Testing

When developing a large project one of the most difficult aspects of the process can be proper testing, this section will go through what kinds of tests we performed to ensure that our engine works correctly.

# Testing of engine

**Unit test:** The location of the unit tests for the engine and all its depencies can be found in theSource code in the ##XmasEngine\_Test and ##JSLibrary\_Test projects

Since our engine by itself is not meant to be executed, but rather is meant to have some of its components implemented first. This means that the only kinds of tests that can be performed on the engine are unit tests. The unit tests that are most important to the core functionality of the engine, is the tests concerning the ##EventManager and the ##ActionManager, since these tests shows that the action, trigger and event functionality correctly works. However as we created the engine using a TDD approach we should have tests for almost all classes with business logic contained in them.

One component proved impossible to properly unit test and that was the EIS agent controller, since this component required a connection to properly understand its errors. To perform this test we designed a simple component/functional test, this test is not meant to be executed as part of the unit tests. The test is designed to be executed with an debugger so that the programmer can easily follow if any errors occur during the run.

# Testing of reference implementation

The testing of our reference implementation hinges on the fact that we assume the engine works correctly, thus it is the job of engine-tests to ensure correctness and not our reference implementation. Therefor as the reference implementation complicated logic was located as part of the extensions, which were already tested as part of testing the engine. The only testing that the reference implementation needed was testing of the Goal program which is controlling the agent.

The goal program was tested by taking out individual parts of it such as its path finding, and carefully tested and debugged in the SWI-prolog program. When all parts worked correctly we moved them to the actual goal program and then made a larger scale test of the goal program using the fully running engine. Once the agent correctly had located all packages and stopped as we wanted, we concluded that the reference implementation is working as it should.