12.04.2020

MEF University

Computer Engineering

Programming Studio

Project 2 : Tetris 2048

Team Members :

Melis Akarçay

Arsen Güler

Baran

**Abstract:**

In the project, Tetris and 2048 games were tried to be combined. In this process, a game was created using Tetris's shapes and the 2048's numbers.

**Problem Definition:**

Tetris 2048 is an application consisting of the combination of tetris and 2048 games. Tetris is a game that was popular in the 1990s. The main purpose of Tetris is to place the falling blocks in a horizontal row with no gaps. Some shapes consisting of blocks are used in Tetris. These shape’s names are Z, T, I, J, L, S and O. Each shape consists 4 blocks. As these shapes fall, the user can direct them to the right or left. If appropriate, the shape can be used by applying the rotation function. 2048 is a game that consists of blocks, but it is a game where the number on the blocks is important, not the shape of the blocks. When the blocks containing the same number overlap in the game, these numbers are added and written to the block below. To combine Tetris and 2048, a random number is written into each block of Tetris shapes. When the figures go down, if there is another block below, it looks to see if it contains the same number as the block it is on. If the neighboring blocks have the same number on them, those numbers are added up and written to the block below.

**Problem Solution:**

When creating a game, first of all, it is necessary to plan the interface that the user will access. For this project, StdDraw library was used to create a user interface. A canvas consisting of 12 rows and 8 columns was created. That page shows the game screen, score and information about the upcoming tetromino.

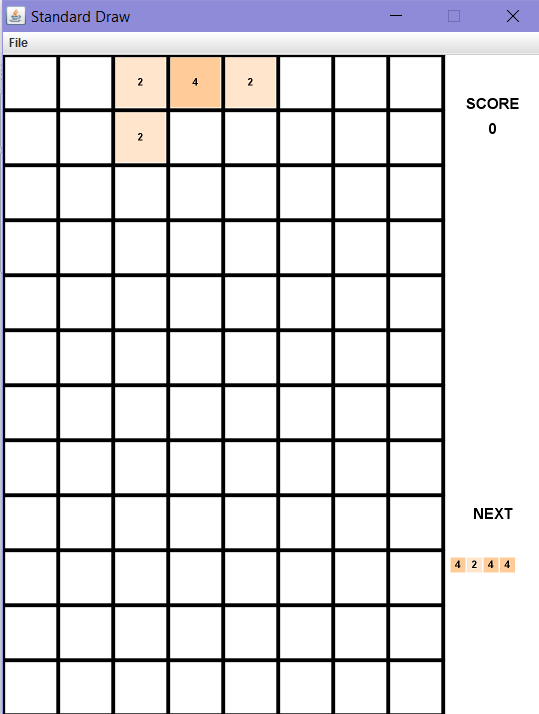


fig 1. Tetris interface

In order to perform Tetris implementation, the block class has been added first. This block class contains the coordinate of each block. Tetrominos are written using these blocks. Each tetromino consists of 4 blocks. The created blocks are stored in an arraylist. These blocks must be given a number value because it combines the Tetris game with 2048. For this, the value variable is added to the tBlock class. (Tetromino example is shown in figure 2.)

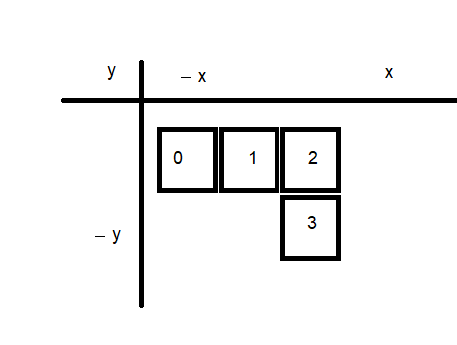


fig 2. Tetromino L

Once the tetromino classes are defined, a function is written that will randomly call these shapes. 7 numbers are transferred to a list in the Tetromino class. Then a random number is selected from this list and the figure corresponding to the number is created. Once the shapes are created, the basic action they need to do is move downward unless otherwise. To ensure the downward movement, the location information of the blocks is changed and they reach the bottom line. Checks if there are any other blocks at the bottom using the flag. Three other important functions are moveRight, rotateTet and moveLeft. To be able to move to the right, the moveRight function must first know the rightmost values. Therefore, this location is determined using the findRightward function. The function checks if the tetromino collides with another shape and goes to the far right end. Using them, the range suitable for movement has been found and the shape cannot be out of this range. RotateTet function is for rotating a shape to right. To make the rotation, the locations of the blocks were changed and redefined for each shape. If the user presses down once, the program determines new locations and changes the isRotated boolean value to true. Thus, the controls are provided more easily when the button is pressed a second time.

