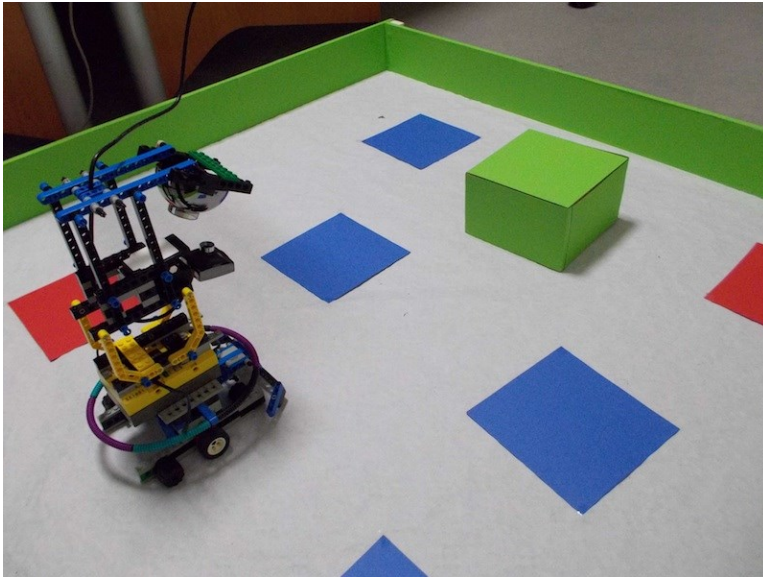


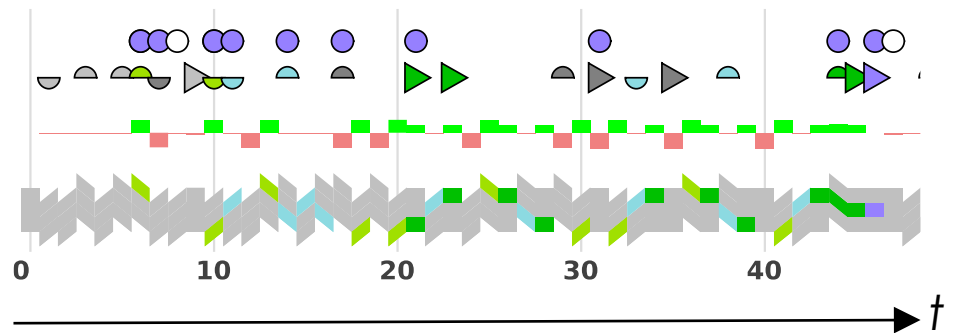
Intelligence Artificielle Développementale



28 Octobre 2022

Olivier.georgeon@gmail.com

<http://www.oliviergeorgeon.com>



Déroulement du cours

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

**Module « IA Développementale » (12h)
Olivier Georgeon**

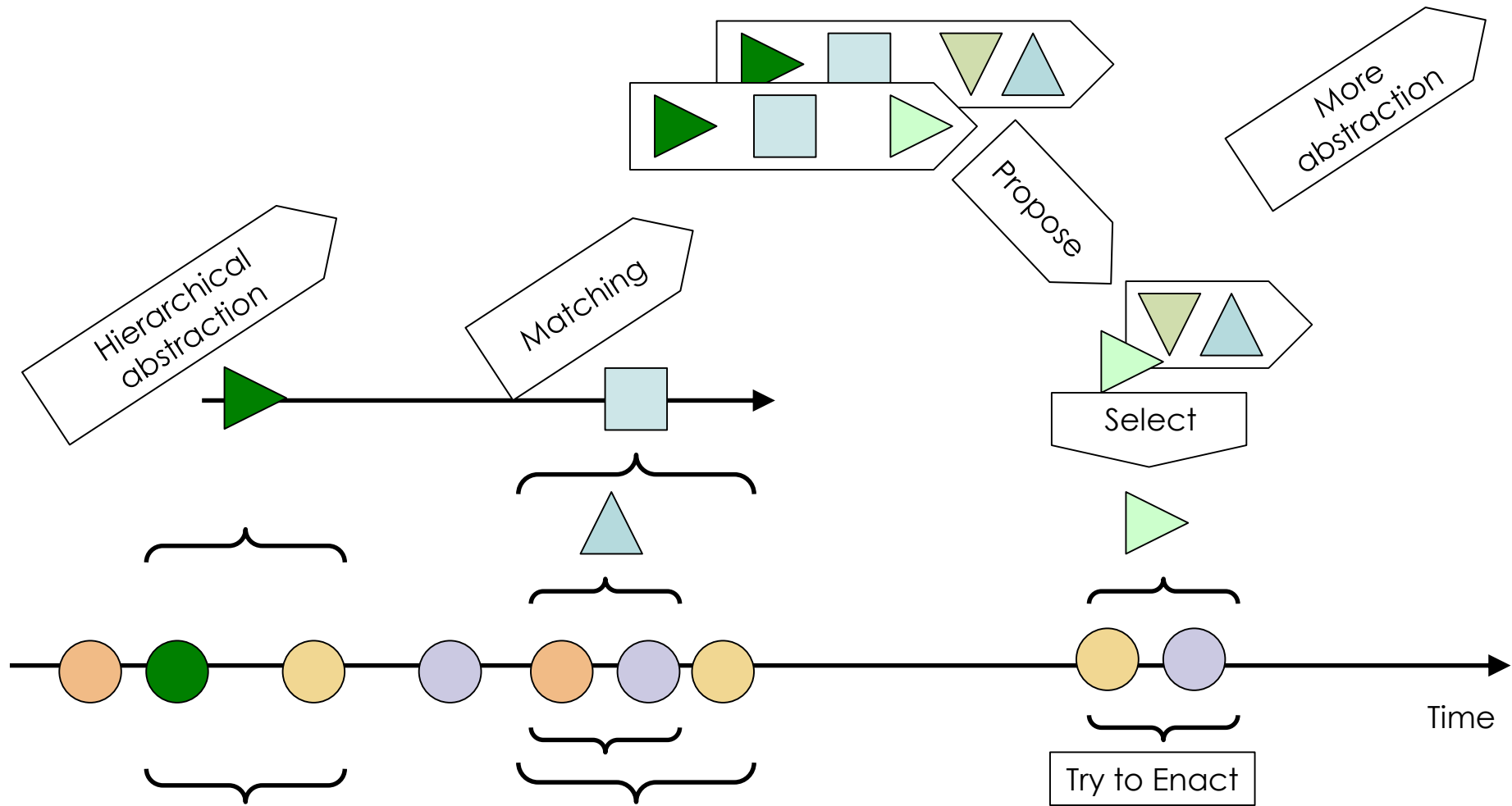
- Vendredi 14 octobre: 3h
- Vendredi 21 octobre: 3h
- Mercredi 26 octobre: 3h
- **Vendredi 28 octobre: 3h**
- Contrôle des connaissances:
 - TD par groupe de 2: 40% de la note de contrôle continu
 - Examen final: 7 points sur 20

