

# G54DIA: Designing Intelligent Agents

## Lecture 12: Coursework 2

Brian Logan  
School of Computer Science  
[bsl@cs.nott.ac.uk](mailto:bsl@cs.nott.ac.uk)

# Outline of this lecture

- description of the coursework
- project resources
- some hints to help get you started
- tutorial arrangements

# Second coursework

- the second coursework involves the design and implementation of a multi-agent system
- comprises 50% of the assessment
- assessed by a report describing your agents and the associated code
- submissions due **Wednesday 23<sup>rd</sup> of March**

# Coursework outline

- Coursework 2 involves designing and implementing a *multi-agent system* consisting of **at least 2 agents**
- task environment is as before: environment contains wells and stations, stations generate tasks
- but now you need to develop a team of tankers delivering water
- Java ‘*multidemo*’ package provided as a starting point (also includes some bug fixes)

# 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 multi-agent architectures and coordination mechanisms for water collection and delivery strategies:
  - 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*

# The coordination 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

# Coursework requirements

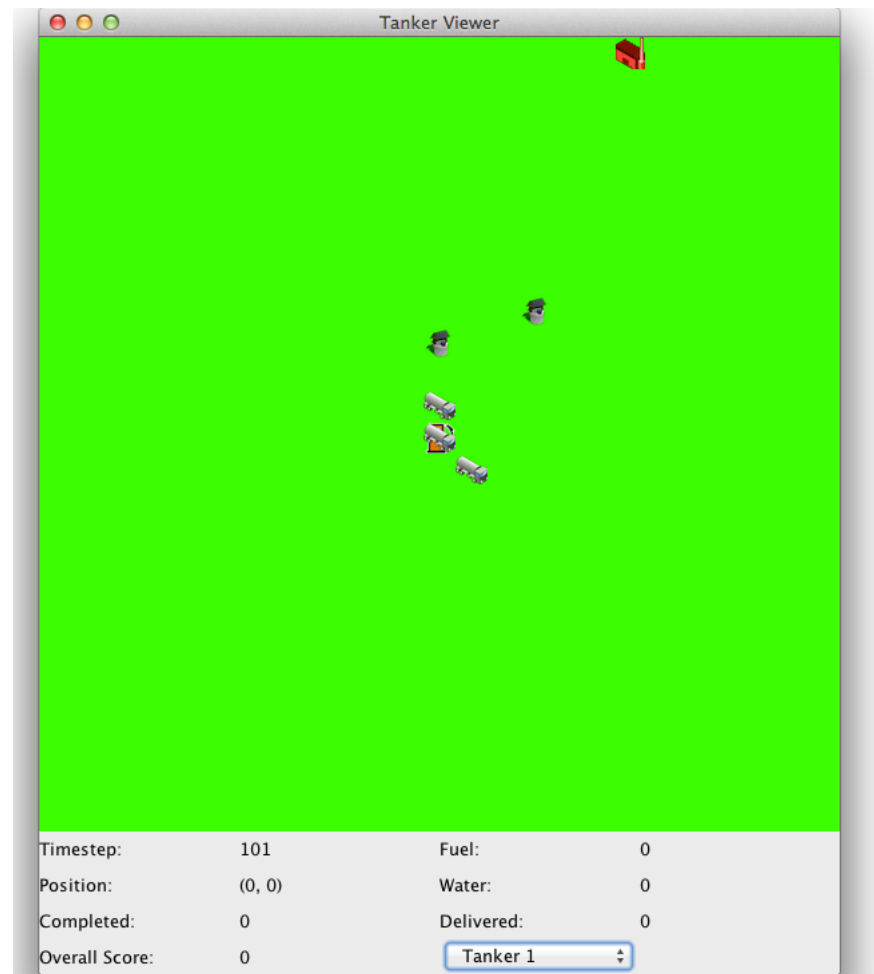
- a standard task environment is given as part of the coursework
- you *must* run your agents in this task environment for purposes of comparison
- you must implement a team consisting of at least 2 agents and include in your report data of the team's performance



# Project resources

- *Java multi-agent demo package* (Package uk.ac.nott.cs.g53dia) with multidemo and multilibrary
  - a simple environment containing randomly distributed stations/tasks and wells
  - an example “fleet” of Tankers (not a complete solution)
  - GUI allows you to centre the display on a particular Tanker

# Multi-demo environment



# Assessment

- **submission** containing a report (pdf) describing your agents and the associated code
- **due** Wednesday 23<sup>rd</sup> of March (midnight UK time)
- 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 multidemo examples (or write your own DemoTanker and DemoSimulator 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

- feedback on Coursework 1 and SET/SEM: 11:00 on the 11<sup>th</sup> March
- individual tutorials:
  - 11:00 & 15:00 on the 7<sup>th</sup> of March
  - 15:00 on the 8<sup>th</sup> of March
  - 11:00 and 15:00 on the 14<sup>th</sup> of March
  - 15:00 on the 15<sup>th</sup> of March
  - 11:00 on the 18<sup>th</sup> of March

# Questions ...