Test Plan for TicTacToe

1. Introduction

   This plan specifies the tests that Software Architects intends to execute to investigate the correctness of the Tic Tac Toe application. The application is distributed and consists of four major threads of activity executing on as many as four different platforms.

2. Test Items

   The three major processes that comprise the application are the major test items. In addition, the Game Board, which is a Java Bean, deserves special attention because of the special characteristics of the standard.

3. Tested Features

   All of the current features of the application will be tested. In addition, the APIs of the major classes will also be tested.

4. Features Not Tested

   None

5. Testing Strategy and Approach
   1. Syntax

      DPACT abstract classes are used as the base classes for developing test classes. The methods correspond to test suites, scripts and cases.

   2. Description of Functionality

      At the system level, the functionality is an electronic version of the TicTacToe game.

      Each class provides a specified functionality that is described in its documentation.

   3. Arguments for tests

      Mouse clicks

   4. Expected Output

      It is expected that the appropriate symbol will appear in the selected slot on both Game Boards. When the game completes in either the win/lost or tied states the appropriate message is printed on the Game Board.

5. Specific Exclusions

   None

6. Dependencies

   The DPACT test classes inherit from the DPACT base classes.

   Each DPACT class uses reflection to access the internals of the objects that comprise the TicTacToe game.

7. Test Case Success/Failure Criteria

At the system test level, test cases result in one of three terminal states for the game: exited, won/lost and tied.

6. Pass/Fail Criteria for the Complete Test Cycle

   Every primary use case must be successful for the application to pass. If any test associated with a primary use case fails, the system test fails.

7. Entrance Criteria/Exit Criteria

   Class testing will be conducted on a continuing basis during development. Integration testing will begin when at least one complete protocol between two classes has been developed. System testing will be initiated when developers/integration testers certify that a game has been completed.

8. Test Suspension Criteria and Resumption Requirements

   Tests will be sequenced to exercise progressively more of the system. Testing will be suspended when program faults prevent the tester from moving further into the system.

9. Test Deliverables/Status Communications Vehicles

   The DPACT-based test classes and the test report are the primary deliverables. The test report will be web-based so that developers can track progress of the testing effort and begin to repair the application before the testing process is complete.

10. Testing Tasks

    The primary tasks are:

    1. Test case selection
    2. Test class construction
    3. Test execution
    4. Test result evaluation
    5. Test report development

11. Hardware and Software Requirements

The software can be run on either unix or Windows based systems. The tests are derived from the DPACT framework.

12. Problem Determination and Correction Responsibilities

    The tester is responsible for completing a test report that identifies those tests that have resulted in program failure. The developer is responsible for addressing each deficiency identified in the test report.

13. Staffing and Training Needs/Assignments

    Fortunately all of our staff are experienced game players and need no further training to use the application. However, they do need training on calculating the possible combinations of moves.

14. Tests Schedules

    Testing will be completed by 6pm April 19th.

15. Risks and Contingencies

    There are a large number of combinations, (2^8*1 = 256) of moves that can be made by the two players. There is a risk that a few of these combinations will be repeated because they are the "natural" sequences. Many combinations will not be tested at all. This risk can be mitigated by writing down all the combinations and using a specific one in each test.

16. Approvals

    This plan has been reviewed and approved by the development team leader and the QA department representative.