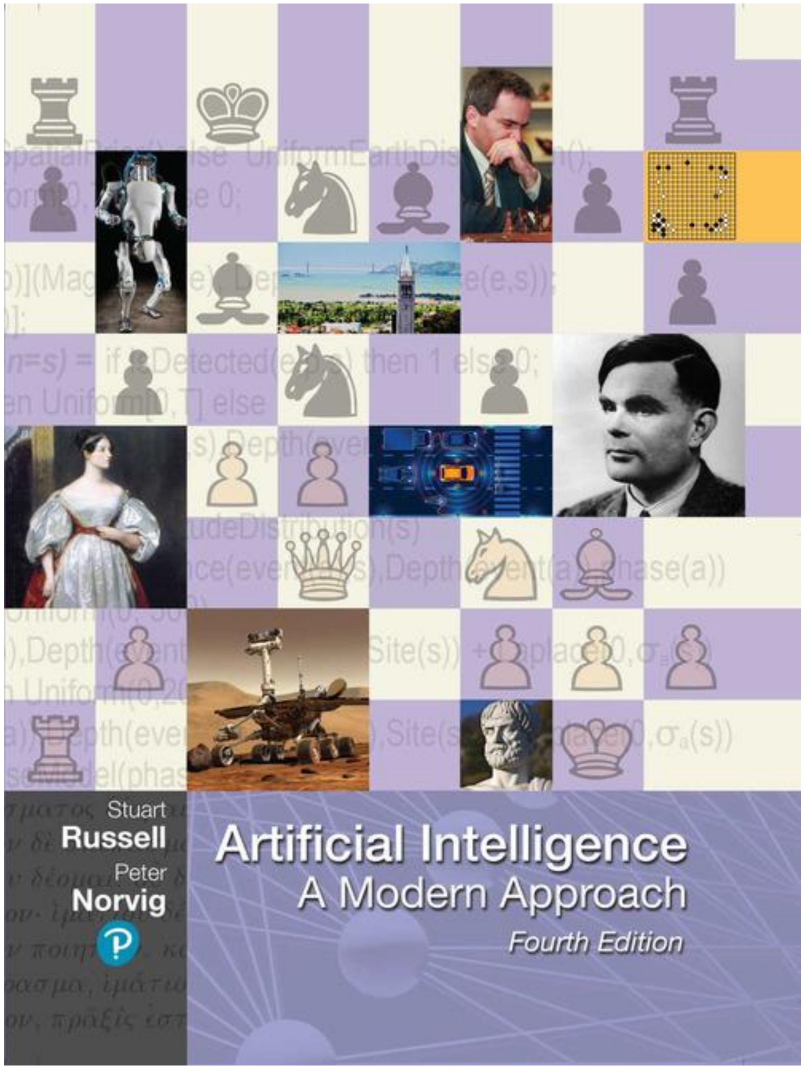
Artificial Intelligence Fundamentals

2024/2025



"Machine intelligence is the last invention that humanity will ever need to make."

– Nick Bostrom

Exam Projects

Outline

- Exam Projects
  - Why?
  - How to create a team
  - How to build the project
  - How to write the project report
  - How to present it
  - How to submit it

- Project proposals and themes

Evaluation

- Team Projects -> **1/3**
  - in-class presentation + notebook report

- Team oral (project intro + 3 questions each) -> **2/3**

Why Projects

- «*Learning by doing*» philosophy

- **1/3** of the exam evaluation will come from projects

- You need to learn how to work in a team: *creativity, collaboration, bigger goals, …*

- You need to learn **how to communicate your work**: orally and in writing

- *It may be hard… but you'll be happy once the course is done!*

How to create a team

- **Team composed by 2-5 members**
  - Exeptions for working/non-attending students
  - The evaluation of the project will be proportional to the number of team members -> less members, more work
  - Try not to choose your close friends -> diverse team will receive a better evaluation
- **Find a team contact reference**
  - For communication with the teacher, submitting the report, etc.
- **Apply with your team here**
  - https://forms.gle/uK5UyMMpqB3LCTQn9

Pearson

How to Build a Project

- **Open project topic**
  - We will suggest some... *but the choice is yours!*

- **Built entirely on GitHub**:
  - All members should be registered and part of the project
  - All members should partecipate equally with a roughly equal number of commits
  - Self-contained project (I should be able to download and run it easily)

- **Only open-source** tools and software allowed
  - If on Windows, I suggest to use Windows Subsystem for Linux (WSL2)

- **The project MUST contain methodologies studied during the course** -> *the more, the better.*

- **It needs to contain a demostrator / empirical evaluation**

Target Project Complexity

- **Make it small**: high complexity is not demanded.

