**OOP REPORT**

Group Member:

Nguyễn Hoàng Đạt – ITITIU19097

Trần Hải Nam – ITITIU19161

Đặng Đức Luân – ITITIU19157

Nguyễn Duy Lập – ITITIU19153

Game: Tetris

**Some things about Tetris:**

Tetris is a classic type of game with gameplay similar to a puzzle game Include blocks of bricks with many different shapes your task is to control the bricks here to fill a line to score points. At the same time if that block fills to the top of the screen, then game over

1. **Rule and Objective:**

* The rule of the game is very simple. Play it until you lose, each time you earn a new high score, it will be saved into your device, you can earn 100 points for a single road clear and 1 point for successfully placing a block. The game will get harder depending on the number of points of your current run.
* The objective is to challenge yourself and get the as many points as possible.

The algorithm map of TetrisDiagram

Description automatically generated

Diagram, box and whisker chart

Description automatically generated

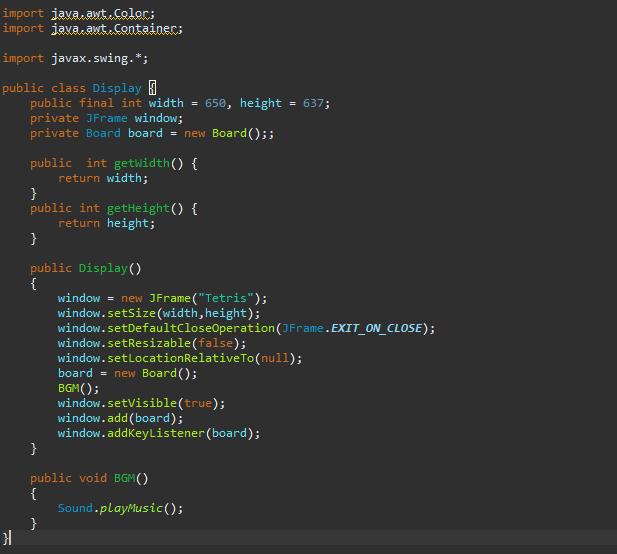
**Class Diagram**

Our game was made entirely using the Swing library from JAVA.

Our game consists of 4 classes: Display, Board, Shape, ReadHighScore, Sound and one interface KeyListener.

1. **DISPLAY CLASS (The foundation of the project)**

* The Display class is used to create GUI and show the game to the user.
* The Display class is also used to play background music for the game.



**2.BOARD CLASS**

* The Board class acts as a panel to store the game.
* The Board is used to update the game

**2.1 Functions in Board class**

**a. Time loop (Tick generator)**

- Because Tetris is a small game, we create a time loop using timer because of its simplicity.

* Create a time loop so that the game can run on it to start counting continuously until losing in order to stop the game and let the game run.
* The delay time is equal FPS /1000 ( FPS = 60).
* Putting update and repaint methods inside an actionPerformed mean that the board can be updated and repainted with the ticker

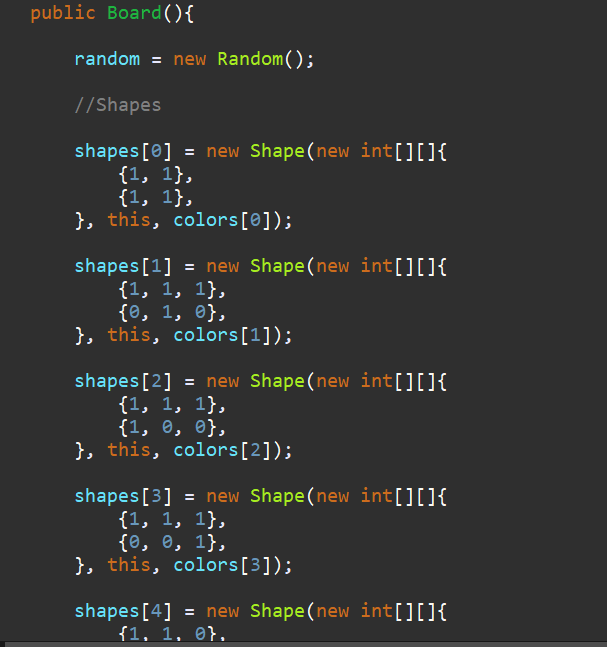


* Update method is also used to check vertical, horizontal collision and collision between shapes.

