

# Sujet 29 : Serious-Game basé sur la simulation multi-agents des ressources halieutiques

**Antonin Carette, Julien Duthoit**

Tuteurs: **Sébastien Picault, Christine Largouët**

Master 1 Informatique - Université Lille1

3 Juin 2015

- ① Intérêt personnel, objectif et problématiques
- ② Les systèmes multi-agents
  - Qu'est-ce qu'un système multi-agents ?
  - Définitions
  - Netlogo et IODA
- ③ La modélisation
  - Un système simple
  - La chaîne alimentaire
  - L'expérimentation
  - L'activité concernant la pêche
- ④ Résultats
- ⑤ Conclusion

## Intérêt personnel, objectif et problématiques

**Intérêt** : curiosité envers les systèmes multi-agents et sujet *fun*.

## Intérêt personnel, objectif et problématiques

**Intérêt** : curiosité envers les systèmes multi-agents et sujet *fun*.

**Objectif** : "Mesurer" les impacts de la pêche sur l'écosystème marin, modélisé au moyen d'un système multi-agents, afin d'informer et de mettre en garde un très large public.

# Intérêt personnel, objectif et problématiques

**Intérêt** : curiosité envers les systèmes multi-agents et sujet *fun*.

**Objectif** : "Mesurer" les impacts de la pêche sur l'écosystème marin, modélisé au moyen d'un système multi-agents, afin d'informer et de mettre en garde un très large public.

**Problématiques** :

- 1 modéliser des ressources halieutiques au moyen d'un système multi-agents,
- 2 implémenter cette modélisation,
- 3 expérimenter et veiller à ce que la modélisation soit la plus stable possible.

# Approche

Ensemble d'entités computationnelles :

- vivant au même moment,
- dotés de comportements,
- agissant dans un système partagé (avec coordination des communications).

Relation collective et individuelle -> intelligence collective et distribuée !

# Approche

Ensemble d'entités computationnelles :

- vivant au même moment,
- dotés de comportements,
- agissant dans un système partagé (avec coordination des communications).

Relation collective et individuelle -> intelligence collective et distribuée !

## Définition par J. Ferber (1995)

"Les systèmes multi-agents ont des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents"

# Définitions

Attendez...

Agents... Interactions... Quesako ??



# Définitions

Qu'est-ce qu'un agent ?

Entité physique ou virtuelle :

- capable d'agir dans un environnement,
- capable de communiquer directement avec d'autres agents,
- possédant des ressources propres,
- capable de percevoir (de manière limitée) son environnement,
- détenant un comportement tend à satisfaire ses objectifs.

# Définitions

Qu'est-ce qu'une interaction ?

Au coeur même des problématiques liées aux SMA !

D'après J. Ferber : "Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques[...]".

## Problème

Comment va-t-on modéliser l'écosystème au moyen d'un système multi-agents, et l'implémenter ?

## Problème

Comment va-t-on modéliser l'écosystème au moyen d'un système multi-agents, et l'implémenter ?

## Solution

Au moyen d'une plateforme standardisée et facile d'utilisation !

# Netlogo

Plateforme de simulation libre et gratuite.

Simulation permettant de prendre en compte différents agents (les *turtles*) vivant dans un environnement discrétisé en *patches*.

<https://ccl.northwestern.edu/netlogo/>

# Netlogo

## Exemple



# IODA

Méthode de modélisation à partir de laquelle a été créée une extension Netlogo - par Sébastien Picault et Philippe Mathieu de l'équipe SMAC (CRIStAL).

# IODA

Méthode de modélisation à partir de laquelle a été créée une extension Netlogo - par Sébastien Picault et Philippe Mathieu de l'équipe SMAC (CRIStAL).

Buts :

- simplifier un maximum le design et la réutilisation de simulations individu-centré,
- expliciter la matrice d'interactions et définir "proprement" les interactions.

[http://www.lifl.fr/SMAC/projects/ioda/ioda\\_for\\_netlogo/](http://www.lifl.fr/SMAC/projects/ioda/ioda_for_netlogo/)



# IODA

## Principes de IODA

Principes de IODA :

- ① une entité est un agent,
- ② tout comportement est représenté par une interaction, composée d'un déclencheur optionnel (*TRIGGER*), d'une condition (*CONDITION*) ainsi que d'une action (*ACTION*),
- ③ ces interactions sont affectées à des familles d'agents, exécutées par un **moteur générique**.

C'est le moteur qui détermine quelles interactions peuvent avoir lieu lors d'un pas de temps.

