**OOP REPORT**

**Name** : Trần Đình Khôi Nguyên - ITITIU19166

Hồ Thị Thu Hòa - ITITIU19120

1. **Purpose:**

- The purpose of this project is to desgin a basic game that was build on foundation of object – oriented. Our group decided to modify a game named “Tetris” follow the principle of this method. This game will shown clearly the object – oriented property, the combination of classes, between class and object. Tetris is a game require the user to arrange different shapes of cubes in order to there is no empty in a line. Although Tetris has appeared for a long time, our Tetris has been developed with new design, speed of the game, not require the configuration of the computer and specific is this game is build based on the Object – Oriented Mehod.

1. **Property of Tetris Game:**

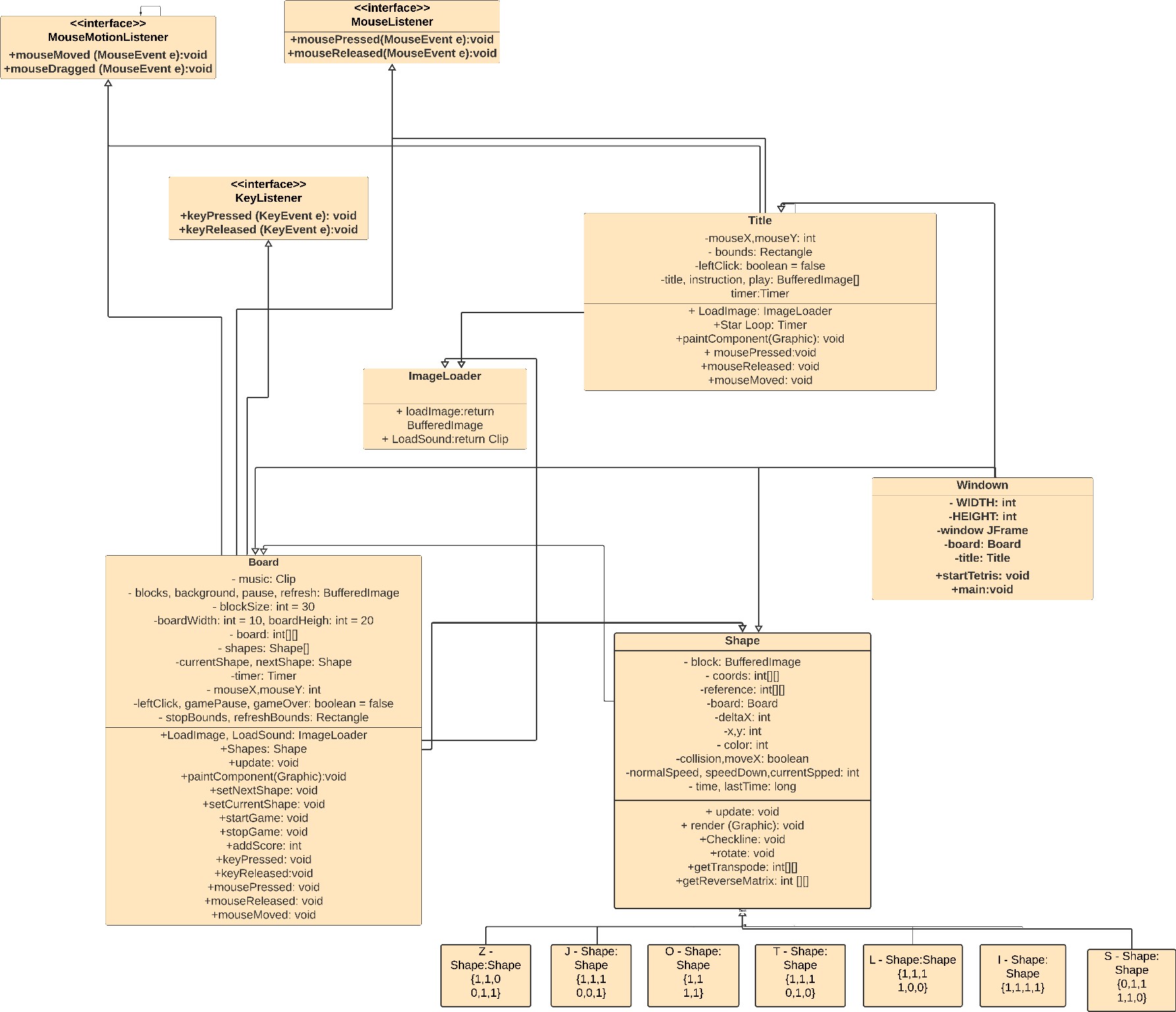
## Goal

The aim in Tetris is simple. You bring down blocks from the top of the screen. You can move the blocks around, either left to right and/or you can rotate them. The blocks fall at the center rate, but you can make them fall faster if you’re sure of your positioning. Your objective is to get all blocks to fill all empty space in a line at the bottom of the screen; whenever you do this, you’ll find that blocks vanish and you get awarded some points.

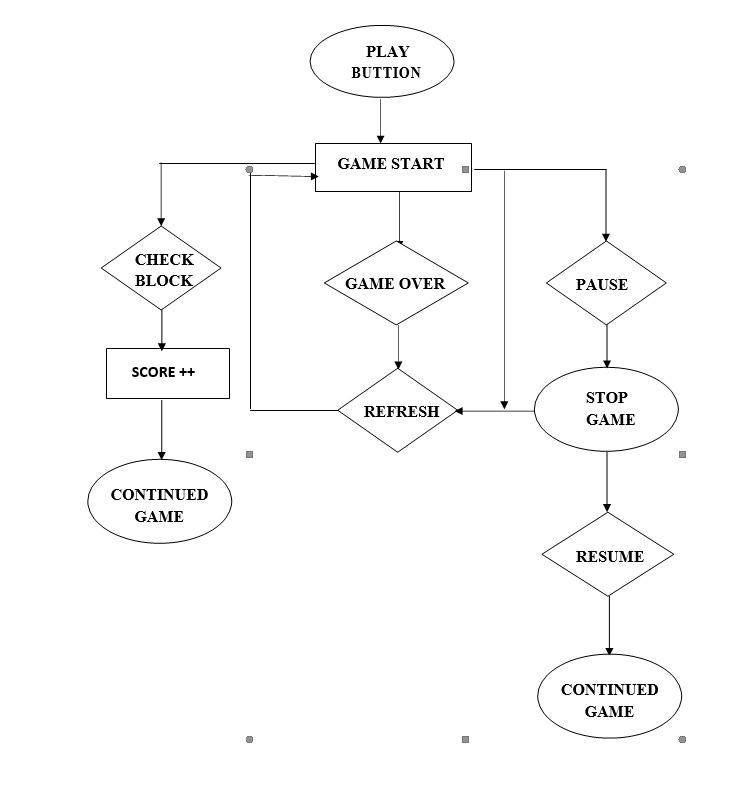
## Rule

Tetris has very simple rules: you can only move the pieces in specific ways; your game is over if your pieces reach the top of the screen; and you can only remove pieces from the screen by filling all the blank space in a line.

