CSCI 1933 Chess Testing Guide
Spring 2023

1 Introduction

This document is intended to give you the information necessary to run the provided unit tests for your project. It is not intended to be a comprehensive guide on JUnit or unit tests in general. If you want to learn more, please visit https://junit.org/junit4/. The provided test files are what your TAs will be using to grade your project, and the tests will contribute a significant portion of your grade for Project 2, so it is obviously in your best interests to make sure your project passes as many tests as possible.

The files you will need to run the tests is TestAllMoves.java, ScoringRule.java, Category.java, WorthPoints.java, and bools.txt. These files should be placed in your source code folder (the src folder in IntelliJ) except for bools.txt which should be placed in the project root folder (the one that contains the src folder). Do not modify any of the other provided files.

2 Getting Started with JUnit

After adding the test files to your src folder, you will need to add the JUnit testing libraries to your project. In IntelliJ, this is as simple as putting your cursor over one of the statements importing a JUnit library in one of the test files (these statements should be red at first), pressing ALT+ENTER, then selection version 4 of JUnit from the drop down menu. **Only version 4 of JUnit will work with the tests given**.

If you have any issues with import the JUnit library, you can start by looking at the official IntelliJ article for importing the libary for testing: https://www.jetbrains.com/help/idea/testing.html#add-testing-libraries.

> **Note:** If there is no option that shows for JUnit version 4. Go to your project folder in IntelliJ, click File → Project Structure → Libraries → click the plus "+" → From Maven → search for "junit" and scroll down and click "junit:junit:4.0", anything between 4.0-4.13 should work → check "Download to" → OK. If you are still having trouble setting up your JUnit testing library, please ask one of the TAs for assistance.

3 What to Understand

The only code you need to understand is in the TestAllMoves.java file. Each test method in the file will be marked with an *@Test* flag. Tests that contribute to your score will have a *@WorthPoints* flag as well, indicating how many points that test is worth. Inside each method, there will be some statements calling methods of the list interface that you implemented and some **Assertions**.

A test only passes if all of the assertions in the definition pass. For example, consider the following test:

```
public void ExampleTest() {
    /*
    Do some stuff here: initialize variables a, b, c, d, and e, call test methods,
        etc.
    */
    assertFalse(a);
    assertTrue(b);
    assertNull(c);
    assertEquals(d, e);
}
```

This test will only pass if a is false, b is true, c is null, and d and e are equal. Otherwise, the entire test fails. You may also see Strings to be printed if an assertion fails as arguments to an assertion; note that they are only there to help explain why the assertion may have failed and are not part of the actual test.

4 Running Tests

To run the tests in IntelliJ, click the green "play" button to the left of the class declaration in the TestAllMoves.java file. This runs all tests in that class. Each test should have an identical green button to the left of it. Click that button to only run that test. After running any test(s), any failed assertions are underlined in red, which should help in debugging. You can also use the debugger in test classes to further assist in your debugging efforts. If you want to run the tests in some IDE other than IntelliJ or without an IDE at all, you will unfortunately have to do some research and figure out how to do this on your own, as the TAs are only knowledgeable about IntelliJ.