ID2209 Distributed AI and Intelligent Agents: Homework 3

avneesh@kth.se, gaiddon@kth.se (Group 6)

November 29, 2016

Task 1: N-Queens problem using N agents representing the queens

Design Notes

1. N jade agents each representing a queen were started. Each queen works on a particular column of the chess board. (Column number and total columns passed as arguments to each agent)
2. The queen agents do not start finding a safe position until every agent knows its predecessor and successor queen. Achieved by subscribing to the yellow pages for agents with specific Ids.
3. Queen agent with Id '0' (thus maintaining column 0) starts a 'step' behavior and finds a safe position. In the first column all positions are safe.
4. The column after finding the safe positions, forwards its positions and all its successor positions to its successor agent.
5. The successor agent then chooses a position in its column and calculates the slope with all the predecessor positions. A position is unsafe if slope is 1, -1, 0 or undefined. If the agent is able to find a safe position, it adds its own position to the position vector and forwards it to its successor. If it does not find a safe position, the agent requests it to predecessor to change its position.
6. The process continues until the last agent is able to find a safe position. Once that happens, it informs all predecessors to finalize their position.
7. As a final step, a chess board agent queries the finalized position from each agent draws the chess board.
8. To allow different solution results, the first safe position in the first column is randomly selected.

Protocol Used

During the development, JADE's FIPA based ACL message passing APIs were used. Chess board used `SimpleAchieveREInitiator` behavior to process responses from `queen agents.`

Test Instructions

1. Download the source code and create an eclipse or netbeans project into the nqueens_jade directory.
2. Add jade.jar as external jar into the project build path

3. Go to queens/TestMain.java and modify following variable's value to set the chess board size (for example, 8 for an 8X8 board, 10 for a 10X10 board etc….)
4. Run queens/TestMain.java as a java application. This will launch the queen agents and the finally the chess board agent to print the output on the screen.
5. The console output will also display the various interactions between queen agents to reach to the final solution.