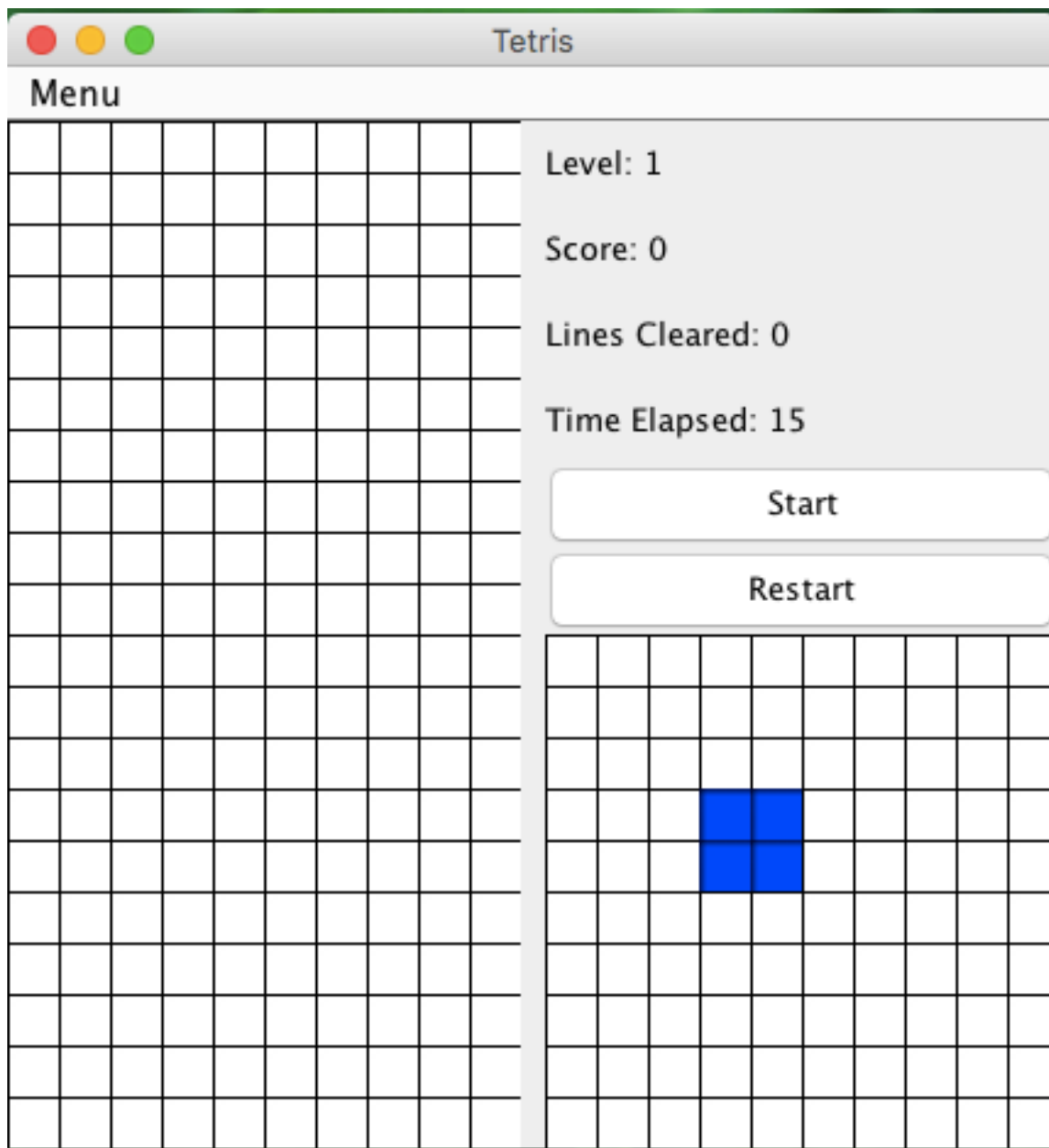

Tetris

Design Report

Dhrumil Patel, Mitul Patel - April 24, 2016



Tetris is a java based application build by using swing libraries. The details about the game can be found here : <http://en.wikipedia.org/wiki/Tetris>. It is a real time game that continuously drops pieces on to a pile. The player is allowed to rotate or move the the pieces left/right to fill up rows in pile. When a row gets completely filled, that row disappears and all the pieces on top of that row falls down. Over time, the pieces starts falling faster. The application is built by Dhrumil Patel (dpate85@uic.edu) and Mitul Patel (mpate204@uic.edu). We are currently in junior year and are pursuing Bachelors degree in Computer Science at University of Illinois at Chicago.