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".

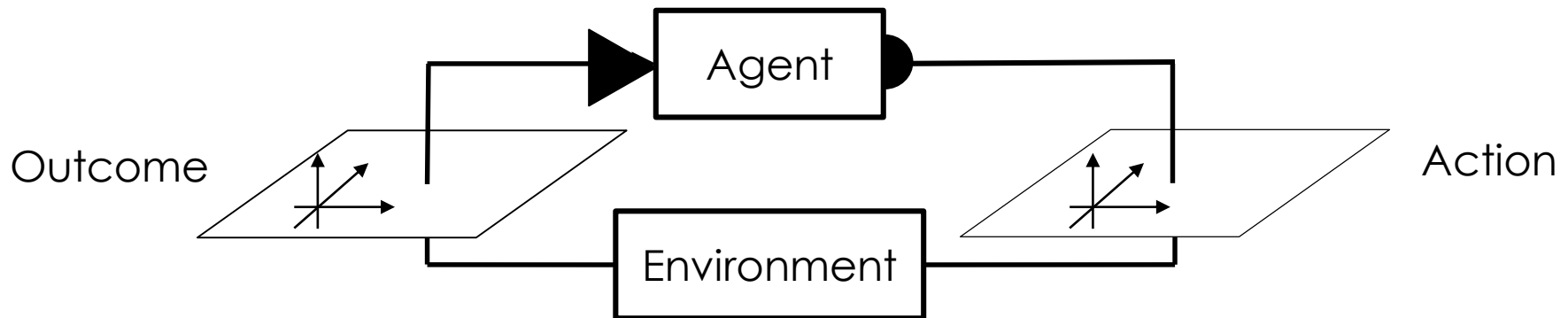
Apprentissage de régularités hiérarchique



