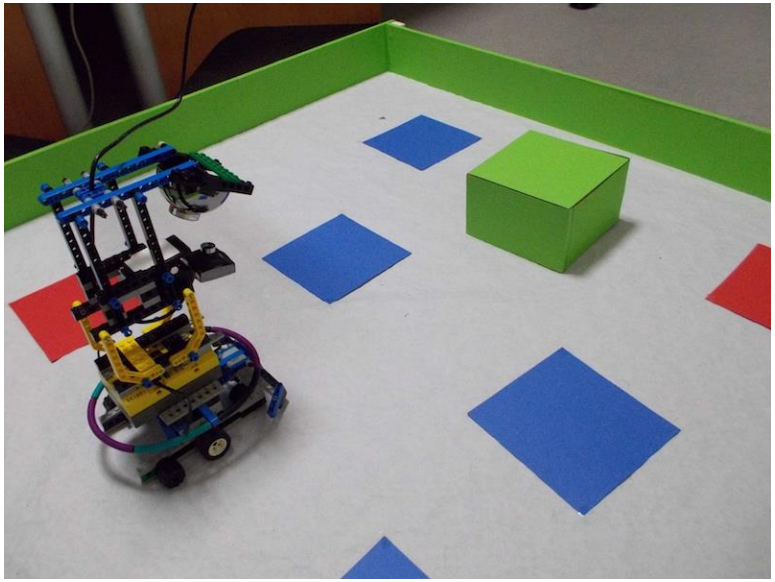
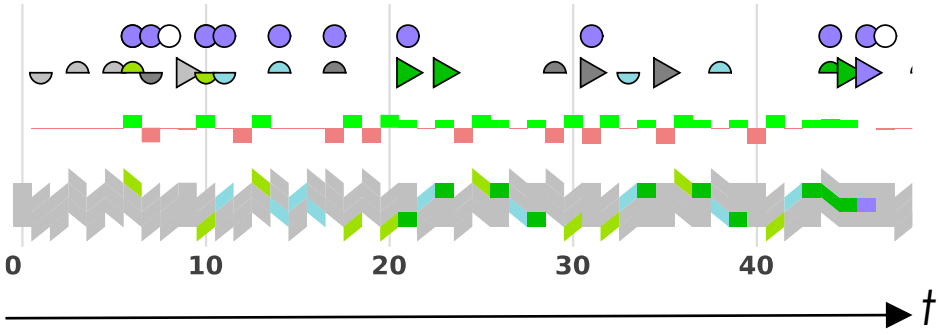
Intelligence Artificielle Développementale



25 Septembre 2023
Olivier.georgeon@gmail.com
http://www.oliviergeorgeon.com



Déroulement du cours

UE « IA et Cognition » (30h): Marie Lefevre

Module « IA Développementale » (12h) Olivier Georgeon

- Lundi 18 septembre: 3h
- **Lundi 25 septembre: 3h**
- Mercredi 27 septembre: 3h
- Lundi 2 octobre: 3h

- Contrôle des connaissances:
  - TD par groupe de 2: 40% de le note de contrôle continu
  - Examen final: 7 points sur 20

Objectifs pédagogiques

Après ce cours, vous serez capables de:

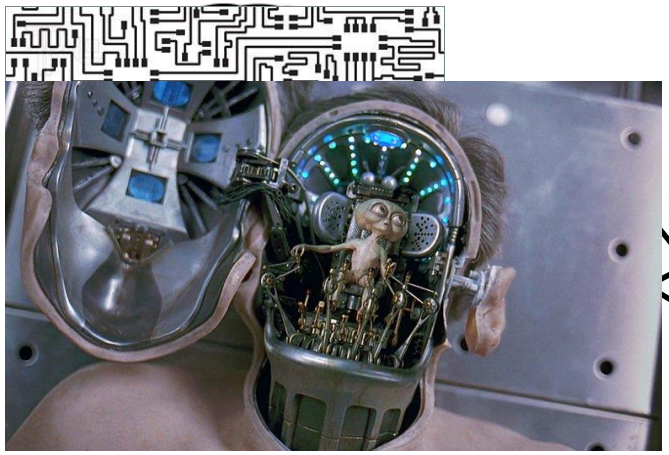
- Cours
  - Expliquer ce qu'est l'IA développementale
  - Différencier IA en domaine modélisé / non modélisé
  - Nommer quelques auteurs de référence dans ce domaine
- TD
  - Implémenter un agent minimaliste dans lequel on ne code pas a priori une ontologie du "monde".

