CS 580 Introduction to Artificial Intelligence — Spring 2022

Thursday 4:30-7:10pm, East Building room 122. See https://cs.gmu.edu/~hrolenok/teaching/cs-580-spr2022/index.html for current syllabus. Be sure to check your @gmu.edu email and blackboard regularly for updates.

Instructor: Brian Hrolenok
@gmu.edu email: hrolenok
Office Hours: TBD

TA: TBD
@gmu.edu email: TBD
Office Hours: TBD

Course description

CS 580 - Introduction to Artificial Intelligence is a 3-credit introductory course intended for graduate students. Artificial Intelligence is subfield of Computer Science which covers the design, implementation, and analysis of computational systems that can be said to reason, learn, or act rationally. This course presents a broad overview of this material using an agent based approach, and has a particular focus on the details of implementation. The class is programming intensive, and a strong background in the programming language of the assignments (Python) will be very useful.

Learning objectives:

- To provide a broad survey of algorithms and approaches in AI.
- To develop a deeper understanding of several major algorithms in AI.
- To develop the design and programming skills that will help you to build computational artifacts that act intelligently.
- To develop the basic skills necessary to pursue research in AI.

Prerequisites. The recommended prerequisites for this course are CS 310 and CS 330, and familiarity in the following topics and strong mathematical maturity will be useful:

- Probability
- Linear algebra
- Calculus

Textbook: *Artificial Intelligence: A Modern Approach* by Stuart Russell and Peter Norvig.
The readings given below are for the third edition, section numbers may be different for other editions. Ideally, readings should be completed before class so you can bring your questions to lecture.

Homework and Problem Sets

This semester, there will be two different kinds of take-home assignments: *homework* and *problem sets*. **Homework** assignments are programming focused, and come with an autograder for you to test your code. They are graded for correctness with the autograder and by manual inspection. Their purpose is to give you hands on practice implementing the algorithms and techniques discussed in class in a concrete setting, and, specifically, experience with the unique challenges of debugging in these domains.

Problem sets are more focused on understanding and exploring the lecture/reading topics from a theoretical and mathematical standpoint. They involve short-answer responses, proofs, open-ended exercises, and occasionally some short programming. These assignments **are not** graded for correctness and they have **no autograder** — any submission before the due date will receive full credit. The purpose of these assignments is to keep you synchronized with the course lectures and readings, give you touchstones for potentially challenging concepts that do not easily fit with the other programming based homework, and to expand on topics that we may not have time to cover in depth during lecture. Some solutions or solution approaches may be covered in lecture, time permitting. Problem set submissions should be a single PDF file unless specified otherwise in the assignment description.

All assignment submissions will be handled through [Blackboard](), and are due by the date and time listed there. Submissions by email will not be accepted.

Collaboration

All of the assignments in this course can be completed individually or in groups of two, **larger groups (3 or more) are not allowed**. If you do work as part of a team, make sure that you indicate this *very clearly* as part of your submission. For homework assignments, each file you submit should have a comment at the top with you and your partner's names and GMU ID numbers. For problem sets, be sure to include both of your names at the top of your submitted PDF. It is important that anything you submit reflects your understanding of the material.

Bug Bounty

The homework assignments this semester have been adapted (with permission) from a similar course offered by Berkeley. The positive side of this is that we have access to some excellent GUI representations, rigorously checked solutions, and robust error checking and debugging messages. The downside is that these assignments were originally developed for Python 2, and have only recently been updated for Python 3. We've done our best to make sure the provided code works out-of-the-box on recent stable versions of Python (whatever version comes pre-installed on your computer, or installed directly from python.org), but there may be some rough edges that we've missed. To help us improve the course for future semesters, and to provide a bonus for engaged students, we're offering a small bug bounty (up to +1% to your semester grade, extra-credit) with the following conditions:

1. You must provide a fix, not just a bug report.
2. Only the first person to report and fix a given bug gets the bounty.
3. The fix must not break backwards compatibility to Python 3.6.
4. Maximum cumulative extra-credit can't exceed 5% of your overall grade.
5. It has to be reproducible on a TA or Instructor machine.
6. We have to agree it's actually a bug.