The purpose of the project is to gain immense knowledge on how objects interact with each other. The swing library is heavily used in the program. There are seven different tetrominoes each made up of 4 blocks. The tetrominoes are referred as I, T, O, L, J, S, and Z. We had to create a base class called Tetromino. Each tetromino has to extend the base class. The details about requirements of program can be seen in the project pdf¹. To achieve such functionalities, we created class for each tetromino which extended the base tetromino class. All those classes were then used in the GUI tetrisGame class. The TetrisGame class handles all the time, level score, restart, and menu options, while the Tetromino classes : I, O, T, J, L, S, and Z inherited the methods form the base class Tetromino according to the requirement.

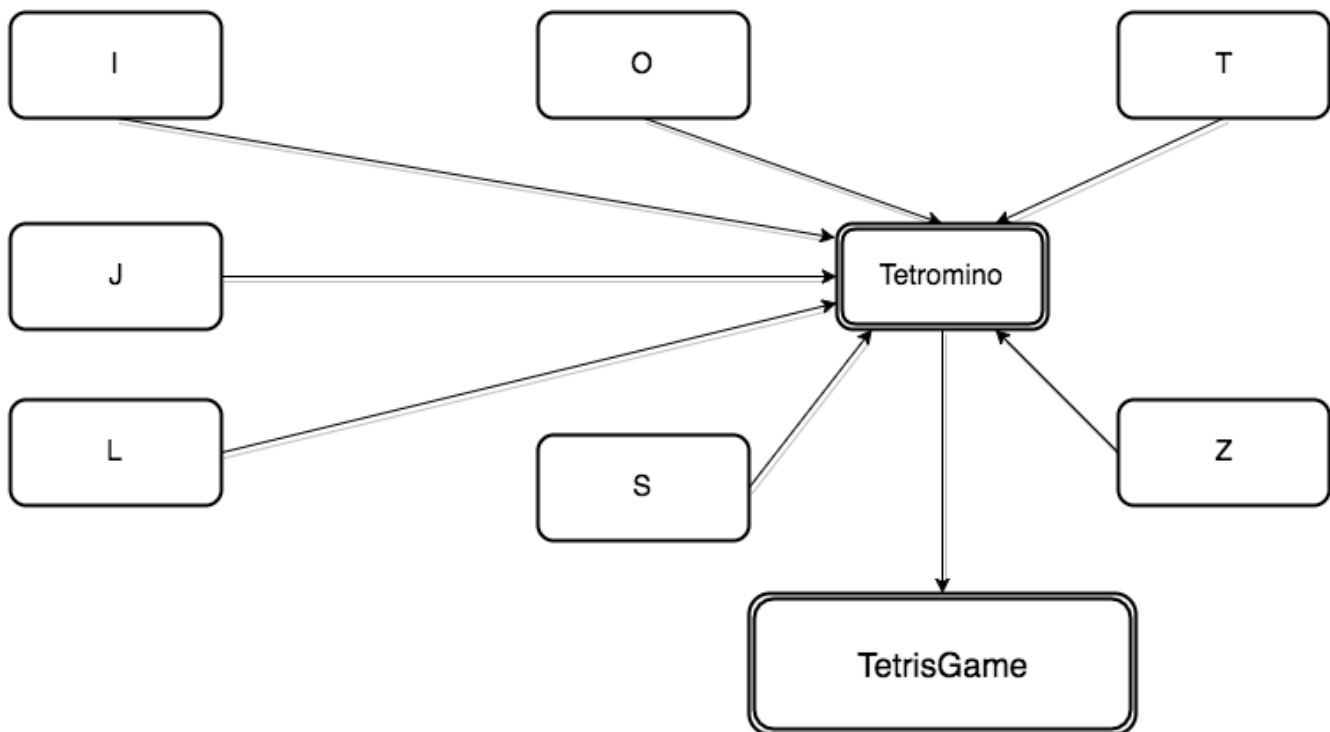
To better understand the classes, a brief description for each class is described in the table below:

