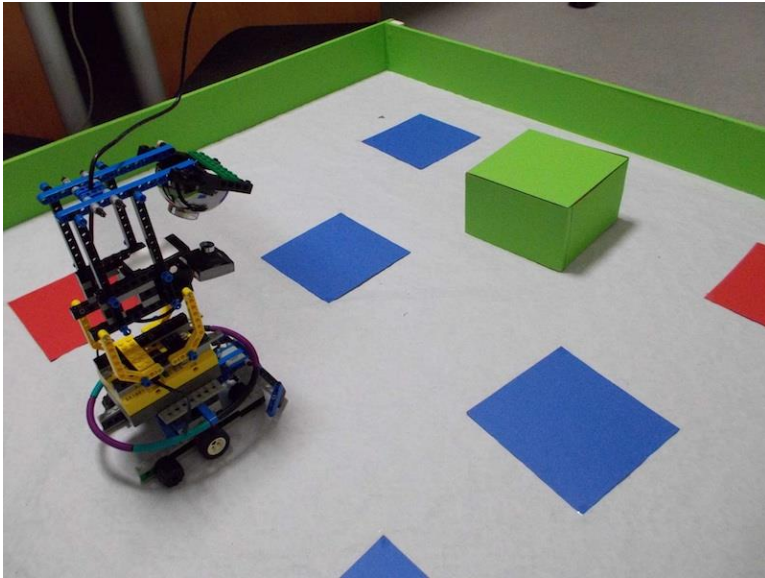


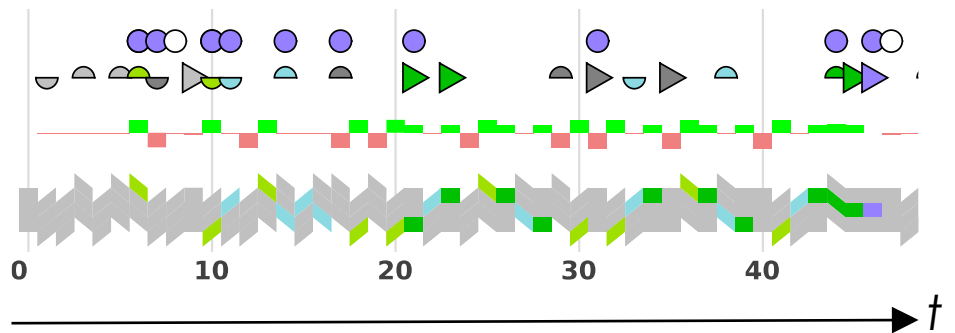
Intelligence Artificielle Développementale



24 Novembre 2021

Olivier.georgeon@gmail.com

<http://www.oliviergeorgeon.com>



Déroulement du cours

UE « IA et Cognition » (30h): Salima Hassas

Module « IA Développementale » (9h) Olivier Georgeon

- Mardi 19 octobre: 3h
- Jeudi 4 novembre: 3h
- Mercredi 24 novembre: 3h

- Contrôle des connaissances:
 - TD par groupe de 2 pour le 10 décembre 2021
 - Une question dans l'exam de l'UE

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

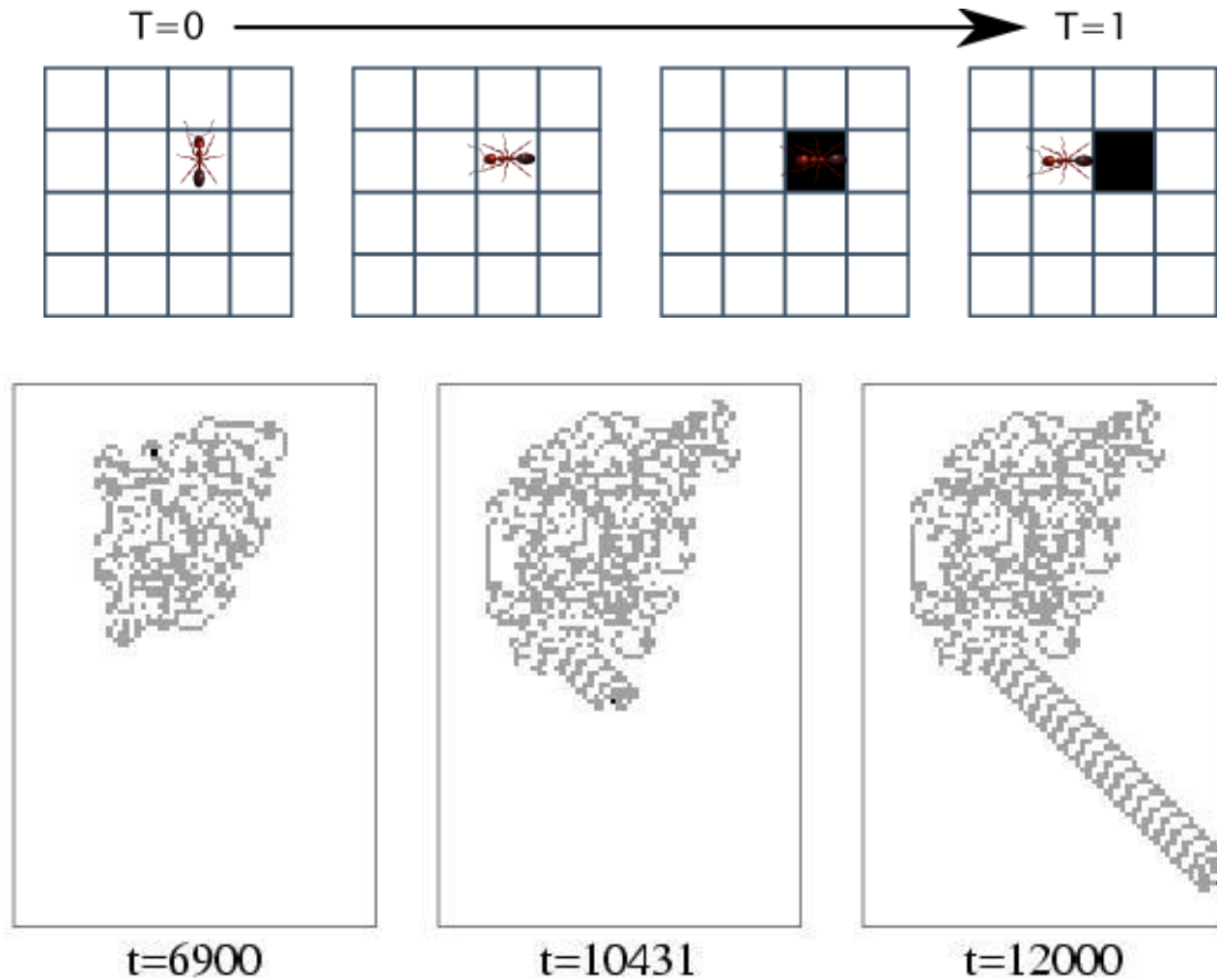
- Résoudre les équations de Newton de N corps interagissant gravitationnellement
- Henri Poincaré

