

# **Rapport de projet - Défi IA**

Simulation d'un système proie-prédateur à stratégies multiples

Antoine Schoonaert, Bastien Schoonaert et Lila Mekki

# Table des matières

1. Aperçu du projet.....	2
1.1. Historique des modifications.....	2
1.2. Contexte.....	2
1.3. Présentation de la solution multi-agents.....	3
2. Modèle conceptuel.....	5
2.1. Agent.....	5
2.2. Environnement.....	6
2.3. Interactions.....	8
2.4. Organisation.....	8
3. Implémentation.....	9
3.1. Du modèle conceptuel au modèle de simulation.....	9
3.2. Implémentation des indicateurs.....	10
4. Évaluation.....	11
4.1. Champ de vision des proies (scénario n°1).....	11
4.2. Nombre de prédateurs et E0 vs. E1.....	12
4.2.1 Nombre de prédateurs avec E0 (scénario n°2).....	12
4.2.2 Nombre de prédateurs avec E1 (scénario n°3).....	12
4.1. Champ de vision des prédateurs (scénario n°4).....	13
4.3. Discussion.....	14
5. Conclusion.....	14
Références.....	15

# 1. Aperçu du projet

Notre projet a pour but de simuler un modèle proie-prédateur. Les proies doivent éviter de se faire attraper par les prédateurs et l'objectif des prédateurs est d'attraper un maximum de proies en un temps (nombre d'itération) imparti. L'objectif de notre travail est d'améliorer la stratégie prédateurs, afin de maximiser le nombre de proies attrapées à l'issue de chacune des simulations. Les proies tendent à se déplacer en troupeau et fuient les prédateurs lorsque l'un d'entre eux est détecté. A l'inverse, les prédateurs vont adopter différentes stratégies afin de contrer celles des proies. Durant ce projet, diverses stratégies seront proposées pour les proies d'une part, et pour les prédateurs d'autre part. Une évaluation des performances de ces dernières, les unes par rapport aux autres, sera effectuée, et l'influence de divers paramètres de la simulation sur les résultats obtenus sera investiguée.

## 1.1. Historique des modifications

05/04/2023 : Version initiale

15/04/2023 : Compréhension de la librairie Maqit Simulator et des fonctions à coder

21/04/2023 : Rédaction de la section 3 et détermination des indicateurs clés de notre simulation

27/04/2023 : Avancement dans l'implémentation de la solution

05/05/2023 : Reprise du code car implémentation trop ambitieuse

10/05/2023 : Avancement dans l'implémentation

12/05/2023 : Actualisation du rapport

14/05/2023: Actualisation du rapport/ Finalisation de l'implémentation

## 1.2. Contexte

- Quel système ? Quelles sont les caractéristiques de l'environnement ?

Notre système est composé de proies et de prédateurs dans un environnement clos. Les prédateurs prennent la décision d'attaquer ou non la proie selon leur position et les données qu'ils ont sur l'environnement, et en fonction de la stratégie choisie. Les proies doivent échapper aux prédateurs, elles peuvent pour cela fuir. L'environnement est statique (il ne change pas au long de la partie, à part les positions des prédateurs et proies) et fermé (les proies et prédateurs ne peuvent s'extraire de l'environnement et aucun autre agent ne peut rentrer dans le système, bien que des proies peuvent disparaître). Une proie est considérée comme prise lorsqu'un prédateur l'attaque. Dans ce cas, elle est supprimée de l'environnement.

- Quelles sont les hypothèses et les contraintes auxquelles votre solution doit répondre ?

L'environnement est fermé, les proies et prédateurs ne peuvent en sortir. Le nombre de proies et de prédateurs est fixé au cours d'une partie. Les vitesses des agents sont fixes et identiques. Il y a un seul type d'obstacle et les proies et prédateurs sont obligés de les contourner.

La proie ne perçoit le prédateur que dans un rayon de perception déterminé. Par défaut, les proies ont un mouvement aléatoire. Elles tendent à se regrouper en troupeau, sans pour autant communiquer directement les unes avec les autres. Il s'agit plutôt de simuler un modèle de type Aoki [\[1-2\]](#) qui permet de simuler l'agrégation de certains animaux durant leurs déplacements (bancs de poissons, nuées, d'oiseaux, ...).

Ce type de modèle permet d'avoir une population de proies qui se déplace sous forme de troupeau, sans pour autant qu'il y ait de communication directe entre des différents membres de cette population, et sans besoin d'avoir de leader désigné. Lorsqu'une proie perçoit un prédateur dans son champ de vision (Field of View ou FoV), elle prend la fuite.

Le prédateur dispose quant à lui de son propre champ de vision, dont l'étendue peut être différente de celle des proies. Lorsqu'un prédateur détecte une proie, il peut s'attaquer ou non, en fonction de la stratégie retenue. Dans le cas d'une stratégie où le prédateur a un choix à effectuer, ce dernier est basé sur une *fonction de coût* qui prend en compte le nombre de proies à proximité, le nombre de proies à proximité des autres prédateurs qui communiquent éventuellement avec lui, et la distance à ces derniers. Les stratégies d'évitement pour les proies et de chasse pour les prédateurs sont définies à l'avance et ne varient pas durant la simulation.

- Pourquoi une solution multi-agents est-elle adéquate pour résoudre ce problème ?

Les proies et les prédateurs sont des individus autonomes qui vont prendre des décisions en fonction de leur environnement proche. Par conséquent, la solution multi-agent est particulièrement adaptée. En effet, chacun des prédateurs et chacune des proies sera modélisée par un agent qui prendra des décisions en fonction des informations qui lui sont localement accessibles. De plus, dans le cadre d'une stratégie plus évoluée, les prédateurs sont également capables de communiquer les uns avec les autres, et de partager une partie des informations locales auxquelles ils ont accès. En fonction du contenu de cette information supplémentaire, les prédateurs peuvent prendre des décisions plus éclairées.

- Quels sont vos indicateurs ? Qu'allez-vous observer ?

Notre indicateur est le pourcentage de proies attrapées à la fin du temps imparti. On pourra en particulier tracer l'évolution de cet indicateur en fonction du nombre d'itérations pour différentes stratégies des proies et des prédateurs, et pour différents paramétrages du système.