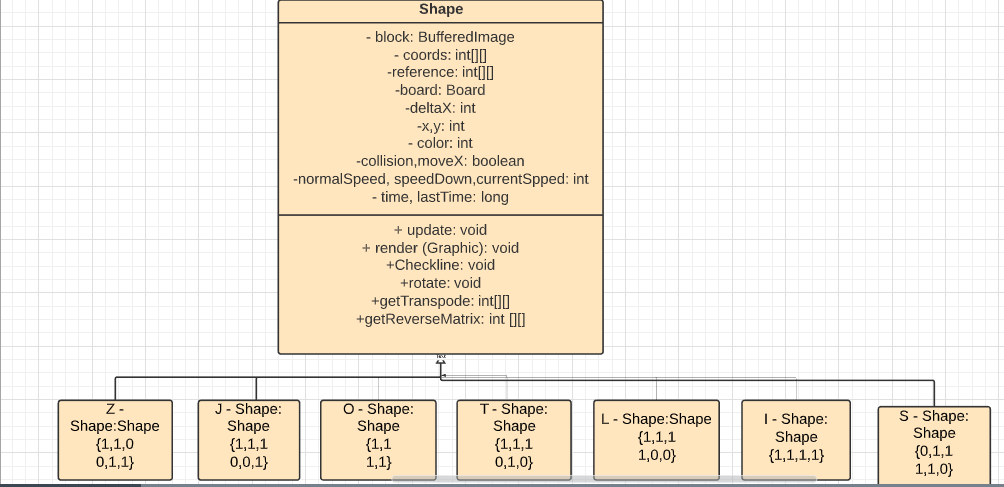
1. **UML Diagram of Tetris Game:**



1. **ALGORITHM of Tetris Game:**



1. **MAIN CLASS – FUNCTION – EXPLAIN:**
2. **Shape Class:**

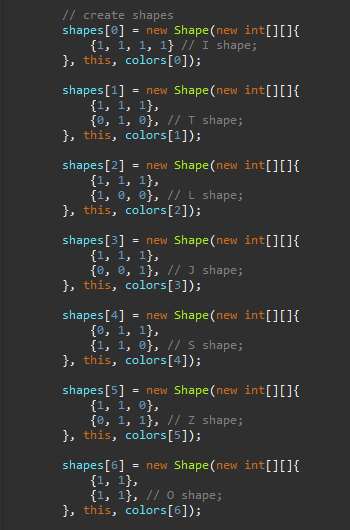


Shape Class is created to desgin others shapes easilly. It is inheritant from the general shape about properties and fucntions.In Shape Class, we use matrix type int do desgin the blocks, put it into xy plane with x,y coordinates. Each piece also have normalspeed, speedDown when user want piece move fast.

Each object of shape also have all functions like: update, render, checkline, rotate, transposed, reverse,etc…In update method, the program reset the speed to normalspeed, check the collision of the shape between two shapes or shape with edges of board, update the score, set the next shape, check the shape is move normal or fast.

There are 7 pieces in standard Tetris. A piece can be rotated 90˚ counter- clockwise to yield another piece. Enough rotations get you back to the original piece.

Code :



A block is represented by the coordinates of 6 squared – blocks then we color what block has value is 1. Each piece has own coordinates.

* + I – Shape is {1,1,1,1}
  + Z – Shape is {1,1,0

0,1,1}

* + S – Shape is {0,1,1

1,1,0}

* + T – Shape is {1,1,1

0,1,0}

* + J – Shape is {1,1,1

0,0,1}

* + L – Shape is {1,1,1

1,0,0}

* + O Shape is {1,1

1,1}

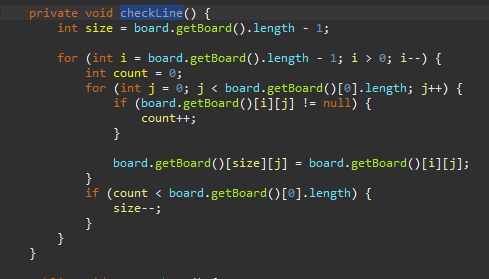
**CHECK COLLISION ALGORITHM:**

In check collision method, we check if the pieces is prevent from moving by other pieces or the edges of the board.

If the position of the block of the current piece that has the value is the same position with the any block of previsous piece or after moving 2 piece has the same position of any blocks of this, or it reach the limit of the board, piece can not skip moving anyway and stop at this coordinate. Collision return false value.

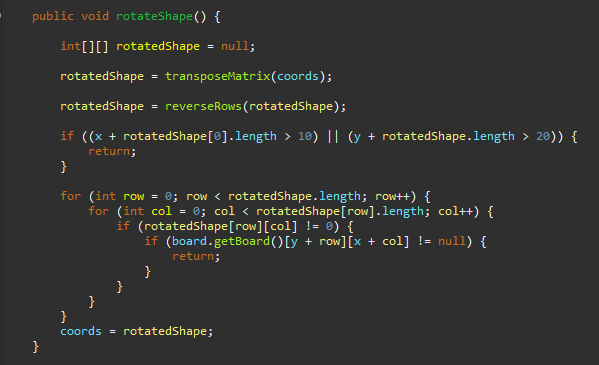
**CHECK LINE ALGORITHM:**

Code :



The checkline method is to remove the shape when full of line. We use the nested if to check each line of the board. If the line is full, remove this line.