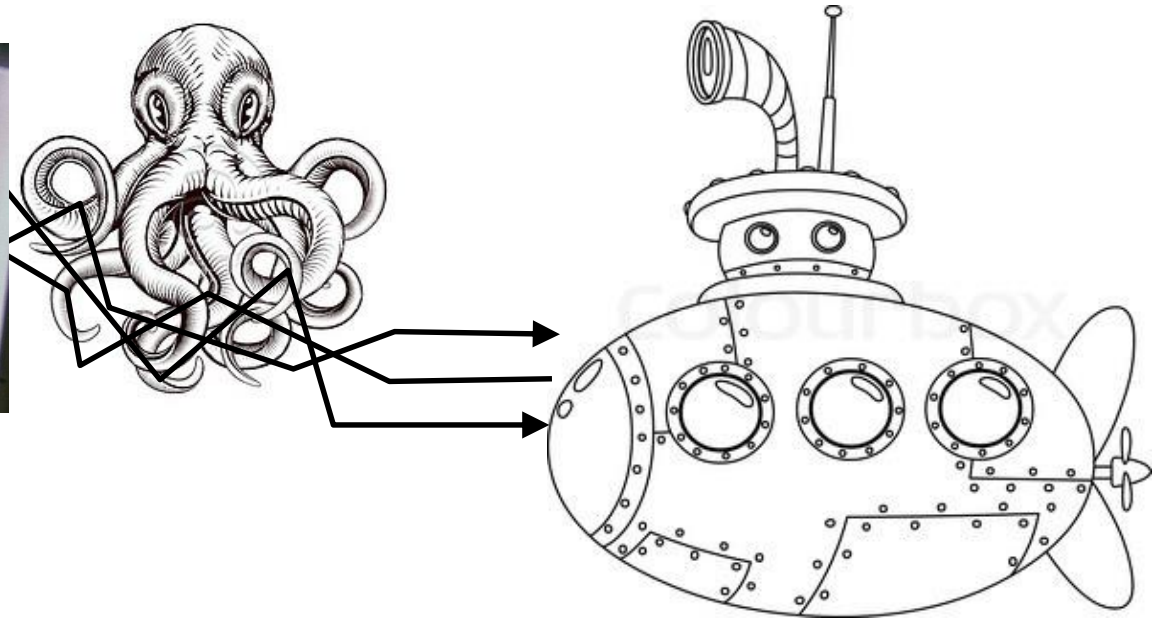
Séance 2: Plan

- **Séance précédente**
  - La critique ontologique de l'IA
  - La critique téléologique de l'IA
- **Intelligence artificielle dans un domaine non modélisé a priori**
  - Inversion du cycle d'interaction
  - Déterminisme et cognition
- **Travaux pratiques**
  - Agent 1
  - Agent 2

Inversion du cycle d’interaction

IA sans domaine modélisé a priori



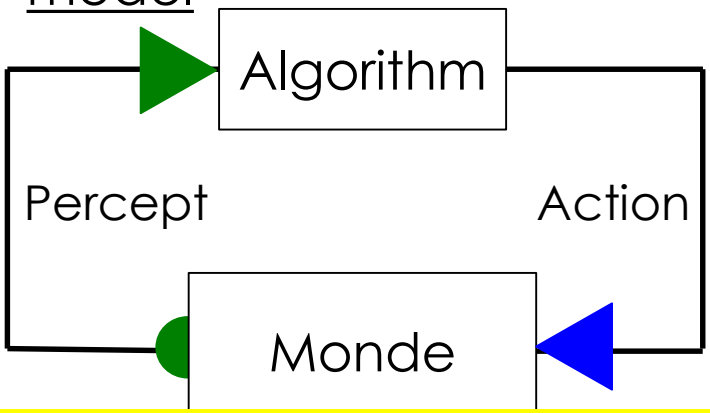


O'Regan & Noë (2001)
A sensorimotor account of vision and visual consciousness

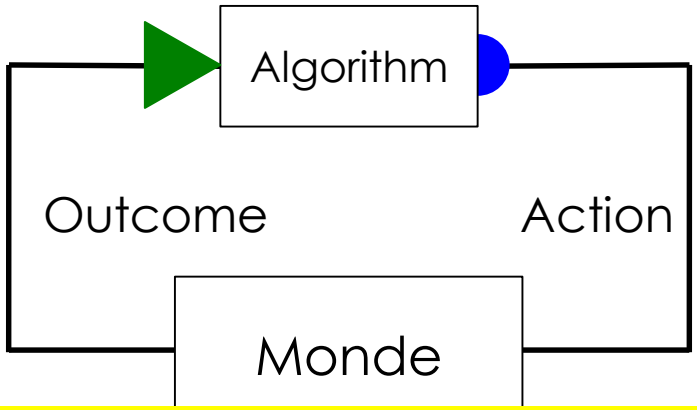
Inversion du cycle d'interaction

- "The problem of AI is to build agents that **receive percepts from the environment and perform actions**" (Russell et Norvig, 2003, p. iv)
- By observing the **structure of the changes** that occur **when they press various buttons and levers** (O'Regan & Noë 2001, p. 940).