### 1.3. Présentation de la solution multi-agents

On propose ci-dessous une présentation du modèle multi-agent ainsi que des stratégies qui seront mises en œuvre pour chacun d'entre eux.

- *Agents* : Les agents (proies ou prédateurs) prennent les décisions dans notre système. Les proies cherchent à se regrouper et se déplacent aléatoirement dans l'espace tant qu'ils ne détectent pas de prédateur, et fuient ces derniers. Les prédateurs se déplacent selon la stratégie choisie, et ont pour but d'attraper les proies. Une proie attrapée n'existe plus dans l'environnement.
- *Environnement* : L'environnement a deux dimensions. (1) La dimension physique contient des obstacles infranchissables et offre un terrain de jeu au sein duquel les différents agents peuvent évoluer et mettre en œuvre leurs stratégies. La vitesse des agents est identique pour tous. En revanche, les proies sont autorisées à se déplacer en diagonale lorsqu'elles sont pourchassées, ce qui n'est pas le cas des prédateurs. (2) La dimension de l'espace de données permet aux prédateurs de communiquer des informations au sujet de leur environnement local, selon la stratégie choisie.
- *Interaction* : Les proies n'interagissent pas directement les uns avec les autres. Leur comportement est régi par un modèle dépendant de l'environnement local de chacun des individus de la population

considérée. Aucun message n'est échangé entre les proies. En revanche, selon la stratégie choisie, les prédateurs peuvent communiquer à chaque itération (voir stratégie **S2** ci-dessous) et partager un indicateur calculé à partir du nombre de proies dans leur environnement local. Cet indicateur sert aux autres prédateurs à décider s'il convient de se rapprocher d'un autre prédateur ou de continuer à pourchasser une proie éventuelle. La décision est prise après l'évaluation d'une fonction de coût.

- *Organisation* : Les proies ont plusieurs stratégies afin d'éviter les prédateurs et se déplacent en général dans 4 directions (connexité N4) [3].
  - **E0** : La proie cherche à s'éloigner du prédateur le plus proche.
  - **E1** : La proie cherche à se diriger vers la position de son environnement local qui lui permet d'être le plus éloigné de tous les prédateurs situés dans son FoV.

Quelle que soit la stratégie utilisée, les proies utilisent l'algorithme  $A^*$  [4] pour décider du chemin optimal pour fuir. Dans ce cas uniquement, les proies sont autorisées à se déplacer en diagonale (connexité N8), ce qui permet d'augmenter leurs chances d'échapper aux prédateurs. Ces derniers ont quant à eux trois stratégies qu'il est possible d'adopter et se déplacent dans 4 directions (connexité N4) uniquement.

- **S0** : Le prédateur se déplace aléatoirement dans l'espace. Si une proie se trouve à proximité immédiate, alors il peut l'attraper.
- **S1** : Le prédateur repère la proie la plus proche dans son FoV et se déplace vers elle. Le plus court chemin est calculé à partir de l'algorithme  $A^*$ .
- **S2** : Chaque prédateur  $a$  associe un score à son environnement local de la façon suivante :

$$S_a = \sum_{i=1}^{n_p} 1/d_i \quad (1)$$

avec  $n_p$  le nombre de proies dans l'environnement local et  $d_i$  la distance du prédateur à la  $i$ -ème proie. Ce score est ensuite partagé à l'ensemble des prédateurs. Chaque prédateur  $a$  peut alors évaluer la fonction de coût suivante :

$$F_a(b) = S_b/d(a, b) \quad (2)$$

avec  $d(a, b)$  la distance dans l'environnement global entre les prédateurs  $a$  et  $b$  et  $F_a(a) = S_a$ . La maximisation de la fonction  $F_a$  permet de déterminer le comportement du prédateur  $a$ .

- Si  $\text{argmax}(F_a) = a$ , alors le prédateur  $a$  suit la stratégie **S1**.
- Si  $\text{argmax}(F_a) = b \neq a$ , alors le prédateur  $a$  va déterminer au sein de son environnement local, en utilisant l'algorithme  $A^*$ , le chemin lui permettant de se rapprocher le plus de la position du prédateur  $b$  dans l'environnement global.

Pour obtenir des déplacements plus imprévisibles et simuler un comportement plus *organique* des agents, chacune des proies suivant les stratégies **E0** ou **E1** et chacun des prédateurs suivant les stratégies **S0** ou **S1** ont entre 10% et 20% de chance de se déplacer aléatoirement vers un emplacement valide et accessible.

Conceptuellement, la stratégie d'évitement des prédateurs par les proies **E0** peut être vue comme l'application d'une simple force répulsive. En revanche, la stratégie **E1** est plus subtile en ce sens qu'elle nécessite le calcul de cartes de distances [5] avant d'appliquer l'algorithme A\* sur des graphes correspondant à la pondération des environnements locaux des proies par ces mêmes cartes de distance. Il s'agit néanmoins ni plus ni moins que de déterminer le chemin quasi-optimal permettant de s'éloigner le plus possible des prédateurs, en minimisant le risque de se faire attraper. La dernière section permettra de quantifier l'écart de performance de ces deux stratégies pour ce qui est de la survie des proies.