**b. Shape of blocks.**

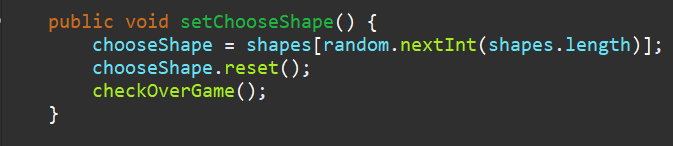
* Tetris shapes is one of the most important component of the game.
* Each block is represented by their coordinates. 1 = fill, 0 =null

For example, {1,1,1} is the representation of the T shape.  
 {0,1,0}

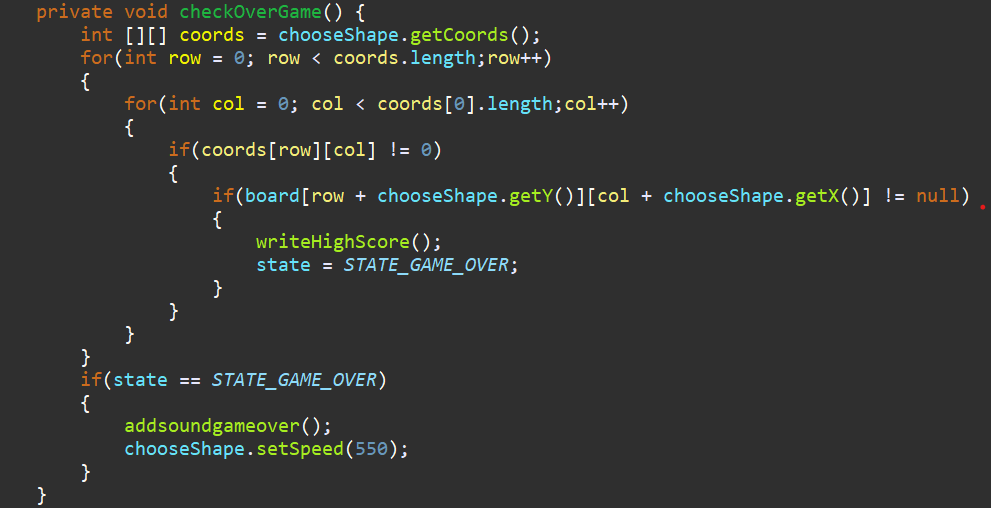


**c. Choose shape**

* This method help randomize and reset the shape

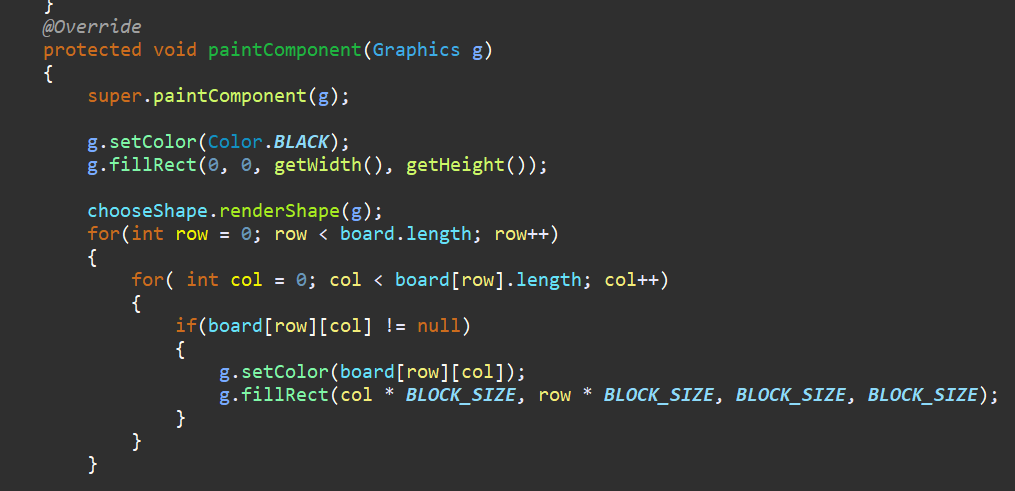


* Called the checkOverGame method to end the game and write the high score (if have) into the text file and also set the shape speed back to default, in this case is 550 milisecond.



