COMP3411 Course Introduction

Aims

Artificial Intelligence is concerned with the design and construction of computer systems that "think". This course will introduce students to the main ideas and approaches in AI - including agent architectures, search techniques, game playing, neural networks, machine learning, evolutionary computation, probabilistic reasoning, logical inference, robotics and communication.

Prerequisites

COMP2711 or COMP2911 or COMP2011.

Related Courses

After completing this course, students with a continuing interest in Artificial Intelligence should consider enrolling in one of these courses:
COMP3431 Robotic Software Architecture
COMP4411 Experimental Robotics
COMP4416 Intelligent Agents
COMP4418 Knowledge Representation and Reasoning
COMP9417 Machine Learning and Data Mining
COMP9444 Neural Networks
COMP9517 Machine Vision
or a 4th Year Thesis in an AI-related area.

Note: students taking COMP3411 cannot also receive credit for COMP9414.

Learning Outcomes

Students successfully completing this course will have a working knowledge of the AI methods presented, and will be able to demonstrate their knowledge by explaining certain features or aspects of these algorithms, and by describing how the techniques would be applied to particular problems. Students will gain practical experience, through the assignments, of what is involved in designing and implementing a functional AI system.

Teaching

There will be three hours of lecture per week, plus one hour of tutorial. The major AI algorithms and learning techniques will be presented in lectures and illustrated on sample problems, along with historical background and theoretical motivation. Guest lectures may be given in the latter part of the session by AI researchers within the School, on current areas of active interest.

Tutorials give students a chance to clarify the ideas mentioned in lectures and practice their problem-solving skills in a small (and hopefully more personal) class with the assistance of a tutor. Students are expected to prepare for and actively participate in tutorials. Most tutorials will also include one or two questions of a speculative nature - which can lead to more in-depth discussion of particular topics, depending on the interests of the students.

Assessment

The assessable components of the course are:

Component	Mark
Assignments	40%
Written Exam	60%

Further details about the assignments will be posted on the Course Web site. Programming assignments give the students an opportunity to put into practice the ideas and approaches that have been presented in lectures and discussed in tutorials. They may, for example, involve writing a program to:

- enable an agent to act in a simulated environment
- solve a problem using search techniques
- play a game
- apply a machine learning algorithm
- enable communication or co-operation between agents

In order to pass the course, students will need to score:

- at least 16/40 for the assignments
- at least 24/60 for the exam

- a combined mark of at least 50/100

Course Web Site

http://www.cse.unsw.edu.au/~cs3411

Staff

Staff Name	Role	Email	Extension
Alan Blair	Lecturer-in-Charge	blair "at" cse.unsw.edu.au	9385-7131
Joel Veness	Tutor	joelv "at" cse	

Units of Credit

This course is for 6 units of credit.

Parallel Teaching

COMP3411 is not taught in parallel with any other course.

Plagiarism

All work submitted for assessment must be your own work. We are aware that a lot of learning takes place in student conversations, and we don't wish to discourage this. However, it is important, both for those helping others and those being helped, not to provide/accept any programming language code either electronically or in writing - since it is apt to be used exactly as is, and lead to plagiarism penalties for both the supplier and the copier of the code. Write something on a piece of paper, by all means, but tear it up or take it away with you when the discussion is over.

In addition, soliciting another person to write code for you - either in person or through the Internet - is never permitted. Generally, the copying of code already available on the Internet is also forbidden. If you find some piece of "standard" code in a textbook, or on the Internet, which you would like to adapt and incorporate into your own assignment, you must email the lecturer in charge to ask if it is permissible to do so in the particular circumstances - in which case the source would have to be acknowledged in your submission, and you would need to demonstrate that you had done a substantial amount of work for the assignment yourself.

When evidence of plagiarism is found, the students involved will be dealt with according to School Policy, which provides serious penalties particularly in the case of repeat offences:

> http://www.cse.unsw.edu.au/people/studentoffice/policies/yellowform.html.
> http://www.cse.unsw.edu.au/help/doc/primer/node42.html
> http://www.cse.unsw.edu.au/~chak/plagiarism

Schedule

These are the provisional lecture times and locations:

Time	Location
Tue 11-1	CLB 2
Thu 11-12	CLB 5

Resources

The recommended textbook for this course is:

> Stuart Russell and Peter Norvig, *Artificial Intelligence: a Modern Approach*, Prentice Hall.

If possible, you should try to get the 3rd Edition, which has just been released (2009). However, if you happen to have the 2nd Edition (2003), you will probably be ok.

The following books might also serve as additional reference material:

> Nils J. Nilsson, *Artificial Intelligence: a New Synthesis*, Morgan Kaufmann, 1998, ISBN 1-55860-467-7.
>
> Valentino Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, MIT Press, 1984, ISBN 0-262-52112-1.

Links to electronic resources will be provided on the Course Web page throughout the session.

Course Evaluation and Development

Student feedback on this course will be obtained via electronic survey at the end of session, and will be used to make continual improvements to the course. In response to feedback from 2009, we hope to include a machine learning component in one of the assignments for 2010.

Students are also encouraged to provide informal feedback during the session, and to let the lecturer in charge know of any problems, as soon as they arise. Suggestions will be listened to openly and constructively, and every reasonable effort will be made to address them. For example, in 2007, the choice of programming tools and environment for the final assignment was substantially influenced by feedback from the first assignment.