KTH- ID 2209 – Distributed Artificial Intelligence and Autonomous Agents

Course Coordinator: Mihhail Matskin
Teacher Assistants: Shatha Jaradat & Lorenzo Corneo

HOMEWORK 3 – Group 6: Alessio Russo, Mumtaz Fatima

Task1

The solution for task1 is based on a recursive algorithm, the backtracking algorithm, which can be found at Math 188, Winter 2001, Prof.Tesler.

The backtracking algorithm is a recursive algorithm that is based on the following rationale: first we notice that we can only think of rows or columns without considering both, thus the problem reduces of a factor n. Based on that the rationale is “if in the current row we can’t place a queen, go back to the previous row and move of one place”.

The problem of implementing this behaviour in a multiagent system is hard, first we need to implement the recursion which is easily done by using a stack object that memorizes all the steps.

The protocol is as follows:

1. Each agents move based on its id. The lower the id, the sooner it acts.
2. When an agent makes a move, it's sent to all the other agents with a multicast message
3. The move is added to the stack, the id of the current player is updated
4. The agents receive the msg, if the new id of the current player is their id they make a move, otherwise wait
5. If a move can't be made, a msg of undo is sent, the id of the currenplayer reduced of one.
6. The agent with higher id has the duty to determine if the game is over

The line argument used to run the program is the following one:

```
-gui -agents
"Q0:hw3task1.QueenAgent(0,4);Q1:hw3task1.QueenAgent(1,4);Q2:hw3task1.QueenAgent(
2,4);Q3:hw3task1.QueenAgent(3,4);"
```

For 5 agents:

```
-gui -agents
"Q0:hw3task1.QueenAgent(0,5);Q1:hw3task1.QueenAgent(1,5);Q2:hw3task1.QueenAgent(
2,5);Q3:hw3task1.QueenAgent(3,5);Q4:hw3task1.QueenAgent(4,5);"
```

etc...

Task2

1. Use of a third agent, in the main container to create the containers - new agents
    a. Problems at using the command line
    b. easy to implement and used only at the beginning, does not use resources
2. An initial Seller: S1 in the auctioneer container
    a. clones (S2) and move them to Museum1,mUseum2
3. An initial Agent AG1 in museum1 and AG2 in museum2
    a. Both clones an agent that moves to the other museum
4. Auction is executed locally
5. S1,S2 go back to their home container, contact each other and decide the best price (the higher the better)

To run the project we just need to execute the third agent:

```
-gui -agents ext:hw3task2.Simple
```