

Dialogue autour de la robotique à l'école primaire

(Pierre-Yves Oudeyer, Didier Roy, Thomas Guitard)

Ce document expose les objectifs d'apprentissage d'une découverte de la robotique à l'école primaire. L'objectif de ce texte est de déterminer quels sont les concepts et les mécanismes fondamentaux que nous souhaitons faire découvrir aux enfants d'école primaire.

La forme qui est utilisée ici est un dialogue entre deux enfants. L'un, ayant suivi le parcours de découverte de la robotique, explique à un second enfant (ou un adulte) ce qu'il a compris et essaie de transmettre ce nouveau savoir. Il faut imaginer que cette discussion a lieu en présence du robot Thymio, de cartes de programmation en carton qu'ils peuvent se montrer, et du logiciel VPL qui permet de programmer le robot, et qu'elle peut être accompagnée de petites démonstrations.

Ce texte pourra servir à éclairer les activités d'IniRobot, à faciliter les explications des concepts mis en œuvre, et à fournir des indicateurs pour évaluation les acquisitions.

Qu'est-ce-qu'un robot ?

Un robot, c'est une **machine** qui peut bouger, et parfois peut aussi déclencher des lumières ou produire des sons, en réaction à ce qui se passe autour d'elle.

Les robots peuvent avoir différentes formes et servir à différentes choses. Par exemple, il y a de gros robots dans les usines qui construisent des voitures, et qui ressemblent à de gros bras en fer avec une pince au bout ou un outil pour souder. D'autres robots un peu de la même forme sont utilisés pour aider les chirurgiens à opérer et soigner les gens. Il y a des robots à roues ou à chenilles qui servent à aller explorer les autres planètes, comme le robot Curiosity sur la planète Mars. Il y a des robots humanoïdes, avec deux jambes, deux bras et une tête, qui existent dans les laboratoires, et qui ressemblent un peu aux robots dans les films, mais ceux dans la réalité sont beaucoup moins perfectionnés. Les robots peuvent aussi être petits et utilisables par les enfants, comme le robot Thymio.



Robot sur roues



Robot médical



Robot sur chenilles



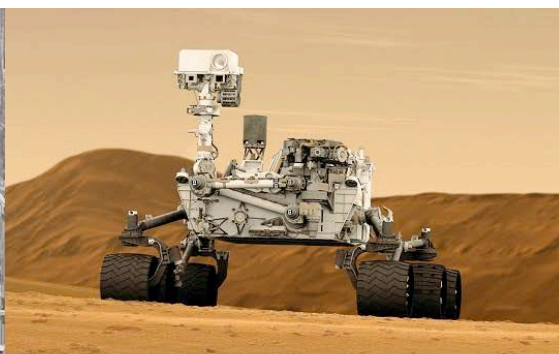
Robot humanoïde



Robot - chien



Robot pour la construction automobile
(bras articulé)



Robot d'exploration spatiale
(Curiosity)

Comment un robot peut-il détecter ce qui se passe autour de lui ?

Pour détecter ce qui se passe autour de lui, le robot a des capteurs. Un **capteur**, c'est une petite boîte qui peut détecter si quelque chose est présent ou disparaît autour de lui. C'est par exemple une petite caméra qui lui permet de voir si un objet ou un mur s'approche, ou un micro qui lui permet de détecter s'il y a du son, ou tout simplement un bouton qui lui permet de savoir si quelqu'un appuie dessus. Quand le robot détecte un objet, ou un son, ou alors qu'un bouton est appuyé, on dit qu'il détecte un **événement**.

Et les capteurs du robot, c'est un peu comme nos sens: nos yeux nous permettent de détecter ce qui est autour de nous, notre peau de toucher les objets qui nous entoure et détecter leur matière, nos oreilles d'entendre les sons, notre langue de détecter les goûts des aliments, notre nez de sentir les odeurs.



Exemple de capteur (laser)
sur un robot aspirateur

Qu'est-ce qui fait bouger le robot ?

Pour bouger le robot utilise des moteurs qui peuvent par exemple faire tourner des roues pour le faire avancer. Le robot peut faire tourner chaque moteur à des vitesses et dans des sens différents, ce qui lui permet d'aller en avant, en arrière, à droite, à gauche, etc. Les moteurs du robot, c'est un peu comme nos muscles, ce sont des parties du corps qui nous permettent de bouger nos bras, nos jambes, notre tête, ou de parler.

Ce machin que tu appelles « Thymio », c'est un robot ?

Oui c'est un robot. Il a été créé par une école scientifique et technologique, l'EPFL (École Polytechnique Fédérale de Lausanne) et l'École Cantonale d'Art de Lausanne. Il ne ressemble pas du tout à un humain, mais plutôt à une boîte avec des roues. On peut mettre un feutre dans le trou qui est au centre pour le faire dessiner, on peut le décorer pour qu'il ressemble à un animal et on peut même réaliser des constructions en accrochant des LEGO dessus. Grâce à ces constructions on peut par exemple le faire marcher sur deux pattes ou encore faire avancer un bateau.