Apprentissage spatial et temporel

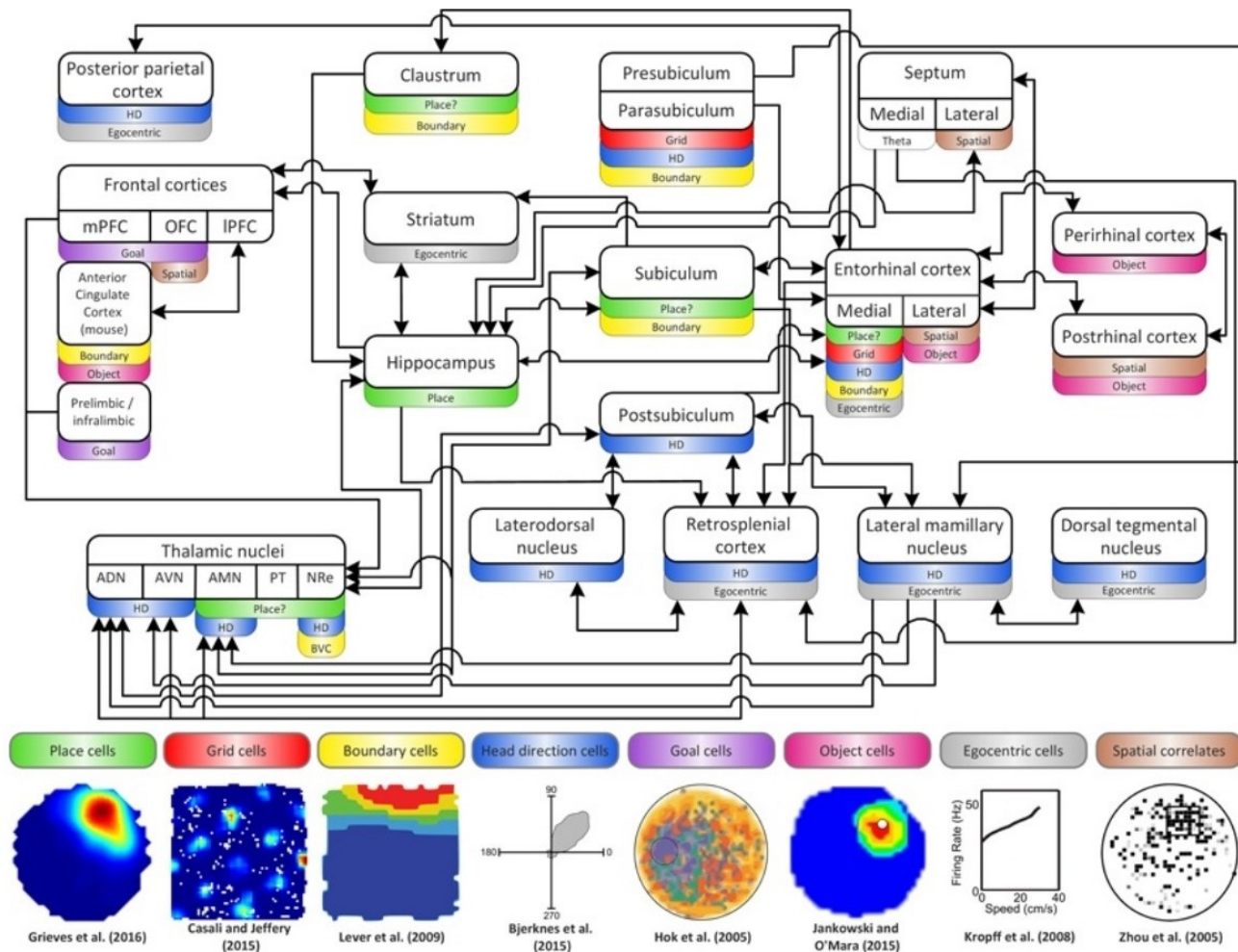
Control loops with spatial information

- Outcome is not representational but contains spatial data
 - Measure of displacement
 - Position of **points of interactions** between the robot and other objects



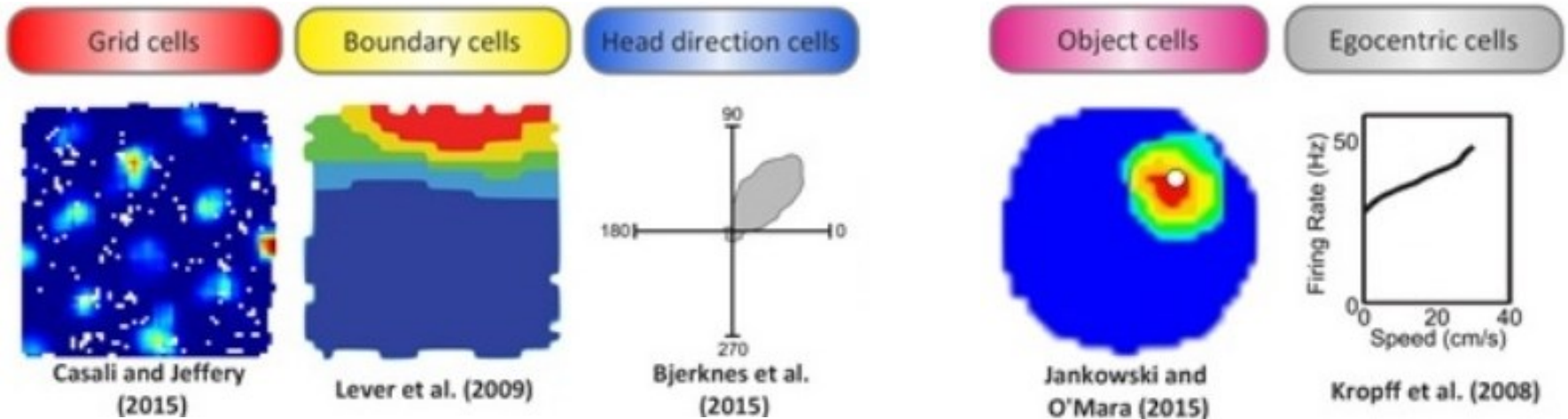
- Proposal:
 - Spatial data is processed by **predefined spatial cognitive structures**
 - The capacity to infer objects from spatially-localized interactions may open the way to more abstract learning
 - Friston & Buzsaki (2017) *The functional anatomy of time: What and when in the brain*

