Test Plan
Mastergoal Machine Learning Environment
Version 0.2

Submitted in partial fulfillment of the requirements of the degree of MSE

Alejandro Alliana
CIS895 – MSE Project
## Change Log:

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Revision</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/05/07</td>
<td>Alejandro Alliana</td>
<td>0.1</td>
<td>Initial version</td>
</tr>
<tr>
<td>04/02/08</td>
<td>Alejandro Alliana</td>
<td>0.2</td>
<td>Minor corrections.</td>
</tr>
</tbody>
</table>
Index

1. **Test plan identifier** ........................................................................................................... 2

2. **Introduction** .................................................................................................................... 2

3. **References** ..................................................................................................................... 2

4. **Test item** .......................................................................................................................... 2

5. **Features to be tested** ......................................................................................................... 2

5.1. **Use Case “Create New Experiment”** ........................................................................... 2

5.1.1. SR 1.1: Create Experiment (Critical) .............................................................................. 2

5.1.2. SR 1.1 Select Learning Technique (Critical) ................................................................. 2

5.1.3. SR 1.2 Select Terms for experiment (Critical) ............................................................. 2

5.1.4. SR 1.3 Setup Search Algorithm (Critical) ...................................................................... 3

5.2. **Use Case “Load experiment”** ...................................................................................... 3

5.2.1. SR 2.1: Load experiment (critical) .................................................................................. 3

5.2.2. SR 2.2 Save Experiment (Critical) ................................................................................. 3

5.2.3. SR 2.3 Edit Experiment (Critical) ................................................................................... 3

5.2.4. SR 2.4 Experiment Format (Critical) ............................................................................. 3

5.3. **Use Case “Train Strategy”** ............................................................................................ 3

5.3.1. SR 3.1: Train Strategy (Critical) ..................................................................................... 3

5.3.2. SR 3.5: Rules of play (Critical) ..................................................................................... 3

5.3.3. SR 3.6: Correct search (Critical) .................................................................................... 3

5.4. **Use Case “Export strategy”** .......................................................................................... 3

5.4.1. SR 4.1: Export strategy (non critical) ............................................................................. 3

5.5. **Use Case “Play Game”** .................................................................................................. 4

5.5.1. SR 5.1: Play a game (Critical) ....................................................................................... 4

5.5.2. SR 5.2: Rules (Critical) .................................................................................................. 4

5.6. **Use Case “Explore Game”** ........................................................................................... 4

5.6.1. SR 6.1: Explore a game (Critical) ................................................................................... 4

6. **Features not to be tested** .................................................................................................. 4

7. **Approach** .......................................................................................................................... 4

7.1. Unit testing .......................................................................................................................... 4

7.2. Component testing and system testing ............................................................................. 4

7.3. Documentation Testing ...................................................................................................... 4

8. **Item pass / fail criteria** ...................................................................................................... 5

9. **Suspension criteria and resumption requirements** ............................................................... 5

10. **Test deliverables** ............................................................................................................. 5

11. **Test tasks** ........................................................................................................................ 5

11.1. **Test suite 1: Use Case “Create New Experiment”** ....................................................... 5

11.1.1. Test case 1.1: ............................................................................................................ 5

11.1.2. Test case 1.2: ............................................................................................................ 6

11.1.3. Test case 1.3: ............................................................................................................ 6

11.1.4. Test case 1.4: ............................................................................................................ 7
11.2. Test Suite 2: Use Case Load experiment. ................................. 7
  11.2.1. Test case 2.1: ................................................................. 7
  11.2.2. Test case 2.2: ................................................................. 8
  11.2.3. Test case 2.3: ................................................................. 8

11.3. Test Suite 3: Use Case “Train strategy” .................................. 8
  11.3.1. Test case 3.1: ................................................................. 8
  11.3.2. Test case 3.2: ................................................................. 9
  11.3.3. Test case 3.3: ................................................................. 9

11.4. Test Suite 4: Use Case “Export strategy” ................................. 10
  11.4.1. Test case 4.1: ................................................................. 10

11.5. Test Suite 5: Use Case “Play Game” ...................................... 10
  11.5.1. Test case 5.1: ................................................................. 10
  11.5.2. Test case 5.2: ................................................................. 11

11.6. Test Suite 6: Use Case “Explore Game” .................................. 11
  11.6.1. Test case 6.1: ................................................................. 11

12. Schedule ............................................................................. 11
1. Test plan identifier

MMLE Test Plan Identifier 0.2

2. Introduction

This document provides a test plan for the validation of the Mastergoal Machine Learning Environment. This document presents a set of test suites that the software must pass to be considered complete. Test stopping and resumption criteria as well as testing tasks are also specified.