Pour ce qui est des stratégies mises en place par les prédateurs, la stratégie **S0** correspond pratiquement à une marche aléatoire [6] et peut, dans une certaine mesure, être assimilée à un processus Markovien [7]. Cela n'est toutefois pas réellement le cas, puisque l'aspect aléatoire est suspendu lorsque le prédateur détecte une proie à proximité dans l'environnement N4. Il s'agit bien entendu d'une stratégie peu performante, mais qui obtient malgré tout des résultats non-nuls, bénéficiant principalement de la tendance qu'ont les proies à se regrouper sous forme de troupeau, ce qui diminue parfois considérablement leur capacité à fuir rapidement. La stratégie **S1** diffère de la stratégie **S0** en ce sens qu'elle consiste pour un prédateur à atteindre le plus rapidement possible la proie la plus proche à partir du moment où elle a été identifiée au lieu d'espérer que cette dernière se retrouve naturellement dans son voisinage N4. Il s'agit donc d'une stratégie considérablement plus efficace, puisque les prédateurs se mettent en chasse des proies et les poursuivent tant qu'elles sont dans le FoV. Enfin, en ce qui concerne la stratégie **S2**, la marche aléatoire est complètement effacée à partir du moment où l'un des prédateurs possède une proie dans son champ de vision. A chaque instant, le prédateur doit choisir entre se rapprocher d'un congénère et chasser en meute, ou poursuivre sa proie. Ce choix est déterminé par l'évaluation de la fonction de coût défini par l'équation (2). La stratégie **S2** est en particulier adaptée lorsque la population de proies diminue, et que certains prédateurs peuvent se retrouver sans la moindre proie dans leur environnement local. Le temps d'inactivité des prédateurs est ainsi réduit de manière significative.

## 2. Modèle conceptuel

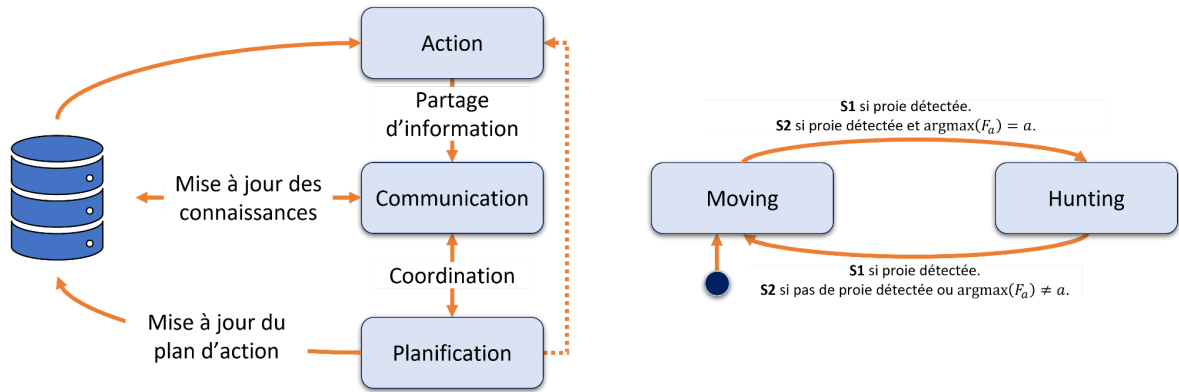
### 2.1. Agent

Nos agents sont les proies et les prédateurs. On considère que les prédateurs peuvent communiquer à n'importe quelle distance. Ils adaptent leur comportement, selon la stratégie sélectionnée, pour attraper le plus de proies possible.

Les prédateurs se comportent selon le modèle suivant et illustré par la figure 1 :

- *Perception* : Vision des proies, des prédateurs et des obstacles dans un certain FoV dont l'étendue est définie au préalable. Perception des positions dans l'espace global de tous les autres prédateurs et des scores  $S_a$  définis par l'équation (1) associés à leurs environnements respectifs.

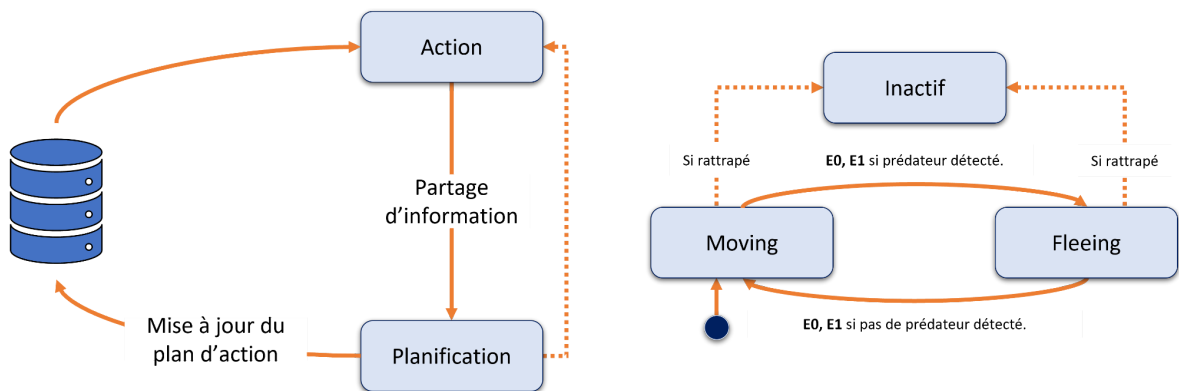
- *Connaissances* : Chaque prédateur connaît la stratégie définie, mais n'a pas de connaissance de son environnement autre que celle contenue dans le FoV. Il a également connaissance des positions de ses congénères et des scores  $S_a$  que chacun d'entre eux attribue à son environnement local.
- *Processus de décision* : Le prédateur prend une décision en fonction de la stratégie **SX** décidée en amont de la simulation.
- *Modèle d'action* : Un prédateur peut effectuer deux actions: *moving* et *hunting*.



**Figure 1** : Description conceptuelle du modèle d'agent pour le prédateur.

Les proies, quant à elles, se comportent selon le modèle suivant et illustré par la figure 2 :

- *Perception* : Vision des proies, des prédateurs et des obstacles dans un certain FoV dont l'étendue est définie au préalable.
- *Connaissances* : Chaque proie connaît la stratégie définie, mais n'a pas de connaissance de son environnement autre que celle contenue dans le FoV.
- *Processus de décision* : La proie prend une décision en fonction de la stratégie **EX** décidée en amont de la simulation.
- *Modèle d'action* : Une proie peut effectuer deux actions: *moving* et *fleeing*.