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.

- Inutile d'utiliser la Fonction Random() pour générer des comportements imprédictibles.
- Hervé Swirn



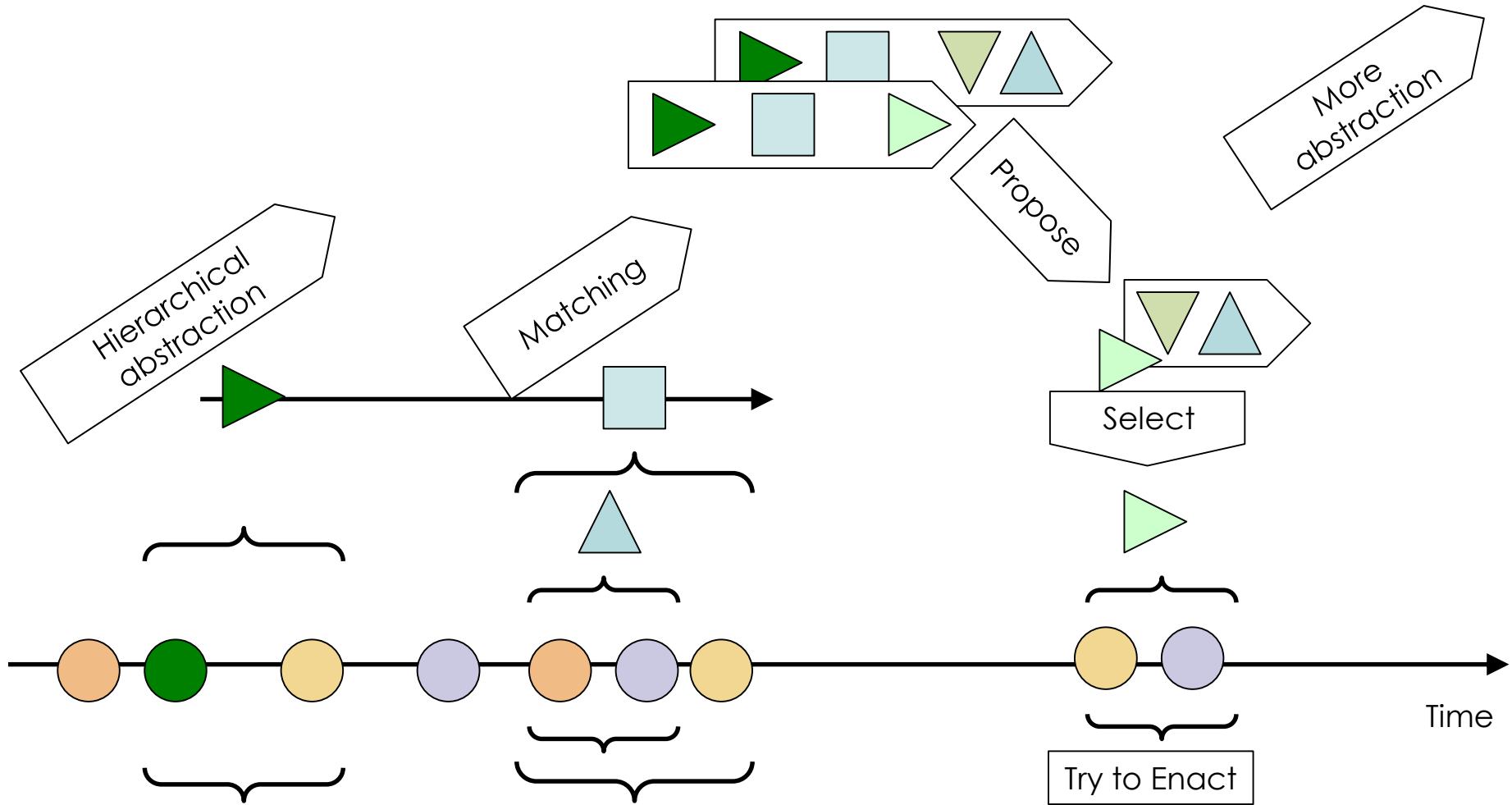
Un système déterministe peut « s'individualiser »

