Group 6: Khanh Phan, Shaista Usman, Yi Niu

CPSC 5210-01 Seattle University

**Milestone 2**

1. **Project Description:**

The source project chosen by our group is a classic game called Tetris written in Java. In this game, a random block of different shapes will appear from time to time, allowing the user to rotate the blocks, move left, move right or move down. The goal is to fill as many lines of the grid as possible to earn points. Everytime the user makes a move, the program will actively check to make sure it’s not blocked by the current state of the grid before executing the user's action.

We use Eclipse and IntelliJ IDEs as well as J-Unit framework for our test suite.

Github URL:<https://github.com/Glank/Java-Games/tree/master/Tetris>

1. **Test Instructions:**
2. **Build Script:**

In order to automate the building of the Tetris source code and the Junit tests and simulate an environment conducive for regression testing, a shell script that compiles all the source code and their corresponding tests was designed. The script was tested and successfully run on the CS1 server with the Java Development Kit (version 11.0.6) already available there. Additionally, JUnit (version 4.13) and hamcrest-all (version 1.3) JAR files were installed and placed in a “lib” folder within the directory that contained all the codes. The build script further sets up the Classpath variable to point to the locations of these JAR files. This is a dependency and it is crucial for these JAR files to be installed on the server, for the tests to be compiled and run successfully. Since there was a great deal of dependency between the Tetris source codes, the \* wildcard was used in the script to compile all the source codes and Junit tests simultaneously. We further took inspiration from the Battleship Project to counter the issue with CS1’s memory constraints and used the memory flag “-J-Xmx512m” to limit the memory used by the Java Virtual Machine and aid in execution of the codes**.**

Usage:

* Use an FTP client like WinSCP to upload “Java-TetrisTest-master.zip” to any location on SU’s CS1 server
* unzip Java-TetrisTest-master.zip
* cd Java-TetrisTest-master
* ./buildTestSuite.sh

Link to Zip on GitHub:

<https://github.com/IrenaNiu/Java-TetrisTest/Java-TetrisTest-master.zip>

Link to script on GitHub:

<https://github.com/IrenaNiu/Java-TetrisTest/buildTestSuite.sh>

1. **Regression Testing:**

The runTestSuite script was written to automate the execution of the JUnit tests and simulate regression testing. The objective of this script is to ensure that whenever changes are made to the source code, the existing functionality can be tested in an automated manner against the predesigned unit tests and their outcome can be evaluated and assessed with minimum effort. The run script simply executes all the JUnit tests and presents their results, i.e. whether these passed or not. This script enables the tests to be run multiple number of times and additionally send their results as an email to a specified recipient. The script further logs the time taken to execute the tests and prints these timestamps in a log file for further analysis. This script was also successfully run on the CS1 servers.

Usage:

./runTestSuite <numIter> [emailRecipient]

- <numIter> - Number of times to run the test suite. Range: [1,10000)

- [emailRecipient] - Optional e-mail address to notify with test results

Link to script on GitHub:

<https://github.com/IrenaNiu/Java-TetrisTest/runTestSuite.sh>

1. **Stress Testing:**
2. **Test Result:**
3. **Code Coverage:**
4. **Lessons Learned:**