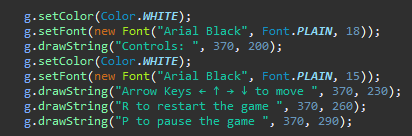
**e. paintComponent and Graphics.**

* Draw everything including the scoreboard, borders, texts, shapes and numbers to the board



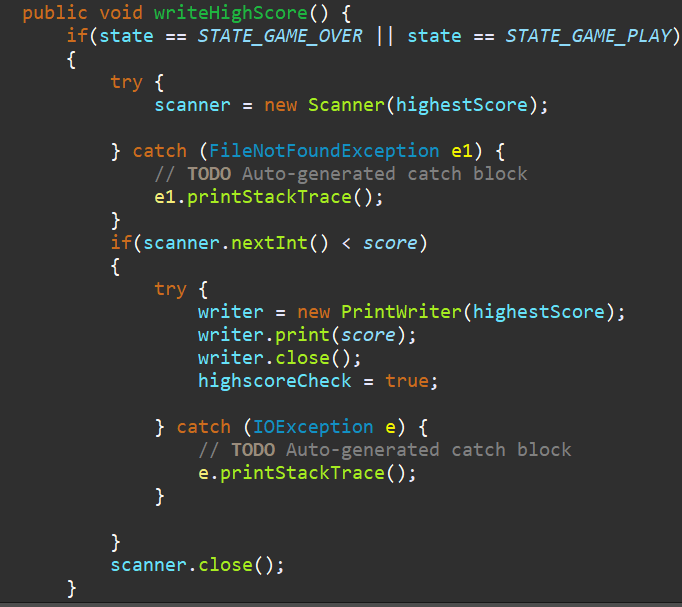




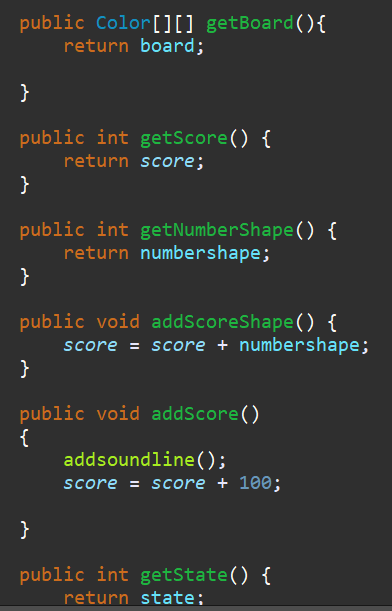


**f. Score save and calculate**

* Check whether the score saved in the text file is larger than the current score.

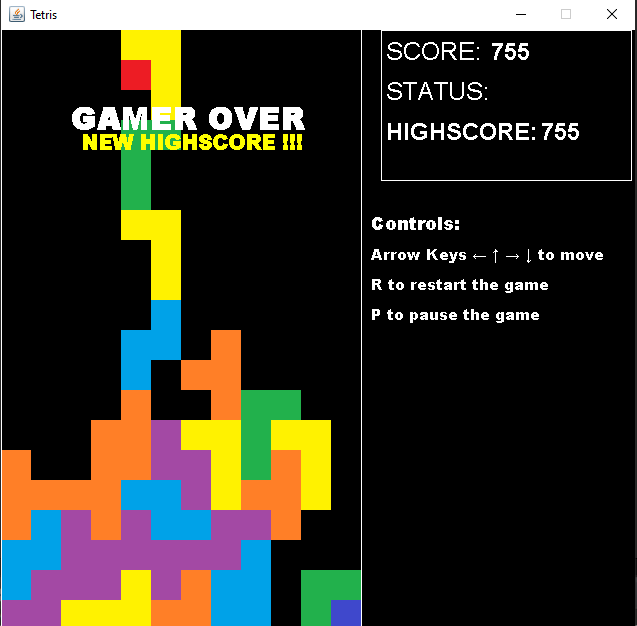


* Set value for different type of score. Score for successfully placing a place is 1 (addScoreShape()) and score for getting a full row is 100 (addScore()).



