G54DIA: Designing Intelligent Agents

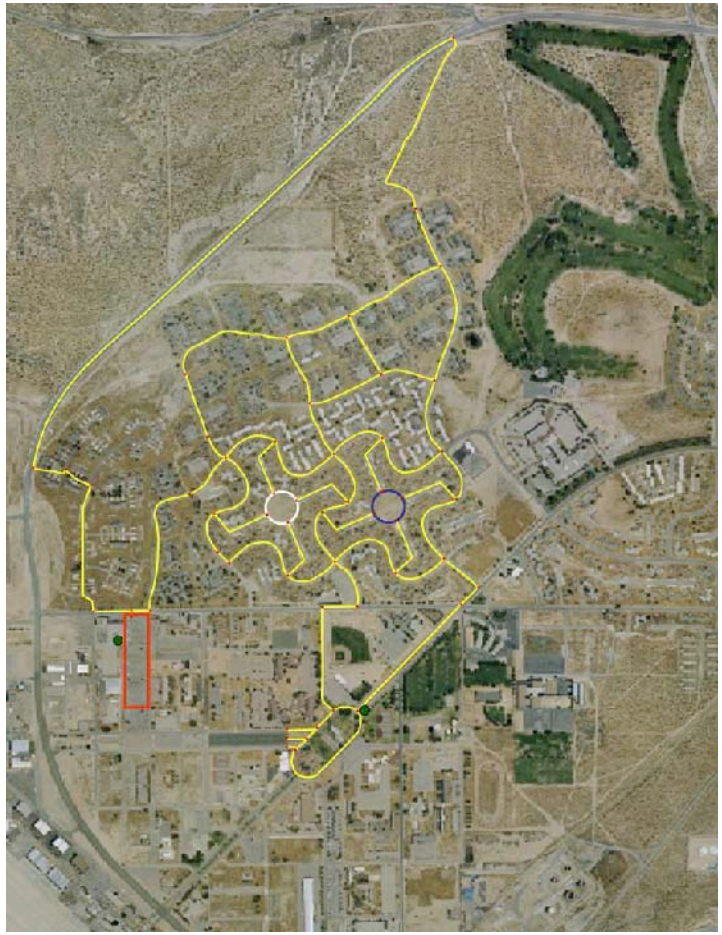
Lecture 1: Introduction

Brian Logan
School of Computer Science
bsl@cs.nott.ac.uk

Outline of this lecture

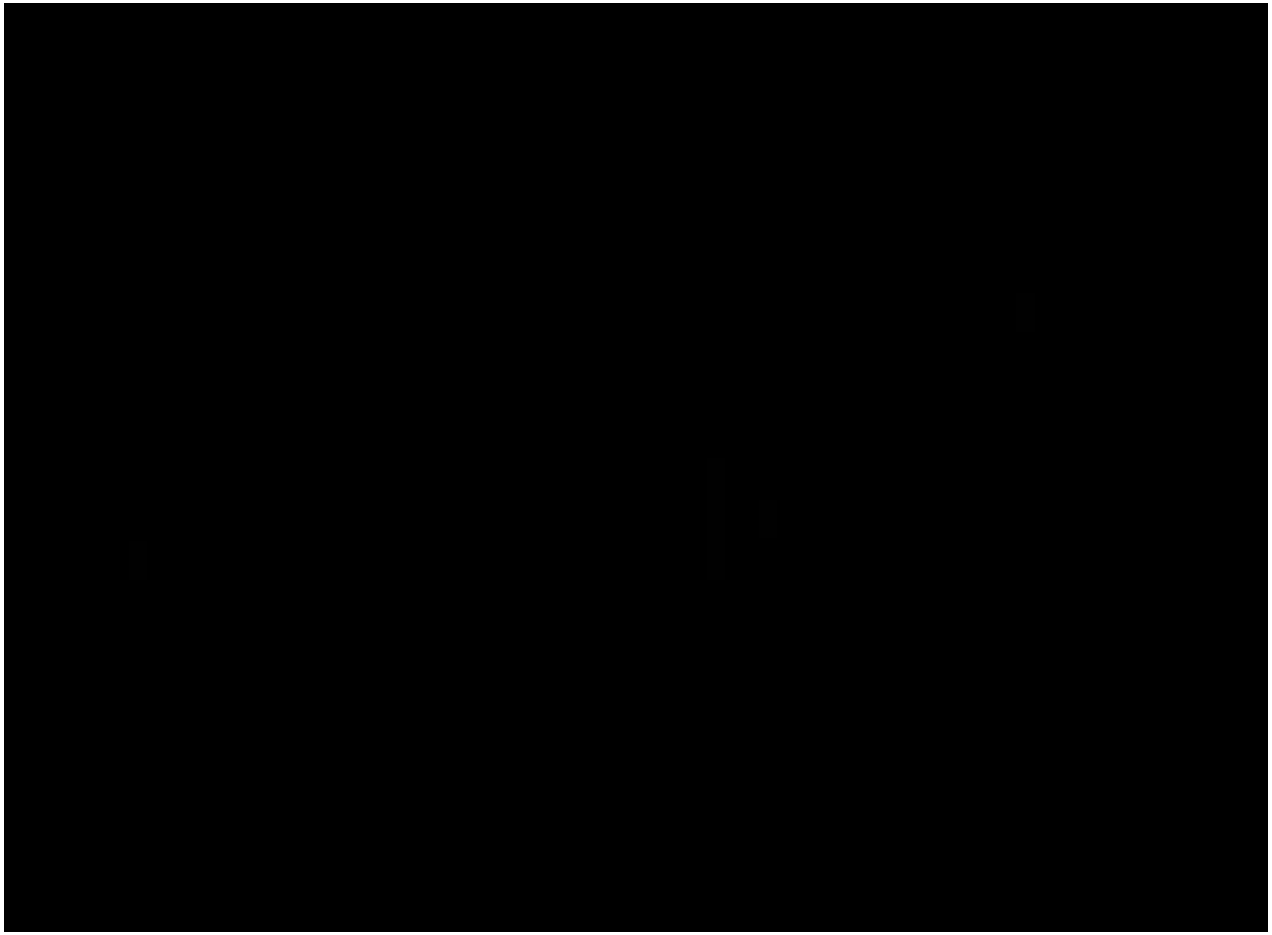
- what is an agent
- why study intelligent agents
- designing intelligent agents
- outline of the module

