

CS 520

Extra credit assignment

Due: **Thursday December 16, 2021, 11:59 PM (a little before midnight)** via [Moodle](#). You may work with others on this assignment but each student must submit their own write up (with their name at the top), clearly specifying the collaborators (also at the top). The write ups should be individual, not created jointly, and written in the student's own words.

How to get started

1. Clone the repository with the following command: `git clone -b v3.0.0 https://github.com/LASER-UMASS/cs520`
2. Read the provided *README* in the *tictactoe* folder.
3. Use the commands to document, compile, test, and run the application from that folder.
4. Familiarize yourself with the original application source code contained in the *src* folder: `RowGameApp`, `src/controller/*.java`, `src/model/*.java`, and `src/view/*.java`.

Command design pattern [Approximately 1/2 points]

The Row game app will apply the command design pattern (https://en.wikipedia.org/wiki/Command_pattern) where the set of commands are: move, reset.

Proposed extension [Approximately 1/4 points]

The proposed extension is to support undo. You should write pseudocode for how to design and implement an "undo move at block row, column" command. (You can think of undo move at block row, column as being the inverse operation of the move at block row, column.)

0.0.1 Manually written test case [Approximately 1/4 points]

For your proposed extension, you should write pseudocode for the test case for the "undo move at block row, column" command.

Here is the expected test case format:

1. setup // In JUnit, @Before and check pre-conditions
2. execution // In JUnit, call the constructor or method
3. validation // In JUnit, check post-conditions
4. cleanup // In JUnit, @After

You may reuse any of the pre- and post-conditions already in the `TestExample` class.

Automated test generation [Approximately 1/2 points]

The toradocu test generation tool takes as input a Java program and generates unit tests based on the javadoc comments:

- Project: <https://github.com/albertogoffi/toradocu>
- Paper: <https://dl.acm.org/doi/10.1145/2931037.2931061>

Code review [Approximately 1/4 points]

You should perform a code review of this automated test generation tool. What are 3 best practices being followed? For each best practice, you should provide an example from the code.

Automatically generated test cases [Approximately 1/4 points]

1. In the model package, which constructors or methods should generate unit tests based on their javadoc comments?
2. Does the TestExample test suite already cover SOME of the automatically generated test cases? If so, which?
3. If you created a new test suite containing the manually written test cases (in our case TestExample) and the automatically generated test cases, do you generally expect the code coverage to increase? Why or why not?

Deliverables

Your submission, via [Moodle](#), must be a single document (plain text or PDF) named *extraCreditAssignment.txt* or *extraCreditAssignment.pdf*, containing:

1. Your full name and any collaborators (both at the top)
2. Proposed extension
3. Manually written test case
4. Code review
5. Automatically generated test cases answers