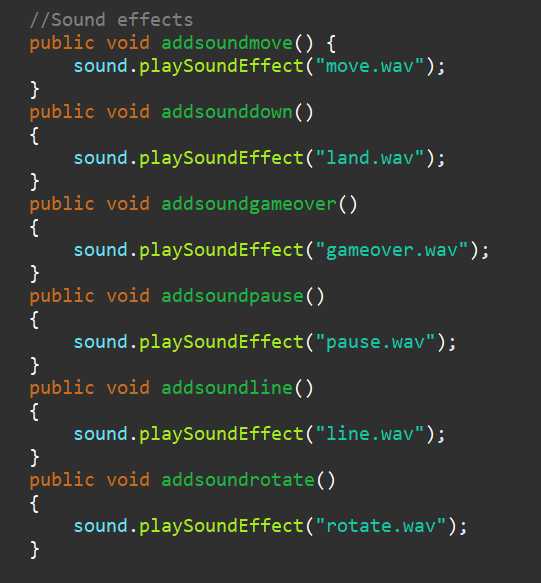


* writeHighScore method will check if the current score is higher than the highscore. If score > highscore, it will set the highscore = score and set the highscoreCheck variable to true -> activate the highscore announcement



1. **Sound effects**

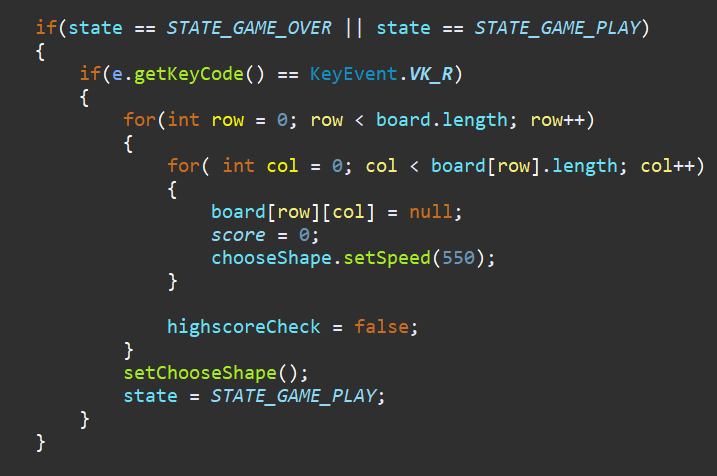
* Play different type of sound for specific action.

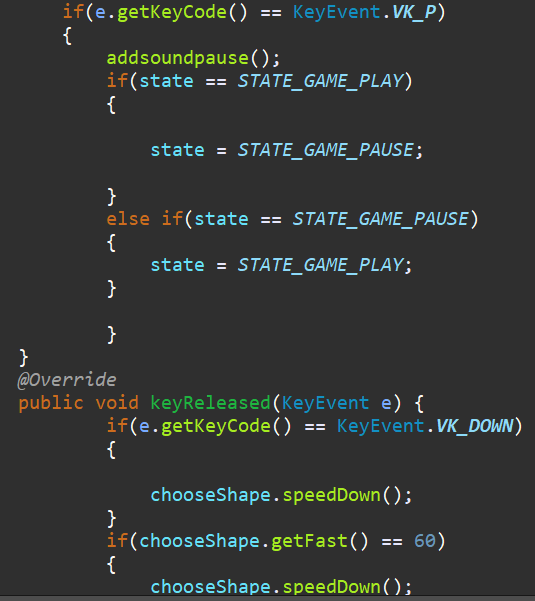


1. **Controls**

* Override medthods from the KeyListener interface that help manipulating the delay variable and xMove variable to help block drop down faster, move left and right, pause and restart the game through keyboard inputs.

