# Evaluation: Instructions

Hédi Hadiji

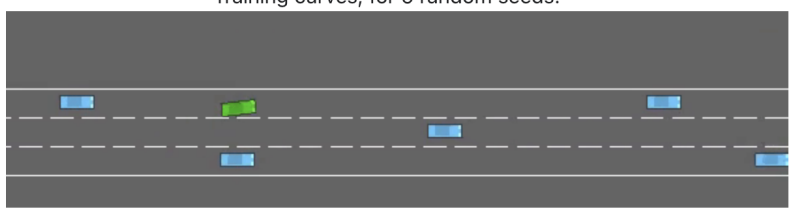
CentraleSupelec
*hedi.hadiji@l2s.centralesupelec.fr*

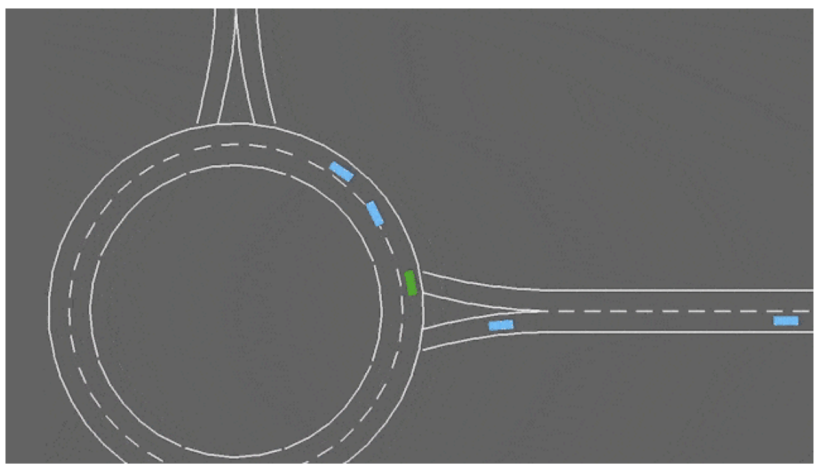
Mention IA
24-25

# Goal of the project: Implement Highway agents

http://highway-env.farama.org/

Training curves, for 5 random seeds.





Train little cars to drive in different road environments.

# Instructions

- Form groups of at most 4 students
- Register the groups by April 1st
- **Individual reports** detailing individual contributions
- Give the report by April 22nd

Registering the groups means declaring the members of the groups and uploading two configuration files specifying your environment of choice (see next slides).
For each group, you will train 3 agents

- One a pre-specified environment that I will provide
- One an environment of your choice, with *continuous actions*
- One an environment of your choice, using standard implementations from the StableBaselines library

# Config files

Parameters of an environment are defined via a config file.
Read the Documentation to see how to use these config files
In particular: they define

- the type of observations (e.g. the speed of other vehicles, whether the positions are relative to the agent or absolute on the grid, etc.)
- the type of actions ('meta-actions' like 'go to top lane', 'continuous action' specify throttle and steering angle, see the doc)
- the rewards (how bad a collision is, stepping out of lanes)

but also the size of the lanes, the size of the screen, etc.

# Task 1: Pre-specified environment, your implementation

With the configuration file given in 'config.py'. Implement an instance of a DQN from scratch (you may use the code of Lab sessions). Document the training perfomance and the behavior of your agent. Describe and explain the different phases in learning you observe.

the config file will be uploaded on March 16th

# Task 2: Continuous actions, your implementation

Choose an environment and configure it to continuous actions. Implement the algorithm of your choice, possibly using code from Lab sessions. Document the difference in behavior with your results on highway with discrete actions.

# Task 3: Your choice of environment, Stable-baselines

Choose a third environment and use the StableBaselines library to train an existing algorithm. Build your code on the examples given here

# Task 4: An extra-experiment

In one the environements defined above, choose an experiment to study an aspect that interests you in the task and algorithm. Example questions

- what is the impact of hyperparameters on performance?
- does your trained algorithm generalize to other environments with modified parameters?
- can you make your algorithm learn more 'safely'?
- Does your algorithm implicitly predict the outcomes of the MDP? If so, is it accurate?
- what happens if you have your trained model drive two cars at the same time?

Be creative! For this part, it's okay to have mostly negative results. Any reasonable attemtp to solve the problem is good (e.g., an attempt to have less crashes during training, even if it ends up not working).

# Grading

Out of 20:

- 9 points for solving the environments (3 per task). Providing an agent and code that consistently obtains high reward. The code should run, be legible, and the outcome clear.
- 6 points (2 per task) for describing the algorithm and the training procedure. All techniques used to make the learning happen, and to monitor the agents should be described.
- 5 points on the extra experiment. Formulate a clear question/hypothesis, describe and perform an experiment to try to answer the question.