**Test Plan**

# **Objectives**

## Purpose

This document describes the plan for testing the Finite State Machine generator. This test plan document supports the following objectives:

* + Identify existing project information and the software to be tested
  + List recommended test requirements
  + Recommend and describe testing strategies
  + Identify expected resources and testing efforts
  + List deliverable elements of the testing activities

## Scope

This plan describes testing the program via unit and manual testing.

The GUI will be tested for the following, existing functions:

1. Creating a new state
2. Creating a new accept state
3. Creating a new link
4. Moving a state node and its associated links
5. Export .PY file
6. Create new project
7. Save project
8. Load existing project (.FXML)

The external interfaces to the following devices will be tested:

1. Local PC
2. Local terminal

The most critical performance measures to test are:

1. Response time for creating state nodes
2. Response time for creating links
3. Response time for moving state nodes
4. Correct coordinate recording for state nodes
5. Correct coordinate calculation for links
6. Correct pane management to prevent memory leaks
7. Any errors generated through expected, normal use

# **Requirements for testing**

### State node integrity testing

Verify state nodes are being stored in the node store instance.

Verify state node data matches expected data for each node processed.

Verify state node coordinates match expected coordinates on screen.

Verify state node correctly saved Pane instance reference.

### Link integrity testing

Verify links are being stored in the link store instance.

Verify link end points match expected data.

Verify link coordinates match expected points on state nodes.

Verify links correctly save Pane instance reference.

### User Interface Testing

Verify creation of new projects.

Verify closing of projects.

Verify ease of adding new state nodes.

Verify ease of adding links.

Verify ease of moving state nodes.

Verify saving projects.

Verify loading projects.

Verify exporting projects as .PY files.

### Performance Testing

Verify response time to create a new project.

Verify response time to create a new state node.

Verify response time to create a new accept state node.

Verify response time to create a new link.

Verify response time to move a state node containing a link.

Verify response time to save a project.

Verify response time to load a project.

Verify response time to close a project.

### Load Testing

Verify system response with 20 linked state nodes exported into .PY file.

# **Test Strategy**

## Testing Types

### *Data Integrity Testing*

The state and link stores must be tested to verify state node and link information is being recorded correctly and can be accessed.

|  |  |
| --- | --- |
| **Test Objective:** | To ensure that data stores record and retrieve data correctly without corruption. |
| **Technique:** | * Create two new state nodes * Verify nodes were saved correctly * Create a new link between nodes * Verify nodes were updated correctly * Verify link was recorded correctly * Move state node * Verify node and link data was updated correctly |
| **Completion Criteria:** | * All data matches expected values * All data updates as expected |

### *Function Testing*

The state and link stores must be tested to verify state node and link information is being recorded correctly and can be accessed.

|  |  |
| --- | --- |
| **Test Objective:** | To ensure all navigation, data entry, data processing, and data retrieval functions as intended. |
| **Technique:** | * Verify all menu options work as expected * Verify GUI buttons work as expected * Verify nodes can be moved as expected * Verify links follow nodes as expected * Verify link coordinates are calculated as expected * Verify saved project loads as expected * Verify exported .PY file works as expected * Verify exported .FSM file is generated as expected |
| **Completion Criteria:** | * All files are generated as expected * All navigation functions as expected * All GUI button operations function as expected |

### *Performance Profiling*

The state and link stores must be tested to verify state node and link information is being recorded correctly and can be accessed.

|  |  |
| --- | --- |
| **Test Objective:** | To ensure the system executes and performs with minimal errors. |
| **Technique:** | * Attempt to create two similar state nodes * Attempt to create two similar links * Attempt to move state nodes * Attempt to load an invalid project file * Attempt to process an invalid .FSM file * Attempt to use an incorrect output file |
| **Completion Criteria:** | * System handles all errors internally with minimal notification * System reports no unexpected or fatal errors |

### *Load Testing*

The state and link stores must be tested to verify state node and link information is being recorded correctly and can be accessed.

|  |  |
| --- | --- |
| **Test Objective:** | To ensure the system executes and performs with minimal impact to the user’s overall computing experience. |
| **Technique:** | * Create 20 linked nodes * Attempt to move nodes * Attempt to save project * Attempt to export as .FSM * Attempt to export as .PY * Attempt to load project |
| **Completion Criteria:** | * System is verified to perform as expected * All data is added and updated as expected * Processing does not excessively impede user experience |

### *Security Testing*

The state and link stores must be tested to verify state node and link information is being recorded correctly and can be accessed.

|  |  |
| --- | --- |
| **Test Objective:** | To ensure that projects being saved, loaded, or exported cannot put users at excessive risk. |
| **Technique:** | * Attempt to load invalid project * Attempt to process invalid .FSM file * Attempt to manipulate data stores directly * Attempt to inject within state node name * Attempt to inject within link node name |
| **Completion Criteria:** | * System processes all input as expected * System encapsulates and sterilizes inputs to strings * System handles errors quietly |

## Tools

The following are recommended tools for testing the system

Documentation / Notes

Microsoft Word

Test casing / Script writing

NetBeans IDE

## Roles

Chris Loos, Justin Butler

All members are responsible for testing and logging as described.