Codebreakers
Testing Report

The testing process for the finite state machine required a set of tests for each element of the model. This includes the state class, path class, label class, and the automaton that is comprised of these three classes. Testing had to be conducted before implementation was completely finished. To accomplish the testing, a test dummy was created for the state in the model. This allowed other members of the team to continue writing and testing their own parts of the code. The dummy state had a rudimentary implementation that allowed other members of the team's code to pass their own tests.

The state class was expected to be able to store and retrieve paths between the current state and any outgoing states. Each state also has its own type (starting, normal, end), position on the screen, and a label. These attributes were set up, exercised, and tested by the tests written. Once the final implementation of the path class was finished, the other members of the team were able to update their testing classes using the final implementation of the path class.

Aside from the automaton, the path was perhaps the most difficult class to test, as it required many other objects, such as states, labels, and an automaton to test them on. This is where the test dummy for the state would have been of use. The path test cases exercised the path's capability of setting labels, types, and retrieving points. Although a simple component of the automaton, the label class underwent testing as well. The tests written exercised name and type capabilities of the label.

Finally, the automaton was to be tested. These tests encapsulated all of the other classes in the model. The automaton was exercised in its ability to add and remove states,

which in turn, exercised the state class, path class, and label class. The code was refactored until all tests ran successfully.