Representation of space in the brain



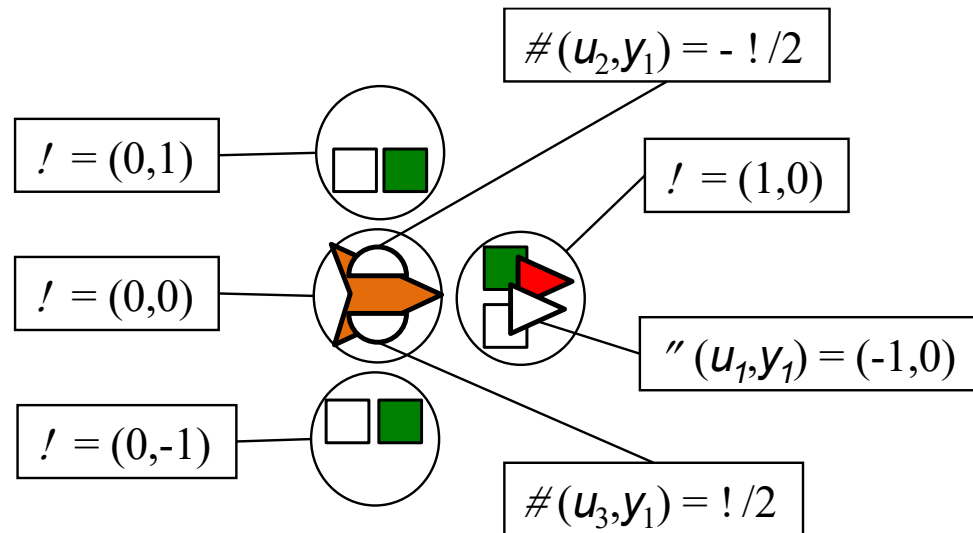
Grievens & jeffery (2017), *The representation of space in the brain*

Sources of inspiration



Grieves & jeffery (2017), *The representation of space in the brain*

Spatial exemple 1

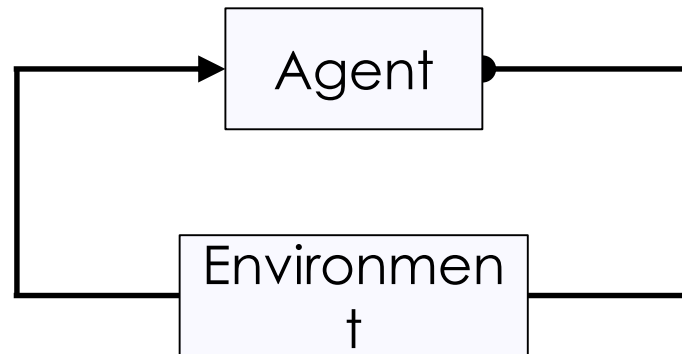


Enacted interaction:



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

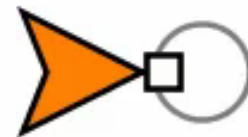
Spatial transformation $\tau = (0,0)$



Intended interaction

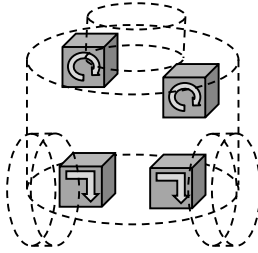
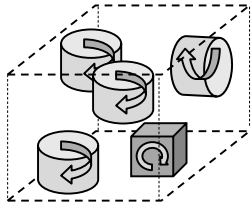
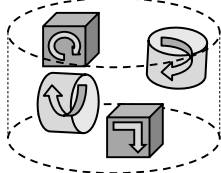
Spatial Example 1

<https://youtu.be/HCDf3Vzl7GM>



Cognitive architecture

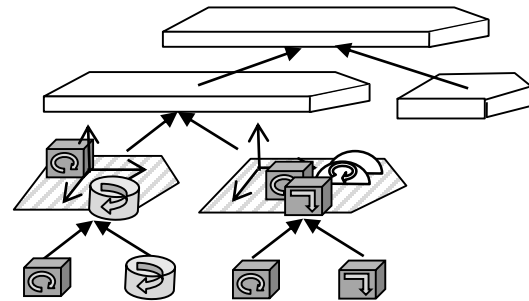
Phenomenal ontology
(long-term memory)



3. Activate

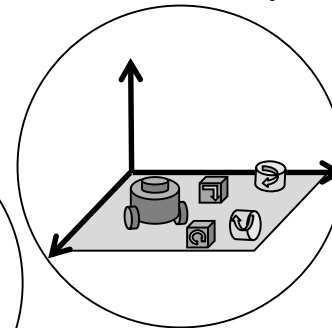
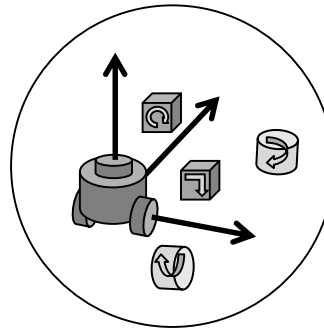
2. Construct

Hierarchical bottom-up
Learning
(long-term memory)



Allocentric spatial
short-term memory

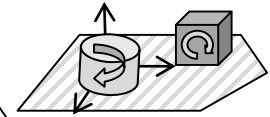
Egocentric spatial
short-term memory



4. Afford

5. Simulate

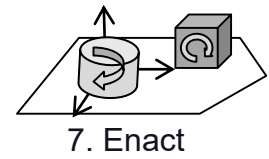
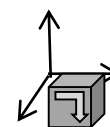
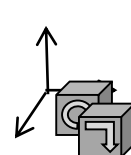
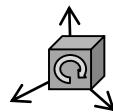
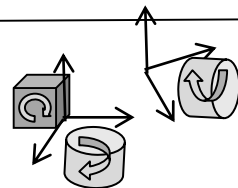
Interaction
selector