DARPA 2007 Urban Challenge



- build an *autonomous vehicle* capable of driving in normal traffic conditions
- vehicles must be able to negotiate intersections, make left (i.e., right) turns, merge with traffic, reverse, park etc.
- all traffic regulations must be obeyed
- 50 other cars (with human drivers) on the ‘urban’ course
- six hours to complete a list of five (previously unseen) tasks
- $2,000,000 prize … so how do we build one?

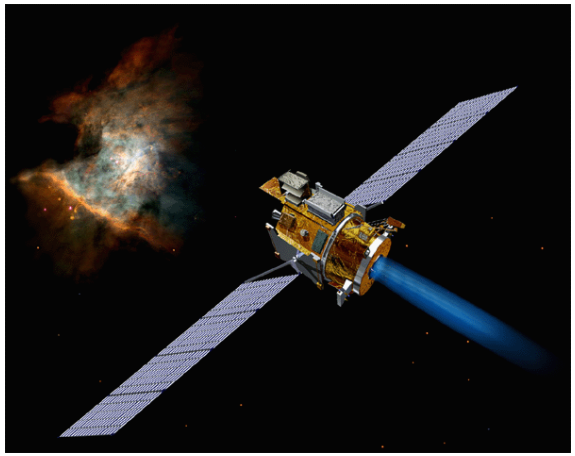
DARPA 2007 Urban Challenge



Example agents: robots

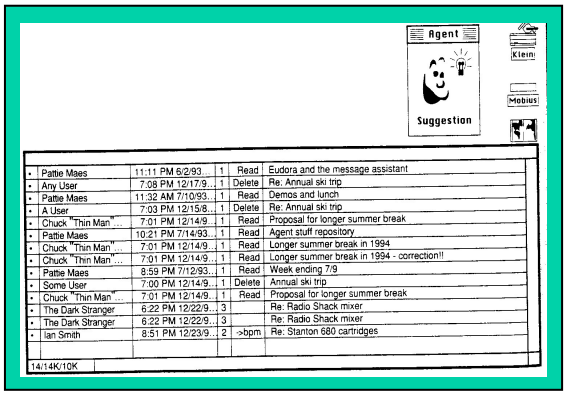
- **Genghis:** a six-legged walking robot

- **Deep Space 1:** an agent-controlled NASA space probe



Example agents: softbots

- **Maxims:** a software agent that learns what to do with different types of email message by watching the user