- **Re-use**: you can use all available assets you want: libraries, other exams project pieces, scripts in the internet, etc. but make sure to mention it in the report.

- **I look for**: *creativity, good ideas, good aiming & planning, quality, attention to details*.

- **Rule of thumb**: you need to spend about 150 hours to pass the course (6 CFU):
  - about 50 in frontal lectures
  - About 50 to study the book
  - About 10 to study additional materials
  - About 40 on the project (e.g. 3 hrs/week for 3 months, 5 days if full-time)

How to Write a Report

- **The notebook report is key for evaluation**:
  - The evaluation will be based on this + github repo (put the link inside!) + presentation
  - Attention to details is key, length is not (make it 5 pages max), it should contain both descriptions and runnable code on Google Colab

- **The report should contain**
  - *Introduction, related works, methodologies, assessment, conclusion*
  - Appendix: team contributions, github metrics, relationship with the course

How to Present it

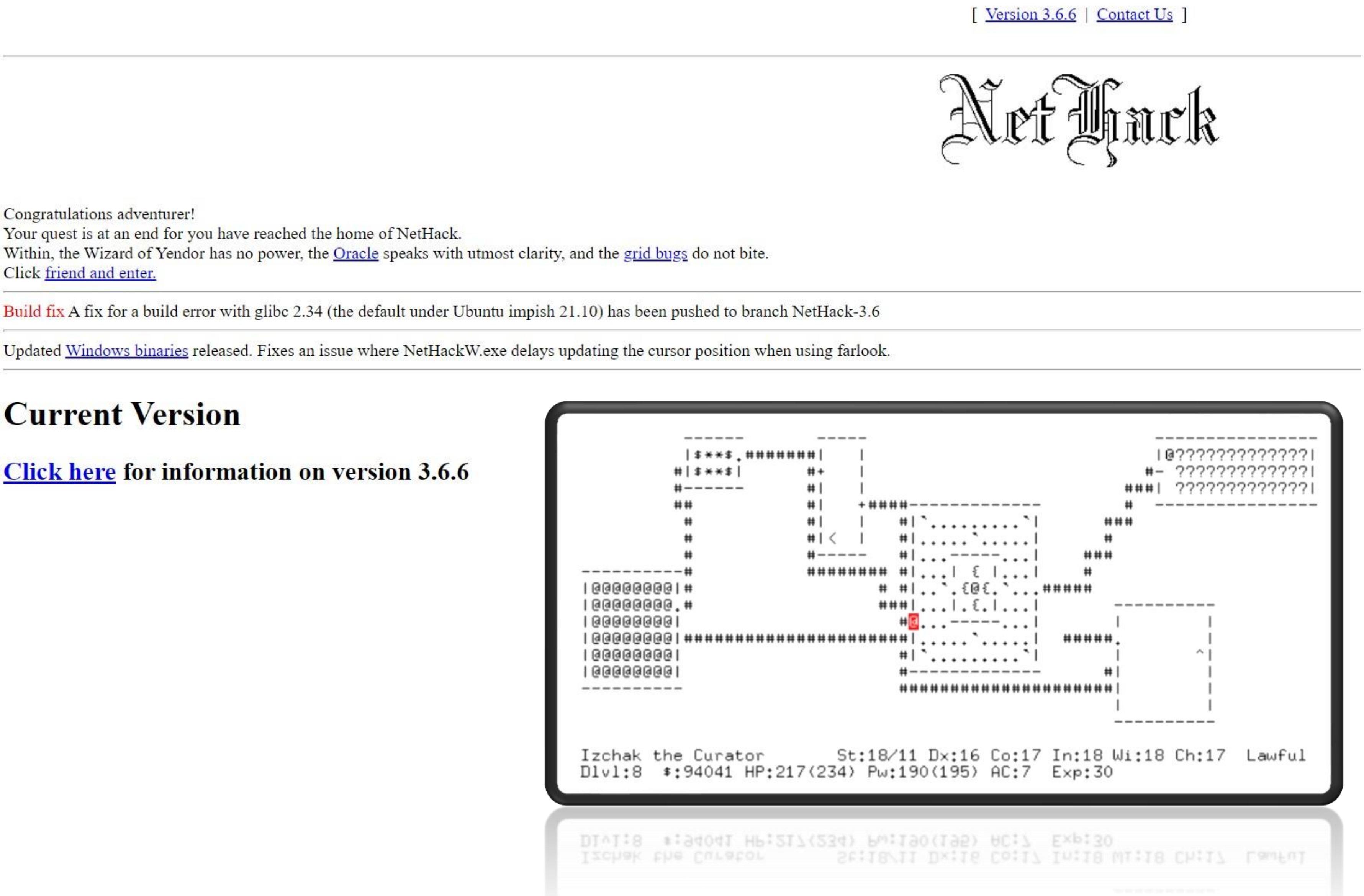
- **You'll get the chance to present your project in December**
  - Main idea, dev. plan, issues, partial results, etc.
  - 5-20 minutes team presentation depending on the teams number
  - All members should take part to the slides preparation and oral presentation
  - The exact schedule will be decided with the team ref contacts
- **Basic tips to present**
  - Face your audience, look them in the eyes
  - Don't put hands in pockets, in front of your mouth
  - Do not rush it, be calm
  - High energy is key!!!