3. References


4. Test item

The project follows a modern software process. It is expected that testing will be carried out on every iteration. On the construction phase, the source code is expected to be tested at the unit level using automatic testing environments.

5. Features to be tested

This section specifies the features to be tested.

5.1. Use Case “Create New Experiment”

5.1.1. SR 1.1: Create Experiment (Critical)
The user must be able to create new experiments.

5.1.2. SR 1.1 Select Learning Technique (Critical)
The user must be able to select from different learning techniques for the experiment.

5.1.3. SR 1.2 Select Terms for experiment (Critical)
The user must be able to select which terms must be used for the current experiment
5.1.4. SR 1.3 Setup Search Algorithm (Critical)
The user must be able to setup a search algorithm, maximum depth for the search algorithm and maximum time to be spent in searching.

5.2. Use Case “Load experiment”

5.2.1. SR 2.1 Load experiment (critical)
The system shall allow the user to load previously saved experiment. The properties of the loaded experiment should match those specified by the data in the saved experiment.

5.2.2. SR 2.2 Save Experiment (Critical)
The system shall allow the user to save an experiment. The data saved must match the properties of the saved experiment.

5.2.3. SR 2.3 Edit Experiment (Critical)
The system shall allow the user to modify the information of an experiment.

5.2.4. SR 2.4 Experiment Format (Critical)
The system shall define a format to be used when saving and restoring the experiments. This format should allow extensibility to allow saving experiments with different types of learning techniques.

5.3. Use Case “Train Strategy”

5.3.1. SR 3.1 Train Strategy (Critical)
The program shall provide a command to train a strategy from an existing experiment. Different learning components will train their strategy differently.

5.3.2. SR 3.5 Rules of play (Critical)
The system shall enforce the rules at every game played during training.

5.3.3. SR 3.6 Correct search (Critical)
The systems search algorithms should return the best move according to the evaluation criteria and the search algorithm behavior.

5.4. Use Case “Export strategy “

5.4.1. SR 4.1 Export strategy (non critical)
The system shall provide a mechanism to export a learned strategy to a file in external format.
5.5. **Use Case “Play Game”**

5.5.1. **SR 5.1: Play a game (Critical)**

The system shall provide a GUI to play games. This user interface can be used to try the learned strategies. The games can be played between Human Players and or Computer program players.

5.5.2. **SR 5.2: Rules (Critical)**

During a game, the system shall enforce the application of the rules at all times.

5.6. **Use Case “Explore Game”**

5.6.1. **SR 6.1: Explore a game (Critical)**

The system shall provide a GUI to explore the games played during training.

6. **Features not to be tested**

The performance of the system is an item that is not going to be tested for the first version.

7. **Approach**

The system is going to be tested following the traditional approach of unit, component and integration testing. Automated testing suites are going to be used where possible.

7.1. **Unit testing**

The system is going to be tested at the unit level using the *CppTest* tool which allows automatic execution of test cases to prove for regression bugs.

7.2. **Component testing and system testing.**

After the software has been tested for bugs at the unit level, integration testing is going to be performed at the component and system levels.

7.3. **Documentation Testing**

At the completion of the project, the user documentation will be assessed against the program and the requirements. These tests will ensure that no features are missing, all the documents are synchronized and the contents can be understood easily and unambiguously.
8. Item pass / fail criteria

Each test suite will be considered successful when the results are as specified in the vision document and failure otherwise.

9. Suspension criteria and resumption requirements

In case of failure of a test suite, the failed test cases should be recorded in the test log along with the description for failure. Testing should be suspended and errors fixed unless it requires a design change, in which case the fix can be delayed until the next iteration.

When a new version of the system is available after a bug fix, all previous tests will be rerun to ensure program changes have not inadvertently created regression bugs.

10. Test deliverables

The project will maintain a test log including the pass/fail result of each test, as well as reasons for failure, date, time and a recommended solution.

The project must also maintain a list of the open bugs with an expected date of fix.

11. Test tasks

11.1. Test suite 1: Use Case “Create New Experiment”

11.1.1. Test case 1.1:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 1.1 SR 1.1 Create Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check create a new experiment</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The system must be started and loaded.</td>
</tr>
<tr>
<td>Actions</td>
<td>• The experimenter selects the Train Strategy option.</td>
</tr>
<tr>
<td></td>
<td>• The experimenter inputs the experiment data,</td>
</tr>
<tr>
<td></td>
<td>o Experiment name (string), date (valid date), and obs.</td>
</tr>
<tr>
<td></td>
<td>Data (string)</td>
</tr>
<tr>
<td></td>
<td>• Searching data</td>
</tr>
<tr>
<td></td>
<td>o Search algorithm (select from combo box), search depth</td>
</tr>
<tr>
<td></td>
<td>[1..15], max time [0..60]sec.</td>
</tr>
<tr>
<td></td>
<td>• Learning parameters:</td>
</tr>
</tbody>
</table>
6

