# Reinforcement Learning Project

Welcome to your Reinforcement Learning project! **Join in groups of a maximum of 5 students** on a project focused on developing an RL agent capable of solving an environment for decision-making in Autonomous Driving. **The project deadline has been set to the 2nd of June.**

Autonomous Driving has long been considered a field in which RL algorithms excel, and this project aims to leverage the power of RL to create an intelligent agent that can solve the Farama's foundation "highway-env" project, namely the Highway environment (refer to https://highway-env.farama.org/environments/highway/).

## Project Requirements:

- The **environments observation's** format can vary according to our preference, namely Kinematics, Grayscale Image, Occupancy grid and Time to collision (refer to https://highway-env.farama.org/observations/). In your solutions you should use only 2 of these types.
- The **agents actions** can also vary, as continuous actions, discrete actions and discrete meta-actions (refer to https://highway-env.farama.org/actions/). In your solutions you should use only 2 of these types.
- As for the **algorithms to use** any algorithm is valid (seen or not in class), with a minimum requirement of 3 different algorithms used.
- Apart from the environment observation types and agent action types you **must use environment's configuration provided** in the annexed notebook!

Note: Your delivery should comprise **4 solutions** to the highway environment (corresponding to the combinations of the two environment observation's types and the two agent's action types), in which you just need to **use one algorithm for each combination** (knowing that you need to use at least 3 different algorithms).

## Project Objectives:

- Train an RL agent to solve the Highway environment: The primary objective of this project is to develop an RL agent that can maximize the reward given by the highway environment (refer to https://highway-env.farama.org/rewards/), which leverages to maximize speed while minimizing crash risk!
- Optimize decision-making using RL algorithms: Explore different RL algorithms to train the agent. Compare and analyse their effectiveness in learning and decision-making capabilities in the context of the environment.
- Explore and expand on the reward system: Although you should evaluate your agent with the reward function provided by the environment, you could/should expand it to better train your agent.
- Enhance interpretability and analysis: Develop methods to analyse the agent's decision-making process and provide insights into its strategic thinking. Investigate techniques to visualize the agent's evaluation of chess positions and understand its reasoning behind specific moves.

### Extra Objectives:

- Investigate transfer learning and generalization: Explore techniques for transfer learning to leverage knowledge acquired in related domains or from pre-training on large chess datasets. Investigate the agent's ability to generalize its knowledge.
- Explore multi agent approaches: The environment allows you to use more than one agent per episode. Explore multi agent alternatives to improve your learning times and overall benchmarks.

# Deliverables

## Code

- **Four RL Agents**: This deliverable consists of a Jupyter notebook capable of replicating your reported results.
- **Documentation**: Provide a well-documented Jupyter notebook that includes the implementation of the RL agent, chess environment, and any supporting scripts or utilities attached. The documentation should cover the setup instructions, usage guidelines, and explanations of key components and algorithms used in the project.

## Report

A report consisting of a maximum of 10 pages or 3000 words (without cover/bibliography) with:

- An introduction describing your overall approach to the problem, the issues associated with the overall sheer number of states and more.
- A methodological description of all different learning algorithms used and the respective reasoning for their choice.
- A comprehensive evaluation that analyses the performance of the RL agent against the different learning algorithms.
- Don’t forget to develop useful visualizations to provide a better understanding of the agent's decision-making process.

# Project Evaluation

- The usage and argumentation of different learning methods will be the focus of the project’s evaluation process. Choose appropriate learning algorithms for each observation and/or action type and argue their relevance and usefulness in solving the problem.
- While the overall performance of your RL agent is not to be part of the evaluation, it is rather essential to understand how different learning methods improve the performance of your agent.
- It is important that you develop a comprehensive evaluation to the extent to which the agent's reasoning can be understood. Assess the clarity and interpretability with visualizations.
- You can achieve the maximum score without the Extra Objectives, which means in practice groups can have “more than 20” in the project.