# IODA

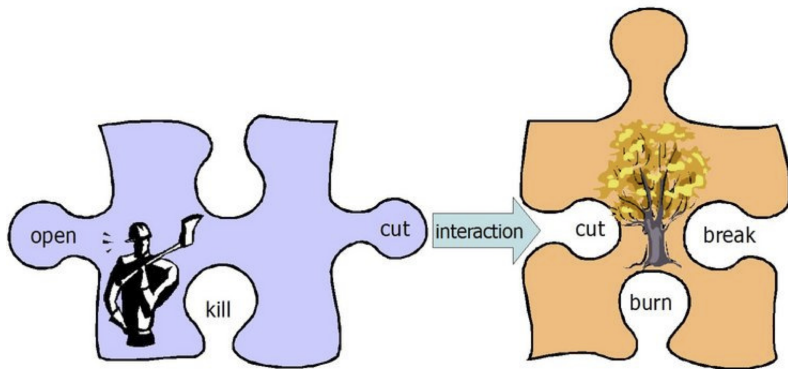
## Fonctionnement du moteur générique IODA

Pour un pas de temps :

- ❶ les interactions de mises à jour (*UPDATE*) réalisables sont exécutées par chaque agent donné,
- ❷ puis chaque agent susceptible d'agir sur les autres choisit une interaction de la façon suivante :
  - ❶ perçoit ses voisins (subissant les actions des autres agents),
  - ❷ filtre les cibles d'interactions qu'il peut effectuer,
  - ❸ évalue les déclencheurs et conditions de ces interactions,
  - ❹ sélectionne une des interactions réalisables (de priorité maximale),
  - ❺ exécute les actions correspondantes, présentes dans la déclaration de l'interaction.

# IODA

## Exemple



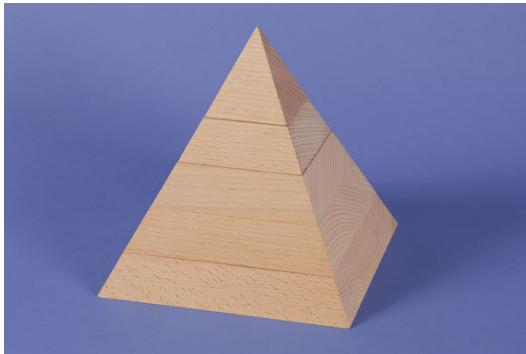
# IODA

## Exemple

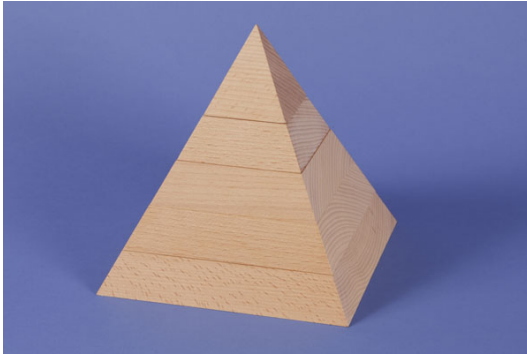
---

Woodcutter	cut	80	Tree	1
Woodcutter	open	60		
Woodcutter	kill	0		
Tree	cut	100		
Tree	burn	80		
Tree	break	60		

# Modélisation

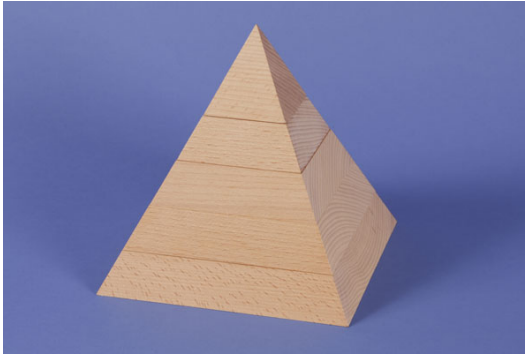


# Modélisation



"Oh non, il est devenu complètement fou !"

# Modélisation



Modélisation empirique et incrémentale!!!

# Modélisation

Découpage de la modélisation en 4 parties :

- la modélisation d'une base,
- l'ajout de la chaîne alimentaire,
- l'expérimentation,
- l'ajout de l'activité concernant la pêche.



# Modélisation

**Idée** : modélisation d'un système proie/prédateur simple :

- un agent proie : le plancton,
- un agent prédateur : les sardines.

# Modélisation

**Idée** : modélisation d'un système proie/prédateur simple :

- un agent proie : le plancton,
- un agent prédateur : les sardines.