Thymio dessine



Thymio décoré en poisson



Thymio bipède

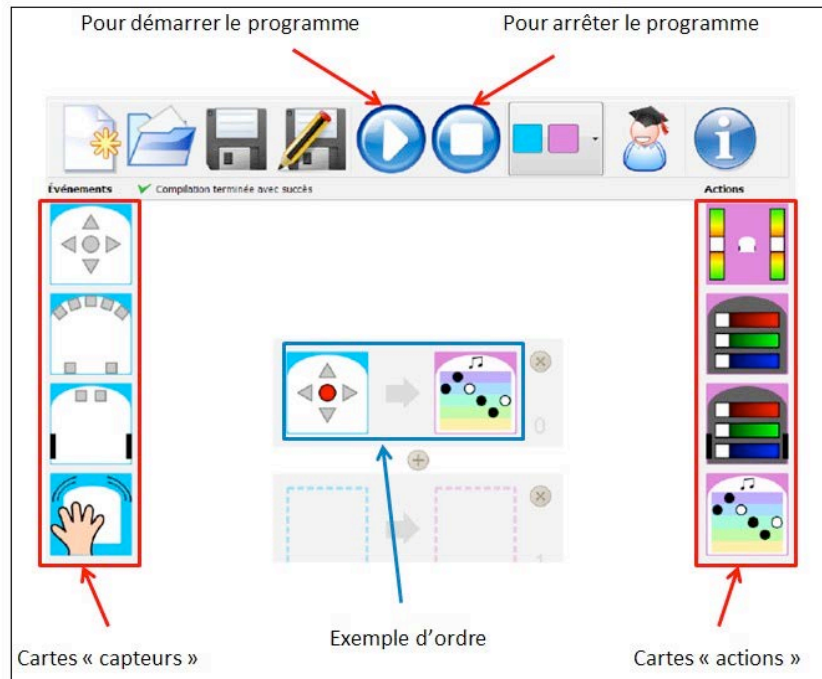


Thymio fait du bateau

Comment un robot décide quelle action faire quand un évènement est détecté ?

C'est un humain qui lui explique, en lui donnant des **instructions**, des ordres. Mais le robot ne comprend pas quand on lui parle avec notre langue habituelle, en français par exemple. Il faut lui parler dans une langue spéciale, qui lui permet de comprendre nos instructions. Les langues que comprennent les robots s'appellent des "**langages de programmation**". Il y en a plusieurs sortes.

Par exemple, pour dire au robot Thymio ce qu'il doit faire, on utilise un langage qui s'appelle « VPL », dans lequel il y a des petites cartes avec des dessins qui représentent des événements qu'il peut détecter (carte "capteurs"), ou des actions qu'il peut faire (carte "actions").

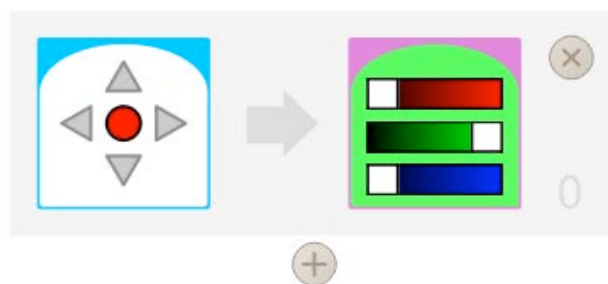


Interface du logiciel permettant de programmer Thymio (en VPL)

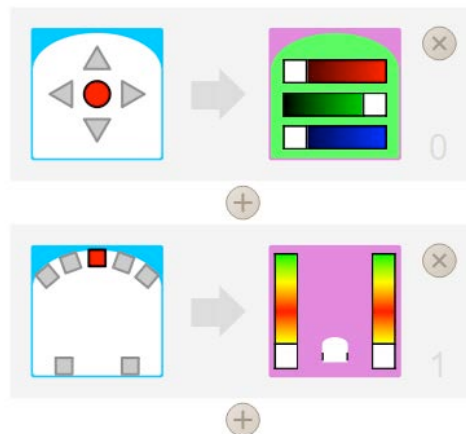
Ces cartes permettent de donner au robot des **instructions**. Par exemple, si on veut que le robot allume sa lumière verte quand on appuie sur le bouton rond, on prend la carte "capteur" qui correspond au capteur "bouton rond appuyé", et la carte "action" qui correspond à "allume la lumière verte", et on place ces deux cartes dans deux emplacements au centre qui forment alors ce qu'on appelle une règle "SI ... ALORS ...". Cette règle "**SI** bouton rond appuyé **ALORS** allumer la lumière verte" est une instruction qu'on donne au robot qui lui permet de comprendre que **SI** son bouton rond est appuyé **ALORS** il doit allumer sa lumière verte.