The last item above does not necessarily exclude display bugs, "text bugs", or typos, but these should rise to the level of significantly increasing the chances of confusion for future students if left unaddressed. For the most part, this bounty is restricted to the autograder provided with homework assignments, but particularly egregious typos on the syllabus, slides, or problem sets *may* also be considered. **Important note**: it is extremely easy to assume that the autograder is failing a particular test-case because of a bug in the grading script. **This is almost certainly not the case**. Especially for the search homework, many common mistakes in your own code can lead to *almost* but not quite correct implementations. When in doubt, assume the autograder is correct and bring your questions to office hours. To avoid a pile-up at the end of the semester, only bounties submitted by week 11 can earn extra-credit this way.

Late Policy

You have five free late days to be used at your discretion throughout the semester, for either homework or problem set submissions. That means you might turn in one assignment two days late or two different assignments one day late, etc. To avoid an accounting nightmare, you can't pick and choose when to apply late days: a late day is "used" as soon as you submit an assignment late. The first late day is used when you submit one minute after the due date. Two late days are used when you submit 24 hours and one minute after the due date. A third late day is used 48 hours and one minute after the due

date, etc. This also means you do not have to let us know ahead of time if you're using a late day. After the free late days are exhausted, you will receive a 20% penalty per day.

Exams

There will be two exams in this class, a midterm and a final (cumulative). The date for the midterm may differ from what is listed below, but the intent is for it to be roughly halfway through the semester. The date for the final is fixed by the registrar (see the registrar's [final exam schedule](#)). You are allowed one 8.5x11in sheet of notes, front and back. There will be no make up for either exam unless previously arranged *(well in advance)*.

Grading policies

Your TA and I will strive to provide you reasonably detailed and timely feedback on every assignment and exam. If you have any questions about any of your grades please reach out to us, either by coming to scheduled office hours or via your "@gmu.edu" email address. If there is an error with your grade, please contact us within a week of when feedback is returned, otherwise we might not be able to change it.

Point breakdown:

- Problem sets: 10% (2% each)
- Homework: 30% (7.5% each)
- Midterm: 25%
- Final: 35%

Academic Integrity
All of the assignments you submit in this class must represent your own work. There are absolutely solutions to some of these assignments on the Internet. Do not use them. If you can find them, so can we, and we will check. It's probably a bad idea to even look for them, even for "an idea of how to start" because once you see **one** solution, its hard not to think of it as **the** solution. If you do wind up "just looking", make sure you document it in your code. Otherwise, you run the risk of appearing to misrepresent someone else's work as your own. When in doubt, **be explicit** about where the code came from.

Accommodations

If you need academic accommodations, please make sure you contact the instructor at the beginning of the semester or as soon as possible. Also make sure to contact GMU's Disability Services, available online (https://ds.gmu.edu, ods@gmu.edu) and by phone (703.993.2474), which coordinates all academic accommodations. After you have contacted ODS, you still need to contact the instructor so that appropriate arrangements can be made.

Schedule

The following is the tentative schedule for the spring 2022 semester. Please check your email and Blackboard regularly for any changes, as this website may not be updated immediately.

Week 1
Introductions and Logistics. **Reading**: Ch 1-2

Week 2 — Coding refresher problem set due *(tentative)*.
Uninformed search. **Reading**: Ch 3.1-3.4

Week 3
A* search and heuristics. **Reading**: Ch 3.5-3.6

Week 4 — Search problem set due *(tentative)*.
Search in different domains. **Reading**: Ch 5.1-5.3, 5.5

Week 5 — Search homework due *(tentative)*.
Markov Decision Processes. **Reading**: Ch 17.1

Week 6 — Reinforcement Learning problem set due *(tentative)*.
Value Iteration and Q-learning. **Reading**: Ch 17.1, Ch 21.3.2

Week 7
Midterm. **Reading**: none.

Week 8
Spring break, no class.

Week 9 — Reinforcement Learning homework due *(tentative)*.
Probability and state uncertainty. **Reading**: Ch 13

Week 10
Bayes Nets. **Reading**: Ch 14.1-14.5

Week 11 — Bayes nets problem set due *(tentative)*.
Intro to Machine Learning. **Reading**: Ch 18.1, 18.2, 18.6.1, 18.6.2.

Week 12 — Filtering homework due *(tentative)*.
Classification, k-NN, Logistic Regression. **Reading**: Ch 18.6.4, 18.8.1.

Week 13 — Linear Regression problem set due *(tentative)*.
Decision Trees/Neural Networks. **Reading**: Ch 18.3/18.7

Week 14
Clustering and k-Means, Gaussian Mixture Models and EM. **Reading**: Ch 20.3

Week 15 — Machine Learning homework due*(tentative*.
Extra topics and review.