Inputs from the user are required to use the move and rotation functions. I/O operations and the main stream in the game are defined in the Main class. In the main class there are three main functions. First one is drawGameEnvironment. It creates the design of the canvas. The second function is deleteRows. deleteRows performs the line deletion, which is the purpose of tetris. To do this, the function navigates all the blocks and checks whether the blocks are full. When checking, it uses a counter and when it has a value of 8, it realizes that a row is full. When counter is 8, this value is checked with a number flag. As a result, the filled row is deleted and the next rows are printed below. Last function is mergeBlocks. mergeBlock function sums the value of overlapping blocks containing the same number. Location information is also used to control the blocks [fig 4].

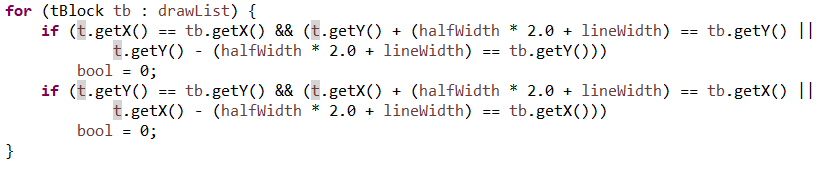


fig 4. Connected block control

The expected change in the blocks after the controls are provided as in Figure 5.

