ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Homework 3

Course Coordinator:
Mihhail Matskin

Course Assistant:
Cosar Ghandeharioon
Shatha Jaradat

Óttar Guðmundsson

Xin Ren

2017-11-21

Task1

How to run

Import the folder as a new java project in Eclipse.

Recommend: Start a new agent platform and create a new stuff.Container. Pass an integer argument to create an NxN Chessboard. After 5 seconds, it will add a new Queen on a 5 seconds interval and ultimately provide a solution.

Optional: Start a new agent platform and create a new Chessboard with an integer argument that determines the size of the chessboard. Default value is 8. Then, start a new Queen with integers parameters N,X – N being the integer passed to the chessboard and ID being the X position of the queens, ranging from 1 to 8.

Chessboard

The Chessboard starts up by deciding how big the board should be based on an argument. It registers to the DF agent so it can keep track of the queens on the board that should be displayed. It then creates a GUI of a chessboard that is similar to an NxN array and two buttons. *Refresh image* simply redraws the chessboard and then *Request another solution* which sends an inform message to the last queen about reordering for another solution.

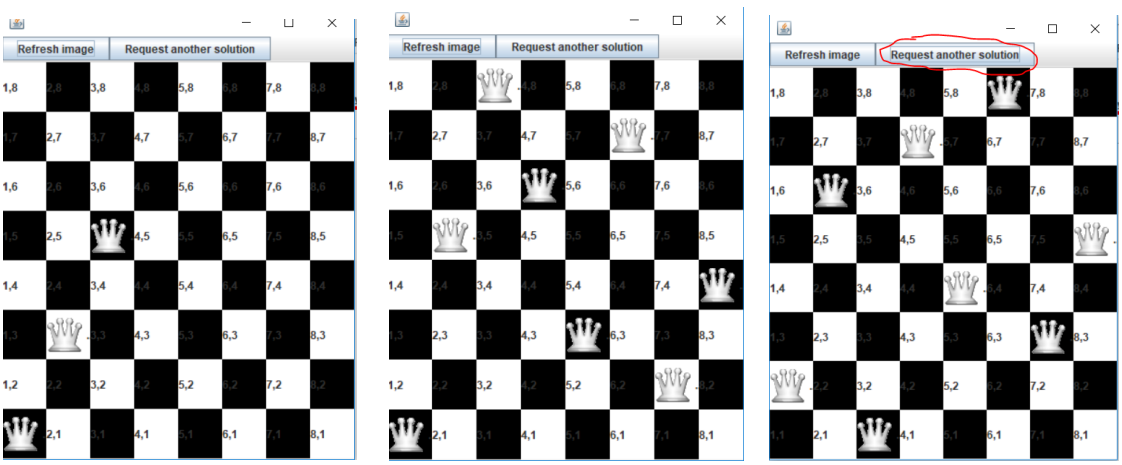
Queen

A Queen starts up with two integers as argument, the former one being the size of the chessboard and the latter is the x position of the chessboard. It keeps track of its current position and the list of the other queens that have already been added. When it starts it adds all possible y positions to an array. It then receives a message from the queen that was added before it in the form of "1,1_2,2.." which represents all positions already taken. The queen then calculates all possible positions based on the string received, and places itself in the first available save position. When it is done, it concats its own position to the message string and forwards it the chessboard, which then draws all the queens. If there is no position available, it sends an abort message to the previous queen and asks it to relocate itself to another position. If that queen can relocate, it does so and notifies the next queen about it. If it cannot relocate, it sends to its previous queen and so on, until a save location is found. Thus, the message is traversed back to a queen that has an extra available position.

Other classes

Controller

A simple class that creates a new Chessboard on the fly and Queens on a specific interval. Used for debugging and demonstration, since manually inserting the input can be costly and frustrating.



Task2

How to run

Import the folder as a new java project in Eclipse. Start a new agent platform create a new clone.Container with no arguments.

Container

The container agent starts with fetching the current main container and the current destination. It then creates a new Auctioneer and then two Buyers (with no arguments) in two separate containers and starts every agent. To see how the container runs, read through *task2_log.txt*

Auctioneer

The auctioneer is initialized just as it did on homework 2, except that the most of the functions is done in the afterMove function. This is done to reset the auctioneer every time it switches containers to only get the highest prices from the buyers in that container. When it has finished traveling between all possible containers created by the original Container, it will write out the highest price found in the console.

Buyer

The buyers behave just as they did in homework 2, except that most of the former initialization is copied to the afterClone function.

Other classes

The other classes are the same that were used in homework 2.