

Othello Testing Methods

Several frameworks exist for testing JavaScript code. I have used JsTestDriver, primarily out of familiarity (I contributed to the project at my internship at Google last summer). One can run the tests by downloading the .jar file, invoking it to start a server, pointing browser(s) at the server to “capture” them for code execution, then running the tests from the command line. It is not necessary to include the test code in a HTML file; the framework autodetects test functions and runs them. Also, by capturing multiple browsers, one can automatically test cross-browser compatibility. The test code is in the tests/ directory of the project.

I wrote tests for any code that appeared problematic (such as the code in board.js that enforces the rules of the game), as well as code that was in pure functions with no side effects as those can be verified in a straightforward manner. I tried to move as much logic as possible into pure parts of the code to test them more easily.

Automated tests were used to verify most of the backend, primarily the heavy lifting code in board.js and its related files. Since the objects and functions in these files are not associated with any GUIs, they can be tested in isolation.

The GUI code was tested manually by visual inspection of index.html. This was necessary because it is not easily possible for code to determine if a GUI is displayed properly.