

G54DIA Coursework 1 Description

The first coursework involves the specification, design and implementation of a simple agent.

Coursework requirements

The problem consists of a 2D environment, in which a single agent must collect and deliver water to *stations* (customers). Stations periodically generate *tasks* – requests for a specified amount of water. The environment also contains a number of *wells* from which water can be collected. The goal of the agent is to deliver as much water to as many stations as possible in a fixed period of time.

Task environment

The standard task environment is defined as:

- the environment is an infinite 2D grid that contains randomly distributed stations and wells
- stations periodically generate tasks – requests for a specified amount of water
- tasks persist until they are achieved (a station has at most one task at any time)
- the maximum amount of water required by a station is 10,000 litres
- wells contain an infinite amount of water
- there is a single fuel station in the centre of the environment that contains an infinite amount of fuel
- a run lasts 100,000 timesteps
- an agent's visual range is limited and it can only see stations and wells within 12 cells of its current position
- if a station is visible, the agent can see if it has a task, and if so, how much water is required
- agents can remember the location of stations, tasks and wells and can accurately navigate to a remembered location
- the agent can take water from a well and deliver it to a station
- moving around the environment requires fuel, which the agent must replenish at a (single) fuel station
- the agent can carry a maximum of 100 litres of fuel and 10000 litres of water
- the agent starts out in the centre of the environment (at the fuel station) with 100 litres of fuel and no water
- the agent moves at 1 cell / timestep and consumes 1 litre of fuel / cell
- filling the fuel and water tanks and delivering water to a station takes one timestep
- if the agent runs out of fuel, it can do nothing for the rest of the run
- the success of an agent in the task environment is determined by its score
- the agent's score is determined by the amount of water delivered \times number of (completed) deliveries

The task environment should not be modified or extended. All other decisions regarding software design and implementation strategy are up to you. However you will be given guidance and feedback on your project in individual tutorials.

You must implement an agent that completes the task in the specified task environment, and must include in your final report an evaluation of the performance of your agent (average score over at least ten runs).

Resources

A Java demo agent package (package uk.ac.nott.cs.g54dia) is provided as a starting point for the project work. This provides an implementation of the standard task environment and a very basic agent that chooses actions at random.

The project work is supported by individual and group tutorials. The individual tutorials cover the design and implementation of your agent. The group tutorials cover the use of the Java demo agent package.

Assessment

Assessment is by means of project work and reports.

What is required

- A report describing the implemented agents and the associated source code
 - The report should document the specification, design and implementation of the agents, and should include a summary of any background material or code used in the project. The report should not exceed 5000 words excluding references.
 - The code implementing the agent.

Coursework 1 constitutes 50% of the assessment.

How it will be marked

Marking will be based on:

- the capabilities of the implemented agent system, including the quality of the specification, design and implementation;
- the degree to which the specification, design and implementation are clearly documented in the report, including any relevant background material used in the design and implementation of the agent; and
- clarity of presentation in general (including grammar, spelling and punctuation).

Extra credit will be given for submissions that demonstrate a clear understanding of the relationship between the specified task environment and the architecture of the implemented system.

Assessment criteria and typical requirements for specific grades are provided (see attached).

Submission dates

Submissions are due on Friday the 27th of February.