6. Try

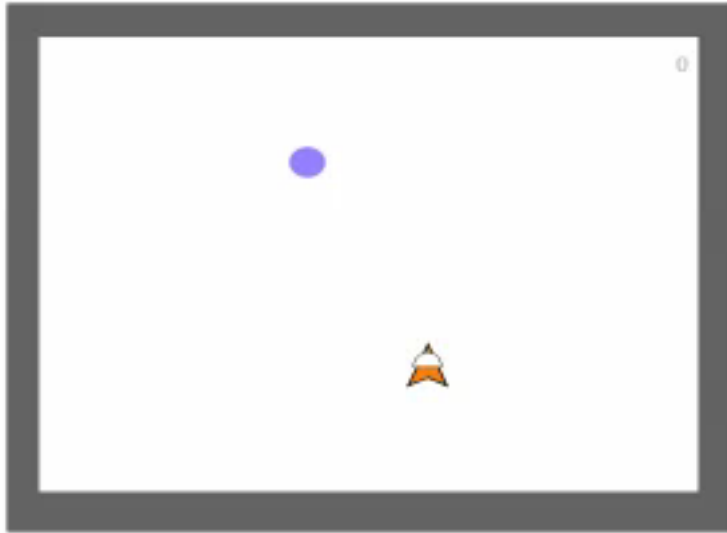
1. Activate

Interaction
timeline



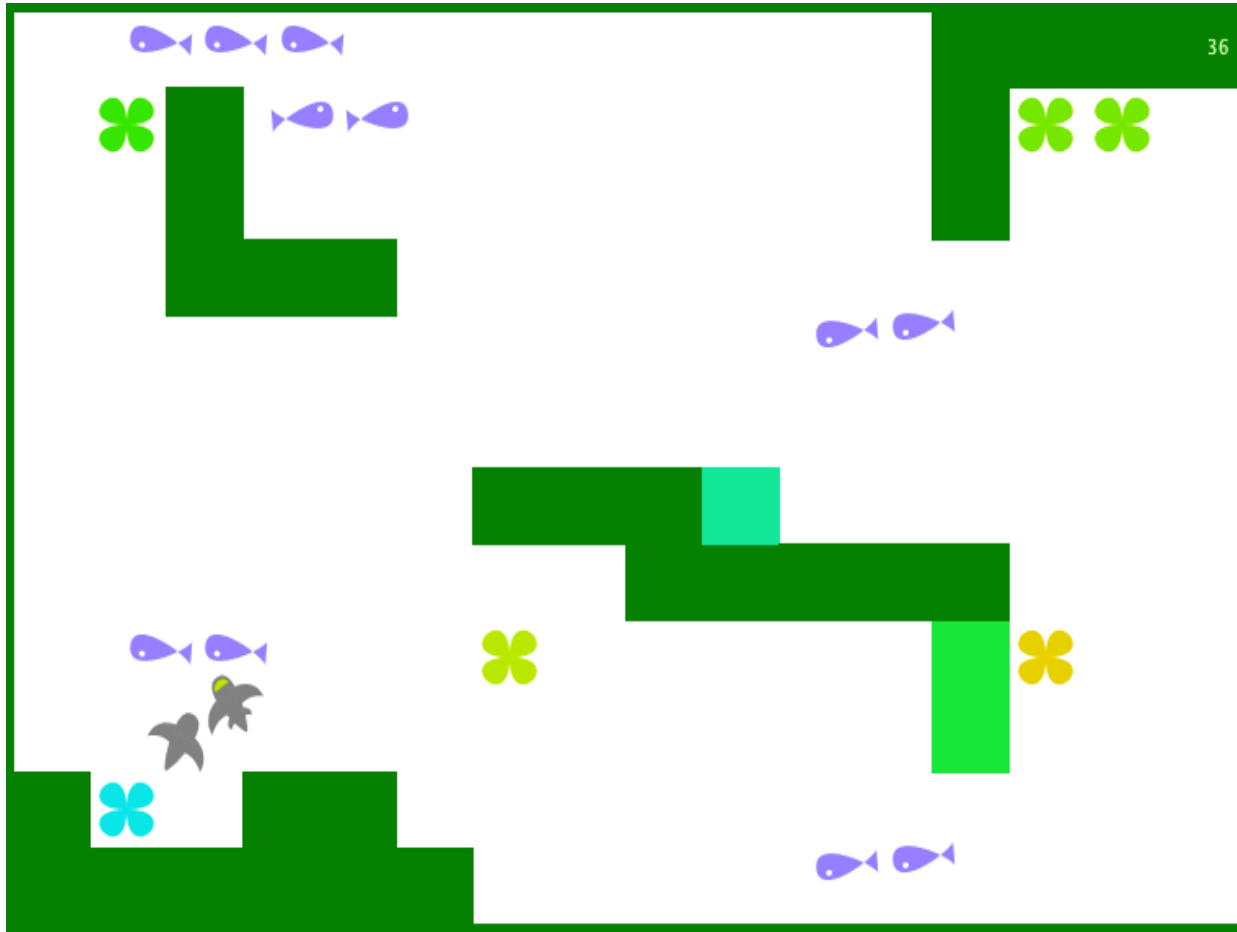
7. Enact

ECA agent exemple 2