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

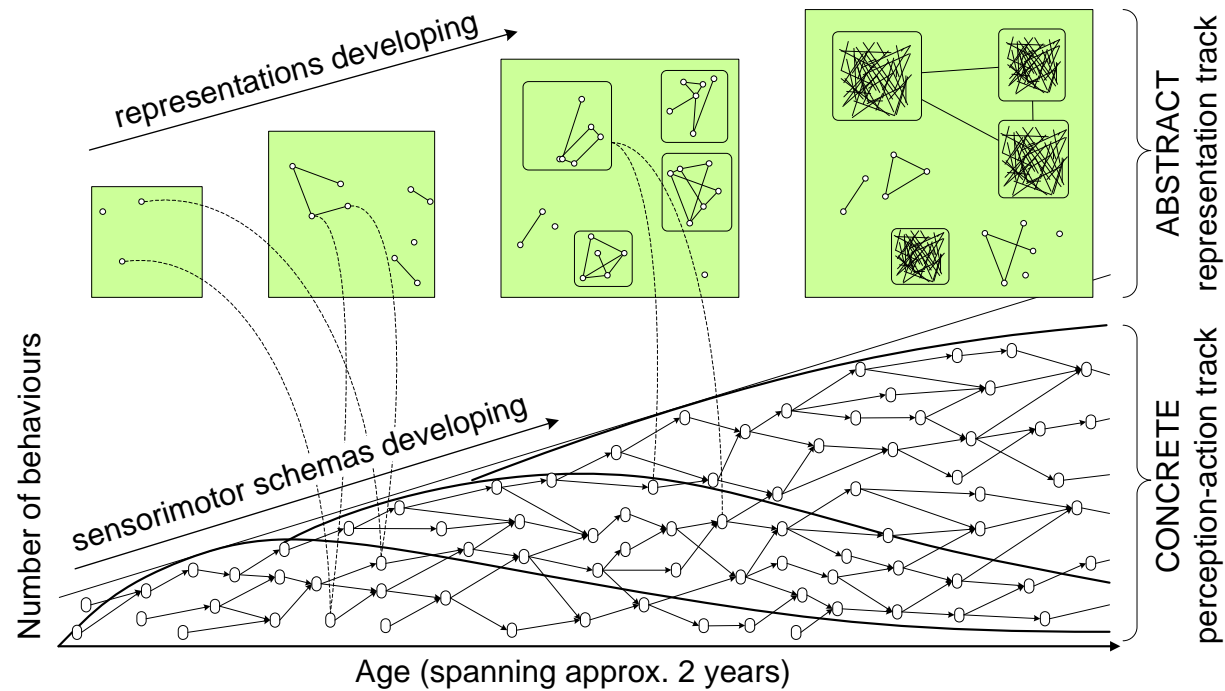
Emergence de « macro-propriétés »

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

Apprentissage de régularités hiérarchique



Construire des connaissances à partir de régularités d'interactions (schèmes sensorimoteurs)



Conceptual diagram of developmental learning (Guerin, Krüger, and Kraft, 2013)

Inversion du cycle d'interaction

- **Neurosciences**

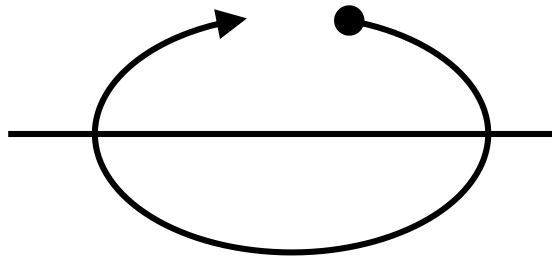
- Active inference
 - Friston (2017) Active inference, a process theory.
 - <https://youtu.be/WzFQzFZiwzk?t=1265>
- O'Regan – Laming
 - On the distinction between “sensorimotor” and “motorsensory” contingencies (2001)

- **Robotique**

- Rolf Pfeifer
 - From perception to action: The right direction? (1994)
- Asada & Nagai
 - *Predictive learning of sensorimotor information as a key for cognitive development.* (2015)

Couplage structurel

On parle de couplage structurel chaque fois que survient une histoire d'interactions récurrentes responsables d'une congruence structurelle entre deux systèmes ou plus.
(Maturana & Varela, 1989, p. 78)

