# Introduction to Artificial Intelligence

Course syllabus, Fall 2024

Prof. Gilles Louppe
g.louppe@uliege.be

LIÈGE université

# Us

This course is given by

- Theoretical lectures: Gilles Louppe
- Exercise sessions: Gilles Louppe, Gérôme Andry, and student instructors
- Programming projects: Arnaud Delaunoy

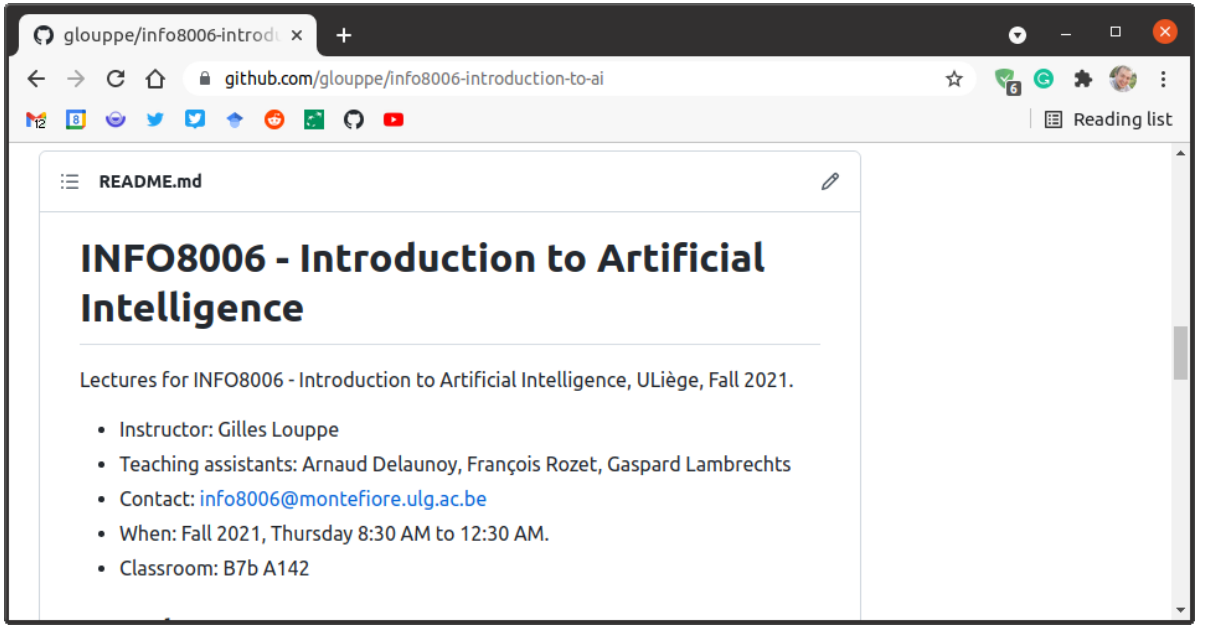
Feel free to contact us at info8006@montefiore.ulg.ac.be or on Discord for help.



# Materials

The schedule and slides are available at github.com/glouppe/info8006-introduction-to-ai.

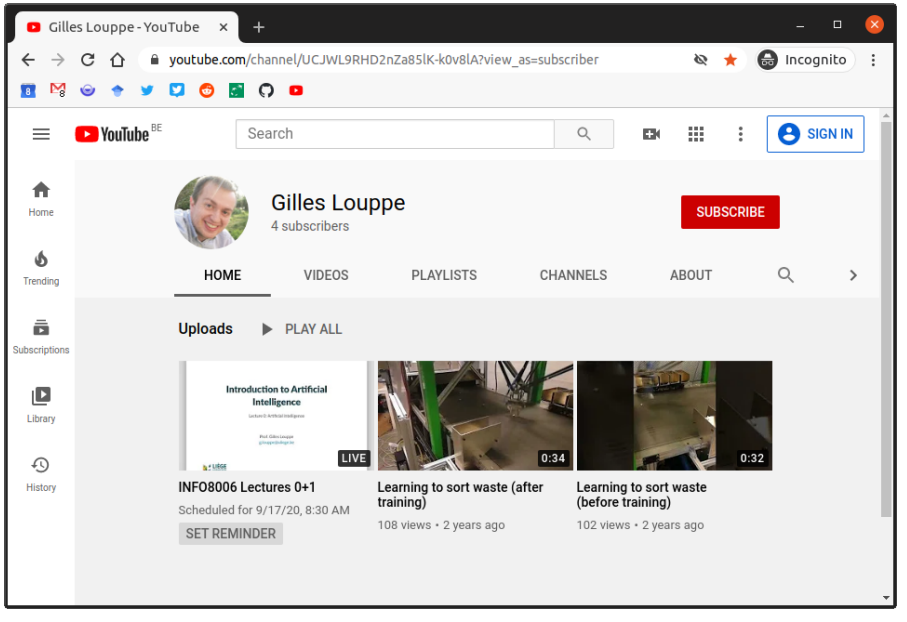
- In HTML and in PDFs.
- Minor updates up to the day before the lesson.