How to Submit it

- **Once you have completed the project**
  - The team ref. can submit it via email to vincenzo.lomonaco@unipi.it
    - **1 files**: the computational *notebook exported in pdf*
  - Use the subject **"[AIF Project Submission] – X Team**"
  - Once approved it will be *valid 1 year* and you'll get access to the oral exam in the ordinary exam sessions.
- **Orals**
  - The oral session will be based on the teacher questions about the project + ordinary course questions
- **Top-3 projects**
  - will receive a public acknowledgement on the course website, can be continued as Master dissertations or scientific publication

1° Project Theme: NetHack

By *Luigi Quarantiello*



[Version 3.6.6 | Contact Us]

Congratulations adventurer!
Your quest is at an end for you have reached the home of NetHack.
Within, the Wizard of Yendor has no power, the Oracle speaks with utmost clarity, and the grid bugs do not bite.
Click friend and enter.

Build fix A fix for a build error with glibc 2.34 (the default under Ubuntu impish 21.10) has been pushed to branch NetHack-3.6

Updated Windows binaries released. Fixes an issue where NetHackW.exe delays updating the cursor position when using farlook.

Current Version

Click here for information on version 3.6.6

```
          ------       -----                            ----------------
          |$**$.#######|   |                            |@?????????????|
         #|$**$|      #+   |                           #- ?????????????|
         #------      #|   |                         ###| ?????????????|
         ##           #|   +####-------------        #  ----------------
          #           #|   |   #|`.........`|      ###
          #           #|<  |   #|.....`.....|      #
          #           #-----   #|...-----...|    ###
----------#           ######## #|...| { |...|    #
|@@@@@@@@|#                  # #|..`.{@{.`....#####
|@@@@@@@@.#                  ###|...|.{.|...|       ----------
|@@@@@@@@|                     #@...-----...|       |        |
|@@@@@@@@|######################|.....`.....|  #####.        |
|@@@@@@@@|                     #|`.........`|       |       ^|
|@@@@@@@@|                     #-------------      #|        |
----------                     #####################|        |
                                                    |        |
                                                    ----------

Izchak the Curator       St:18/11 Dx:16 Co:17 In:18 Wi:18 Ch:17  Lawful
Dlvl:8  $:94041 HP:217(234) Pw:190(195) AC:7  Exp:30
```

NetHack

Useful references:

- YouTube video (super nice introductory video)
- The NetHack Learning Environment (paper)
- MiniHack (paper)
- NetHack Challenge (witht python notebooks)
- Alt.org server (play in the browser, telnet, ssh)

Some ideas

Some Ideas for the Projects

- ***Smart* pathfinding**: implement an agent able to reach one or multiple goals while minimizing the number of steps required, being aware of its surroundings (*e.g.* the actions of monsters). The agent should be able to react to dangerous situations while performing its plan;
- **Advanced combat system:** define a module to manage the combat system. The agent should be able to correctly use the different types of weapon available (*e.g.* swords, bows, wands).
- **Inventory management**: define an agent able to manage its inventory.
- **Floor recognition:** NetHack provides different types of floors and rooms (*e.g.* shops). The task is to implement a module that can recognize the current floor/room, in order to behave accordingly;
- **State representation:** Represent the states of the game, possibly taking into consideration also *n* previous states. It can be done using several methodologies, *e.g.* HMM, Markov Chains, Bayesian Networks, RNN, LSTM, Attention modules, etc. This encoding can be used to decide the action to take, or for next-state prediction;
- **Logic Programming**: Define a knowledge base using Prolog, to represent the game dynamics, by exploiting the information present on the NetHack Wiki; it can be done in a custom environment, defined with MiniHack.