**Besoins** : implémentation des agents et des interactions.

# Modélisation

## Implémentation des premiers agents

Des caractéristiques propres, pour chaque agent :

- **une biomasse,**
- un nombre maximal de stock de nourriture (peut être égal à 0),
- zéro, une ou plusieurs cible(s),
- une biomasse cible (ce dont il a besoin pour [sur]vivre),
- une stratégie de migration,
- etc...

# Modélisation

## Implémentation des premiers agents

Interactions définies dans la **matrice d'interactions**, et implémentées dans le **fichier d'interactions**.

# Modélisation

## Exemple

---

sardines	Reproduce	100		
sardines	Eat	80	food	0.5
sardines	Die	0		
food	Grow	0	UPDATE	

# Modélisation

## Exemple

---

sardines	Reproduce	100		
sardines	Eat	80	food	0.5
sardines	Die	0		
food	Grow	0	UPDATE	

"L'agent *sardines* va adopter pour le *tick* *x* le comportement *Eat* avec une priorité de 80, sur un agent *food* à distance 0.5 de lui."

# Modélisation

## Exemple

---

```
interaction Reproduce
  trigger    canReproduce?
  condition  stillAlive? not:hungry? target:stillAlive?
  actions    reproduce eat
end

interaction Eat
  trigger    hungry?
  condition  stillAlive? target:stillAlive?
  actions    eat
end

interaction Die
  trigger    hungry?
  actions    die
end
```

# Modélisation

## Exemple

---

```
to-report agent::hungry?  
  report biomass-req > 0  
end  
  
to-report agent::stillAlive?  
  report biomass-f > 0  
end  
  
to-report food::PREY_stillAlive?  
  report food::PREY_actualFood? > 0  
end  
  
to-report sardines::PRED_stillAlive?  
  report agent::stillAlive?  
end
```



# Modélisation

## Premiers résultats

Mini-objectifs :

- modélisation et implémentation d'une base saine,
- modélisation d'un écosystème basique et général,
- devancement des problèmes liés à la suite de l'implémentation.

# Modélisation

## Premiers résultats

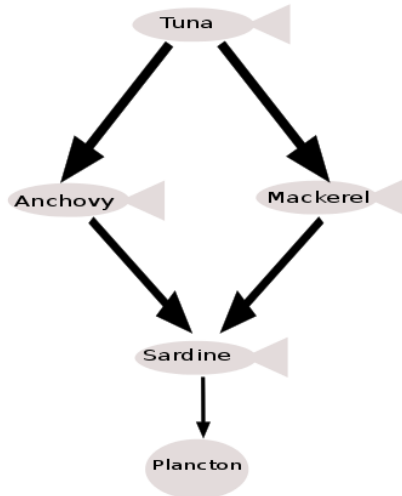
Mini-objectifs :

- modélisation et implémentation d'une base saine,
- modélisation d'un écosystème basique et général,
- devancement des problèmes liés à la suite de l'implémentation.

Partie longue et difficile !

# La chaîne alimentaire

## Modélisation



# La chaîne alimentaire

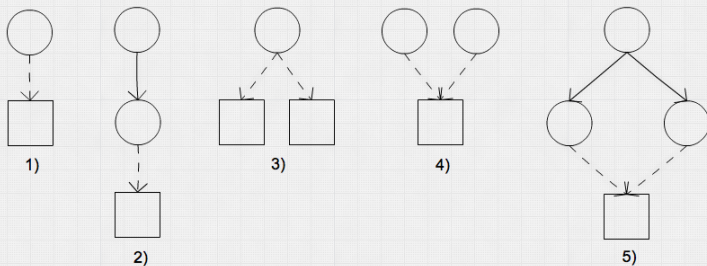
## Nouvelles problématiques

De nouveaux défis :

- nouvelles espèces,
- les mêmes interactions,
- nouvelles configurations.

# La chaîne alimentaire

## Les nouvelles configurations



## Problème

Modifications -> écosystème instable -> impossible de mesurer l'impact de la pêche.