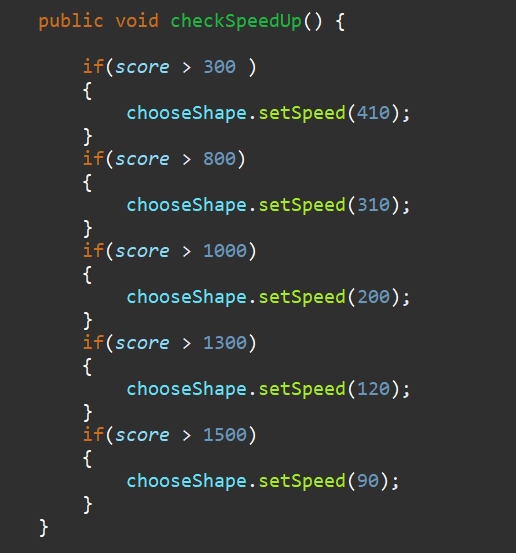






1. **Speed Up**

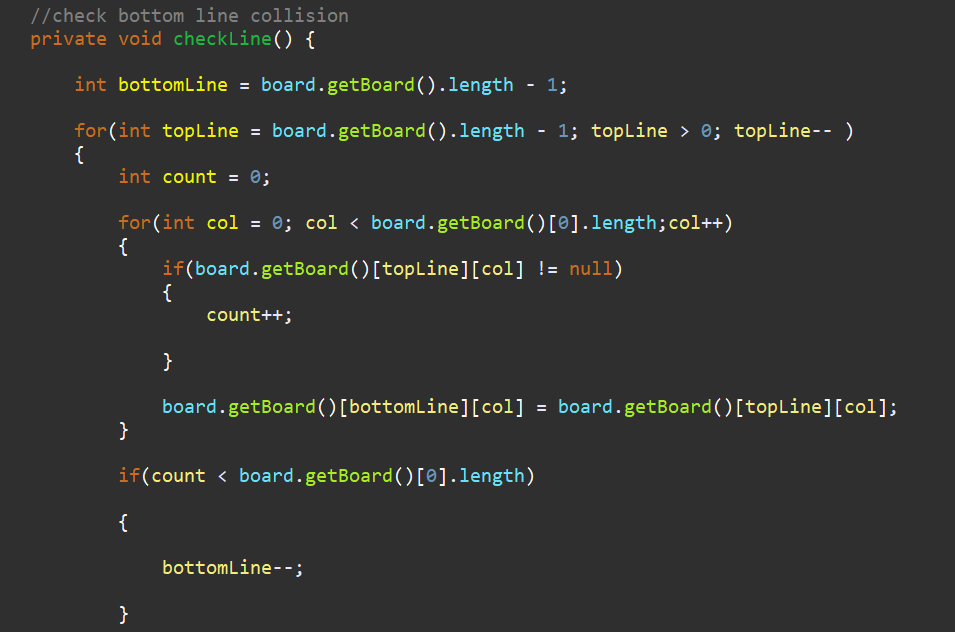
* Make the block fall faster after a certain score achievement by lowering the shape delay time.



**3. SHAPE CLASS**

* Shape class is use to check collision between shape and the x, and y axis border, and collision between other shapes.
* Shape is also use to calculate value of shapes for rotation, score, and move left and right value.

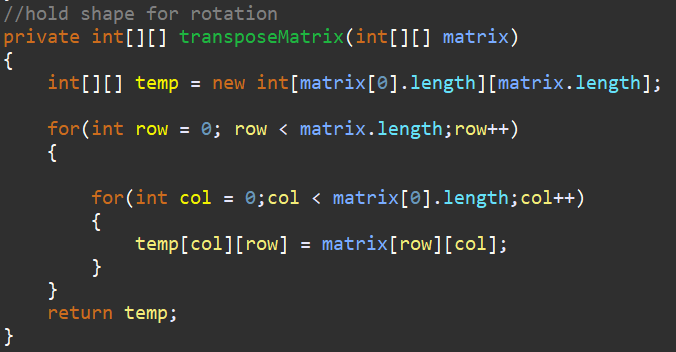
**3.1**. **Functions in Shape Class**

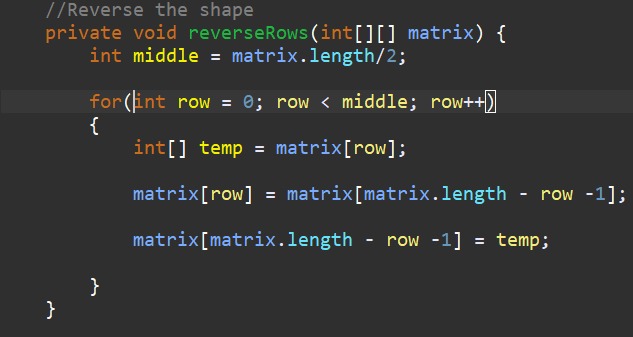


* checkLine method is used to remove a full line, it create a temporary line board.getBoard()[bottomLine][col] which copied the value of the line board.getBoard()[topLine][col], and check if both of them have value then increase the count. If the count is not equal with the width of the board then it mean that the line is not full line and nothing happen. But if the count equal the width of the board then temporary whose have the value of next line with fall down and replace the full line that got removed.