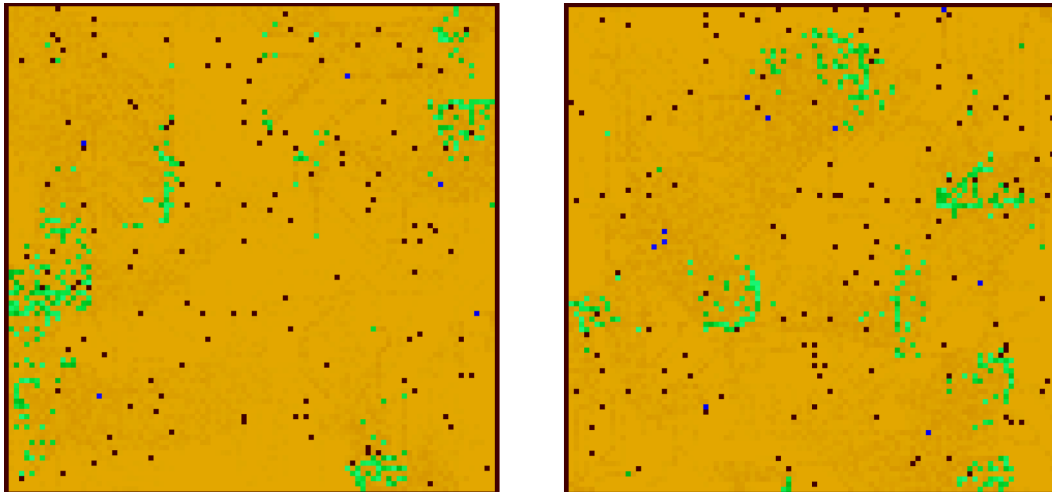


**Figure 2** : Description conceptuelle du modèle d'agent pour le prédateur.

## 2.2. Environnement

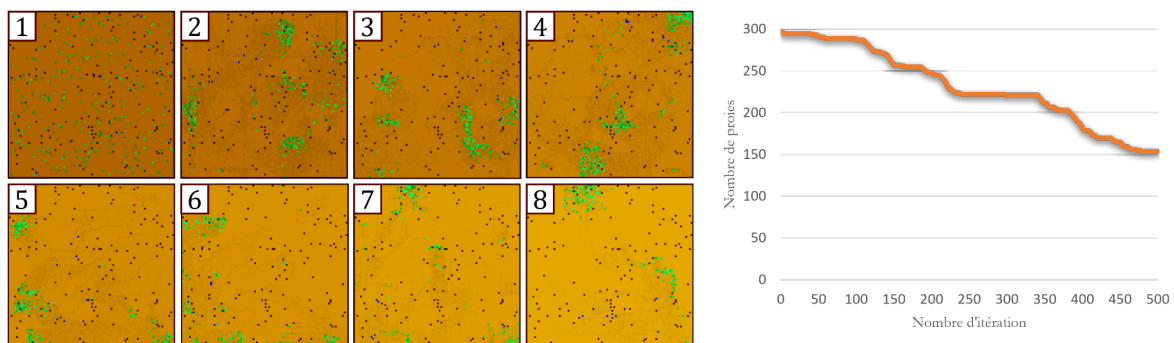
L'environnement est un espace fermé, contenant un certain nombre d'obstacles placés aléatoirement, ainsi qu'un ensemble d'agents, constitué de deux sous-ensembles: l'un de proies, et l'autre, de

prédateurs. Il permet aux proies de se structurer sous forme de troupeaux, en leur fournissant une information partielle quant à leur environnement local, et permet aux prédateurs de prendre des décisions concernant leurs actions. Les communications sont autorisées entre tous les prédateurs, quelle que soit la distance qui les séparent. Les proies ne communiquent pas entre-elles.



**Figure 3 :** Exemple de représentation graphique de l'environnement (300 proies, 10 prédateurs, grille de taille  $96 \times 96$ , stratégies **E1** et **S2**). Légende : ■ Background, ■ Obstacle, ■ Proie, ■ Prédateur.

La figure 3 propose des exemples d'environnements au sein desquels les agents peuvent évoluer. Les proies et les prédateurs laissent des traces au sol après leur passage, qui disparaît avec le temps. Une idée était d'autoriser les prédateurs à *pister* les proies à partir de leurs traces, mais cette fonctionnalité n'est pas encore implémentée. On peut constater que les proies se regroupent bien naturellement sous forme de troupeau, alors que ces dernières sont distribuées de manière uniforme dans l'espace au moment de la génération de l'environnement. On peut également remarquer des groupes de prédateurs en train de chasser sur l'image de droite (au milieu en haut et au milieu gauche). En effet, les stratégies utilisées pour générer ces simulations sont **E1** pour les proies, et **S2** pour les prédateurs, ce qui encourage ces derniers à se regrouper afin de chasser les plus gros troupeaux.



**Figure 4 :** Exemple d'évolution de l'environnement (à gauche) et de la population de proies durant une simulation (à droite) avec les mêmes paramètres que pour la figure 3 (300 proies, 10 prédateurs, grille de taille  $96 \times 96$ , stratégies **E1** et **S2**). Les images de 1 à 8 illustrent l'état du système de 1 à 500 itérations.



Lors de l'initialisation de la simulation, les obstacles, puis les agents, sont placés aléatoirement de manière uniforme dans l'espace (nettement visible dans la vignette 1 de la figure 4).. Le nombre d'obstacles est fixe, de même que le nombre de prédateurs. En revanche, lorsqu'une proie est attrapée par un prédateur, elle change de *statut* et devient inactive. Elle est alors supprimée de l'environnement. La figure 4 permet d'illustrer ce phénomène. En effet, la courbe d'évolution de la population de proies durant cette simulation montre de fortes évolutions lorsqu'un troupeau est attaqué. À ce moment, les agents correspondants aux proies qui ont été attaquées sont désactivés et supprimés du système. Il n'interviennent plus durant toute la suite de la simulation.

