Final Test Report
CSC-260: Large Scale Software Development
David Annex, Blair Hagen, Jacob Karaul, Matthew Silfin

Save and Load Tests

Saving Same File with Different Names: Save the same FSM with two different names and see if the display shows identical FSMs.
Result: Test passes and saves and loads identical FSMs.

Saving File, Loading it, and Resaving: Save FSM and load it back, then resave to make sure that everything that is saved is loaded and everything that is loaded is saved.
Result: Tests passed and both saves were identical.

Action Tests

Default Action: Tests that the default action of a state is NoAction.
Result: The default action is NoAction.

Add Action: Tests that adding an action to a state actually adds the action.
Result: The action is added successfully.

Change Action: Tests that changing an action on a state that has already had an action applied to it changes the action correctly.
Result: The action is changed from the old applied action to the new one.

Open and Select Action Menu (manual): Test if selecting state opens action menu then selecting an action properly connects the action to the state.
Result: Menu opens, selected actions save properly in both formats, sound does not always work with Ubuntu.

Theme Tests (All manual tests)

Load Default Theme: Load in a default theme to change the colors of the FSM.
Result: Blue theme changes the colors of the FSM as intended.

Save Theme: Change the colors of the FSM and save that theme with the name you give it.
Result: New theme can be saved with the given name.

Load Saved Theme: Load a custom theme that has been previously saved.
Result: Saved theme changes the colors of the FSM when applied.

Save Same Name: See if two custom themes can be saved under the same name.
Result: Two themes can have the same name. This should not happen as it causes confusion and instead an error should appear when this is attempted.

Select Colors: Select specific colors to change the visuals of the FSM without saving a new theme.
Result: Selected colors properly change when applied without having to save the changes.

Simulation Tests (All manual tests)

Multiple Paths: Tests for what the simulation does when there are multiple possible paths, only one accepting path.
Result: Goes to the first possible transition, does not check all the transitions.

No Next Transition: Tests what happens when the simulator puts in a transition weight that does not exist.
Result: When the step is taken with the nonexistent transition, the simulation ends.

Standard Simulation: Tests a standard string simulation.
Result: Gets to the last state, stays highlighted showing there are no more inputs.

Below are tests results from previous test report

View Tests

Self Loop: Testing that a self loop is created, shows properly, and loads and saves as intended.
Result: Saves proper location coordinates of state and knows that there is a self loop transition connected to it. Display is correctly a self loop.

Empty FSM: Save an empty machine as the correct file and the file will load into an empty FSM as well.
Result: Saves the file with no information and is able to load a new FSM with the empty FSM file.

Single State: Tests if a single state can be made and displayed correctly and saved and loaded as well.

Result: state is saved and loaded properly with information with no ghost transitions or weights.

State to State Loop: Test if the states can display transitions going back and forth between the two states while being readable by the user and save properly.
Result: Each side of the transition is for weights in opposite directions and displays as such. Saves as with the proper transitions connected to each state.

One State to Many: Test connecting one state to multiple other states.
Result: Saves and loads in the proper format and when loaded will display the correct format for the states and transitions in the GUI.

Special Characters in Names: Test the save and load and subsequent display of when states and transitions have names and weights with special characters.
Result: The file will save with all the proper information in the format it should but loading with some special characters causes issues with displaying all the saved states and transitions.

One State to One State with Many Transitions: Test functionality with multiple weights on the same transition.
Result: The file is saved and loaded properly but the GUI overlays the arrows showing the weight of transitions which could be changed for a workaround.

One State to One State with Many Transitions Workaround: Attempt at solving the issue of transitions being over layed on top of eachother in the display.
Result: Saving the secondary transitions with spaces behind it means that the transitions are displayed correctly on the GUI but the saving method is changed.

Document Tests

Add State and Has State: Add a state and check that the document now has that specific state in it.
Result: Added state with name "A" and the FSM could tell that the specific state was added and know that it has that state.

Add and Remove State: Add a state and remove the same state.
Result: State "A" is added then removed. Knowing that add works, checking after that the FSM does not have any states in it after removing the state comes out to be true.

Remove and Has State: Add multiple states and remove one then check for both of the original states added.

Result: Adding state "A" and "B" then removing "B" produces the expected has state output of having A still in the FSM but not B.

Add Duplicate State: Add two states with the same name and make sure it does not make a duplicate state.
Result: Adding state "A" twice only adds one state with name A to the FSM, the number of states stays as one state, and the FSM is considered equivalent to an FSM that only had state "A" added to it once.

Add Transition: Add an transition with given weight between two states.
Result: Adding an transition produces the desired transition from one state to the other.

Remove Transition: Adds an transition between two states then removes that same transition.
Result: Added an transition, checked that it was added correctly, then removed the same transition correctly leaving no transitions connecting the two states.

Number of Transitions: Checks that the FSM properly tracks the amount of transitions connected to a specific state.
Result: Created two states, checked that the number of transitions was zero, added an transition and the number of transitions properly went to one, the removed the transition and the number went back to zero.

Notes on Design:

- The themes do not save from one FSM to another. This would be included in future versions of the FSM.