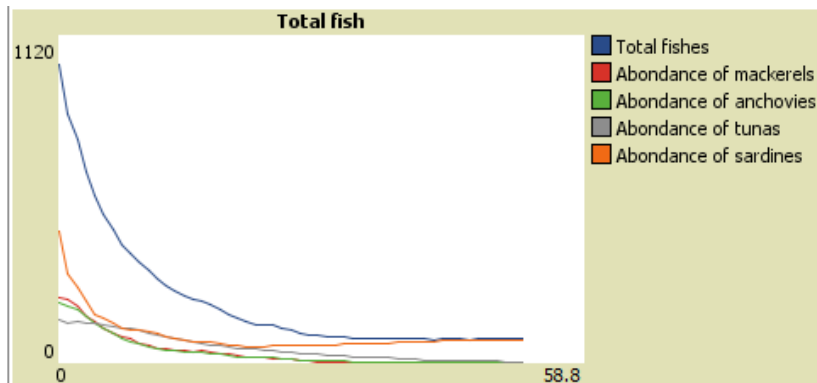
## Problème

Modifications -> écosystème instable -> impossible de mesurer l'impact de la pêche.

## Solution

Une série d'expérimentations.

## Evolution de la biomasse avant les expérimentations





# Expérimentation

## Problème

Enormément de paramètres entrent en jeu

- stratégie de migration,
- consommation de nourriture,
- rapidité de reproduction,
- mortalité,
- stock de nourriture.

Grand nombre d'expériences possibles !

# Expérimentation

## Problème

Enormément de paramètres entrent en jeu

- stratégie de migration,
- consommation de nourriture,
- rapidité de reproduction,
- mortalité,
- stock de nourriture.

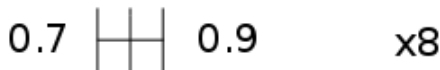
Grand nombre d'expériences possibles !

## Solution

- Choix de 3 caractéristiques : consommation, reproduction et mortalité.
- Un protocole de recherche pour des recherches efficaces.

# Expérimentation

## Méthodologie



# Expérimentation

## Outils utilisés

Experiment

Experiment name

Vary variables as follows (note brackets and quotation marks):  

`[ "food-consumption-rate" [0 0.2 2] ]  
[ "mortality-rate" [0 0.1 1] ]  
[ "reproduction-rate" 0.1 0.2 0.4 0.5 ]`

Either list values to use, for example:  
[ "my-slider" 1 2 7 8 ]  
or specify start, increment, and end, for example:  
[ "my-slider" [0 1 10] ] (note additional brackets)  
to go from 0, 1 at a time, to 10.  
You may also vary max-pccor, min-pccor, max-pycor, min-pycor, random-seed.

Repetitions   
run each combination this many times

Measure runs using these reporters:  

`count tunas with [toda:alive? => true]`

one reporter per line; you may not split a reporter across multiple lines

☒ Measure runs at every step  
if unchecked, runs are measured only when they are over

Setup commands:

`setup`

Go commands:

`go`

Stop condition:  
`not any? tunas with [toda:alive? => true]`

Final commands:  
run at the end of each run

the run stops if this reporter becomes true

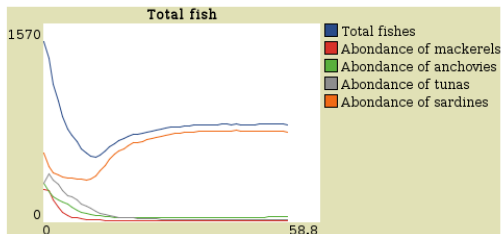
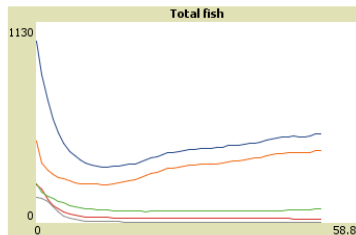
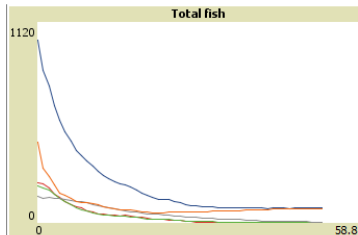
Time limit   
stop after this many steps (0 = no limit!)

OK

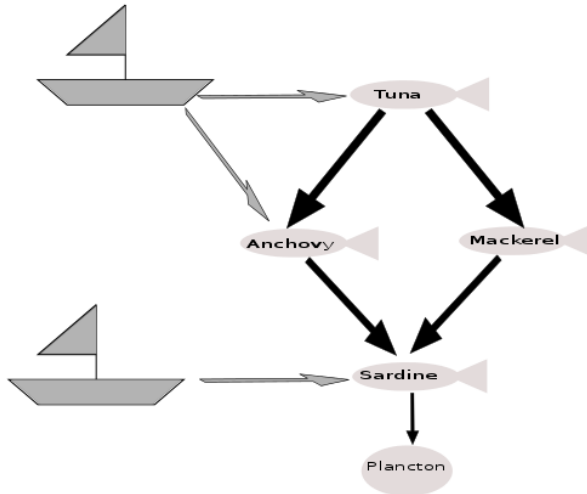
Cancel

# Expérimentation

## Évolution



## Intégration à la chaîne alimentaire



# L'activité concernant la pêche

## Problématique

Deux nouveaux agents nécessaires :

- les ports :
  - possèdent un ou plusieurs bateaux,
  - déterminent si les bateaux peuvent prendre la mer.
- les bateaux :
  - spécialisés,
  - autonomes,
  - stocker du poisson.