- **Quakebot:** an agent that plays Quake and can beat most human players

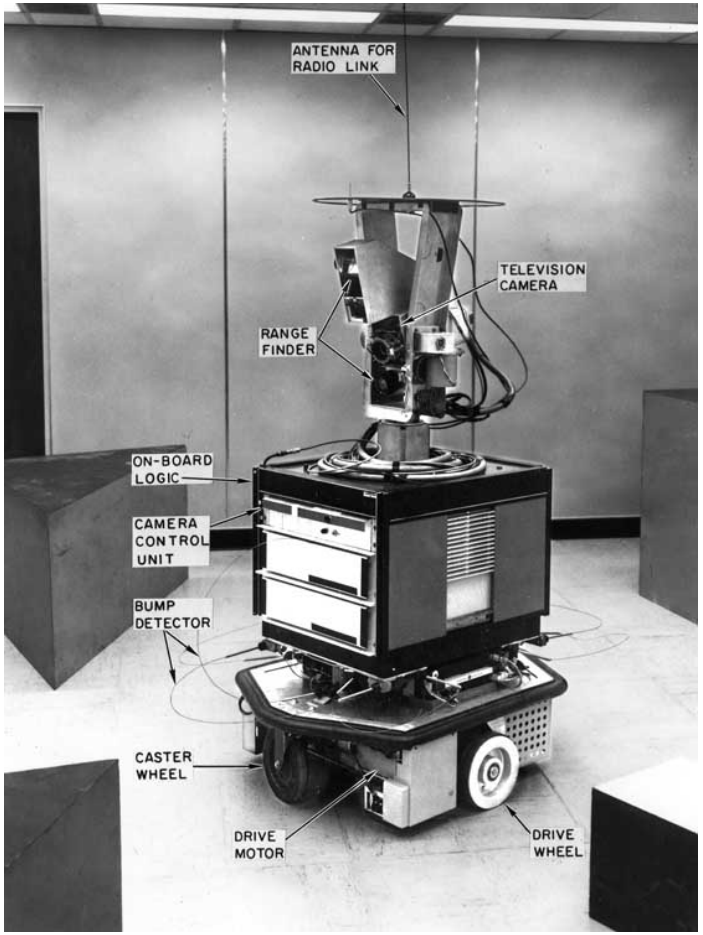
A selective history of AI

- the understanding of intelligent behaviour in animals, humans and artificial systems was the original and is the ultimate goal of AI

- early AI projects combined several capabilities, such as sensing, problem-solving and action, in a single system

Shakey the robot (1966–1972)

- Shakey was the first mobile robot to reason about its actions
- multiple sensors (TV camera, a triangulating range finder, and bump sensors)
- connected to DEC PDP-10 and PDP-15 computers via radio and video links
- programs for perception, world-modeling, and acting (simple motion, turning, and route planning)



Fragmentation of AI

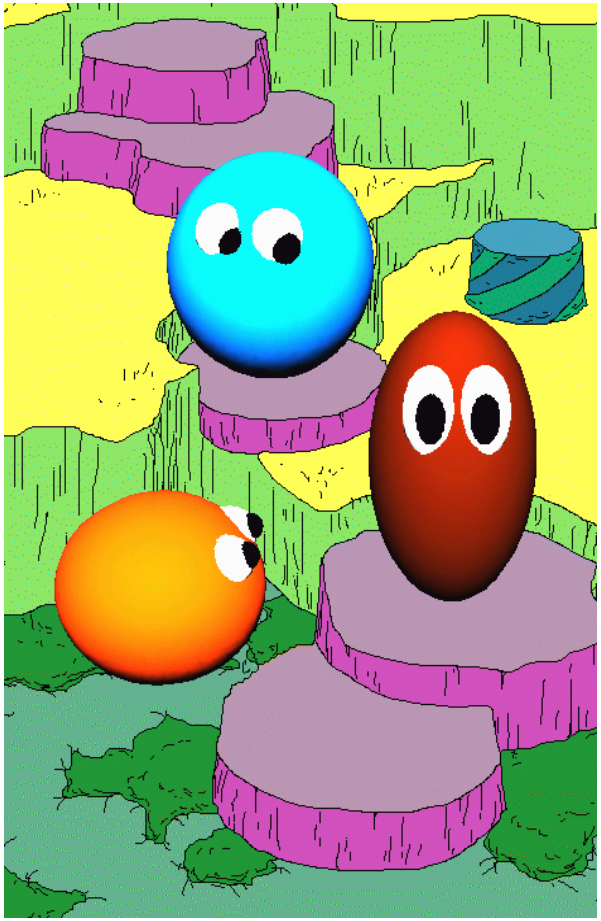
- from the 1970's AI fragmented into sub-disciplines each looking at a small part of the overall problem of intelligence, e.g.:
  - vision
  - natural language processing
  - planning
  - knowledge representation
  - and many others …
- while a lot of good work has been done there are dangers in this approach