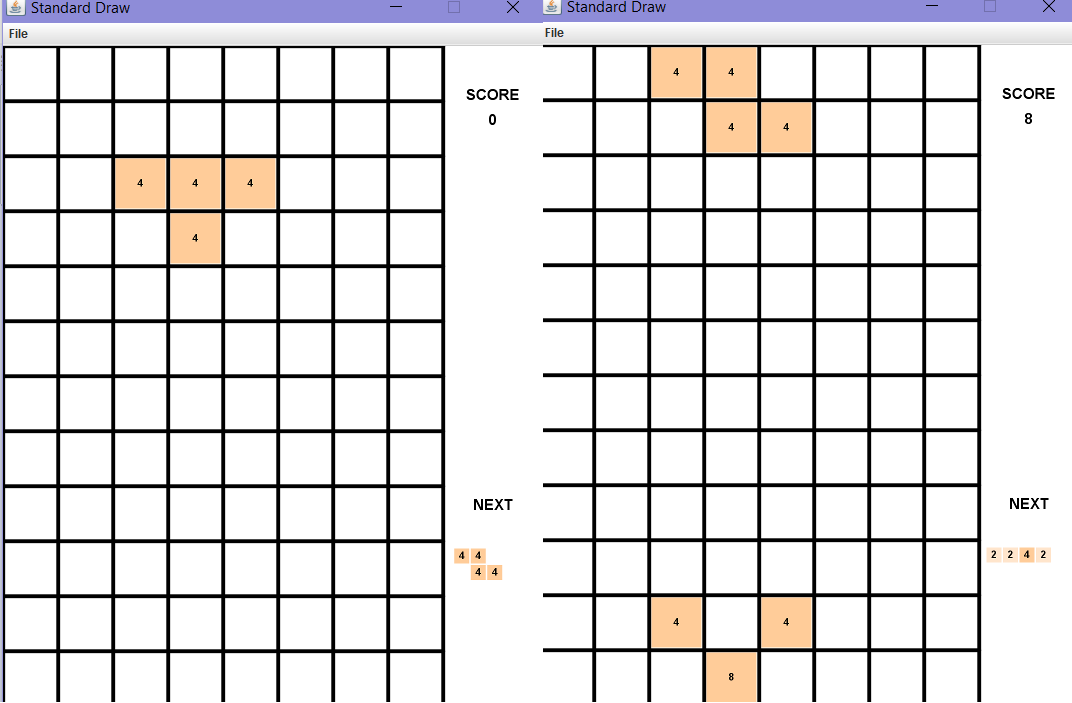


fig 5. Merging the blocks

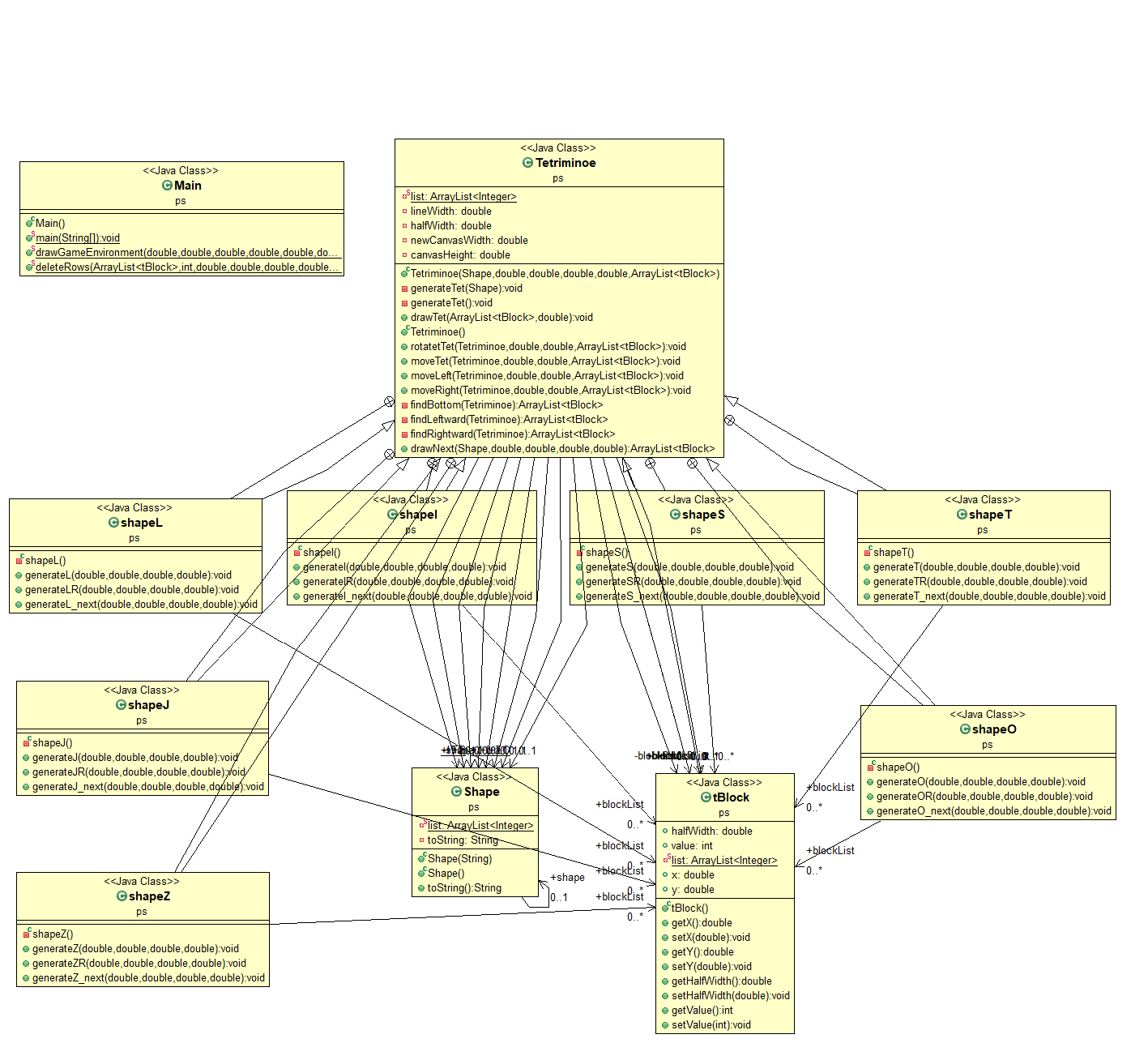


fig.6 UML Diagram

**List of Achivements:**

During the project, we first learned about the StdDraw library. Since we use three classes tBlock, Shape and Tetromino, we have gained a lot of experience with class relationships and their correct use in functions.

**References:**

[1] Bodnar, J. (n.d.). Java Tetris. Retrieved from <http://zetcode.com/tutorials/javagamestutorial/tetris/>

[2] Introcs.cs.princeton.edu. n.d. *Stddraw*. [online] Available at: https://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html

[3] <https://github.com/arsenguler/Tetris2048>