Exemple de programme à une instruction
(association d'un événement à une action)
exprimé en français



Si on veut, on peut donner au robot plusieurs instructions. Par exemple, on pourrait en plus lui dire "SI tu détectes un objet sur le devant, ALORS recule". Comme cela, il aurait en tout deux instructions. S'il arrive au robot des événements prévus par les cartes capteurs qui sont dans ces instructions, alors il déclenche l'action qui correspond. L'ensemble des instructions données au robot s'appelle un "**programme**".



Exemple d'un programme à deux instructions
langage VPL (Visual Programming Language)
spécifique au robot Thymio

D'accord, mais ces cartes, elles sont comment ? En carton ?

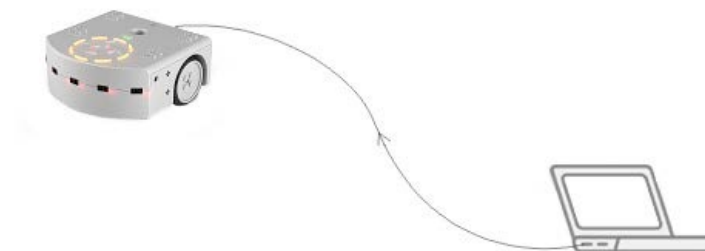
Quand on veut réfléchir à quelles instructions donner au robot, on peut en effet utiliser des cartes en carton, et construire des règles par exemple sur une table ou sur le tableau. Mais le robot ne peut ni voir ni lire ces cartes en carton. Alors on utilise des cartes spéciales dans un logiciel de programmation sur un ordinateur. Pour Thymio, ce **logiciel** s'appelle VPL.

Sur l'écran, on a les différentes cartes, et avec la souris on doit les positionner pour faire des règles. Quand c'est fini, on clique sur le bouton "Play" et le robot les **exécute**.



Mais l'ordinateur qu'on utilise, il est à côté du robot, comment le robot peut-il voir ce qu'on lui a demandé avec le logiciel ?

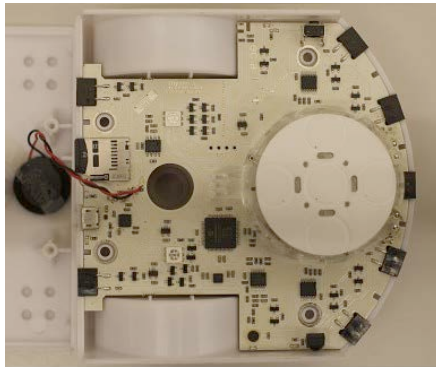
Parce que le robot est relié à l'ordinateur par un câble. Quand on appuie sur le bouton "Play", le logiciel sur l'ordinateur envoie par le câble au robot les instructions qu'on a construites.



Transfert du programme dans le robot

Si on débranche le fil entre le robot et l'ordinateur, le robot peut-il encore suivre les instructions ?

Oui, parce que dans le robot, il y a un **mini-ordinateur**, qui fait plusieurs choses. D'abord, il peut mémoriser les instructions qui viennent de l'autre ordinateur par le câble. Donc si on enlève le câble, le robot s'en souvient quand même. Ensuite, ce mini ordinateur est connecté (avec des fils électriques) aux capteurs, aux moteurs, aux lumières, aux haut-parleur. En permanence, il reçoit des informations des capteurs, et cela lui permet de détecter s'il arrive des évènements prévus dans les règles qu'on a construites. Si c'est le cas, le mini ordinateur envoie alors aux moteurs, ou aux lumières ou à ses haut-parleurs, les actions à effectuer.



Micro ordinateur



Moteurs (en jaune)

Ce mini-ordinateur est un peu comme ceux qu'on utilise avec des écrans et des claviers, comme ceux aussi qui sont dans les téléphones ou les tablettes. Si on démonterait nos ordinateurs ou nos téléphones à la maison, on trouverait dedans des petites plaques vertes **électroniques** sur lesquelles sont soudées des petites puces noires. Ces petites puces noires, on les appelle des "**processeurs**", et elles contiennent des circuits souvent très compliqués de fils électriques. C'est elles sur l'ordinateur qui contrôlent ce qui s'affiche sur l'écran, comment faire les calculs qu'on lui demande, comment se connecter à internet, et quoi faire quand on tape des choses sur le clavier ou qu'on bouge la souris. C'est elle aussi qui ont des mémoires. Et c'est ce type de carte électronique avec des processeurs qu'il y a aussi dans les robots comme le Thymio.