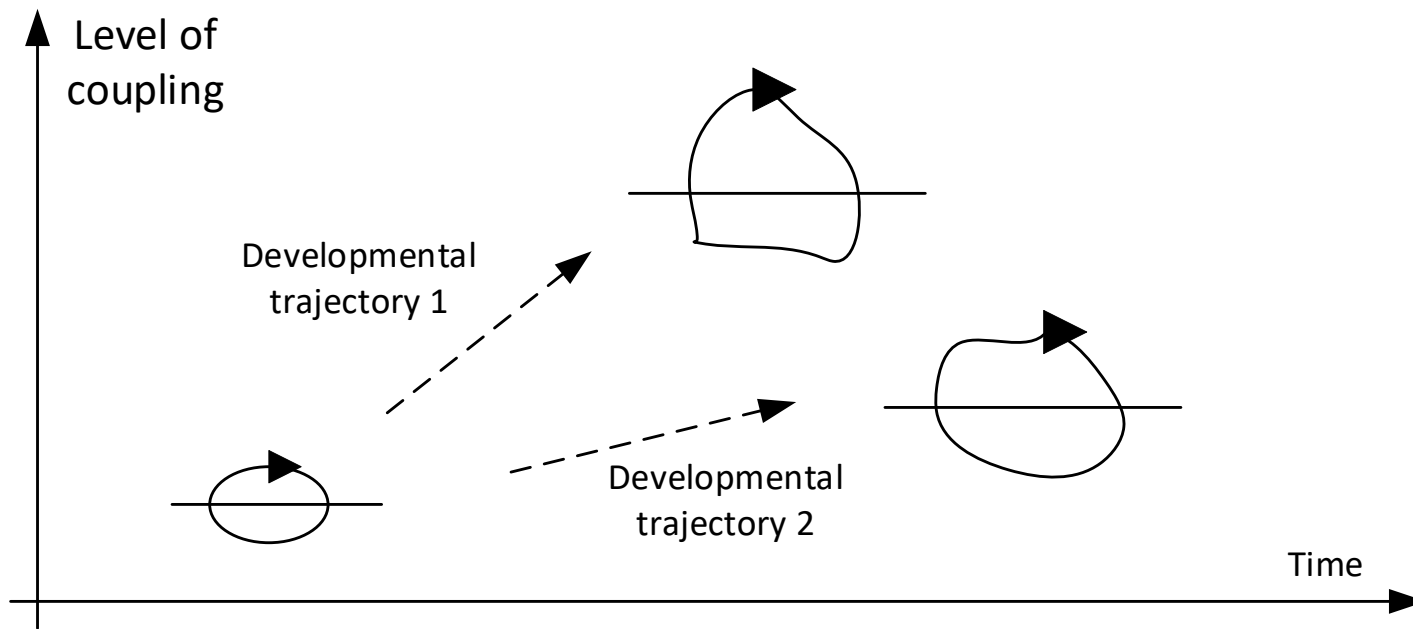


Couplage

Sujet
/
Objet

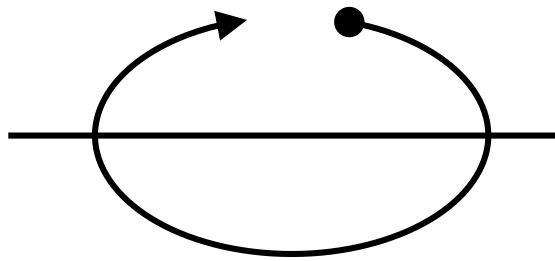
Apprentissage développemental

Capacité d'un sujet a développer individuellement sa structure interne et son couplage avec son environnement. (Théories de Piaget).



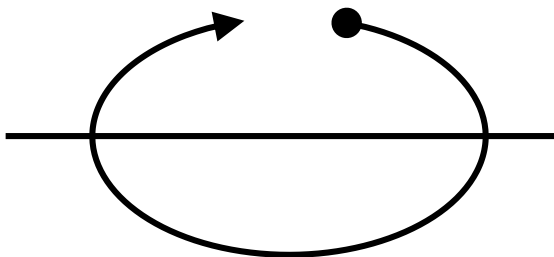
Constitutive autonomy

Dans un système artificiel: dissocier
« couplage cognitif » vs « couplage physique »