## 2.3. Interactions

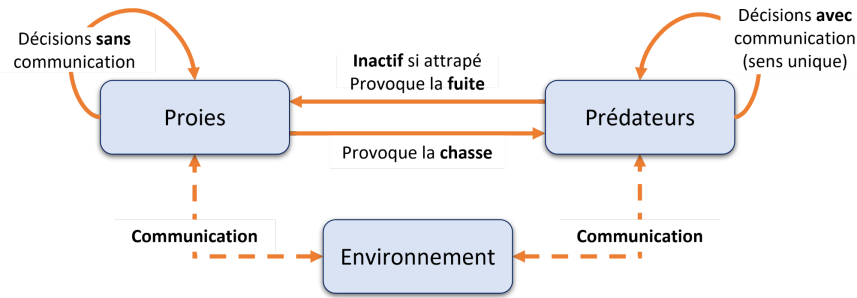
Tous les agents présents au sein de l'environnement se déplacent de manière autonome, en fonction des stratégies définies en amont du lancement de la simulation. Cela conduit à un certain nombre d'interactions entre les agents (proies et prédateurs).

- Le mouvement des agents : En particulier le mouvement des proies, regroupés en troupeaux, peut considérablement diminuer la mobilité de chacun des individus lorsqu'il s'agit de fuir face à un prédateur. Bien que les proies soient, dans ce cas précis, autorisées à se déplacer en diagonale, leur mobilité est malgré tout d'autant plus réduite que la taille des troupeaux est importante. De fait, un grand nombre d'agents dans l'environnement local d'une proie entraîne des difficultés à trouver un chemin optimal pour s'échapper. Ces problèmes sont bien moins présents pour ce qui est des prédateurs, dont l'objectif est précisément d'atteindre les proies.
- La communication entre prédateurs : Les prédateurs interagissent à chaque itération dans le cadre de la stratégie **S2** en communiquant à leurs semblables le score  $S_a$  qu'ils attribuent à leur environnement local. Dans ce cas, et surtout lorsque le nombre de troupeaux diminue, les prédateurs ont tendance à chasser en meute.

## 2.4. Organisation

L'organisation du système repose sur un modèle non hiérarchisé pour les deux types d'agents: proies et prédateurs. Les proies évoluent de manière autonome en suivant un certain nombre de règles qui les conduit à se regrouper sous forme de troupeau et à se déplacer de manière globalement aléatoire au sein de l'environnement. Ce phénomène d'agrégation de la population de proies est une conséquence des règles qui s'appliquent au niveau individuel et ne nécessite pas de communication parmi les proies. De même, en cas de fuite, les proies adoptent la stratégie du "chacun pour soi" et ne communiquent pas davantage entre-elles.

Les prédateurs, quant à eux, communiquent dans le cadre de la stratégie **S2**, mais cette communication est à sens unique. Il s'agit, pour chacun des individus constituant la population de prédateurs, d'actualiser les informations concernant son environnement local propre. Libre ensuite à chacun des prédateurs de prendre une décision, à la vue de ces nouvelles informations, sans pour autant qu'il ne soit nécessaire d'en informer ses semblables.



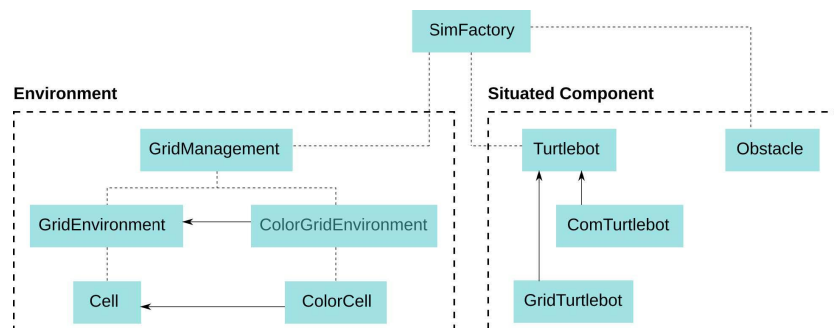
**Figure 5** : Représentation schématique de la hiérarchie et des interactions entre agents au sein du système.

### 3. Implémentation

Dans cette partie, nous détaillons la façon dont notre solution a été implémentée. Elle repose en partie sur la librairie Maqit Simulator. Néanmoins, il s'est rapidement avéré que cette dernière n'offrait pas autant de flexibilité et de souplesse qu'il aurait été nécessaire afin de répondre à nos besoins. On pensera, en particulier, au fait que cette librairie n'a pas forcément été pensée pour simuler des robots de types différentes, de fait que l'intégration des proies et des prédateurs au sein de l'environnement ont nécessité de modifier quelque peu certaines classes, en particulier pour ce qui est de retirer des agents de l'environnement durant une simulation (le statut *inactif* n'étant pas suffisant). La librairie a donc été décompilée et certaines parties modifiées afin de répondre à nos besoins. On pensera également au fait que la couleur du fond était par défaut hardcodée, ce qui nous empêchait d'offrir une expérience utilisateur aussi satisfaisante que nous l'aurions souhaité.

#### 3.1. Du modèle conceptuel au modèle de simulation

La librairie Maqit Simulator est structurée selon la hiérarchie de classe illustrée par la figure 6. Une simulation est instanciée à l'aide d'une extension de l'objet SimFactory et dépend d'un certain nombre de paramètres. Le tableau 1 liste les paramètres nécessaires au bon fonctionnement du programme. Ces derniers permettent non seulement de définir le nombre d'obstacles et la taille de l'environnement, mais également le nombre d'agents, c'est-à-dire le nombre de proies et de prédateurs qui vont évoluer au sein de la simulation.



**Figure 6** : Vue fonctionnelle du simulateur, ie la librairie Maqit Simulator. (Droits: Fabien BADEIG)

**Tableau 1** : Description des paramètres de l'environnement de simulation

Paramètre	Description
<i>nbPredateur</i>	Nombre d'agents prédateurs
<i>nbProie</i>	Nombre d'agents proie
<i>nbObstacles</i>	Nombre d'obstacles aléatoires dans l'environnement
<i>fodPredateur</i>	Étendue du champ de vision des prédateurs
<i>fodProie</i>	Étendue du champ de vision des proies
<i>stratPredateur</i>	Stratégie suivie par les prédateurs ( <b>S0</b> , <b>S1</b> ou <b>S2</b> )
<i>stratProie</i>	Stratégie suivie par les proies ( <b>E0</b> ou <b>E1</b> )
<i>seed</i>	Seed aléatoire
<i>rows</i>	Nombre de lignes de la grille constituant l'environnement
<i>columns</i>	Nombre de colonnes de la grille constituant l'environnement
<i>nbIteration</i>	Nombre total d'itérations
<i>randomWalk</i>	Paramètre contrôlant le côté organique des mouvements (entre .1 et .2 par défaut)

Les proies et les prédateurs sont représentés par deux classes qui étendent la classe `GridTurtleBot`. Les mouvements correspondant à chacune des 5 stratégies possibles (**E0**, **E1**, **S0**, **S1** et **S2**) sont définis au sein de ces classes. À l'initialisation de la simulation, l'environnement est créé, les obstacles, les proies et les prédateurs sont répartis de manière uniforme au sein de l'espace, dans cet ordre. Des répartitions sous forme de clusters, type Neyman-Scott [8], processus de Gibbs [9] ou de Strauss [10] ont été envisagés, mais n'ont pas été explorés.

