CLASS SYLLABUS

Instructors

Troy McMahon
Office: Hill 227
Hours: Tuesday 2-3 (tentative)
Email: troymcmahon1@gmail.com

Teaching Assistants

To be announced.

Lecture Times and Locations

Tuesday, Thursday 6:40P - 8:00P
Beck Hall Auditorium

Course Website

To be announced

The webpage will be containing updated syllabus information as the semester progresses and a calendar of lectures and topics covered. Homework and project announcements, practice questions for exams and similar information will be available on website.

Prerequisite

CS 205 Introduction to Discrete Structures I

Text and Reading Material

"Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig Prentice Hall Series in Artificial Intelligence

The slides presented during the lectures will cover a limited part of the material discussed in the classroom. The majority of the mathematical notions will be presented on the whiteboard. The students are encouraged to take notes. Homework and exams will be based on the presented material.

Description

The class will introduce fundamental ideas that have emerged over the past fifty years of AI research. It will also provide a useful toolbox of AI algorithms.

The main unifying theme is the idea of an intelligent agent: autonomous computational systems that receive percepts from the environment and perform actions or take decisions. The objective of the class is to teach students how to identify the appropriate technique for designing such intelligent agents for different types of problems.

The class will be split into three modules. Each module will focus on a different set of challenges and the corresponding techniques:

1. Deterministic Reasoning	2. Reasoning under Uncertainty	3. Learning and Reasoning in Unknown Environments
(Heuristic) Search Local Search Adversarial Search Logic-based Inference Planning	Bayesian Networks Hidden Markov Models Kalman and Particle Filters Markov Decision Processes POMDPs	Decision Trees Expectation Maximization Neural Networks Support Vector Machines Intro to Reinforcement Learning

Part 1: Initially the focus will be on autonomous decision-making in deterministic environments. Search techniques (including dynamic programming algorithms, informed techniques like A* and local search methods, such as hill climbing and simulated annealing) will be presented that concentrate on deciding what to do when one one needs to think ahead several steps. The class will then discuss ways to represent knowledge about the world through logic and how to reason logically with that knowledge, especially for problems with constraints. The first part of the class will close with path planning algorithms in continuous spaces.

Part 2: This part will introduce a significant challenge, that of uncertainty. In this section, the intelligent agents do not have perfect knowledge of the world. They depend on probabilistic tools, such as the Bayes law, in order to reason about their state, the world and take decisions. For example Bayesian networks will be presented that provide a systematic way of representing independence and conditional independence relationships in order to simplify probabilistic representations of the world. They can be used in order to achieve inference under uncertainty. Building on these tools, we will present filtering variants of Hidden Markov Models (e.g. Kalman filter, particle filters) that are able to reason under uncertainty when the environment changes over time. Then through the tool of Markov Decision Processes we will show how it is possible to not only infer the world's state under the presence of uncertainty but also to make decisions under uncertainty so as to maximize an expected utility function.

Part 3: The final part of the class involves even more challenging problems, where the environment is not just uncertain but unknown. In order for agents to be able to cope with such problems they depend on learning techniques that generate the knowledge required for decision making. Inductive learning techniques use observations to create simple theories about the world. Statistical learning methods learn probabilistic models of the world when there is uncertainty in the measurements. Neural networks are biologically inspired and provide a way of inferring a world model. Finally, reinforcement learning techniques show how intelligent agents can learn from success and failure, from reward and punishment.

Exams

There will be 2 exams: one midterm and one final. The first exam will cover the material of the first half of the course (approx. first 14 lectures), and the final will cover the last 2/3rds of the course (there will be some overlap with the midterm). Check the tentative schedule on the course website for the date of the midterm. Both exams are in-class exams.

A missed exam draws zero credit. Emergencies will be considered upon submitting a University- issued written verification to the Instructor; for assistance contact your Dean's Office. Also, check the definition of Final Exam Conflicts by SAS.

Assignments

There will be multiple assignments (either 4 or 5). You will be informed in advance when an assignment is due. Typically, each assignment will include opportunities for extra credit.

The assignments include practice questions which are intended to assist the student in mastering the course content. They will also involve programming AI algorithms and testing their efficiency. Typically you will be asked to submit an electronic version of your code, test runs, a typeset report and demo your project to the teaching assistants.

Assignments should be completed by teams of students - two at most. No additional credit will be given for students that complete a homework individually. Please inform the TA about the members of your team. Students can switch teams between assignments but they are not required to do so. When the composition of a team changes, the students need to inform the TAs.

Submission Rules

No late submission is allowed. If you don't submit an assignment on time, you get 0 points for that assignment. Students can submit their assignments electronically via Sakai. Programming results must be demonstrated during prescheduled demo appointments. The project report must be forwarded in advance of the demo.

Grading System

The final grade will be computed according to the following rule **(this is tentative and can change)**:

Assignments 50 points total (approx. 8 to 12 points each depending on difficulty)

Midterm and Exam 50 points total (approx. 25 points each)

Participation up to 5 points bonus

By default your participation grade is 0, i.e.„ if you typically come to the lectures/recitations but you rarely answer questions during the lectures or the recitations, your participation grade will be 0. Positive participation grades will be given to students that actively participate in lectures and recitations.

The mapping of scores to letter grades will be determined at the end of the semester. As a **rough** guide, the following rule may be used for the final grade **(it may be adapted close to the end of the semester)**:

Ranging	Final Grade
A	> 89
B+	80 - 89
B	70 - 79
C+	60 - 69
C	50 - 59
D	40 - 49
F	< 39

Students interested in a recommendation letter by the instructor will be offered one only if they achieve a score above 95 after the completion of the course.

Questions about Grading

If you have a question or complaint regarding the points you received on specific parts of an as- signment, or an exam, submit a request to the TA, stating specifically but very briefly what parts of that assignment you wish to be reviewed. Please refrain from verbal arguments about grades with the instructor or with any of the TAs. We will try to get back to you soon. The deadline for submitting such requests is the last lecture.

Academic Standards

Exams are to be treated as individual efforts. Homework and programming assignments are not to be treated as collective efforts beyond the participation of the team members! Discussions between teams are not allowed on how to solve specific questions in homework. Do not discuss assignments with students that are not currently taking the class.

A severe penalty will be given to any assignment which indicates collusion or cheating. The usual penalty for cheating on an assignment or an exam is failure in the course. At a minimum your grade in the corresponding exam/assignment will be reduced. Stealing another team's listing or having another person "ghost write" an assignment will be considered cheating.

Turning in work without properly citing the sources of the content included in your work is plagiarism. All kinds of sources are included in this definition, even those downloaded from the web, in which case an operable link must be cited. Plagiarism from the web or other sources is considered cheating and has the same effects to those mentioned above. Even with a reference, submitting an answer to a homework question, verbatim from any source and without any contribution on your part, draws zero credit.

You should carefully study the website of Rutgers University on Academic Integrity and the corresponding policy, as well as the corresponding website from the department of Computer Science. Links are available through the course website. Your continued enrolment in this course implies that you have read these policies, and that you subscribe to the principles stated therein.