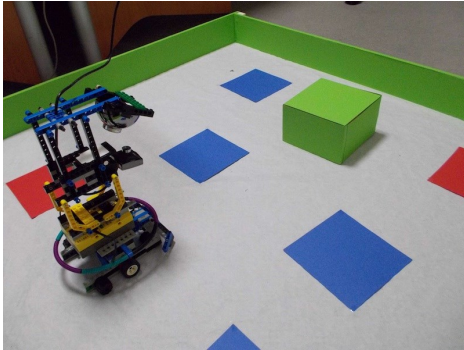


Environnement continu

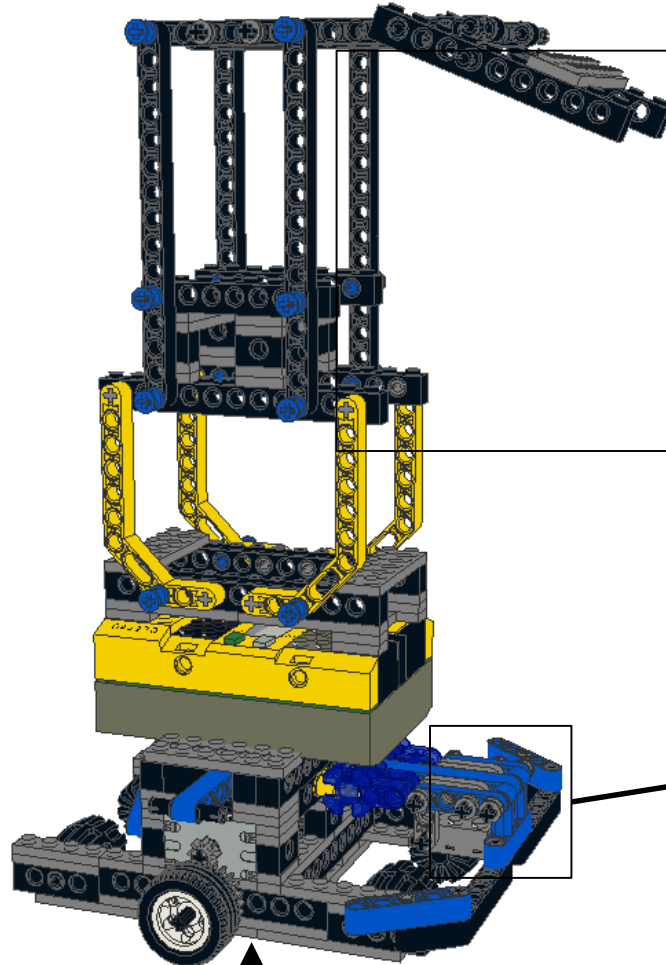
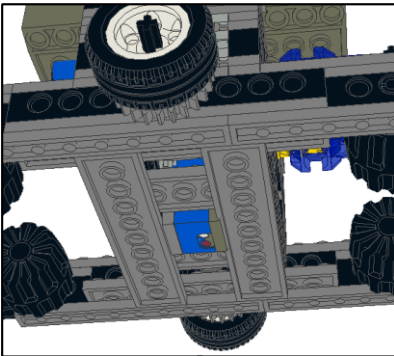
https://youtu.be/LjOck5ts_2g