The ‘whole iguana’

- in addition to work on sub-problems there is a need for work on *complete systems*:
  - the components of the system may make incompatible assumptions
  - we may end up solving the wrong problem, e.g., the ‘scene understanding problem’ in vision
  - the ‘homunculus problem’
- more importantly, at some point we have to understand how all the various bits fit together.

Intelligent agents

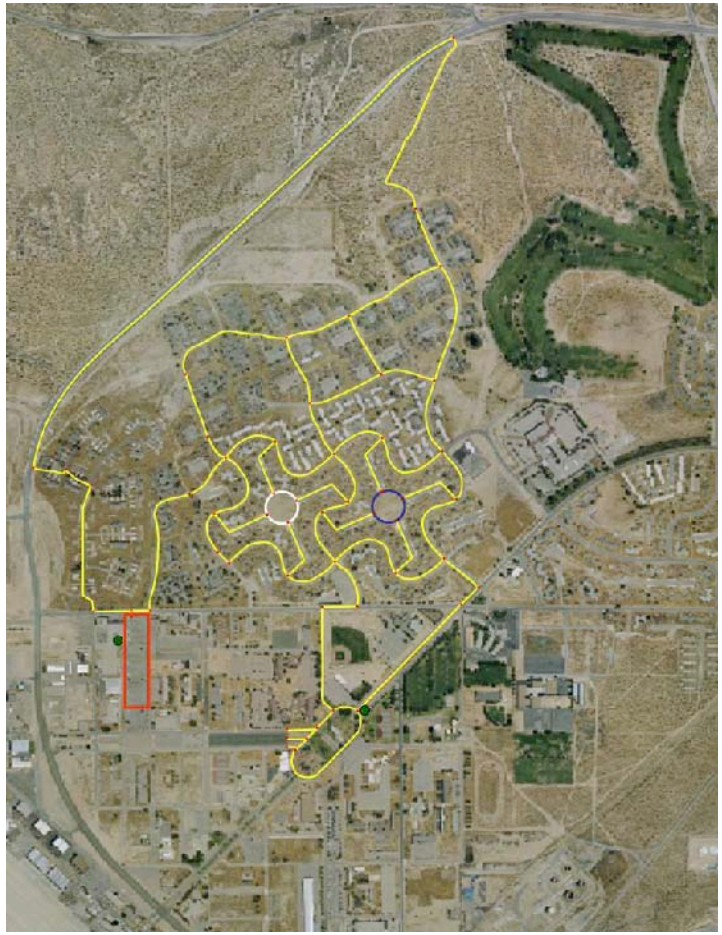
- an *agent* is a complete system which integrates a range of (often relatively shallow) competences
- e.g., the Oz project at CMU developed a range of 'Broad Agents' which integrated:
  - goals and reactive behaviour
  - emotional state and its effect on behaviour
  - natural language
  - memory and inference
- in artificial creatures called 'Woggles' capable of participating in simple childrens' stories



Xavier the robot (1993-2003)

- Xavier is an office delivery robot
- picks up and delivers post, faxes and printouts, returns library books, recycling cans, getting coffee, telling jokes
- determines the order in which to visit offices, plans a path from the current location to the next office to be visited, and follows the path reliably, avoiding static and dynamic obstacles
- responds to commands from a Web interface

DARPA Urban Challenge (2007)



- build an *autonomous vehicle* capable of driving in normal traffic conditions
- vehicles must be able to negotiate intersections, make left (i.e., right) turns, merge with traffic, reverse, park etc.
- all traffic regulations must be obeyed
- 50 other cars (with human drivers) on the ‘urban’ course
- six hours to complete a list of five (previously unseen) tasks

Methodological issues

- working on complete systems (agents) also introduces interesting *new* problems in AI:
  - **design trade-offs:** one part of the system can be made simpler if another part is more complex
  - **integration issues:** how to fit the various capabilities together–is a decomposition which mirrors the traditional sub-fields of AI the right one?