File Name	Brif Description
I.java	Handles the I tearomino. Inherits methods from base class Tetomino. Used in TetrisGame class. Changes the axis which results in rotation according to requirement.
O.Java	Handles the O tearomino. Inherits methods from base class Tetomino. Used in TetrisGame class. Changes the axis which results in rotation according to requirement.
T.java	Handles the T tearomino. Inherits methods from base class Tetomino. Used in TetrisGame class. Changes the axis which results in rotation according to requirement.
J.java	Handles the J tearomino. Inherits methods from base class Tetomino. Used in TetrisGame class. Changes the axis which results in rotation according to requirement.
L.java	Handles the L tearomino. Inherits methods from base class Tetomino. Used in TetrisGame class. Changes the axis which results in rotation according to requirement.
S.java	Handles the S tearomino. Inherits methods from base class Tetomino. Used in TetrisGame class. Changes the axis which results in rotation according to requirement.

¹ <https://www.cs.uic.edu/pub/CS342/AssignmentsS16/proj5s16.pdf>

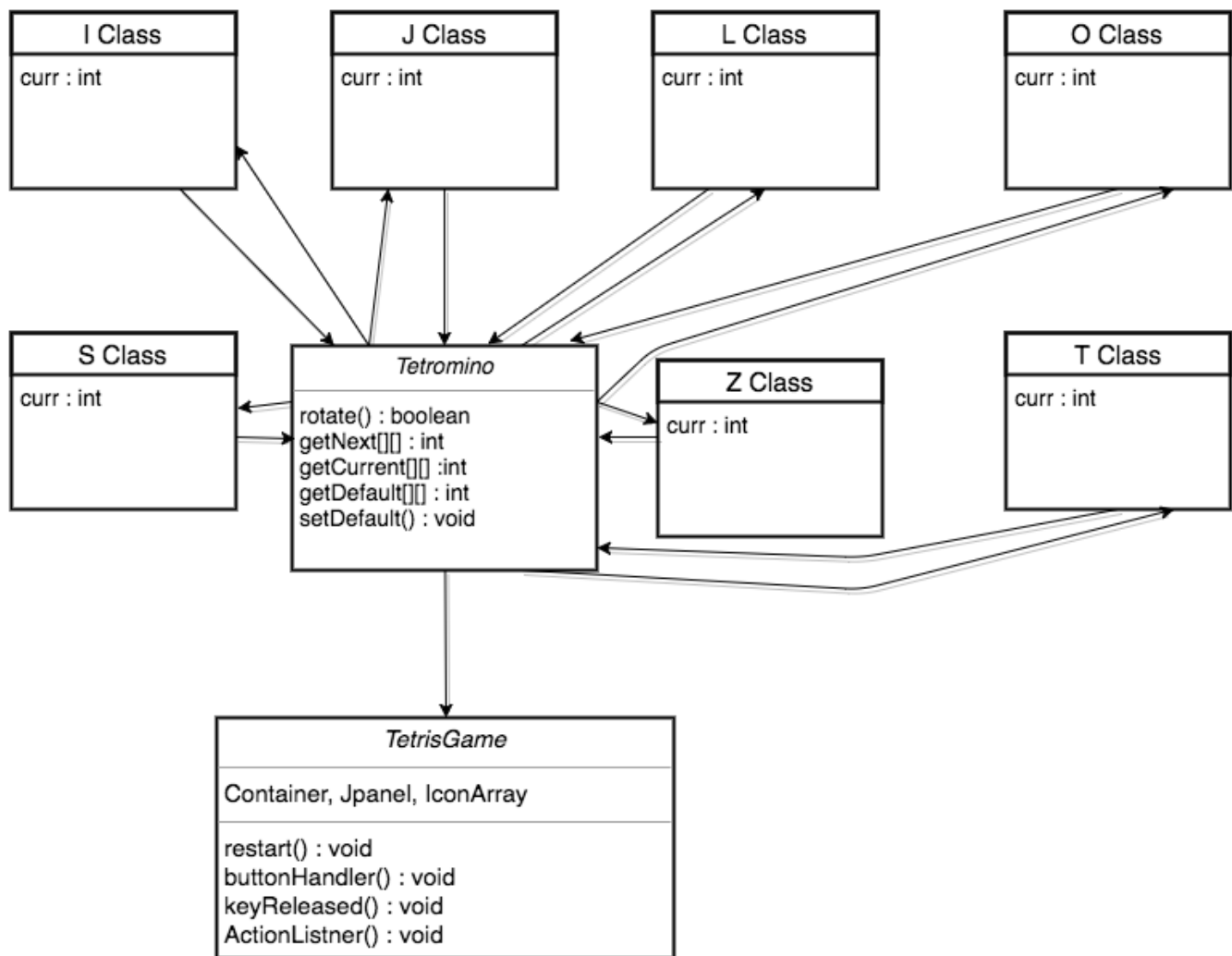
File Name	Brif Description
Tetromino.java	All the tetromino inherits from this class. It contains following methods: Rotate(), getNext(), GetCurrent(), getDefault, setDefault(). More detailed information about each method will be described later in the report.
TetrisGame.java	This class is responsible for score keeping, level management, delay management, GUI, time keeping, checking if the bar is filled or not, removing filled bar, etc.

