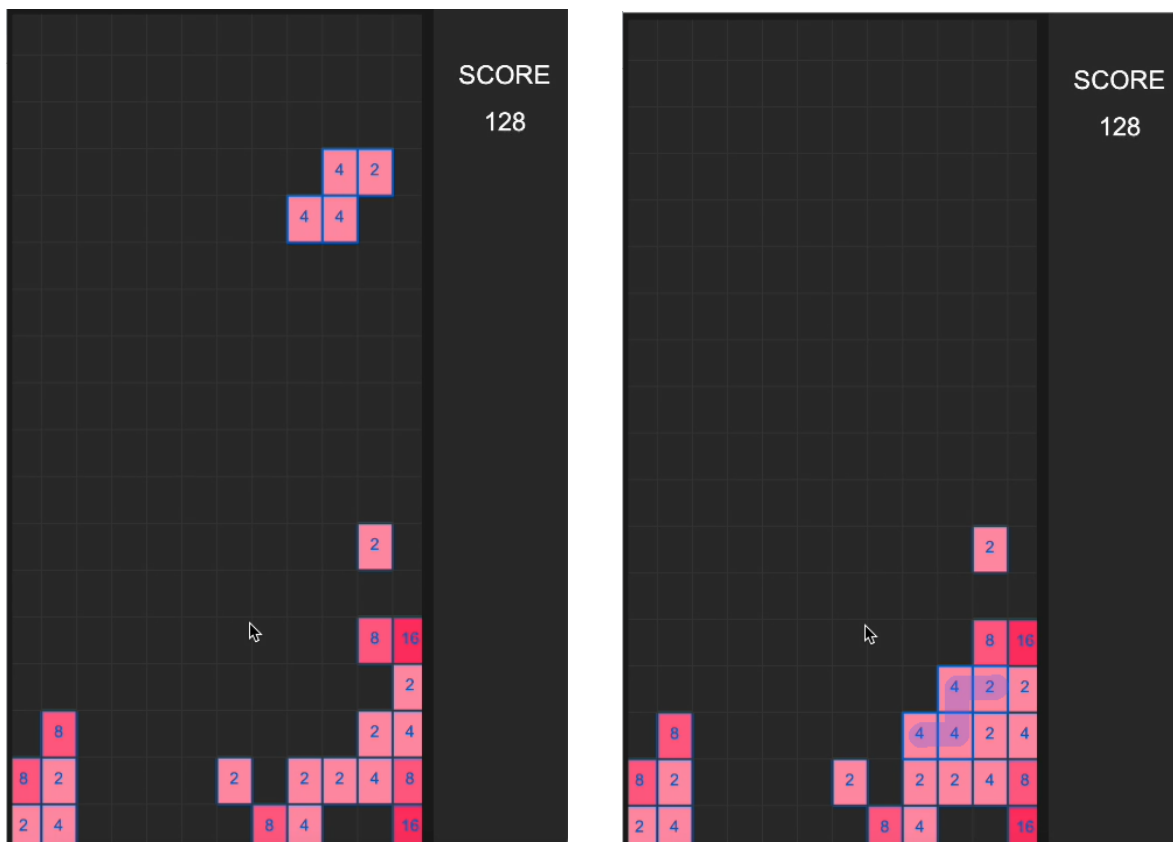
c. Clear Full Row

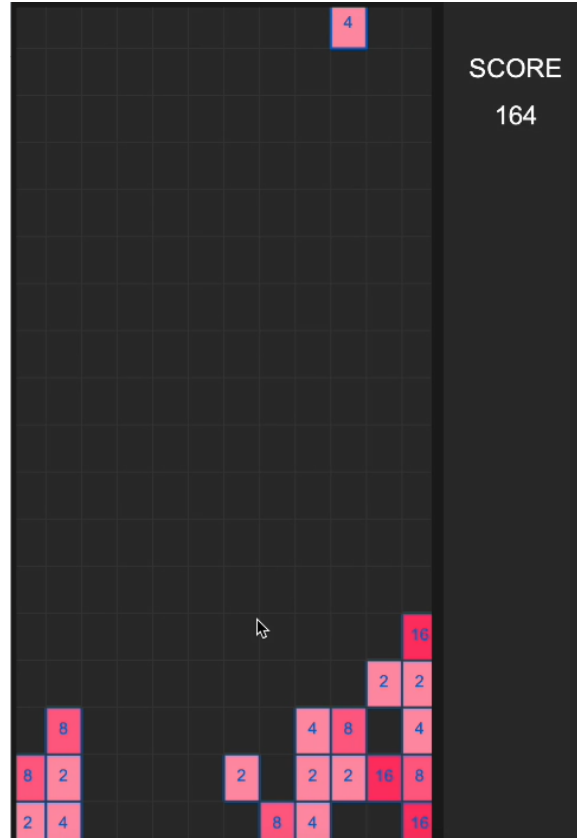
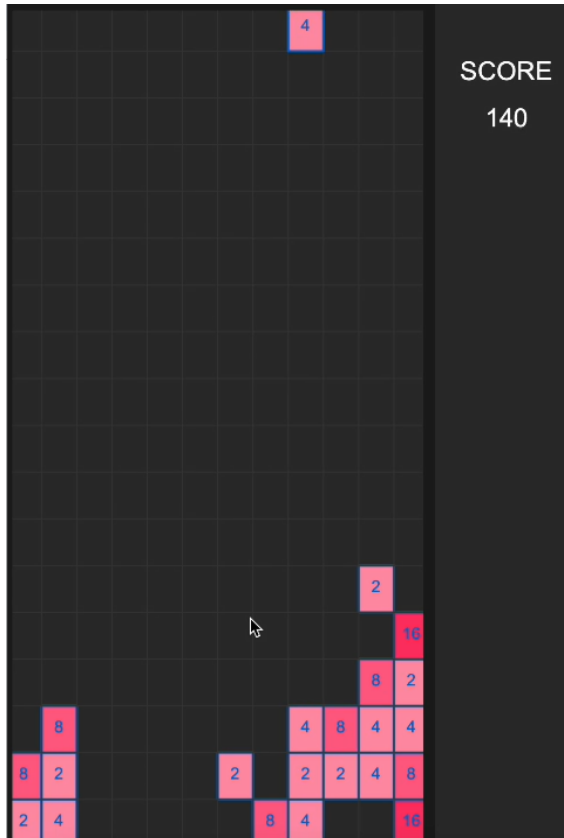


d. Rotation of a Tetromino



e. Merge





4. ACHIEVEMENTS

In this project, he used the principles of Object-Oriented Programming to create modular and easy-to-maintain code, and the StdDraw library was used. By integrating the 2048 tile joining mechanism with the Tetris game, an easy-to-interface gaming experience was offered. A button and a banner were created that easily launches the game for the user. Thanks to this, a user-friendly interface was provided that allows players to navigate. Successfully implemented the Tetris game logic, including the movement of blocks, rotation and scoring. Collaboration was carried out with team members to divide tasks and effectively manage the project timeline. We have created a Java project that demonstrates creativity and the performance of technical skills in game development.

5. REFERENCES

Gökmen, M. (2023). Project 2: Tetris 2048. Lecture

Stackoverflow. (2018). *java - Tetris full row only half-removed*. Stack Overflow.

<https://stackoverflow.com/questions/49737524/tetris-full-row-only-half-removed>

Stackoverflow. (2020, November). *Tetris Piece Rotation Algorithm*. Stack Overflow.

<https://stackoverflow.com/questions/233850/tetris-piece-rotation-algorithm>

tutorialandexample. (2022, December 19). *Tetris Game in Java - TAE*.

Www.tutorialandexample.com. <https://www.tutorialandexample.com/tetris-game-in-java>