G54DIA: Designing Intelligent Agents

Lecture 13: Coursework 2 Description

Natasha Alechina
School of Computer Science
nza@cs.nott.ac.uk

Outline of this lecture

- coursework outline
- minimal requirements
- project resources and tutorials
- assessment
- any questions …

Project outline

- Coursework 2 involves designing and implementing a *multi-agent system* consisting of at least 2 agents
- same setting as before: environment with wells and stations, stations generate tasks, but now you need to develop a team of tankers delivering water
- Java '*multi-demo*' package provided as a starting point (and the stations array list is not available to use anymore)

The problem

- task consists of collecting and delivering water to *stations* (customers)
- environment contains a number of stations which periodically generate *tasks* – requests for a specified amount of water
- environment also contains a number of *wells* from which water can be collected
- the goal of the agents is to deliver as much water to as many stations as possible in the time available
- the score is as before, *but divided by the number of agents*

The objective

- objective is to investigate agent architectures and coordination mechanisms for water collection and delivery strategies, to decide
  - which agent architecture(s) to use given the task environment
  - specialised agents vs homogeneous
  - deciding how to allocate tasks to agents

- aim is not just to build a team that works well, *but to understand why it works well*

Main problem

- adding multiple agents introduces a new problem of *coordination*
    - how should agents explore the environment, and what information should they share with other agents
- how do agents decide which tasks to perform and how to (who should) perform them so that
    - two agents don't try to perform the same task, and the largest number of tasks are achieved

Project requirements

- a standard task environment is given as part of the coursework

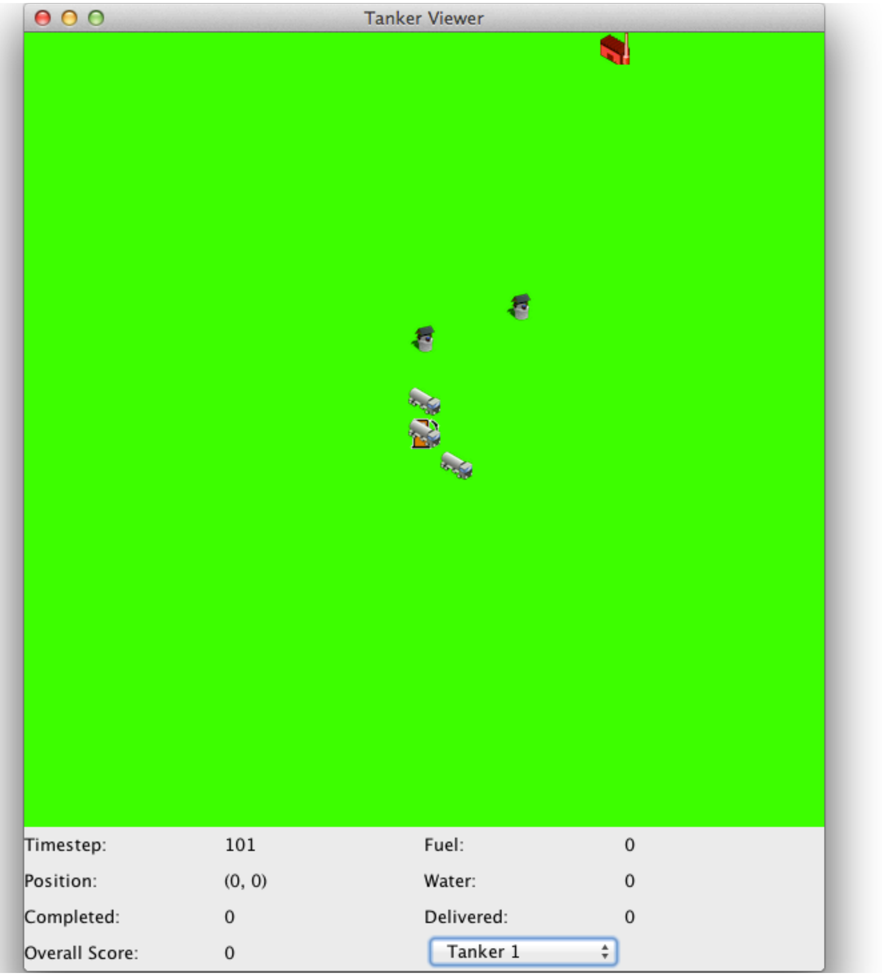
- you *must* run your agents in this task environment for purposes of comparison

- you must implement at least 2 agents and include in your report data of the team’s performance

Project resources

- *Java demo agent package* (Package uk.ac.nott.cs.g54dia) with multidemo and multi-library
  - a simple environment containing randomly distributed stations/ tasks and wells
  - an example "fleet" of Tankers (not a complete solution)
  - GUI allows you to center the display on a particular Tanker

Multi-demo environment



Timestep:	101	Fuel:	0
Position:	(0, 0)	Water:	0
Completed:	0	Delivered:	0
Overall Score:	0	Tanker 1	

Assessment

- **submission** containing a report (pdf) describing your agents and the associated code – due *Monday 30th of March at 23:55*

- constitutes 50% of the assessment

- assessment will be based on the content and clarity of the report and the capabilities of the implemented system

Getting started

- modify the multi-demo example (or write your own Tanker and Simulator classes)
- keep in mind that in your report you will need to classify your system and argue for why you have chosen a particular organisational structure and made the system redundant (or not), specialised (or not) – see Lectures 10 and 11
- you will also need to explain in your report how coordination and task allocation are achieved

Tutorials

- Tuesday 10 March at 9:00 group tutorial
- Tuesday 10 March at 11:00 individual tutorials
- individual tutorials in times of lecture slots, until
- Tuesday 17 March at 9:00 feedback on coursework 1 and SET/SEM

Questions …