The connection of the classes is shown in the diagram :

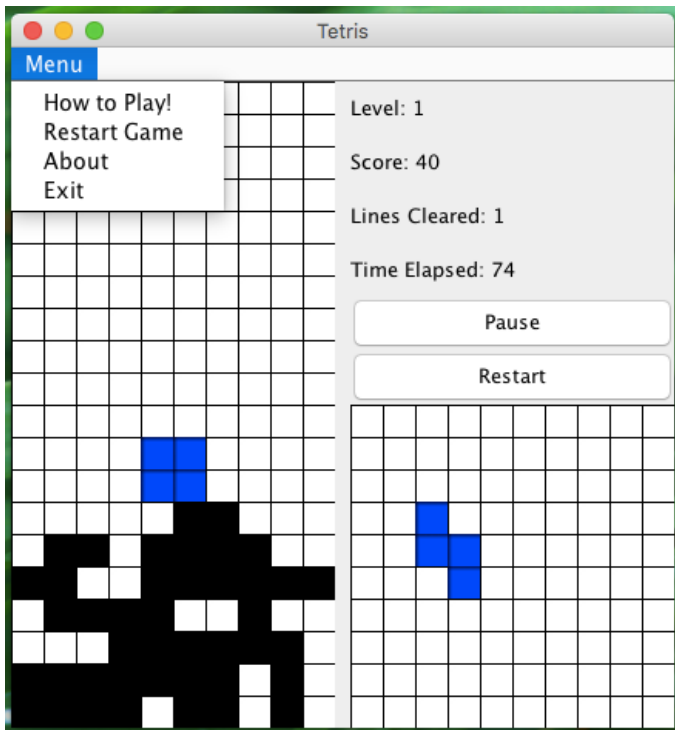


Every Termino gets rotated at 90 degrees. Therefore, when the tetromino is rotated 4 time, it should end up at same postion. The classes correspond to each tetromino handles the rotation. the rotate method is called and axis is changed every time a “rotation” button is pressed. All other operations are handled by the TetrisGame. This makes the design simple and easy to understand. The modification also becomes easy as the programmer will not have to change anything else except for the intended part of the game. The TetrisGame automatically adjusts accordingly. If we need to change the time settings or score difficulty, only the TetrisGame needs to be changed.