*Decisional mechanism
/
Software Interface*

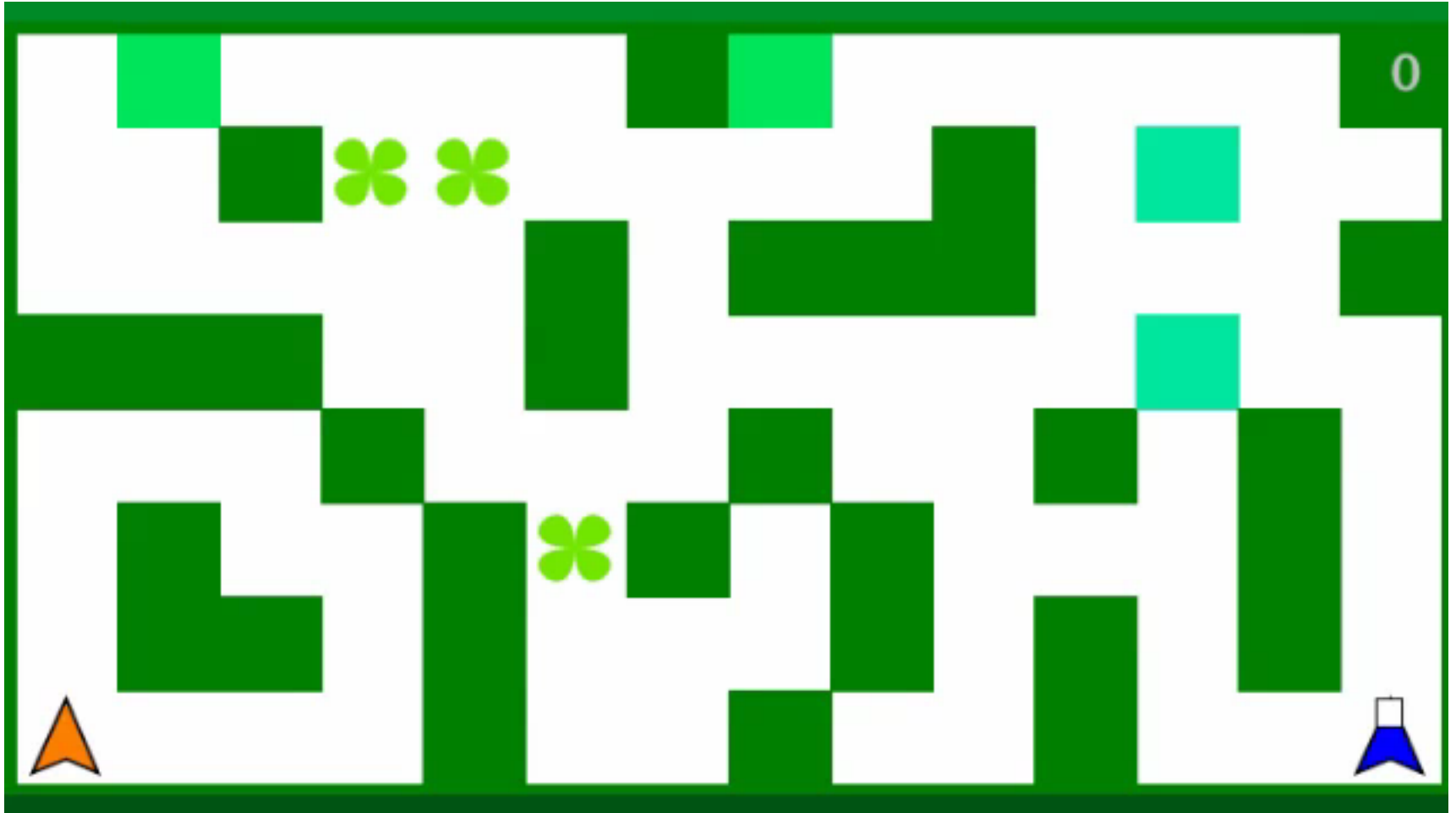
↑
Evolve



*Software
/
World*

Fixe

Environnement plus complexe



<https://youtu.be/q8WkYbt2BxM>

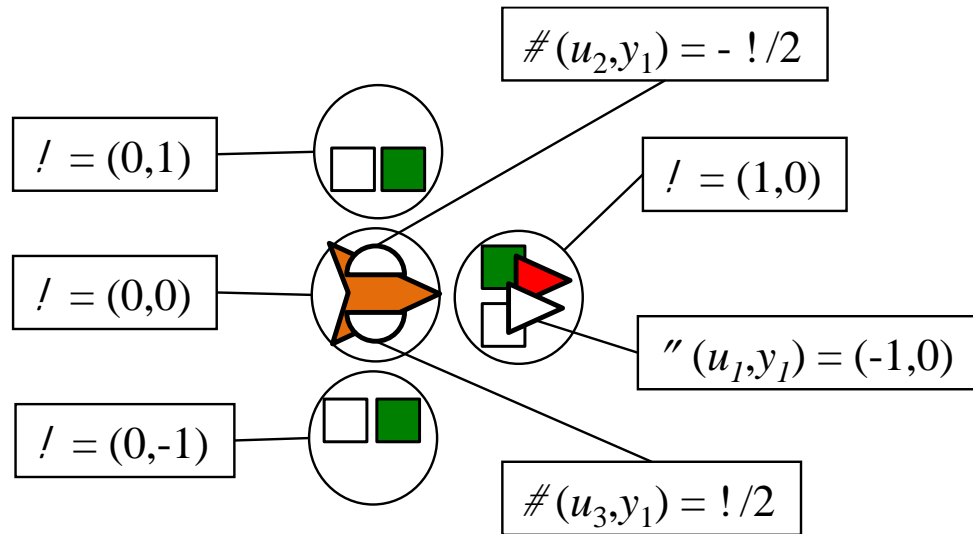
Auto programmation

Un système auto-programmant est un système capable d'apprendre du code qu'il peut ré-exécuter.

Cela pose de nombreuses questions:

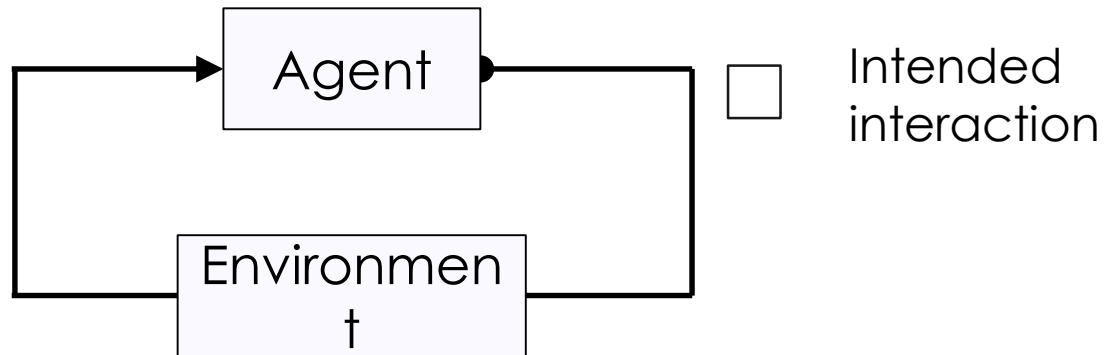
- Quel jeu d'instructions?
- Quel moteur d'exécution?
- Quelle finalité?
 - Pourquoi apprendre un programme plutôt qu'un autre?

Spatial exemple 1



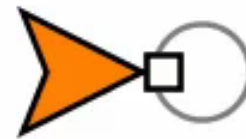
Enacted interaction: ■

Spatial position $\sigma = (1,0)$
 Spatial transformation $\tau = (0,0)$



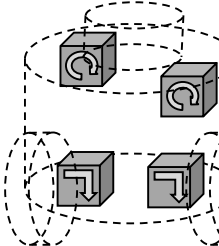
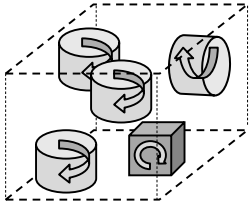
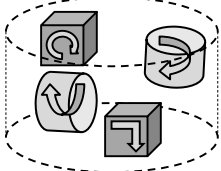
Spatial Exemple 1

