MIT ES.S20 Course Syllabus

Course Description

This course will examine the mathematical structure and strategy for many common games and puzzles, culminating in a project exploring an existing game, an open problem, or a new game design. Using games as a model, we will explore the fundamental ideas behind AI, groups, game theory, computational complexity, probability, and cellular automata. Each class session will involve a discussion of a new kind of game or design and game play.

This course will be run by Robert Sloan (rsloan) and Jayson Lynch (jaysonl). Our faculty supervisor is Erik Demaine (edemaine).

Objectives

Students will learn how to conceptualize and work with games from a mathematical and computational perspective. We will try to play everything we talk about during class, but many of them are difficult (like Monopoly); so some of it will be online.

Materials

There is no official textbook other than the course notes, but you can use the resources provided on our GitHub repository. You may also find the following books useful:

- Winning Ways for Your Mathematical Plays by Berlekamp, Conway, and Guy
- Games, Puzzles, and Computation by Hearn and Demaine

Grading and Expectations

The course is pass/fail, with the grade breakdown as follows:

Attendance	30%
Project idea/proposal	10%
Project completion	60%

We would like consistent attendance; but tell us if something happens.

Weekly Schedule

Because this is on GitHub, this will update with changing dates.

Game Representation/Trees	2/13
Classical Games	2/20
Algebraic Structures	2/27
Dynamic Programming	3/6
Computational Complexity	3/13
AI Search Techniques	3/20
Other Metaheuristics	4/3
Economic Game Theory	4/10
Auctions and Mechanism Design	4/17
Randomized Games	4/24
Inference and Network Models	5/1
Generative Games	5/8
Game Design Strategies	5/15

Final Project Guidelines

You can work on any of these projects in groups of 1-3 students: just try to make sure that the work is divided evenly.

Project Examples

- Research paper on a game

  Pick a game that we did not discuss deeply in class and use it to explore new mathematical or computational strategies. Be sure to include all relevant background information in a final paper which should be at least 15 pages.

- Coding project

Implement a game we've discussed and/or an AI Engine in a language that the course staff understands as an application. Your code should be documented enough to be comprehensible, especially if you implement your own algorithms.

- Create a toy or game

  This could be a physical toy, a board game, a card game or whatever you happen to come up with. You should provide a short write-up with designs to describe the ideas behind what you made (along with the rules…)

- Open Problems

  Work on an open problem in recreational mathematics, which are plentiful especially with less common games. This should look something like the research paper, though probably shorter depending on the conceptual complexity of what you're talking about. Make sure to describe the problem, what you'd like to find out, and its significance because sometimes research problems aren't so easy.

Deadlines

1. Project idea by 3/13

   We approve your project idea – this can be done via email.

2. Project proposal by 4/10

   A formal project proposal detailing everything you plan to do with deadlines. If you have a group project, you should have delegated general tasks.

3. Project final by 5/8

   We will present projects during the last days of classes.