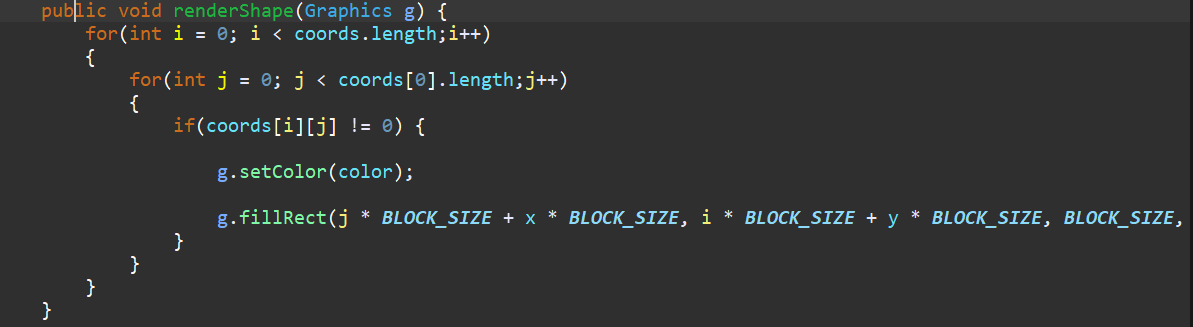


* rotatedShape is used to call the transposeMatrix and reverseRows method
* tranposeMatrix act temporary holder to transpose the matrix of the current shape, rotate it 90 degree and combine with the reverseRows method to achieve the full rotation.



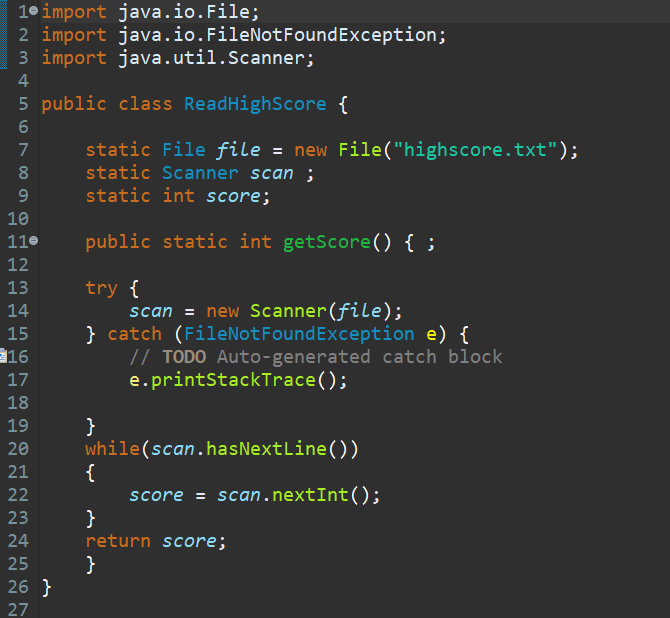


* renderShape is the method that draw the current shape when called.