À chaque itération, la fonction *schedule()* de notre instance de simulation est appelée. C'est cette fonction qui organise la façon dont l'environnement est mis à jour, qu'il s'agisse des mouvements des agents, de la destruction des proies lorsqu'elles se font attrapées par les prédateurs, ou encore de la communication des prédateurs entre eux. En premier lieu, si la stratégie **S2** a été adoptée, alors les prédateurs communiquent à tous les autres le score qu'ils ont attribué à leur environnement local. Chacun d'entre eux prend alors la décision de poursuivre leur proie courante, ou de se diriger vers leur congénère qui se situe dans un lieu plus propice à la chasse. Cette décision est basée sur l'évaluation de la fonction  $F_a$  de l'équation (2), comme décrit au sein de la section 1.3. Ensuite, les prédateurs sont autorisés à se déplacer de manière séquentielle. L'environnement global, ainsi que la perception que les agents ont de leur espace local est mis à jour à chacun des déplacements individuels des prédateurs. C'est alors au tour des proies de pouvoir se déplacer, également de manière séquentielle, à nouveau selon la stratégie adoptée. Il faut noter en particulier que le fait que les prédateurs se déplacent en premier leur fournit un avantage que nous n'avons pas encore été en mesure de quantifier. Durant toute cette étape, si une proie a été rattrapée par un prédateur, son statut passe à *inactif*, ce qui nous permet de l'ignorer durant toute la suite de la simulation, et d'accélérer considérablement le temps d'exécution de cette dernière. Enfin, les environnements locaux de chacun des agents encore existants sont à nouveau mis à jour.

### 3.2. Implémentation des indicateurs

À chaque itération, le nombre de proies ayant été attrapées est enregistré, ce qui permet ensuite de calculer nombre d'indicateurs (la proportion de proies restantes, la proportion de proies ayant été attrapées, la quantité de proies attrapées par itération, ...), sachant que nous nous intéressons en particulier à la proportion de proies ayant été attrapées en fonction du nombre d'itération, et des stratégies mises en place. Une procédure est ainsi implémentée, afin de générer un grand nombre de simulations, avec des *seed* différentes, pour un paramétrage de la simulation donné, et de produire des fichiers de données structurés au

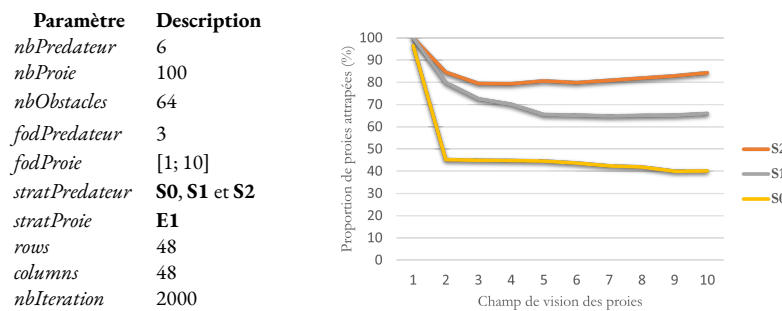
format *.csv* contenant l'ensemble des informations enregistrées durant l'exécution des simulations, ainsi que les valeurs moyenne de notre indicateur, à savoir, la proportion de proies attrapées.

## 4. Évaluation

Dans cette partie, on se propose d'évaluer les stratégies mises en place pour, d'une part, que les proies ne soient pas trop faciles à attraper, tout en conservant leur caractéristique de troupeau et, d'autre part, que les prédateurs puissent maximiser le nombre de proies attrapées. À travers plusieurs scénarios et protocoles, l'influence des paramètres de la simulation est quantifiée, et la qualité des stratégies est évaluée.

### 4.1. Champ de vision des proies (scénario n°1)

Dans un premier temps, on se concentre sur l'influence qu'a le champ de vision des proies sur les résultats de la simulation. Pour ce faire, on fait varier la taille du champ de vision des proies, et on fixe les autres paramètres (différentes stratégies ont toutefois été testées pour les prédateurs). Les résultats sont présentés par la figure 7.



**Figure 7** : Paramétrage du scénario n°1 (à gauche) et présentation des résultats (à droite). Résultats obtenus à partir de 80 simulations différentes.

S'il apparaît clairement que l'étendue du champ de vision des proies a une influence très forte sur les résultats quand celle-ci est réduite, on pourra remarquer que ceci n'est plus vrai à partir d'un certain seuil, qui se situe autour de 2 fois le champ de vision des prédateurs (c'est-à-dire 6 dans ce cas), ce que d'autres scénarios tendent à prouver. De plus, ce simple scénario laisse déjà penser que la stratégie **S2** est supérieure aux deux autres, et ce, de manière très nette.