a) Traditional model



b) Inverted model



La complexité des données d'entrée n'a pas besoin d'etre proportionnelle à la complexité du monde

Inversion du cycle d'interaction

- **Neurosciences**
  - Active inference
    - Friston (2017) Active inference, a process theory.
    - https://youtu.be/WzFQzFZiwzk?t=1265
  - O'Regan – Laming
    - On the distinction between "sensorimotor" and "motorsensory" contingencies (2001)
- **Robotique**
  - Rolf Pfeifer
    - From perception to action: The right direction? (1994)
  - Georgeon O. & Cordier A.
    - Inverting the interaction cycle to model embodied agents (2014)
  - Asada & Nagai
    - Predictive learning of sensorimotor information as a key for cognitive development. (2015)

Deux épistémologies en concurrence

- Hypothèse « représentationaliste » ou « réaliste »
  - Les données d'entrée représente des aspects de la réalité (percepts)
- Hypothèse « constructiviste » ou « interactioniste »
  - Les données d'entrées informent sur les possibilités d'interaction (résultat d'une action)

Déterminisme et cognition

Déterminisme

Je suis un système déterministe

Je suis un système non déterministe

Déterminisme et prédictibilité

Déterministe

- Chaque état du système découle de manière univoque de l'état précédent
- Si on « ré-exécute » le système, il se comportera exactement de la même manière

Prédictibilité

- Possibilité de prévoir l'état futur d'un système

Sources d'imprédictibilité

Indéterminisme

- physique quantique

Incertitude

- Connaissance imparfaite des lois
- Connaissance imparfaite des conditions initiales

Complexité

- Moyens de calcul insuffisants

Irréductibilité computationnelle

- L'algorithme qui simule le système ne peut pas être court-circuité pour prédire directement le résultat à l'étape n.

Imprédictibilité déterministe

Problème des trois corps

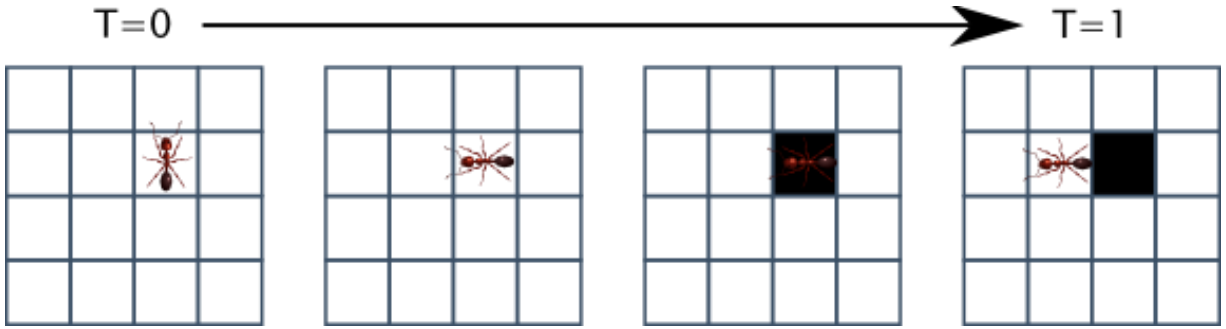
- Résoudre les équations de Newton de N corps interagissant gravitationnellement
- Henri Poincaré