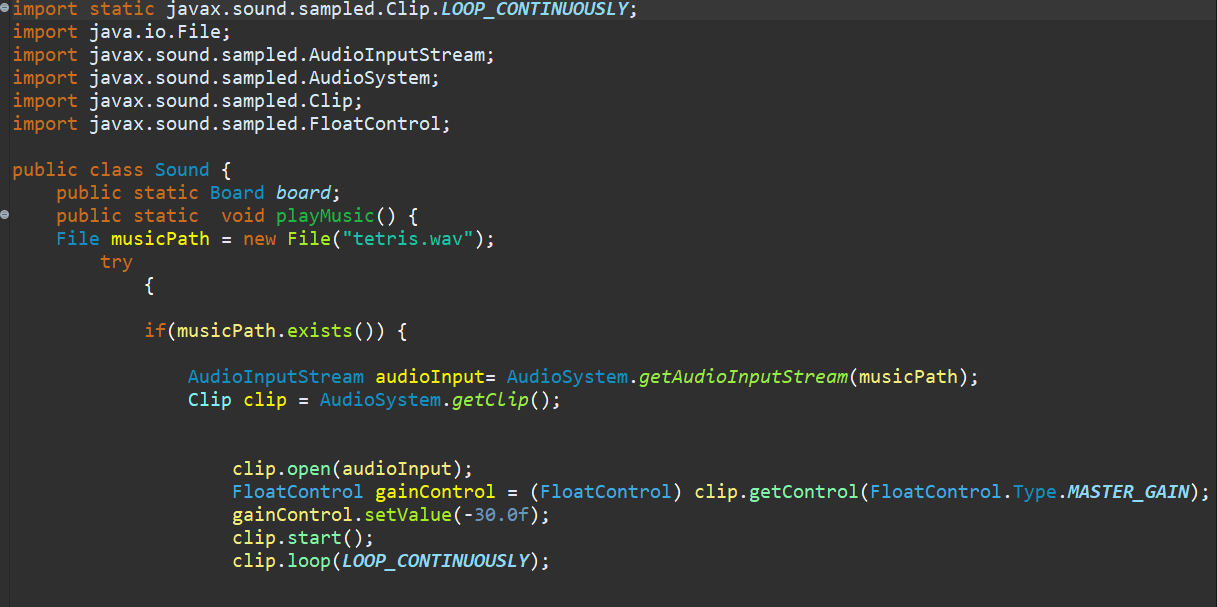
**4.READHIGHSCORE CLASS**

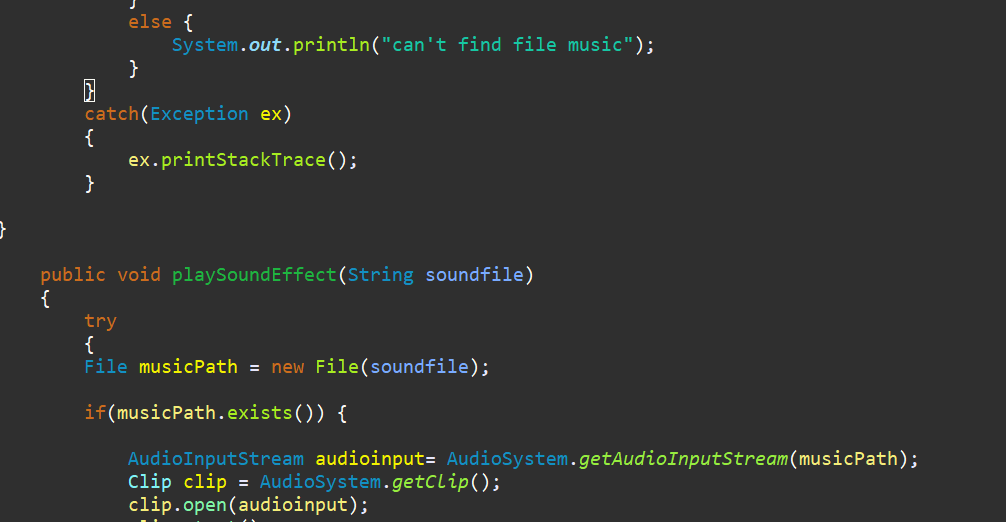
* Use scanner to read and return the current value of the score in the text file.

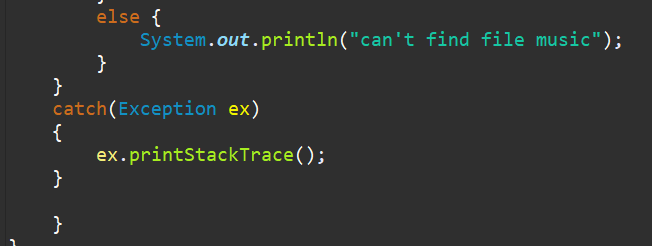


**5. SOUND CLASS**

* Provide method to play and loop music file, and method to active certain sound effects for specific actions.







- THE END -