# L'activité concernant la pêche

## Nouvelles Interactions

---

boats	ToFish	100	anchovies	1
boats	ToFish	100	mackerels	1
boats	ToFish	100	sardines	1
boats	ToFish	100	tunas	1
boats	MoveWithoutFishing	80		
boats	GoToHarbour	80		
harbours	ComputeFoodStock	10	UPDATE	
harbours	ComputeBoatsAvailable	10	UPDATE	
boats	VerifySailing	10	UPDATE	
boats	DecrTick	10	UPDATE	



# Résultats

## Serious-game

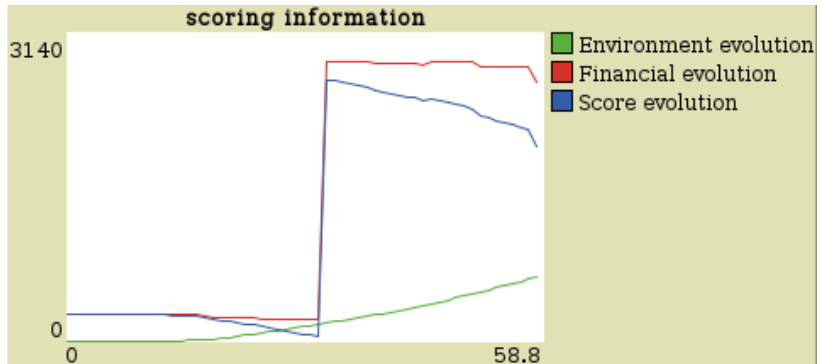
Différents éléments en jeu dans le serious-game :

- aspect économique,
- aspect écosystème.

Le joueur gère sa stratégie pour maximiser son score.

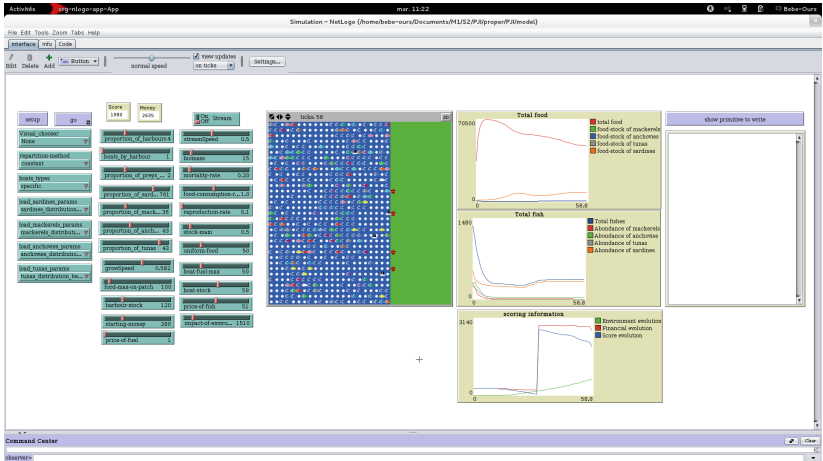
# Résultats

## Score



# Résultats

## Au final...



# Conclusion

## Projet

Découverte de nouvelles technologies et méthodologies :

- SMA et l'approche IODA,
- outils spécialisés dans la modélisation de SMA et la simulation,
- mise en place de stratégies d'expérimentation,
- développer un projet de manière incrémentale.

# Conclusion

## Perspectives

Modèle :

- integration de données réelles et d'éléments environnementaux,
- développer les différents aspects du serious-game,
- mesurer l'évaluation par les impacts indirects de la pêche sur l'écosystème,
- travailler la rejouabilité.

Génie logiciel :

- utilisation du moteur IODA en Python,
- utilisation du projet Netlogo en JavaScript.

# Conclusion

## Personnel

- Compréhension des SMA.
- Travailler sur un sujet de recherche.
- Renforcer notre envie de suivre le Master MoCAD.