### 11.1.2. Test case 1.2:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 1.2 SR 1.2 Select learning technique.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>Experimenter</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>Check the use different learning techniques with the environment.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The system must be started and loaded. The experimenter must create an experiment.</td>
</tr>
</tbody>
</table>
| **Actions** | • The experimenter might select from a list of learning techniques. For the initial version only GAs will be provided  
• The user selects the data button to add the specific data to a learning technique. |
| **Post conditions** | • The system creates a new experiment specific to the learning technique. |
| **Exceptions** | • If the application had an active experiment opened with unsaved changes, the user must receive information about this. |
| **Open issues** | |

### 11.1.3. Test case 1.3:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 1.3 SR 1.3 Select terms for experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>Experimenter</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>Check the use a subset of the terms available for an experiment.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The system must be started and loaded. The experimenter must create an experiment.</td>
</tr>
<tr>
<td><strong>Actions</strong></td>
<td>• The experimenter must select a set of terms to be used for the experiment from the checkboxes of terms in the experiment UI.</td>
</tr>
<tr>
<td><strong>Post conditions</strong></td>
<td>• The system creates a new experiment that uses that set of terms as the terms.</td>
</tr>
<tr>
<td><strong>Exceptions</strong></td>
<td>• If the application had an active experiment opened with unsaved changes, the user must receive information about this.</td>
</tr>
<tr>
<td><strong>Open issues</strong></td>
<td></td>
</tr>
</tbody>
</table>
11.1.4. **Test case 1.4:**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 1.4 SR 1.4 Setup search algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check the use a specific setting for the search algorithms</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The system must be started and loaded.</td>
</tr>
<tr>
<td></td>
<td>The experimenter must create an experiment.</td>
</tr>
<tr>
<td>Actions</td>
<td>- The experimenter must select a set of values to parameterize the search to be used in the experiment</td>
</tr>
<tr>
<td></td>
<td>- Search algorithm (select from combo box)</td>
</tr>
<tr>
<td></td>
<td>- Search depth [1..15]</td>
</tr>
<tr>
<td></td>
<td>- Max time [0..60]sec.</td>
</tr>
<tr>
<td>Post conditions</td>
<td>- The system creates a new experiment that uses the search data just selected.</td>
</tr>
<tr>
<td>Exceptions</td>
<td>- If the application had an active experiment opened with unsaved changes, the user must receive information about this.</td>
</tr>
</tbody>
</table>

11.2. **Test Suite 2: Use Case Load experiment.**

11.2.1. **Test case 2.1:**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 2.1 SR 2.1 Load Experiment SR 2.4 Experiment Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check loading an experiment from a file.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The system must be started and loaded.</td>
</tr>
<tr>
<td></td>
<td>The experiment must exist in a file.</td>
</tr>
<tr>
<td>Actions</td>
<td>- The experimenter selects the train strategy option.</td>
</tr>
<tr>
<td></td>
<td>- The experimenter selects the <em>open experiment</em> option.</td>
</tr>
<tr>
<td></td>
<td>- The experimenter selects the experiment to load.</td>
</tr>
<tr>
<td>Post conditions</td>
<td>- The experiment is loaded into the GUI and the status and results are displayed if any.</td>
</tr>
<tr>
<td>Exceptions</td>
<td>- If the application had an active experiment opened with unsaved changes, the user must receive information about this.</td>
</tr>
<tr>
<td></td>
<td>- There might be a file system error.</td>
</tr>
<tr>
<td></td>
<td>- The file conforms to the experiment file specification</td>
</tr>
</tbody>
</table>

Open issues
11.2.2. Test case 2.2:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 2.2 SR 2.2 Save Experiment SR 2.4 Experiment Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check saving an experiment to a file.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The system must be started and loaded.</td>
</tr>
<tr>
<td>Actions</td>
<td>• The experimenter selects the save option.</td>
</tr>
<tr>
<td>Post conditions</td>
<td>• The experiment is saved into a file.</td>
</tr>
<tr>
<td>Exceptions</td>
<td>• If a required field is not set, the system should display an error.</td>
</tr>
<tr>
<td></td>
<td>• There might be a file system error.</td>
</tr>
<tr>
<td></td>
<td>• The file conforms to the experiment file specification</td>
</tr>
</tbody>
</table>

