ECE 553/653 Neural Networks

# Project 3: Multi-agent reinforcement learning

You will design machine learning algorithms for optimizing the game predators and prey. Your algorithms will be responsible for optimizing the behaviors of predators and prey.

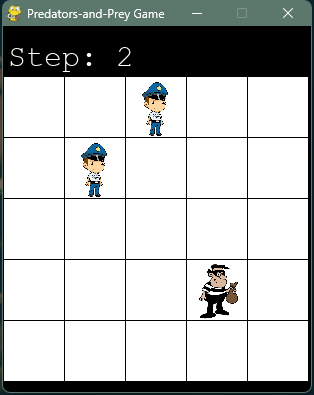
# Requirements

You will write **two** programs: a program that optimizes the behaviors (strategies) of predators, and a program that optimize the behavior of the prey. It is recommended that your programs are written in python.

For the program of the predators (police), you will propose a reinforcement learning (RL) (i.e., Q-learning, deep Q) to optimize the strategy of one predator or multiple predator. You have to design the neural network architecture of predators, and complete the VDN algorithm with reference to the IQL algorithm.

For the program of the prey (thief), you can either optimize the default strategy of the prey or propose a RL method to optimize the strategy of the prey.

# Program Introduction



The Scenario.

The program comprises the following components:

1. **"main.py"**: This file includes the initialization of the environment, Multi-Agent Reinforcement Learning (MARL) parameters, and the selection of game parameters.
2. **"controller.py"**: Responsible for designing MARL algorithms, such as Independent Q-Learning (IQL), Value Decomposition Networks (VDN), and QMIX.
3. **"agent.py"**: This file encompasses the design of predator agents, with a specific focus on the neural network design found in the "Brain.build\_model" function. The provided solution features a basic design with a single Dense Layer connecting state to action. Experimentation with more intricate network designs is encouraged.
4. **"env.py"**: Defines the Predators and Prey Game board.
5. **"results\_predators\_prey"**: This directory stores the training results.

From these components, the following tasks can be identified:

1. **Neural Network Design**: Investigate the "Brain.build\_model" function in "agent.py" for the neural network design. The current solution involves a simple Dense Layer from state to action. Consider implementing more complex network designs to enhance functionality.
2. **Parameter Selection**: Adjust training parameters following the guidance in "main.py" and the provided information in "Readme.md". The basic parameters are outlined in the Readme, but you are encouraged to experiment with other adjustable parameters to achieve better model performance.
3. **VDN Algorithm Implementation**: Complete the implementation of the VDN algorithm in "agent.py". Reference the IQL and QMIX algorithms provided and focus on completing the missing parts in the "controller.VDNreplay" function in "agent.py".
4. **Prey Escape Strategy Design**: Develop a strategy for the prey's escape in the "PredatorsPrey.actor\_prey\_designed" function in "env.py".

For additional code details, please consult the "README.md" file.

# Report

You will write a report for this project. There is no strict format of this report but below are the essential parts:

1. How you design the predator’ neural network architecture.
2. A screen shot of the results in a successful run of your programs.
3. Try different configurations of training and report the results.
4. How you design the prey’ escape strategy.
5. Evaluate the efficiency of your algorithms (e.g. the time used for training, the size of the model).

There are other aspects that you can explore and report, which are up to you. More comprehensive study will receive more credits. Considering multiple predators instead of one predator will also receive more credits.

\* Do not paste a lot of codes in the report. Focus on the design and analysis. Your codes will be submitted separately.

# Submitting your project

You should submit your project in Blackboard. The submission includes: a project report and your source codes. Please put everything in a zip/rar file, name it “FirstName\_LastName\_Project3.zip”.

# Advice

Start early! Do not wait until the last few days.