Reinforcement Learning
Final Project - 2026

The main goal of this project is to summarize the key topics (theoretical & practical) that we have covered in the second part of the course integrating some theoretical and practical aspects.

MiniGrid env.

In this exercise you will solve two variations of the "Minigrid" environment.

MiniGrid is a lightweight, 2D grid-world environment with goal-oriented tasks. The agent in these environments is a triangle-like agent with a discrete action space. The tasks involve solving different maze maps and interacting with different objects such as doors, keys, or boxes.

Minigrid documentation:

https://minigrid.farama.org/environments/minigrid/

In this assignment there are two Minigrid environments:

1. **SimpleGridEnv:**
    - **Description**: An 8x8 empty room.
    - **Goal**: Navigate to the green goal square.
    - **Action Space**: 3 actions (Turn Left, Turn Right, Move Forward)
2. **KeyDoorBallEnv:**
    - **Description:** An 8x8 grid divided into two rooms.
    - **Mission Sequence:**
        i. Find the Key (Left room).
        ii. Open the Door.
        iii. Pick up the Ball (Right room).
        iv. Reach the Goal.
    - **Action Space:** 5 actions (Turn Left, Turn Right, Move Forward, Pickup, Toggle)

In both environments you can add **reward shaping**.

The main difference from the mid semester project is that here the **observation is the image** (pixels based).

Use the following **template notebook** as the foundation for your project. It currently includes a demonstration of the environment using random actions. You must **save a copy** of this notebook and implement your solution directly inside it:

https://colab.research.google.com/drive/17jaQ11Ysl6PYp8tS8-UFmIUR73qapuHZ?usp=sharing

- Show your knowledge in DEEP RL that covers what we have learnt so far.
- Try to “solve” it as fast as you can (less episodes) - this is a competition part.

Note: There are a lot of implementations on the web of these algorithms. We want to see your different considerations while solving the environment.

- Solve it with different algorithms from the DEEP RL part of the course
- Discuss the advantages/disadvantages of the approaches relating to your Minigrid mission.
- Show relevant graphs comparing the different approaches.
- Add preprocessing to the input image.
- Different hyperparameters: Learning rate, Epsilon, replay buffer size, target network update rate, initializations etc.
- Different considerations for the two environments.
- Approaches for Exploration-Exploitation tradeoff.
- Graphs for the training stage.
- For the inference stage - Last 100 episodes “average num of steps to finish episode”.
- Evaluate both the good and bad points of your approach.

Guidelines:

1. You are not allowed to use existing RL libraries. Write the algorithm yourself.

2. You are not allowed to use deep reinforcement learning approaches or RL algorithms that you haven't learnt in the course.

3. For the most successful experiment, in each exercise, Show (**In the notebook and report**) the number of steps it took the agent to solve the environment, and a convergence graph (Rewards\Steps according to the number of episodes). All training outputs will be displayed in the notebook.

4. Add video clips (In the notebook) that show the agent in the middle of the training process and after the learning completed and converged.

5. Even if you did not fully solve an environment - supply graphs\tables describing the average rewards on 100 episodes (after finishing the training phase).
   Submit a final report in pdf format when your ID numbers are included in the file name.

6. You need to supply your code as Google Colab notebooks link. Such that the course team can run it.

7. Write clean code! Separate code cells and make extensive use of text cells. A sloppy notebook will result in a lower grade.

8. Do not change anything in notebook after the final submission date. A notebook that has been run/changed after the submission date will be automatically disqualified.

9. The **final report** must be a professional, self-contained scientific document that fully explains your methodology and results without the need to reference the code.

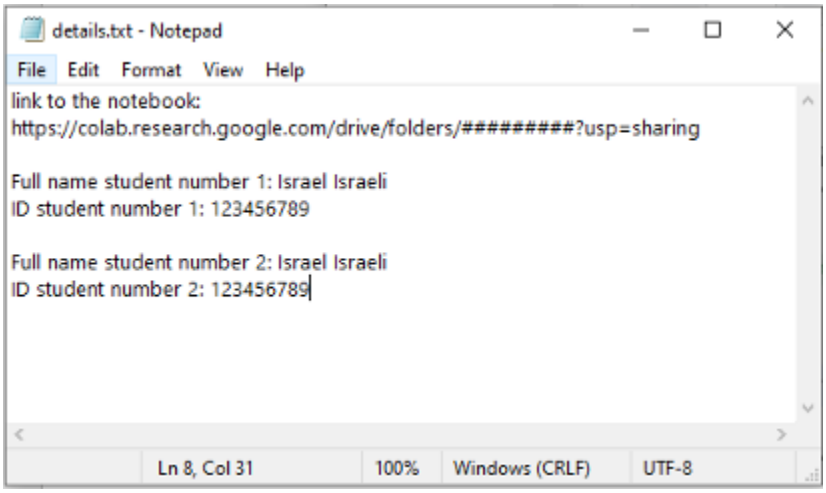
The report will be submitted to the submission box. The file name of the report will contain your IDs as follows: **report_ID1_ID2.pdf**

Do not forget to include the project title, your name and ID in this file.
Max number of pages: 10 (but you don't have to use them all !!).

The notebook will be presented in an orderly and clean manner; it will contain separate code cells and text cells that explain the actions performed.
** **Very important** ** - when submitting, the notebooks will contain **all the outputs** relevant to the training results.

To the submission box you may also submit an **explainer** file that contains instructions and explanations on how to operate your notebook and other relevant details that need to be known to those who want to use your notebook.

Enter your details in the text file named **submit.txt**, the address of your notebook, the names and ID of the two partners, as follows:



To sum up, in the submission box, submit the following files:

1. details.txt
2. report_ID1_ID2.pdf
3. explainer.txt (optional)

Team size
The project will be performed in groups of **2 students**.

Academic Integrity

Team/Student may not copy code from other teams/students. Copying answers or code from other students for a project is a violation of the university's honor code and will be treated as such. All suspicious activity will be reported to the head of the department and the university authorities.

Giving code to another student is also considered a violation. Students are responsible for protecting their own work from copying.

If you build some of your code on existing work and utilize existing code (your own or code found on the web), you must give proper attribution to all existing work that you used and make it clear what you changed and contributed. Any unattributed or uncited work that you use will be considered a breach of academic honesty and dealt with according to the course policy in the syllabus.