# MAS: Activity 3
# Predator - Prey with communication

## Alexandru Sorici

**The Problem:** we continue with the Predator - Prey scenario set up in Activity 2. Please defer to documentation for the previous activity to view specifications of the environment and agent setup. In this activity we introduce an additional fundamental ability of cognitive agents: (direct) communication.

Specifically, the predator agents now have the added ability to send messages to other predators that are known to them. Predators can only send messages to other predators they perceive nearby, or which they *remember* from previous encounters.

The purpose of the game is for the predator agents to eat all prey agents on the map. To show that communication leads to improved strategies for the predators, use your existing Activity 2 implementation (cognitive predator agents with no communication ability) as baseline.

Afterwards, for a chosen map configuration, compare the average number of steps taken to finish the game in Activity 2 with that from this Activity and show that the communication ability leads to lower step counts.

### Specifications

- the agents have four operations: *UP, DOWN, LEFT, RIGHT*.
- prey agents can perceive tiles at a maximum Manhattan distance of 2.
- predator agents can perceive tiles at a maximum Manhattan distance of 3.
- the agent's perception is modelled as a `MyPerceptions` structure, containing information about its current position, obstacles, nearby agents, as well as **the set of messages received from other agents**.
- two agents **can** be located in the same tile

- predator agents **are allowed to use memory** and **communication**.
- prey agents are reactive - they employ a random walk strategy, trying to stay away from predators if they perceive them.

- a prey agent is considered killed in the following conditions:
    - there is at least one predator at a Manhattan distance of 1 to the prey agent
    - there are at least two predators at a Manhattan distance of 2 to the prey agent

Work **only** in the **my** package. If you need anything else, discuss with the assistant.

**To Do 1:** Use your code from the previous activity combined with the new given code (see page 2).

**To Do 2:** Implement in `MyEnvironment` the code for the management of messages. Messages sent at one step will be delivered in the next step.

**To Do 3:** Design and implement a cognitive behavior, involving communication with other agents, for the predators, showing that this leads to a lower step count than without the information exchange ability: improve both **map exploration** (don't stay in the same place, explore different regeions) and **hunting down prey** (converge quickly on prey).

**Source code changes:**

- added the `communication` package containing `AgentMessage`, `AgentID` and `SocialAction`.

- `MyPerceptions.nearbyPredators` is now a `Map` that also contains the `AgentID`s of the predators as keys (and their positions, as values)
    - both `MyPerceptions` constructors are changed accordingly;
    - perception generation for prey is changed accordingly.
    - `MyPrey` code is changed accordingly.

- `MyPerceptions` also contains a `Set` of received `AgentMessage`s. There is a gettern and a new constructor for `MyPerceptions` that initializes the message set.

**Helpful pointers:**

- An `AgentMessage` contains a sender, a destination and some content. The sender and the destination are `AgentID` instances. The content can be whatever you want.

- Generate necessary `AgentID`s using the static method `AgentID.getAgentID(Agent)`.

- For predators, `response` should now return an instance of `SocialAction` instead of `MyAction`. "Social actions" contain the physical action to perform (movement) and also messages to be sent to other agents.

- Messages from the returned `SocialAction` instance must be kept from one step to another in `MyEnvironment.messageBox`. **Careful:** don't store messages from the current step and from the previous step simultaneously in the `messageBox`.

---

Cum să raportați activitatea:

- **la sfârșitul laboratorului:** trimiteți arhiva conform cu instrucțiunile de mai jos.
- **la terminarea taskurilor** aferente laboratorului (înainte de următorul laborator, altfel cu depunctare): trimiteți din nou arhiva, conform cu aceleași instrucțiuni, eventual adăugând ceva la nume.

**Conținutul arhivei: numai** directorul `src`, arhivat într-o arhivă cu numele `PrenumeNume_MAS-N.zip`, unde `N` este numărul laboratorului pe care l-ați rezolvat.

**Cum trimiteți:** trimiteți arhiva în atașament la un mesaj către adresa cs+mas@andreiolaru.ro. Dacă adresa este corectă și există atașament, veți primi un mesaj automat de confirmare.

**Notă:** Folosiți adresa de mai sus numai pentru a trimite activitatea de laborator. Pentru alte probleme folosiți modalitățile de contact indicate la curs.