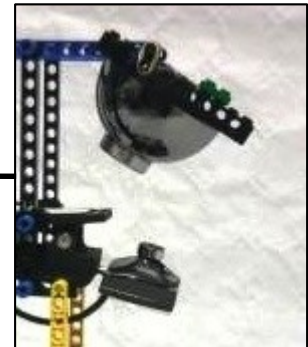
Recherche en robotique



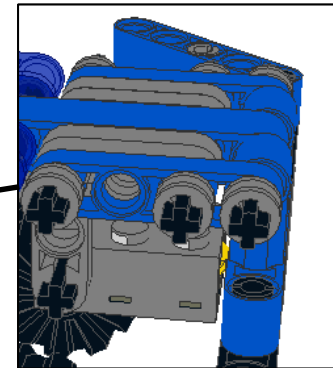
Ground optic sensor



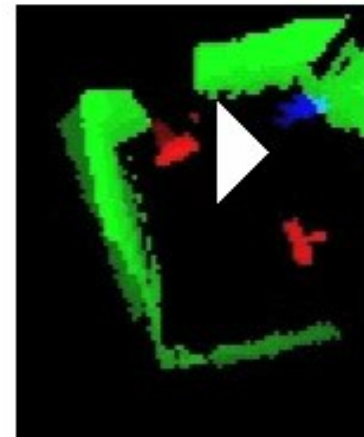
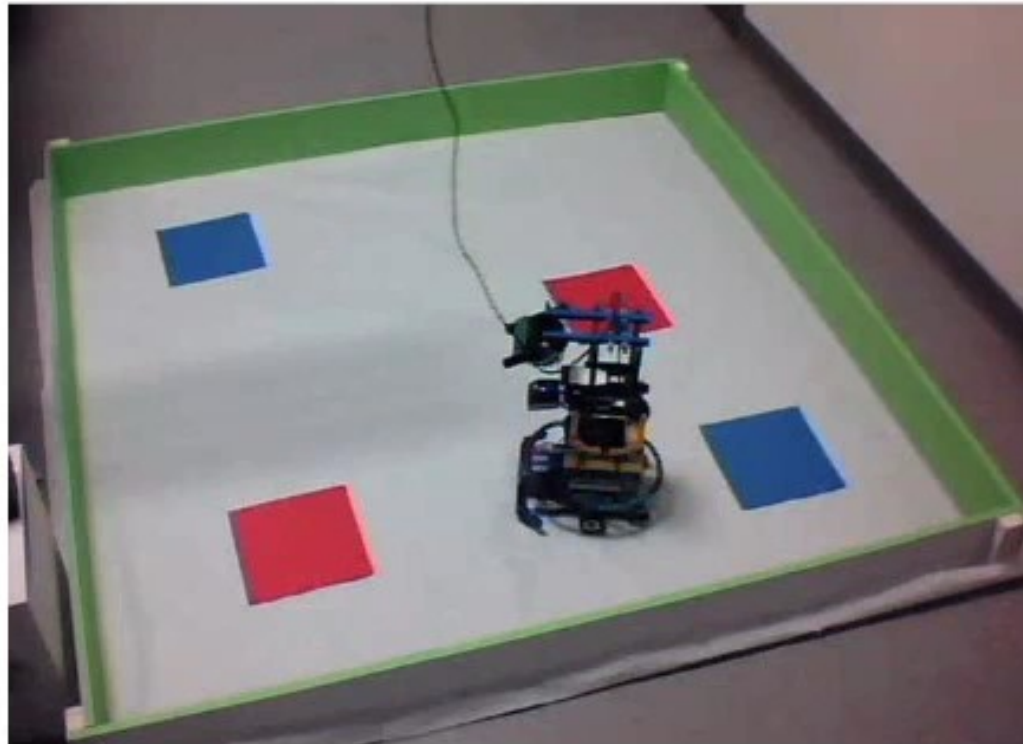
Panoramic camera