**ROTATE ALGORITHM:**

Code :

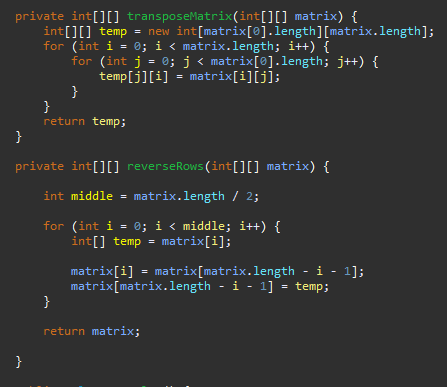
In rotate method, the main algorithm is to Transpose the matrix of the piece. For example, with J – shape {1,1,1

0,0,1} when rotate, this shape become

{1,1,1

1,0,0} L – shape.

Its mean we rotate 90 degree of this piece that make the matrix change follow the Transpose Principle.

To complete the rotate method we combinates getTranspose method with getReserveMatrix together.

1. **Board class:**



**MOVE SHAPE ALGORITHM:**

* **Repaint and update everything**:
  + **Using Timer** : should not use for game. However, this tetris is a small game, so using Timer may make the program easier and more simple
  + **Set Frames per second** ( FPS = 60)
  + **Set delaytime = 1000/FPS**
  + Override the actionPerformed with method repaint(), update()

 By this way, the board can be update and repaint with the timer.

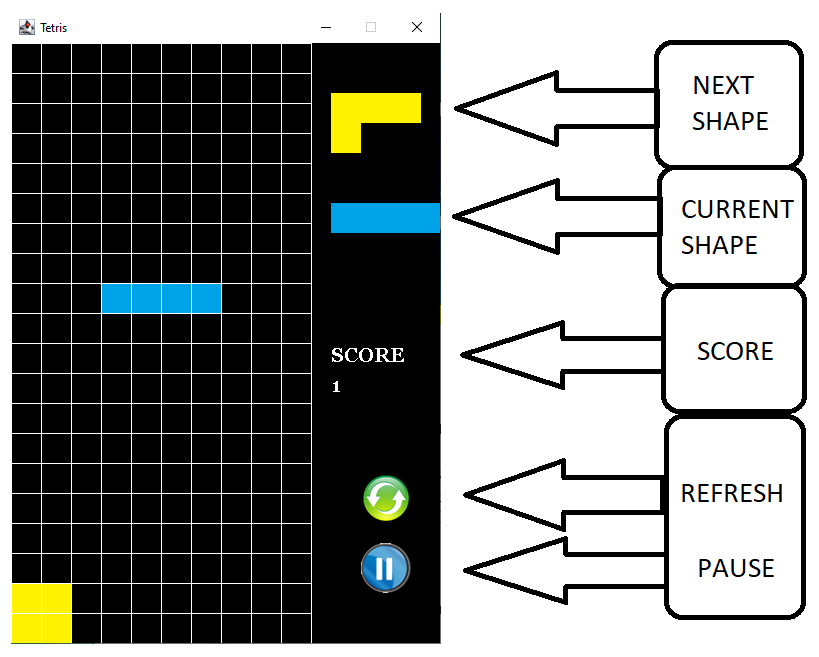
* **Move shapes**:
  + Implement KeyListener
  + Override keyPressed, create setDeltaX variable so that we can control the shapes moving to the left or the right
  + Update method to make the shape just move in the frame
  + KeyReleased to make the shapes move faster, so we create the normalspeed, speeddown, and currentspeed to manage the moving speed.

1. **Result – Limited – Conclusion:**
2. **Result**

Based on principle of OOP Method, our project has been completely with basic rules and properties of Tetris game. The combination of classes and objects in system is relative logically. We also do successfully the display to conect the user with program, it can be control by other input devies like: mouse, keyboard.

Morever, there are **some new features** out of the basic rule is the sound while playing, some button to pause, stop, refresh. The user interface is more attractive.



******

***User Interface of Tetris Game***

1. **Limited**

Beside the success of buil the game with basic rules, our project still has many case that can not be solved:

Do not have Save Game, Save Score, input the Name for the user.This game just for 1 player, still have not modified for 2 players.

There is only 1 level. We have not yet build more upgrade levels.

1. **Conclusion**

Tetris Game that was build by object – oriented method is more easier and locially than traditional – method. This shows cleary polymorphism, inheritance, encapsulation, data abstraction of OOP and the realationship between Shape, Board or other shapes together is linked tightly and systematically. Besides that, learning more knownledge out of the limited of this course is one of the important things to do while performing this project.