11.2.3. Test case 2.3:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 2.3 SR 2.3 Edit Experiment SR 2.4 Experiment Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check editing a previously saved experiment.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The system must be started and loaded.</td>
</tr>
<tr>
<td>Actions</td>
<td>• The experimenter modifies some of the fields in the experiment.</td>
</tr>
<tr>
<td></td>
<td>• The experimenter saves the experiment again.</td>
</tr>
<tr>
<td>Post conditions</td>
<td>• The experiment is saved into a file.</td>
</tr>
<tr>
<td></td>
<td>• The attributes of the experiment in the file match the modified values.</td>
</tr>
<tr>
<td></td>
<td>• The file conforms to the experiment file specification</td>
</tr>
<tr>
<td>Exceptions</td>
<td>• If a required field is not set, the system should display an error.</td>
</tr>
<tr>
<td></td>
<td>• There might be a file system error.</td>
</tr>
</tbody>
</table>

11.3. Test Suite 3: Use Case “Train strategy”

11.3.1. Test case 3.1:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 3.1 SR 3.1 Train strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check obtaining a new strategy (set of new strategies) according to the training options and the learning procedure specified by the learning</td>
</tr>
</tbody>
</table>
### Components

**Preconditions**
- The new experiment must be created or loaded and the training options must be set.

**Actions**
- The experimenter selects the *start training option*
- The program starts training with the specified arguments.
- The program creates the directory structure to hold the strategies that are created.

**Post conditions**
- The learned strategy or set of strategies should be identified by the Experimenter using the GUI.

**Exceptions**
- The experimenter might want to stop training.

### 11.3.2. Test case 3.2:

**Test Case** TC 3.2 SR 3.5 Rules of play.

**Actors** Experimenter

**Goal** Check that the games are played using correct rules.

**Preconditions**
- A game is created and the board set up.

**Actions**
- This test case should be tested creating different board situations and checking that the algorithm creates only valid moves and that it creates all the valid moves.

**Post conditions**
- Invalid moves should be identified.
- All valid moves must be created.
- Only valid moves should be created.

**Exceptions**

**Open issues**

### 11.3.3. Test case 3.3:

**Test Case** TC 3.3 SR 3.6 Correct Search.

**Actors** Experimenter

**Goal** Check that the search algorithm behaves correctly.

**Preconditions**
- A game is created and the board set up.
- Search algorithm has all parameters set.

**Actions**
- This test case should be tested creating different starting board situations and analyzing subsets of the tree to check if the algorithm returns valid results.

**Post conditions**
- The search algorithm should return a correct value according to the search algorithm policy.

**Exceptions**

**Open issues**
- The search is time and ply constrained, this should be accounted for when testing the search algorithm.
11.4. Test Suite 4: Use Case “Export strategy”

11.4.1. Test case 4.1:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 4.1 SR 4.1 Export Strategy.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Check the exportation of the strategy in a format that can be used by other programs.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>• A trained strategy must exist saved in the file system.</td>
</tr>
</tbody>
</table>
| Scenario  | • The experimenter selects the export strategy option.  
  • The experimenter selects the strategy to be exported.  
  • The experimenter selects the location where to export the strategy.  
  • The experimenter selects an exporter algorithm. |
| Post conditions | • The strategy is saved into a file using the exporter algorithm file format. |
| Exceptions | • Leaving some of the fields blank should be reported to the user as an error. |
| Open issues |                                      |

11.5. Test Suite 5: Use Case “Play Game”

11.5.1. Test case 5.1:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 5.1 Play Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Test the ability of a program to play games using a UI.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>• At least one trained strategy exists in the framework</td>
</tr>
</tbody>
</table>
| Scenario  | • The experimenter selects the Play/Explore game option.  
  • The experimenter sets up the game information.  
    o Level [1,2, 3]  
    o Team that starts [1,2]  
    o Max time per play [1..60] sec  
    o Team Name (String)  
    o Type of agent (Human, Computer)  
  • The experimenter sets up the search tree information.  
    o Search algorithm {loaded at runtime}  
    o Max depth [1..15] |
11.5.2. Test case 5.2:
See TC 3.2 Rules of play.

11.6. Test Suite 6: Use Case “Explore Game”

11.6.1. Test case 6.1:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>TC 6.1 SR 6.1 Explore Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Experimenter</td>
</tr>
<tr>
<td>Goal</td>
<td>Allow the experimenter to scroll through moves of a game or to review games played during training.</td>
</tr>
</tbody>
</table>
| Preconditions | ● At least one trained strategy exists in the framework  
|           | ● The game to be explored must be opened in the program. |
| Scenario  | ● The experimenter scrolls through the moves of the game. |
| Post conditions | ● The board gets updated when selecting the previous or next move. |

12. Schedule

Testing should be done throughout the development cycle. A schedule for the testing activities is given in the Project Plan [2]