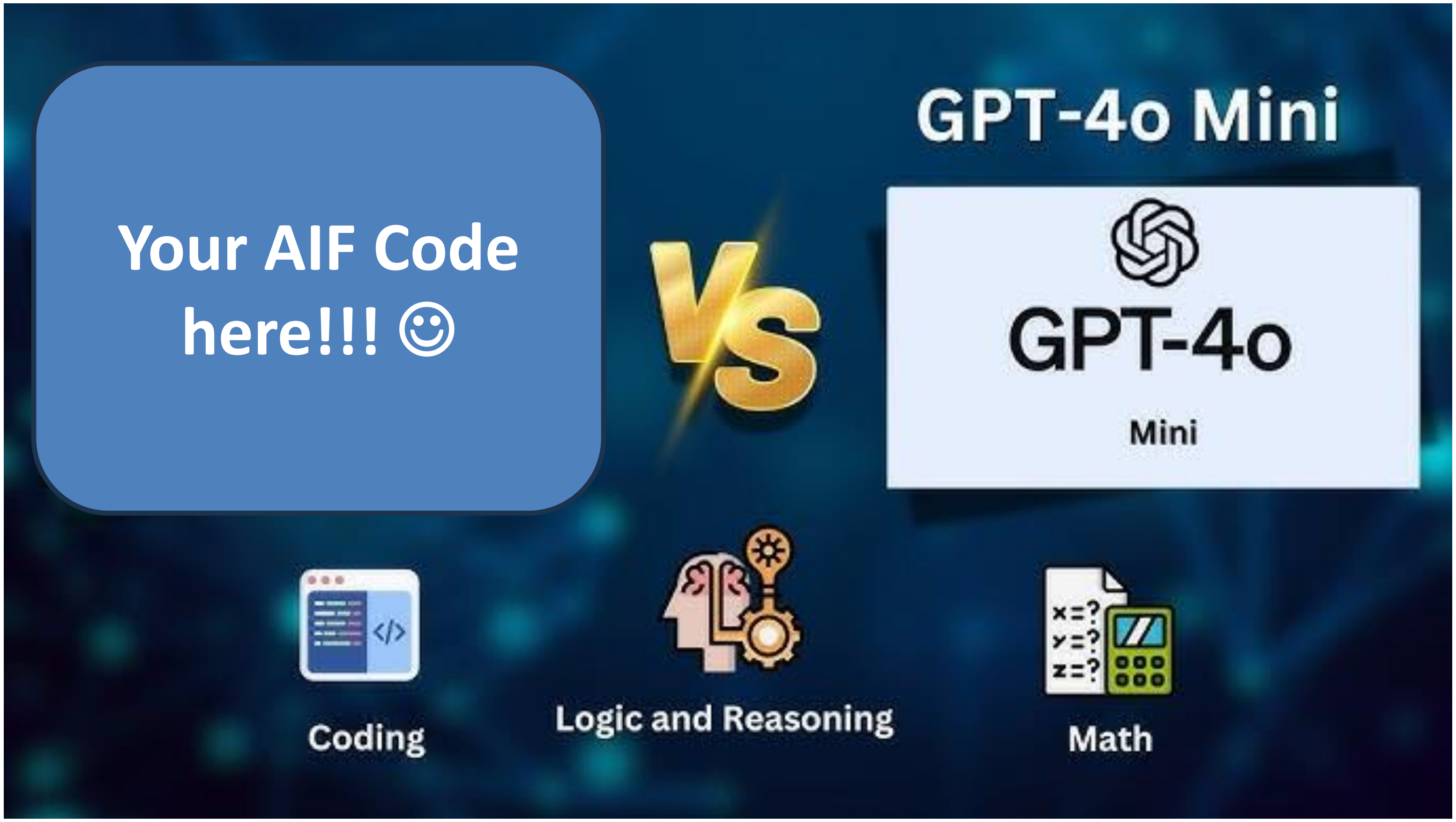
...only the sky is the limit! More details here

2° project theme: Conference of Games (CoG) competitions

By *Elia Piccoli*

3° project theme: Complex Reasoning - Symbolic AI vs GPT

By *Vincenzo Lomonaco*



Your AIF Code here!!! ☺

In the next lecture...

- Problem-solving agents
- Example Problems
- Problem formulation
- Search Algorithms
- Uninformed Search Strategies
- Informed (Heuristic) Search Strategies
- Heuristic Functions