

# G54DIA: Designing Intelligent Agents

CW1 report

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

# The rest of the module

- Group tutorial (and probably time for some individual tutorials) today
- individual tutorials on Friday 15:00-16:00
- **CW1 submission** describing your agent and the associated code – due Monday the 2<sup>nd</sup> of March at 12:00 (slightly extended deadline)
- From 3<sup>rd</sup> of March: lectures on multi-agent systems resume.
- CW2 announced 6 March, deadline 30 March

# Final coursework submission

- **report** *in pdf format* describing your agent—should not exceed 5,000 words (excluding references)
- **code** implementing your agent (in zip format). The agent code should work with the Environment provided in the coursework package.
- submissions should be made electronically via the G54DIA moodle page
- **submissions due 2/03/2014 at 12:00**

# Possible report outline

- your name, email address, student id and “G54DIA final report”
- introduction
- relevant background material
- specification
- design
- implementation
- evaluation
- discussion/conclusions
- references

# Project specification

- the *project specification* states the agent task environment
  - properties of the task and environment
  - percepts and actions available to an agent
- the specification shouldn't say *how* an agent does things, only *what* its environment is like and what it does
- this section can be very short

# Software design

- the *software design* explains how your agent work:
  - what sort of architecture the agent has—how percepts are connected to actions
  - the algorithms used for searching for tasks, deciding which water sources to use etc.
- it should *not* be a list of classes and methods

# Documenting your design

- you need to describe your design and the *reasons for each design decision* clearly in your report
- a good approach is first to say which general type of architecture your agent has (and why)
- then explain the main components or steps in its operation in outline
- then describe each component or step in detail

# Software implementation

- high level description of the implementation:
  - which data structures were used
  - how the algorithms were implemented etc.
  - why the approach adopted was chosen
- try to focus on the ‘interesting’ bits of the implementation
- a full code listing is *not* required

# Evaluation & discussion

- how well does your agent work, e.g.:
  - what score does it achieve (on average)
  - how does this compare to other agents (e.g., simpler versions of your agent)
- why is your solution appropriate for the task environment
- which features of the task environment are critical – how would you expect it to perform in other task environments?

# How it will be marked

- aim of the module is to understand the relationship between an agent's task environment and its architecture
- to do well, you need to develop an agent that works well, and demonstrate that you *understand why they work well*
- marking is therefore based on:
  - the capabilities of the implemented agent, including the quality of the specification, design and implementation
  - the degree to which the specification, design and implementation are clearly documented in the report
  - clarity of presentation in general (including grammar, spelling and punctuation)

# Assessment guidelines

- very broadly, a basic implementation of the minimal requirements (and corresponding report) would gain a pass mark
- 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
- this does not necessarily involve implementing one of the extensions— it is possible to get a first class mark by doing an excellent implementation of the minimal requirements and an excellent report
- full assessment guidelines on the module web page