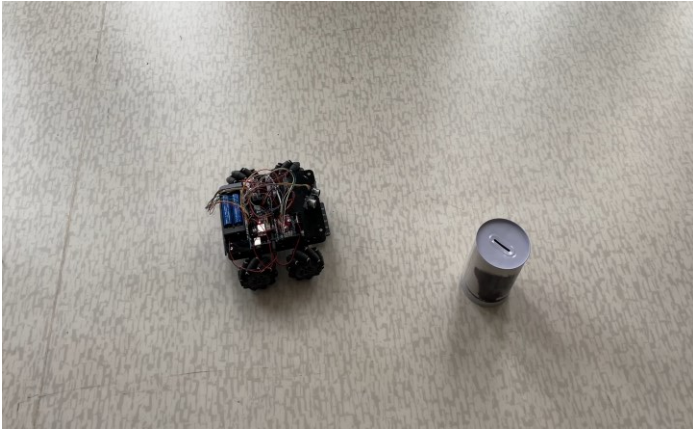
Bumper tactile sensor



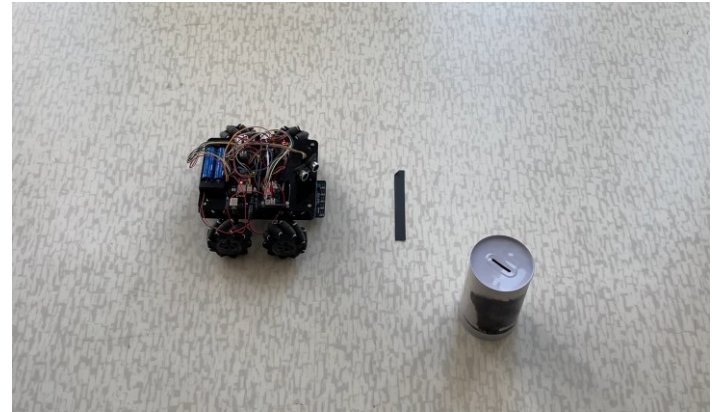
Expérimentation



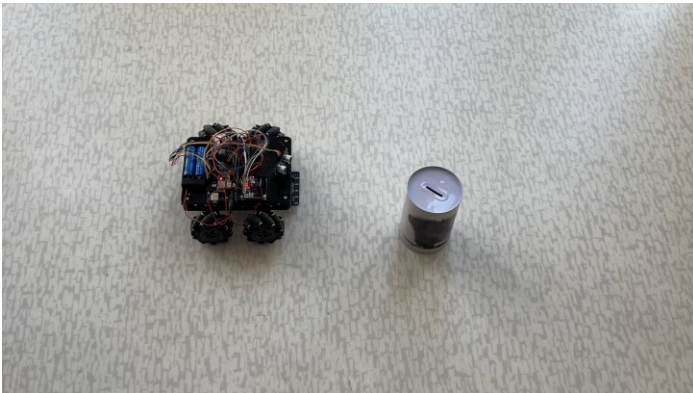
Primitive interactions



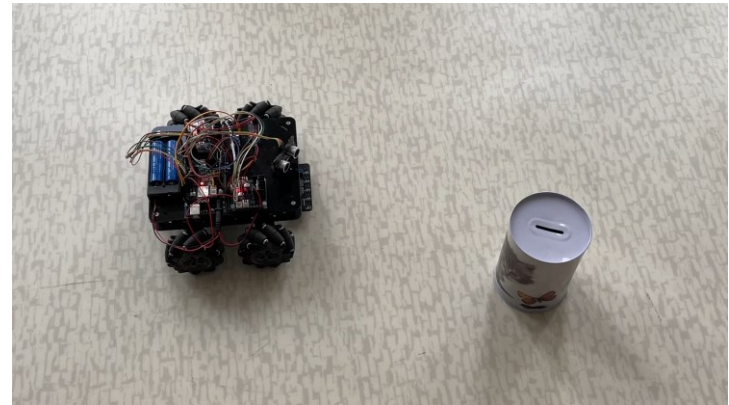
Move_forward with focus



Move forward – line_front

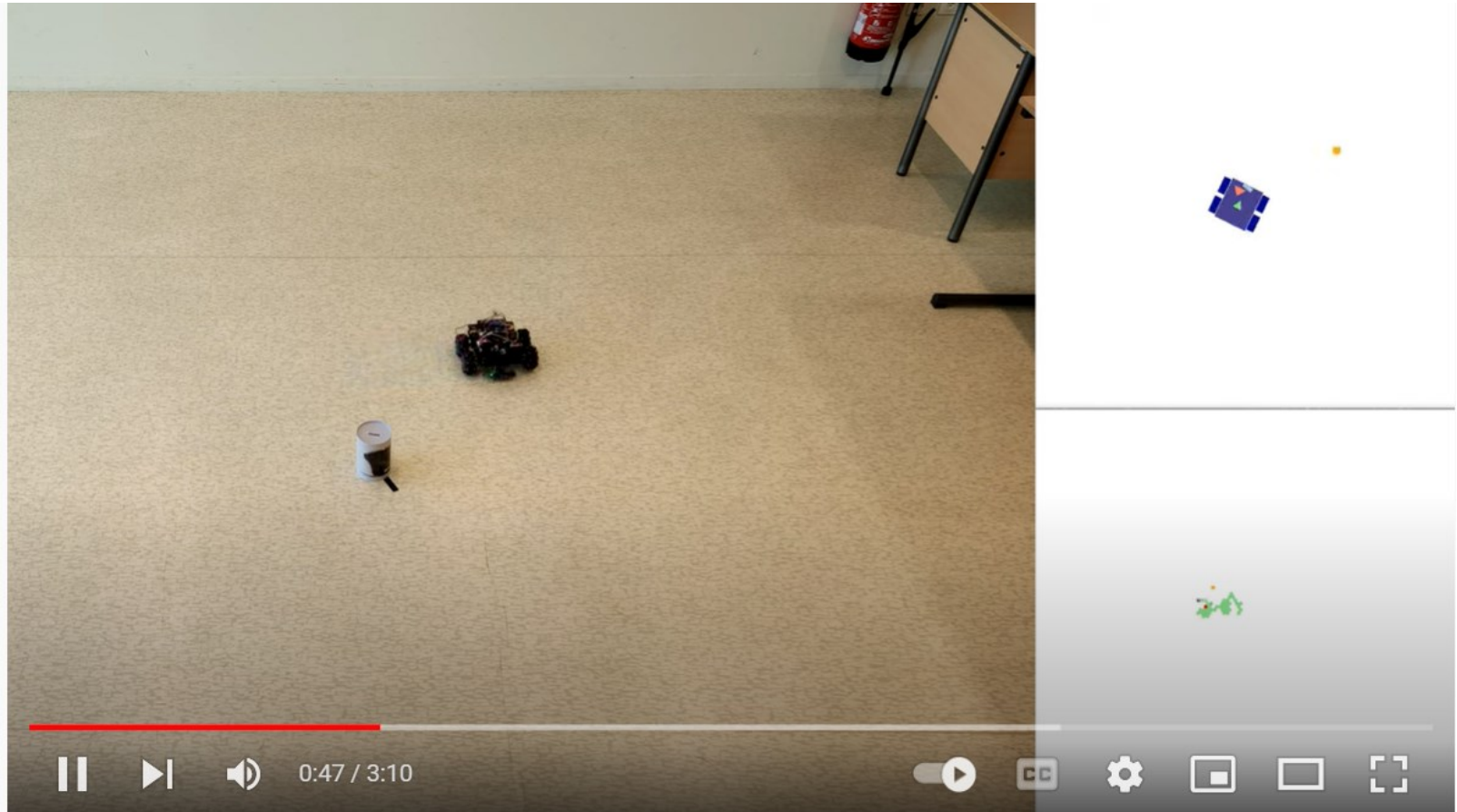


Sweep_right with focus



Scan

Demo 3



<https://youtu.be/W0X60Ami9Rc>
<https://youtu.be/Sue9yMDq0E8>

Travaux dirigés

Séance 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

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 valences 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.