<https://youtu.be/HCDf3Vzl7GM>



Cognitive architecture

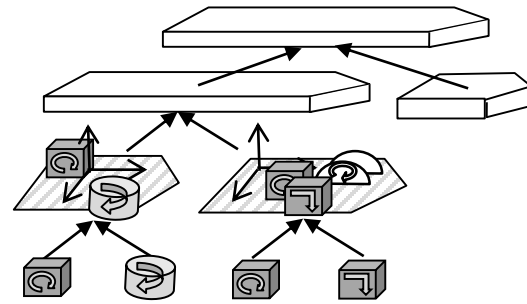
Phenomenal ontology
(long-term memory)



3. Activate

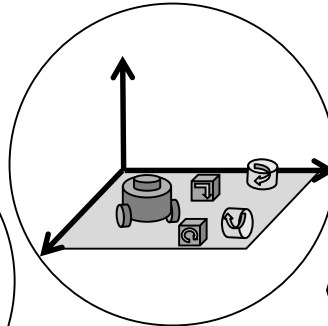
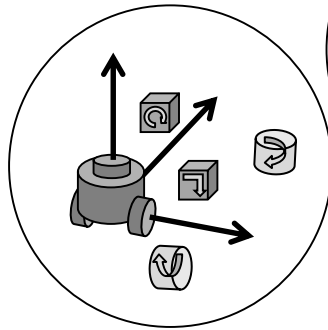
2. Construct

Hierarchical bottom-up Learning
(long-term memory)