Pylyshyn

"[there is a] growing feeling that the study of intelligence cannot be decomposed along such traditional lines as, say, perception, learning, memory, comprehension, reasoning, motivation and so on. … This does not mean that we cannot study problems in these areas, but rather that categorizing them in such terms may not reveal (indeed may cover up) the way in which they individually and collectively contribute to producing intelligent behaviour."

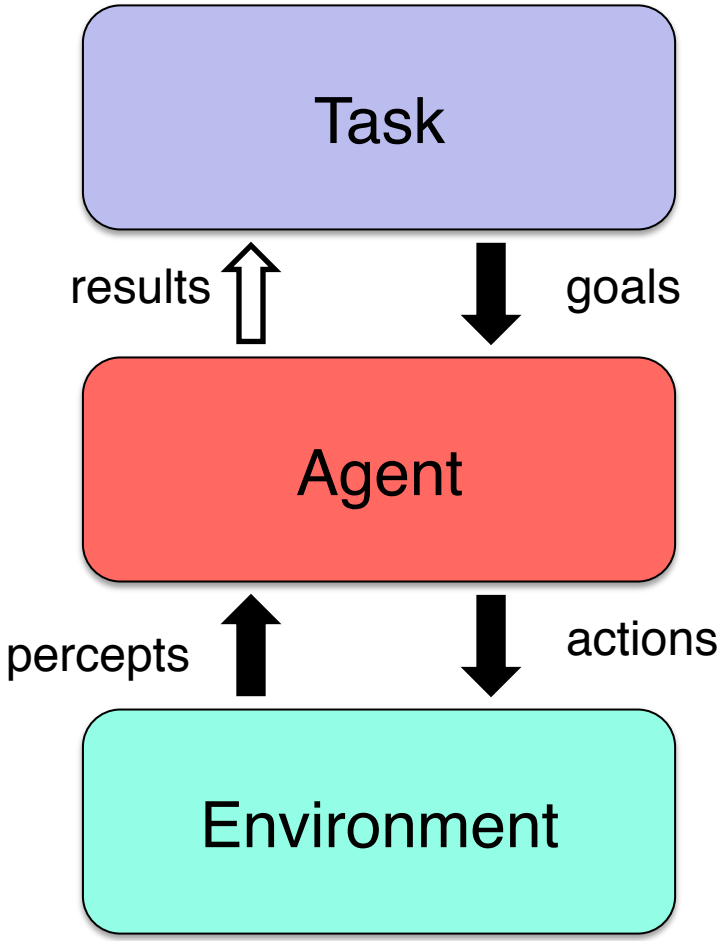
(Pylyshyn 1986)

Designing intelligent agents

- an agent operates in a *task environment*:
  - **task**: the goal(s) the agent is trying to achieve
  - **environment**: that part of the real world or a computational system 'inhabited' by the agent
- the agent obtains information about the environment in the form of *percepts*
- changes the environment by performing *actions* to achieve its goals

