CptS355 - Assignment 6 - Java
Fall 2018

Assigned: Wednesday November 28, 2018

Due: Friday December 7, 2018 @ 23:59pm

Weight: Assignment-6 will count for 6% of your course grade.

Your solutions to the assignment problems are to be your own work. Refer to the course academic integrity statement in the syllabus.

This assignment provides experience with language features of Java that we haven't encountered in previous languages. The goal is to write a simple tetris game where the player tries to match the pieces with same numbers. The player will score points every time there are 3 or more adjacent matching pieces on the game board. The matching pieces will be cleared on the board and some pieces will be moved to remove the gaps. The more pieces are cleared on the board the higher the score will be. The player can earn bonus points by matching the characters of the target string ('i.e., "355").

Turning in your assignment

All code should be developed in the `JavaTetris` directory. When you are done, the directory will contain your source (`.java`) files and object (`.class`) files. (If you use a Java IDE (Eclipse or NetBeans) there may be additional project files.) Before you submit your assignment, please delete all the `.class` files and the project files. You only need to submit the `.java` files.

Please make sure that your `JavaTetris` folder only includes the `.java` files and zip the `JavaTetris` folder. To submit your assignment, turn in your zipped file by uploading on the Assignment6 (Java) DROPBOX on Blackboard (under AssignmentSubmisions menu).

The work you turn in is to be your own personal work. You may not copy another student's code or write any code together with another student. You may not copy code from the web, or anything else that lets you avoid solving the problems for yourself. **Please mention the students with whom you collaborated in a comment in the beginning of your Tetris.java file.**

Getting Started

A skeleton of the code for this assignment is provided on Blackboard (`JavaTetrisSkeleton.zip`). The skeleton code includes the following files:

- `Tetris.java` – Sets up the game. Uses Java Swing to create the game interface and to draw the tetris pieces. Creates the game board, status/score bars, and adds them to the container of the main window.
- `Board.java` - Implements the game logic. The `Board` class provides the methods to draw the board and the tetris pieces, move the falling tetris piece every second, and store the dropped pieces in the board.
- `Piece.java` Implements the `Piece` class. Each tetris piece is a Piece object. The `Piece` class provides the methods to get and set the properties of the tetris piece.

Compiling and Running Your Project

Download and install Java Software Development Kit (JDK) from the link:
https://www.oracle.com/technetwork/java/javase/downloads/index.html

On Windows: Before you run your code on command line, make sure that Windows can find the Java compiler and interpreter by adding the Java installation directory to the Windows 'Path' environment variable.

(Go to Computer -> System Properties -> Advanced system settings -> Environment Variables -> System variables and Edit Path variable under System Variables. Add the Java installation path (e.g. C:\Program Files\Java\jdk-11.0.1\bin) to the **beginning** of the 'Path' variable.)

To run your code on the command line:

In command line, browse to the `JavaTetris` directory and compile the program with the following command:

```
javac *.java
```

and run the game :

```
java Tetris
```

You may alternatively use an IDE (e.g., Eclipse).

Instructor will show how to run the skeleton code on Eclipse in class. (You can download Eclipse at https://www.eclipse.org/downloads/)

Grading

The assignment will be marked for good programming style (indentation and appropriate comments), as well as clean compilation and correct function.

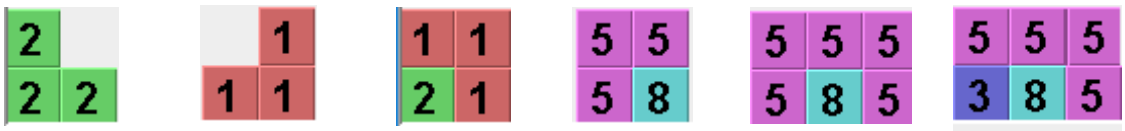
Help with Java

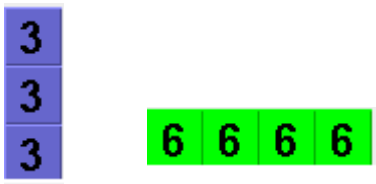
There is extensive documentation for Java on the web. Below link includes a list of Java tutorials.

The Java Language Tutorials (from Oracle - previously Sun)

Assignment Requirements

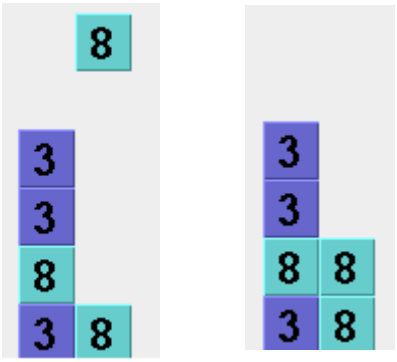
1. (30%) The provided skeleton code adds the dropped pieces to the board, but doesn't update the board if there are any adjacent group of pieces with 3 or more matching pieces. You need to complete the "`updateBoard`" function and add the code to clear all adjacent matching pieces after a piece drops. Note that the matching pieces can appear in many different ways (see below for examples – the following doesn't include all possible cases):





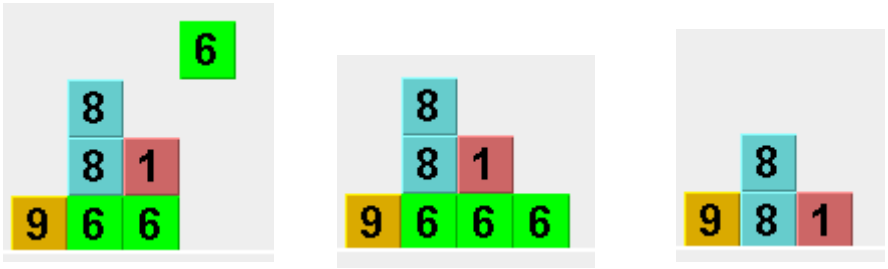
2. (15%) In addition, if the characters of the target string ('i.e., "355") is matched horizontally or vertically, those should be cleared from the board as well. You should clear the board for the following cases:

3. (10%) The matching groups should be cleared from the board shortly after the last matching piece drops. You will be deducted points if you fail to clear some matching groups until the next piece drops. See below for an example.



Both the *8-group* and *the 3-group* should be removed after the third 8 drops.

4. (30%) If there are additional non-matching pieces on top of the cleared pieces, gaps will appear on the board after matching pieces are cleared. Your game should remove those gaps by moving the non-matching pieces down. See below for an example.



5. (10 %) Add a new class "Player" to your project. The Player class should store player statistics in the current game. These include:

   - the number of pieces cleared so far on the board,

- the score that the player has earned in the game.

These information should be displayed on the screen (in the score and status bar).

6. (5%) After a matching group of pieces are cleared, the game score should be updated. The player will earn 5 points for each cleared piece. And for each cleared target (i.e. '355'), the player will earn 20 bonus points.