Allocentric spatial short-term memory

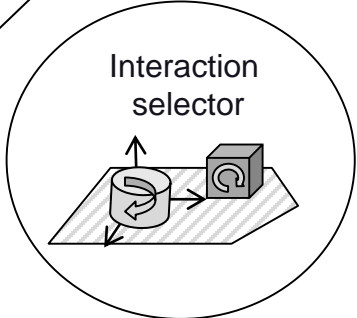
Egocentric spatial short-term memory



4. Afford

5. Simulate

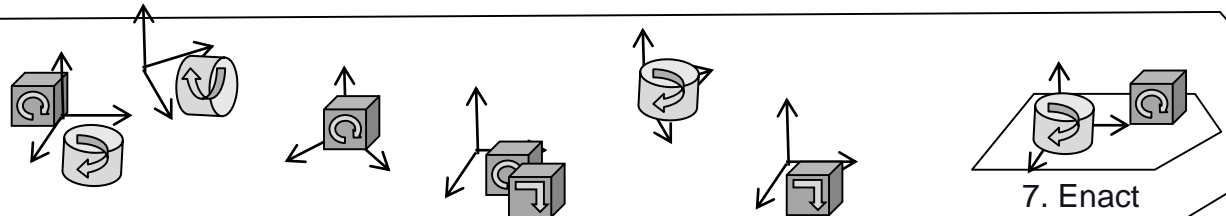
Interaction selector



6. Try

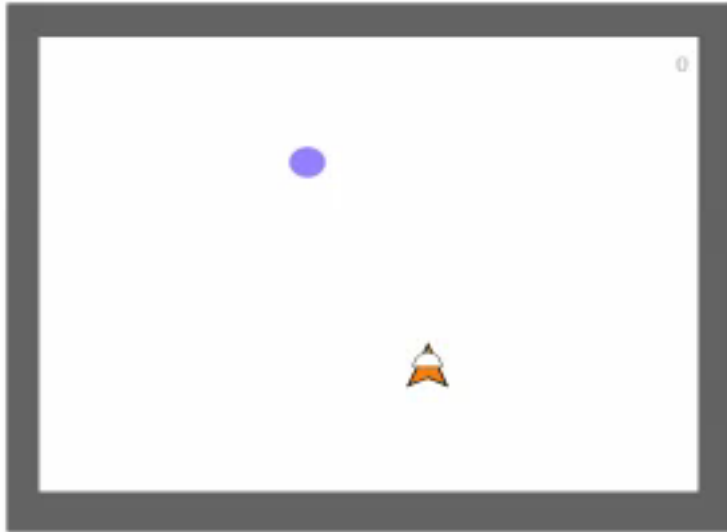
1. Activate

Interaction timeline



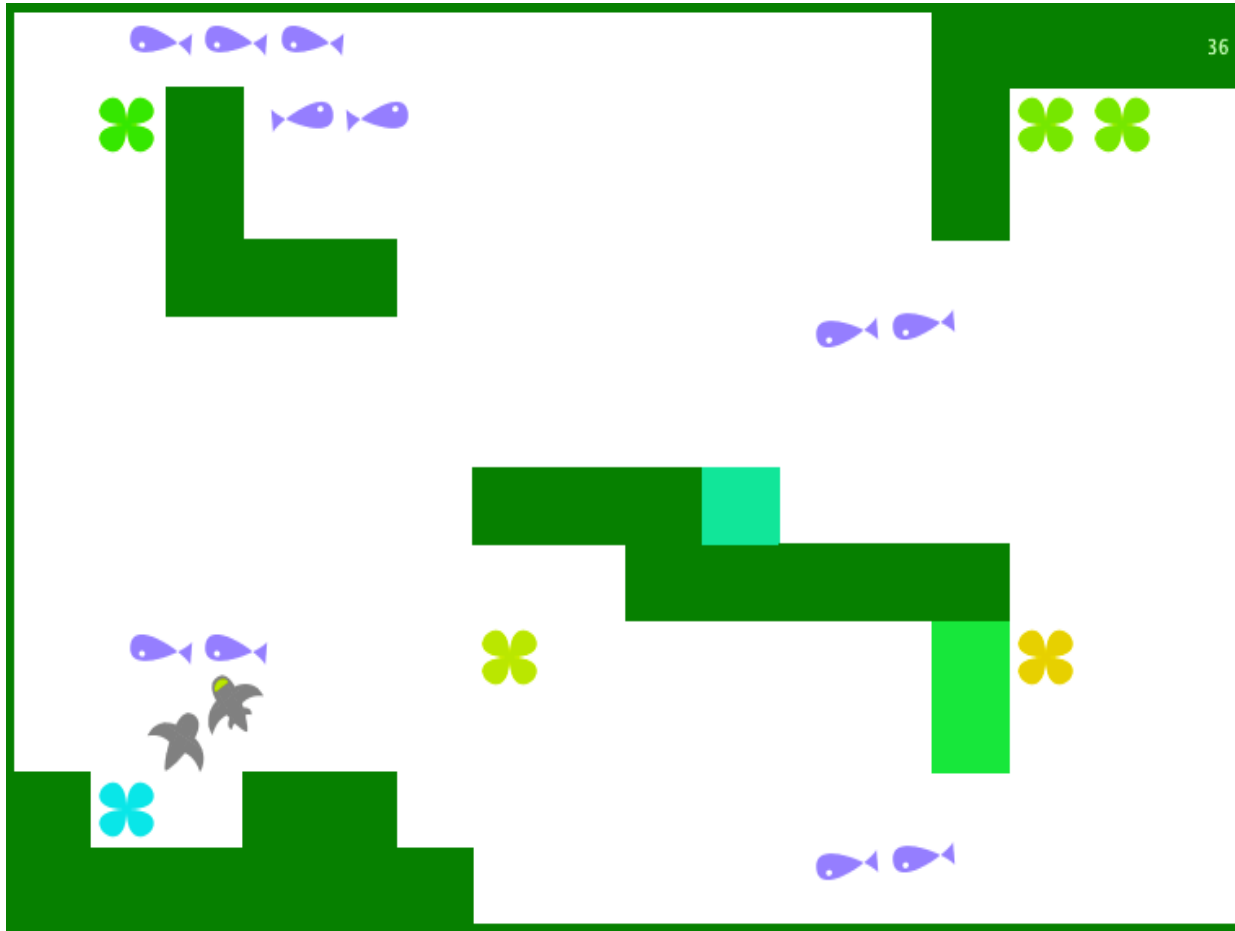
7. Enact

ECA agent exemple 2



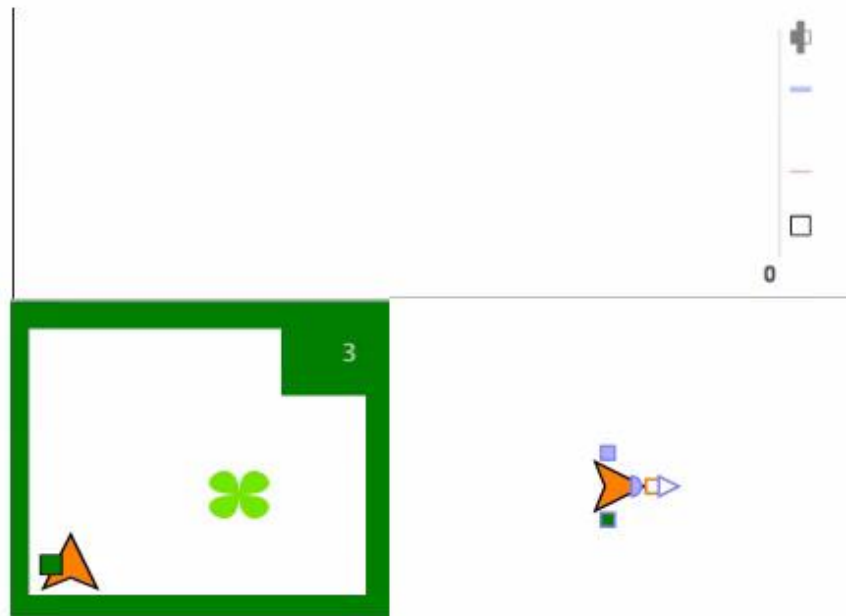
Environnement continu

https://youtu.be/LjOck5ts_2g



From “drives” to “goals”