The inheritance of each class is in the UML diagram below:



The GUI of the game is fairly simple. The half right part contains the information about game. It has the Level, Score, Lines Cleared, and Time Elapsed text. The function of each label is self explanatory. It also has the Start and Restart button. When the game starts, the Start button changes to Pause button. This gives player the functionality to pause the game. The bottom half of the (half right) frame displays the upcoming tetromino. The Time keeps increasing by every second. The next tetromino keeps changes as the game progresses. The left half of the frame is the main part of the game. the tetrominos gets dropped here. Rotation occurs in real time.



The bar consists of drop down menu. The menu contains 4 options. "How to Play!" has the list of keyboard button with corresponding function. "Restart Game" basically restarts the game. "About" consists of relative information about the authors. "Exit" will close the program.

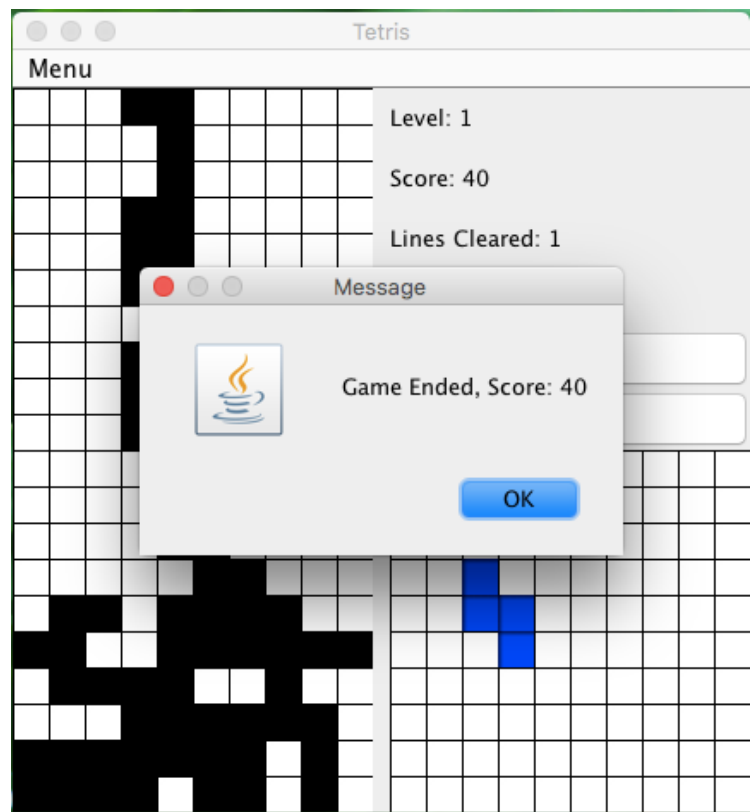
As you can see, the labels on the rights gets updated in real time.

The tetromino converts to black color when it gets fixed. When it is falling it is blue color. White squares are used as background.

When the game ends, a dialog appears having the message :

"Game Ended. Score : XX "

Clicking OK will take you back to the game and no action will be taken. To start the game again, restart needs to be pressed.



The benefit of the program is the user interface. It's very simple and easy to use. Every information needed is listed in the frame. The player / user does not have to follow extra steps to get any kind of information. The program can completely be played from keyboard. If graphical user face needs to be changed, only the TetrisGame class modification is required, To modify a tetromino, the programmer does not have to change every class. Changing the TetrisGame class is risky because having any illogical formula for delay or changing time settings could break the program. It can output garbage score. The blocks might not fall properly.