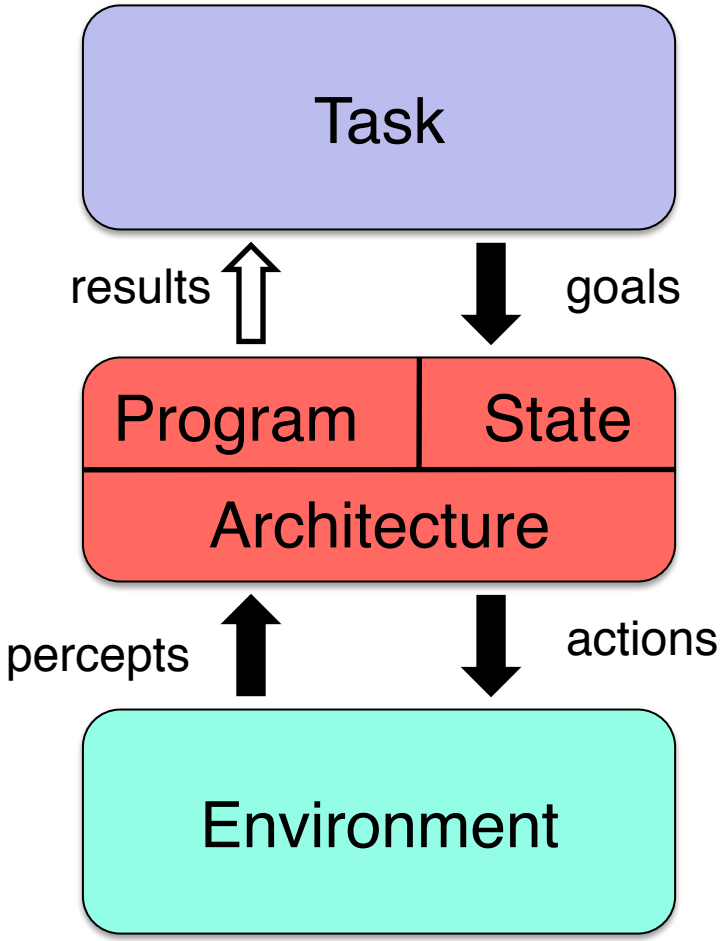
Task environment

- the *task* specifies the *goals* the agent must achieve (and any *results* required)
- the *agent* and *environment* jointly determine:
  - the information the agent can obtain (*percepts*)
  - the *actions* the agent can perform



Components of an agent

- **agent program:** a function which implements a mapping from percepts to actions
- **agent state:** includes all the internal representations on which the agent program operates
- **agent architecture:** a (possibly virtual) machine that runs the agent program and updates the agent state



Implementing the components of an agent

- **agent program:** or *action selection function* may be programmed in a standard programming language (e.g., Java) or in a special purpose *agent programming language* (APL)

- **agent state:** some agents have very little state; in others the state includes representations of the agent's environment and goals, the plans it has for achieving those goals, which parts of the plan have been executed etc.

- **agent architecture:** may be a standard computer or programming language (e.g., JVM) or it may include special purpose hardware (e.g., cameras, robots) and/or software architectures (e.g., APL interpreter)

Agent architectures

- different task environments require different agent architectures
- focus of this module will mostly be on *agent architectures*:
  - what sorts of architectures there are; and
  - which architectures are appropriate for different tasks and environments
- *to build a successful agent we must choose the right architecture for the task environment*

Module aims

- to develop a basic understanding of the problems and techniques of building intelligent agents
- to give an appreciation of the trade-offs inherent in the design of agent-based systems
- to illustrate these through a project involving the construction of a simple agent-based system
- also aims to develop new analysis and design skills appropriate to more complex AI problems

Module objectives

At the end of the module you should be able to:

- judge the appropriateness of different agent architectures for different applications
- analyse the features of a task environment and understand their implications for agent design
- design simple agents able to perform basic tasks in limited domains
- apply basic AI techniques covered in previous modules to implement components of simple agent-based systems
- demonstrate more advanced AI programming skills and familiarity with at least one more advanced AI technique

Plan of the module

- the module consists of two main parts:
  - **lectures:** cover the design of intelligent agents
  - **coursework:** involves designing and implementing a simple agent (or agents)

Provisional lecture plan

Week	Lectures	
1	Introduction & Task environments	
2	Reactive agents I & II	
3	Project description & Getting started with your project	Group tutorials
4	Deliberative agents I & II	
5	Hybrid agents I & II	
6	Multi-Agent systems I & II	
7	Multi-Agent systems III & IV	
8-11	Individual tutorials	
	Easter Break	

Tutorials

- the project work is supported by individual and group tutorials
  - **individual tutorials** cover the design and implementation of your project
  - **group tutorials** cover the use of the *Java agent package* provided as a starting point for the coursework

Project outline

- project involves designing and implementing an agent
- all projects must meet a set of minimum requirements
- projects can be extended in a number of ways, e.g., to multiple agents
- Java ‘demo agent’ package provided as a starting point

Assessment

- **interim report** containing the specification and outline design of your agent – due Friday 1st of March (provisional)

- **final report** describing your agent(s) and the associated **code** – due Friday 11th of April (provisional)

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

Suggested reading

- Wooldridge (2002), *An Introduction to MultiAgent Systems*, Wiley
- Russell & Norvig (2003), *Artificial Intelligence: A Modern Approach*, Prentice Hall (or the 1995 edition)
- Ferber (1999), *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley
- Braitenberg (1984), *Vehicles: Experiments in Synthetic Psychology*, MIT Press
- Arkin (1998), *Behavior-Based Robotics*, MIT Press

Module web page

Web page for the module is available in Moodle

- reading list
- suggested reading for each lecture
- lecture slides –*WARNING: the slides are* ***not*** *'lecture notes' and are not a substitute for taking your own notes or reading the suggested reading for the module*
- tutorial arrangements (later)
- coursework description (later)

The next lecture

Task Environments

Suggested reading:

- for this lecture: Russell & Norvig (2010), chapter 1; Wooldridge (2009), chapter 2

- for the next lecture: Russell & Norvig (2010), chapter 2