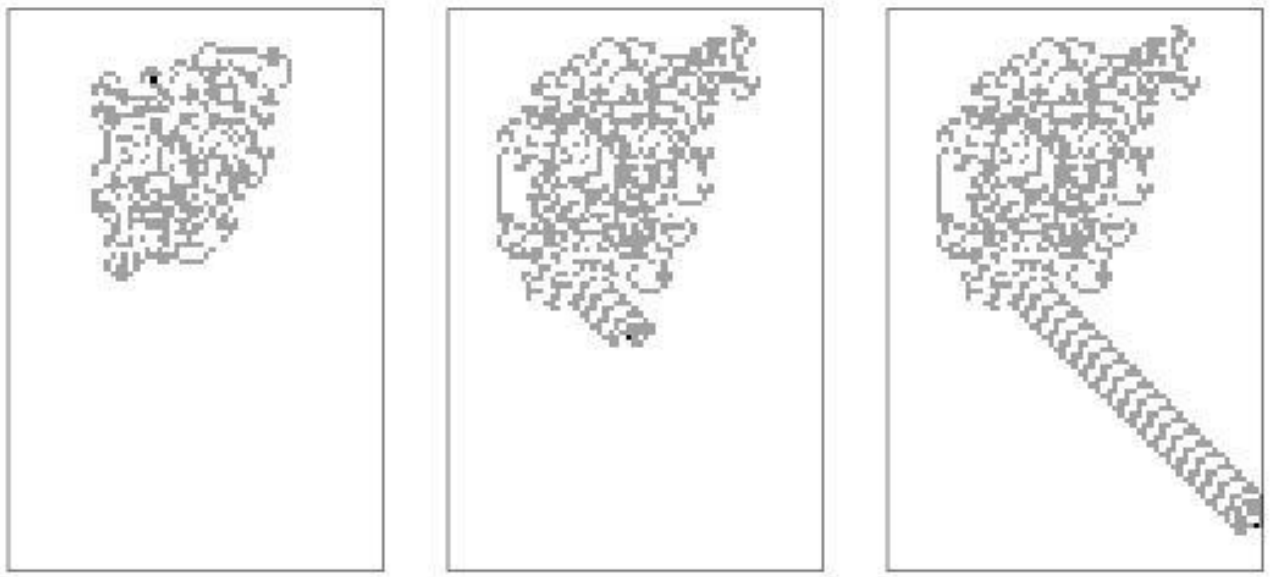
Jeu de la vie de Conway

Fourmi de Langton

- https://youtu.be/qZRYGxF6D3w

Fourmi de Langton





Idées Clés

L'humain est peut être déterministe mais néanmoins libre

- *Théorie évolutionniste de la liberté* (Dennett 2003).

Système déterministe peut être imprédictible.

- *Computatinal irreductibility* (Stehen Wolfram)
- Inutile d'utiliser la Fonction Random()
  pour générer des comportements imprédictibles.

Un système déterministe peut « s'individualiser »

- En fonction des conditions initiales
- En fonction d'expériences individuelles
- *Autonomie constitutive* (Froese & Ziemke 2009).

Emergence de « macro-propriétés »

- Souvent non démontrable mais observable depuis un niveau d'observation supérieur

Travaux dirigés

Séance 1

Sujet

Deux actions possibles $A = \{a_0, a_1\}$

Deux outcome possibles $O = \{o_0, o_1\}$

Quatre interactions possibles $I = A \times O = \{i_{00}, i_{01}, i_{10}, i_{11}\}$

Environnement

- env_1: a_0 -> o_0 , a_1 -> o_1 (i_{01} et i_{10} ne se produisent jamais)
- env_2: a_0 -> o_1 , a_1 -> o_0 (i_{11} et i_{01} ne se produisent jamais)

Implémenter un agent qui apprenne à anticiper son outcome sans connaître a priori son environnement (env_1 ou env_2).

Et qui change d'action quand il commence à s'ennuyer: (quand il a effectué plus de n fois de suite la même interaction)

Produire un rapport d'analyse de comportement basés sur les traces.

Consignes pour les TP

- Par groupe de 2.
- Rendre un seul rapport à la fin .PDF *ou .IPYNB ?*
- Indiquer bien le nom des deux membres du groupe
- Envoyer par mail à olivier.georgeon@gmail.com pour le **vendredi 17 novembre 2023** 23h59
- Pour chaque agent
  - Décrire les principes de l'algorithme que vous avez implémenté
  - Inclure des captures d'écran des traces affichées à la console dans différents environnements
  - Expliquer les comportements obtenus en vous appuyant sur les traces.
- Conclure sur ce que vous retirez de cette expérience et suggestions de comment aller plus loin (Activité 4)

Setup

Suivre la procédure écrite ici https://github.com/OlivierGeorgeon/TestROS/wiki/Implementer-un-agent-rudimentaire

Créer un nouveau projet python dans votre environnement de développement Python favori (par exemple Pycharm) contenant le fichier world.py. Vous avez deux méthode possibles :

- Cloner le repository https://github.com/OlivierGeorgeon/TestROS
- Créer un nouveau projet et copier le fichiers world.py

Exécuter world.py et vérifiez que vous obtenez la trace d'interaction montrée en Figure 1 sur https://github.com/OlivierGeorgeon/TestROS/wiki/Implementer-un-agent-rudimentaire)

Agent 1 – Qui n’aimait pas s’ennuyer

Dans le fichier world.py, modifier la class Agent pour créer l’Agent 1 en suivant les instructions :

https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-1

Tester votre agent dans Environment1 puis dans Environment2 en commentant et décommentant les lignes appropriés (lignes 70 et 71 dans le fichier world.py initial)

Agent 2 – qui préférait les interactions positives

Créer l'Agent2 en suivant les instructions :

https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-2

Tester votre agent dans Environment1 puis dans Environment2 en commentant et décommentant les lignes appropriés (lignes 70 et 71 dans le fichier world.py initial)

Modifier la table des valences d'interaction.