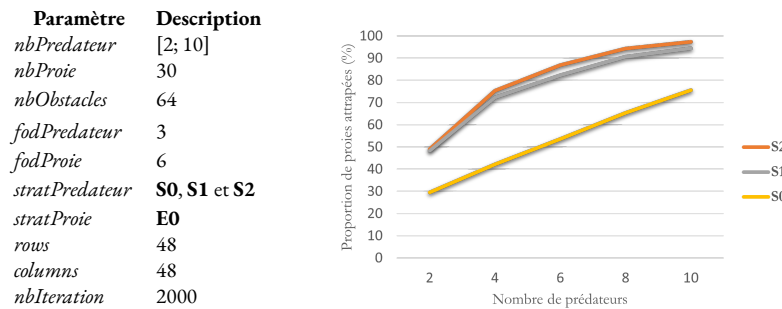
On pourra par ailleurs noter que les résultats tendent à s'améliorer alors que le champ de vision des proies s'agrandit, ce qui est a priori contre-intuitif, car on pourrait imaginer qu'elles voient venir les prédateurs de plus loin. Malheureusement pour les proies, cela signifie également qu'elles repèrent plus facilement leurs congénères, et les troupeaux formés sont donc de taille plus importante, et, de fait, il est plus aisé pour les prédateurs de les attaquer avec succès.

## 4.2. Nombre de prédateurs et **E0** vs. **E1**

Dans cette partie, on propose deux scénarios afin d'investiguer l'influence du nombre de prédateurs, tout en comparant les stratégies d'évitement **E0** et **E1** pour les proies.

### 4.2.1 Nombre de prédateurs avec **E0** (scénario n°2)

On fixe la stratégie **E0** pour les proies et on fait varier le nombre de prédateurs. Cela signifie que les proies ne vont chercher qu'à échapper au prédateur le plus proche, quitte à se mettre à terme en danger. Les trois stratégies possibles pour les prédateurs seront testées.



**Figure 8** : Paramétrage du scénario n°2 (à gauche) et présentation des résultats (à droite). Résultats obtenus à partir de 80 simulations différentes.

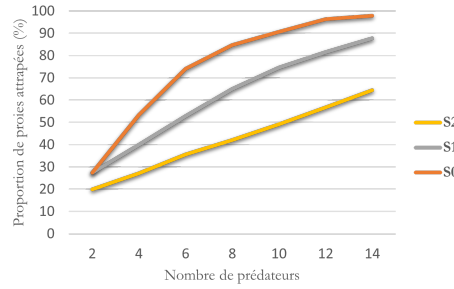
Cette fois encore, on peut remarquer que la stratégie **S2** est supérieure aux deux autres, bien que l'écart soit faible avec la stratégie **S1**. Les résultats obtenus avec la stratégie **S0** augmentent linéairement en fonction du nombre de prédateurs ce qui semble naturel, étant donné qu'avec cette stratégie, les prédateurs ne communiquent pas entre eux et que leur influence sur les proies est *homogène* et *isotrope*, en ce sens qu'il s'agit de comparer une marche aléatoire (**S0**) avec un phénomène d'attraction/répulsion (**S1** et **S2**), ce derniers ayant une influence bien différente sur la dynamique de population à l'oeuvre.

L'influence du nombre de prédateurs est donc plutôt conforme à ce que l'on aurait pu imaginer, c'est-à-dire qu'un nombre plus élevé de prédateurs implique un résultat plus élevé. On notera toutefois que la stratégie **S2** reste au-dessus des autres, alors qu'on aurait pu imaginer qu'une meute dispersée pourrait parfois se révéler plus efficace dans certains scénarios, notamment lorsqu'il reste peu de proies à attraper, ce qui ne semble finalement pas être le cas.

### 4.2.2 Nombre de prédateurs avec **E1** (scénario n°3)

On fixe à présent la stratégie **E1** pour les proies et on fait varier le nombre de prédateurs. Cela signifie que les proies vont cette fois baser leurs décision sur l'ensemble des données à leur disposition, tenter d'atteindre l'endroit qui leur permettra d'être le plus loin possible de tous les prédateurs dans leur champ de vision. Les trois stratégies possibles pour les prédateurs seront à nouveau testées. La figure 9 illustre les résultats obtenus.

Paramètre	Description
<i>nbPredateur</i>	[2; 14]
<i>nbProie</i>	30
<i>nbObstacles</i>	64
<i>fodPredateur</i>	3
<i>fodProie</i>	6
<i>stratPredateur</i>	<b>S0, S1 et S2</b>
<i>stratProie</i>	<b>E1</b>
<i>rows</i>	48
<i>columns</i>	48
<i>nbIteration</i>	2000



**Figure 9 :** Paramétrage du scénario n°3 (à gauche) et présentation des résultats (à droite). Résultats obtenus à partir de 80 simulations différentes.

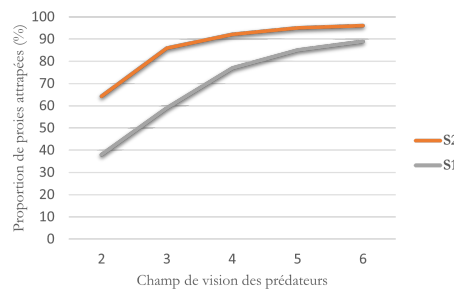
On constate à nouveau que l'influence du nombre de prédateurs est quasi linéaire dans le cadre de la stratégie **S0**. En revanche, il apparaît à présent de manière très claire que la stratégie **S2** est nettement supérieure aux autres dans le cas où les proies agissent avec plus d'intelligence, même s'il nous faut à présent 10 prédateurs pour atteindre un taux de 90% de proies attrapé, qui avait été atteint avec seulement 6 prédateurs quand les proies appliquaient la stratégie **E0**.

On ne peut s'empêcher de penser que la stratégie **S2**, du fait des nombreux calculs nécessaires (communication, mise à jour des environnements, optimisation de la fonction de coût) et de sa complexité à mettre en oeuvre, n'est pas une alternative viable à la stratégie **S1** lorsque les proies ne disposent pas d'une stratégie d'évitement efficace comme **E0**. En revanche, dès que les proies agissent de manière plus intelligente, la stratégie **S2** montre tout son intérêt et se montre jusqu'à 40% plus efficace que la stratégie **S1** (cas de 6 prédateurs). Il est par ailleurs intéressant de noter que lors de ces deux scénarios concurrents, les résultats des stratégies **S1** et **S2** sont identiques lorsque seulement deux prédateurs sont disponibles, et que dans ce cas là, la stratégie **E1** se montre près de deux fois plus efficace que la stratégie **E0**. Cela témoigne également du fait que la chasse en meute n'est pas plus efficace que le chacun pour soi lorsque seulement un ou deux prédateurs sont disponibles. En revanche, dès que 3 prédateurs partent à la chasse, les résultats augmentent de près de 25% pour **S2** par rapport à **S1**.