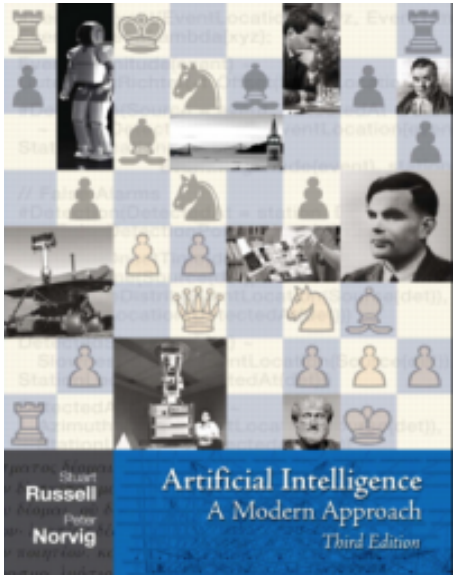
# Videos

Videos from Fall 2020 are available at https://youtube.com/playlist?list=PLLqXZ_E-UXlybvRU7vgaYMTbxZdT73ZFD. They are not up-to-date with the current course, but they may still be useful.

Lectures will also be recorded and made available on MyUnicast.



# Textbook




The core content of this course is based on the following textbook:

> *Stuart Russel, Peter Norvig. "Artificial Intelligence: A Modern Approach", Third Edition, Global Edition.*

This textbook is recommended. It covers both the theory and the exercises.

## CS188

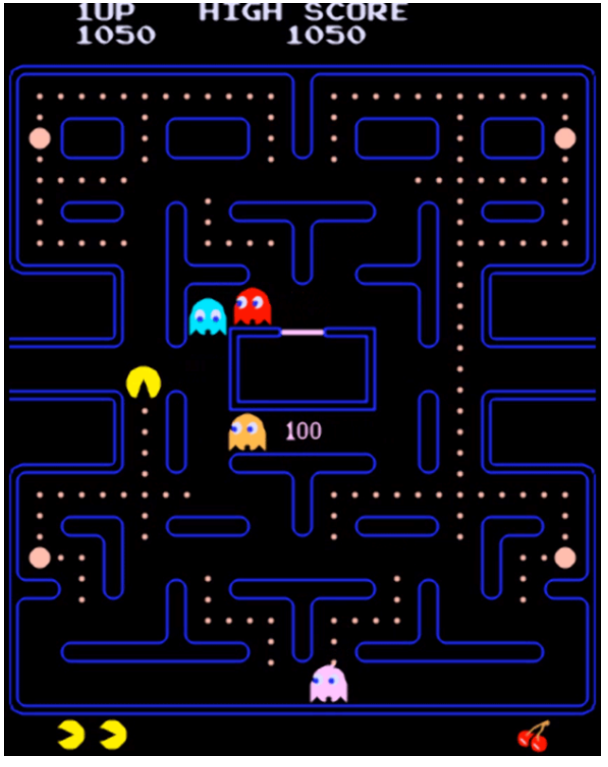
- Some lessons, exercises, and various other materials are partially adapted from CS188 Introduction to Artificial Intelligence, from UC Berkeley.
- Cartoons that you will see in those slides were all originally made for CS188.

# Exercise sessions

- Exercise sessions are held every week after the lecture.
- You will work on exercises by yourself or in small groups, with our help.
  - Your active participation is expected.
  - The exercises are designed to help you understand the materials and prepare for the exam.
  - Use this time to get answers to your questions.
- Solutions are provided for all exercises.

# Programming projects

Implement an intelligent agent for playing Pacman. The project will be divided into three parts, with increasing levels of difficulty.



# Evaluation

- Written exam (60%)
- Programming projects (40%)
  - Project 0: 0% (tutorial of September 26)
  - Project 1: 20%
  - Project 2: 20%
  - Programming projects are mandatory for presenting the exam.

## Evaluation of the projects

- Projects are evaluated automatically on Gradescope.
- Preliminary tests are provided to help you debug your code.
- However, the final grade will be based on additional tests that are not provided.
    - It is part of the learning experience to design your own tests.
    - It is not because the public tests pass that your code is correct.
- The efficiency of your code will also be taken into account, beyond its correctness.

## Honor code

You may consult papers, books, online references, or publicly available implementations for ideas that you may want to adapt and incorporate into your projects, so long as you clearly cite your sources in your code and your writeup. *However, under no circumstances, may you base your project on someone else's implementation.* In particular, the use of large language models (e.g., ChatGPT, Github Copilot) is forbidden*.

Plagiarism is checked and sanctioned by a grade of 0. Cases of plagiarism will all be reported to the Faculty office.

—

*: Unless you build and train your own :-)

Let's start!