https://youtu.be/_5HUb6AvSLg



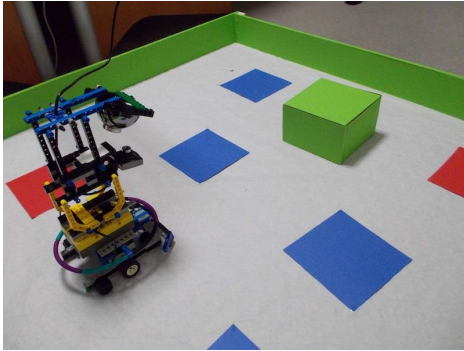
Afforded



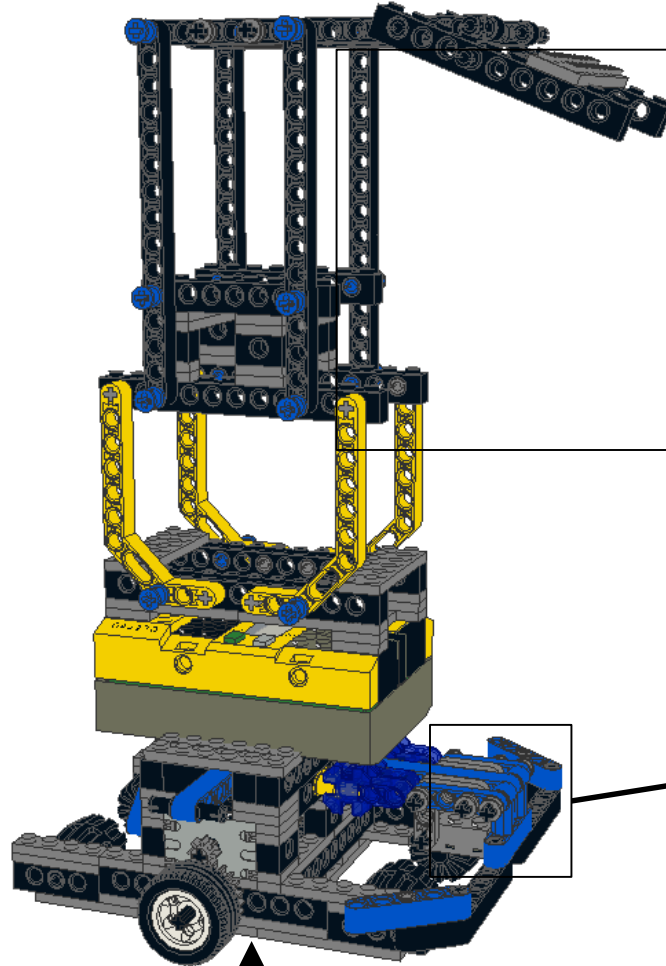
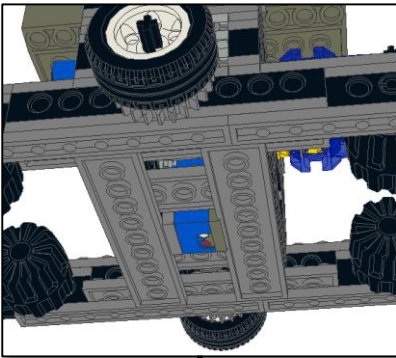
Simulated



Recherche en robotique



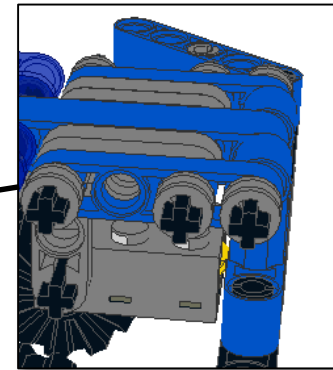
Ground optic sensor



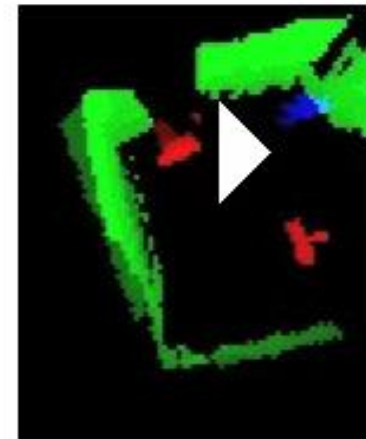
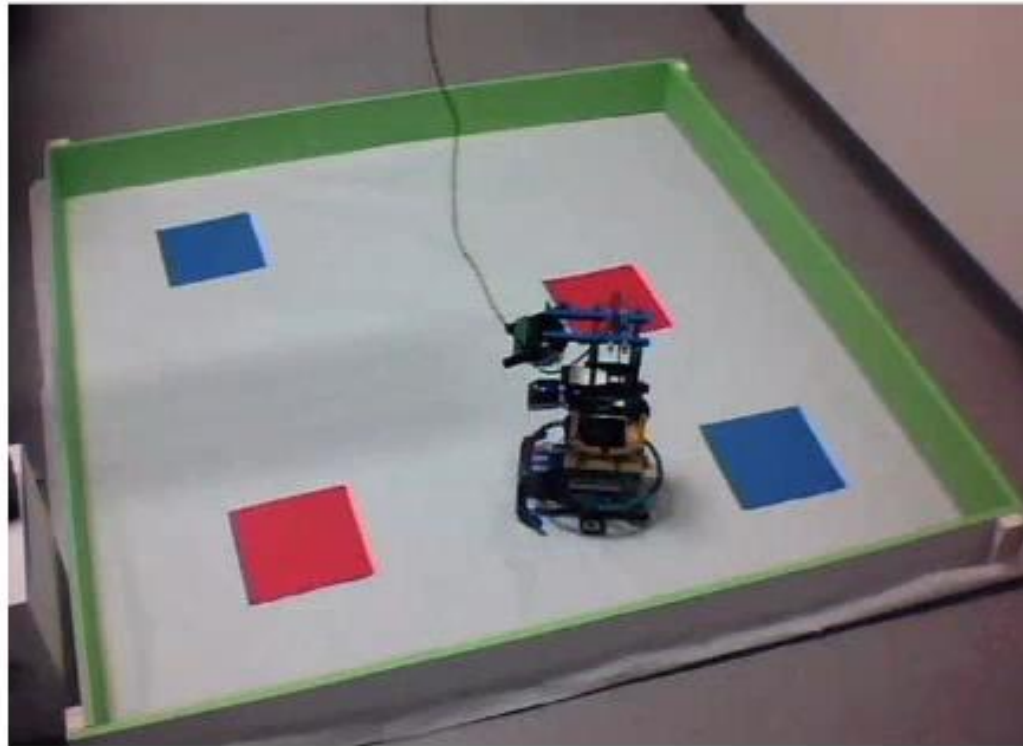
Panoramic camera



Bumper tactile sensor



Expérimentation



Travaux dirigés

Séance 3

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

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

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 valeurs hédonistes d'interaction.

Agent 3

Créer l'Agent3 en suivant les instructions :

<https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-3>

Tester votre agent dans l'environnement TurtlePy

Agent 4

Créer l'Agent4 en suivant les instructions :

<https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-4>

Tester votre agent dans les environnements 1, 2 et 3 et montrer qu'il sais s'adapter à chacun de ces trois environnements.