#### 4.1. Champ de vision des prédateurs (scénario n°4)

On cherche à présent à quantifier l'influence de l'étendue du champ de vision des prédateurs. On utilise la stratégie **E1** pour les proies et les stratégies **S1** et **S2** pour les prédateurs. La figure 10 présente les résultats obtenus.

Paramètre	Description
<i>nbPredateur</i>	6
<i>nbProie</i>	120
<i>nbObstacles</i>	96
<i>fodPredateur</i>	[2; 6]
<i>fodProie</i>	5
<i>stratPredateur</i>	<b>S1 et S2</b>
<i>stratProie</i>	<b>E1</b>
<i>rows</i>	48
<i>columns</i>	48
<i>nbIteration</i>	2000



**Figure 10 :** Paramétrage du scénario n°4 (à gauche) et présentation des résultats (à droite). Résultats obtenus à partir de 80 simulations différentes.

La stratégie S2 tend notamment à réduire le temps que les prédateurs passent à chercher des proies ou de gros troupeaux en permettant à chacun des chasseurs d'obtenir des informations par rapport à l'environnement de ses congénères. De fait, lorsque peu de proies sont disponibles ou que le champ de vision des prédateurs est réduit, on peut naturellement supposer que la stratégie **S2** va se révéler plus efficace que la stratégie **S1**. Les résultats obtenus tendent à confirmer cette hypothèse.

### 4.3. Discussion

Les résultats obtenus à l'aide des 4 scénarios présentés ci-dessus tendent à montrer qu'au-delà de l'influence des nombreux paramètres du modèle, la stratégie **S2** est toujours la meilleure, si l'on ne prend pas en considération les temps de calcul et la complexité du programme. Les gains de performance par rapport à la seconde meilleure stratégie, à savoir S1, peuvent atteindre jusqu'à 50%, ce qui est considérable. Le cas de la stratégie **S0** n'est même pas évoqué tant ses performances sont faibles. D'un autre côté, si l'on se place du point de vue des proies, le succès de la stratégie **E1** face à la stratégie **E0** est encore plus éclatant, avec cette fois des gains de performance, c'est-à-dire une proportion de proies ayant échappée aux prédateurs, supérieurs à 100%, ce qui est remarquable.

On aurait aimé avoir le temps d'explorer d'autres pistes, comme par exemple, faire en sorte que les prédateurs se répartissent mieux au sein de leur environnement lorsqu'ils chassent en meute, afin de maximiser le nombre de proies attrapées. On aurait pu imaginer qu'il communiquent des informations sur le terrain lui-même, et cherchent ainsi à collecter de l'information afin d'obtenir une représentation partielle de l'environnement global. On aurait également aimé pouvoir faire en sorte que les prédateurs puissent pister les proies qui laissent des traces au sol durant un temps limité, la profondeur des traces étant une indication du temps depuis le passage. Toutes ces idées seront mises en œuvre dans une prochaine version de notre simulateur.

## 5. Conclusion

Le problème que nous cherchions à résoudre était la mise au point de stratégies efficaces dans le cadre d'une simulation de type proie-prédateur. Un ensemble de cinq stratégies ont été proposées, trois pour la prédateurs et deux pour les proies, avec des stratégies assez naïves et basiques comme la marche aléatoire de **S0**; des stratégies plus subtiles, basées sur des recherches de chemin minimaux avec  $A^*$ , comme **E0** et **S1**; des stratégies nécessitant le calcul efficace et rapide de cartes de distances comme **E1**; ou encore des stratégies combinant tous les aspects précédents, et nécessitant la définition d'une fonction de coût à maximiser, afin de trouver une solution optimale, comme **S2**.

Les gains de performance des stratégies les plus efficaces par rapport aux moins efficaces sont de l'ordre de 100%, et de l'ordre de 25% en moyenne quand on se compare aux solutions intermédiaires. La métrique utilisée est la proportion de proies ayant été attrapées (ou le complémentaire). Enfin, nous avons proposé des pistes d'amélioration de notre programme que nous espérons pouvoir mettre en œuvre dans un futur proche.

# Références

- [1] Ichiro AOKI, *A Simulation Study on the Schooling Mechanism in Fish*, NIPPON SUISAN GAKKAISHI, 1982, Volume 48, Issue 8, Pages 1081-1088.
- [2] I. Aoki, *Internal dynamics of fish schools in relation to inter-fish distance*, Bull. Jap. Soc. Sci. Fish, 50(1984) 751 – 758.
- [3] Jonker, Pieter (1992). *Morphological Image Processing: Architecture and VLSI design*. Kluwer Technische Boeken B.V. pp. 92–96.
- [4] P. E. Hart, N. J. Nilsson et B. Raphael, *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*, IEEE Transactions on Systems Science and Cybernetics SSC4, vol. 4, no 2, 1968, p. 100–107.
- [5] Maurer, Calvin, Rensheng Qi, and Vijay Raghavan, *A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 2, February 2003, pp. 265-270.
- [6] Joseph Rudnick et George Gaspari, *Elements of the Random Walk : An introduction for Advanced Students and Researchers*, Cambridge University Press, 2004, 346 p.
- [7] Gagniuc, Paul A. (2017). *Markov Chains: From Theory to Implementation and Experimentation*. USA, NJ: John Wiley & Sons. pp. 1–235.
- [8] Neyman, J. and Scott, E.L. (1958) *A statistical approach to problems of cosmology*. Journal of the Royal Statistical Society, Series B 20, 1–43.
- [9] Dereudre, D. (2019). *Introduction to the Theory of Gibbs Point Processes*. In: Coupier, D. (eds) Stochastic Geometry. Lecture Notes in Mathematics, vol 2237. Springer, Cham.
- [10] D.J. Strauss. *A model for clustering*. Biometrika, 63:467-475, 1975.

Lien Github : <https://github.com/meckilila/Defi-IA.git>