Carte électronique

D'ailleurs, pour construire des logiciels sur les ordinateurs ou des applications sur les téléphones, comme par exemple des jeux vidéos, il faut là aussi expliquer à l'ordinateur ou au téléphone ce qu'il doit faire en utilisant un langage de programmation. Par exemple, dans un jeu vidéo, pour lui dire comment faire bouger le personnage qu'on contrôle en fonction des boutons sur lesquels on appuie, et comment faire bouger les monstres qui essaient de l'attrapper.

Est-ce-que ce sont des puces électroniques qui sont aussi dans notre corps et nous permettent de décider quelle action faire en fonction de ce que l'on perçoit ?

Non. Dans notre corps, nous avons le système nerveux, et en particulier le cerveau et la moelle épinière. Dans ce système nerveux, nous avons des milliards de petites cellules nerveuses, qui sont connectées entre elles de manière très compliquée. Comme sur les puces électroniques, elles forment des circuits et échangent des informations avec des signaux électriques. Mais ces circuits de cellules nerveuses sont très différents des circuits électroniques dans les ordinateurs. En particulier, ce qu'ils font n'est pas décidé par un programme informatique qui a été écrit par quelqu'un d'autre. Et on est encore loin de comprendre entièrement comment ils fonctionnent. Ce qu'on sait, c'est qu'ils sont connectés aux sens et à nos muscles, et que c'est grâce à eux que nous pouvons apprendre, former des souvenirs et choisir quoi faire en fonction de ce que nous percevons et de ce qu'on a appris. C'est aussi ces circuits nerveux qui nous permettent de réfléchir ou de rêver.



Exemple de circuit nerveux du corps humain

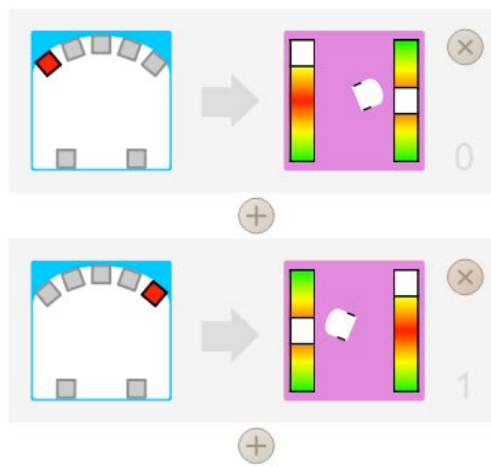
Revenons à nos robots. Quand on construit un programme, est-ce-que l'ordre des règles qu'on donne au robot a de l'importance ?

Oui, cela lui permet de savoir quelle action faire quand plusieurs règles lui disent de faire des actions contradictoires. Dans VPL, plus la règle apparaît près de la fin dans le programme, plus elle est prioritaire.

Par exemple, imaginons le programme suivant:

Règle 1: "Si tu détectes un objet sur ta gauche, tourne à droite"

Règle 2: "Si tu détectes un objet sur ta droite, tourne à gauche".

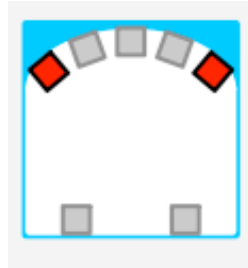


Si à un moment le robot détecte en même temps à la fois un objet sur sa gauche et un objet sur sa droite, va-t-il tourner à gauche ou à droite ?

Pour décider, il regarde quelle règle apparaît en dernier: c'est la 2, donc il va tourner à gauche.

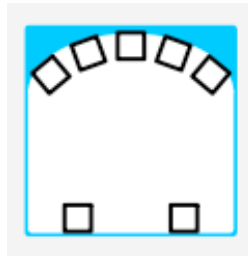
Est-ce-qu'on peut demander au robot de ne déclencher une action que quand il détecte plusieurs événements en même temps ?

Oui, par exemple avec les cartes des capteurs d'obstacles. Si dans une règle on utilise une carte de capteur d'obstacle et que l'on coche en rouge sur la carte à la fois les capteurs à droite et à gauche, cela permet de dire au robot "applique cette règle seulement si ces capteurs détectent un obstacle en même temps".



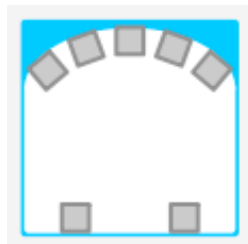
Est-ce-qu'on peut demander au robot de faire une action quand il ne détecte rien ?

Oui. Le fait de ne rien détecter est aussi une forme d'évènement, et on peut l'utiliser dans les règles. Si dans une règle on utilise une carte de capteur d'obstacle, et que l'on coche en blanc tous ses capteurs, cela permet de dire au robot "applique cette règle seulement si tous les capteurs ne détectent rien".



J'ai vu que sur les cartes de capteurs, on pouvait aussi laisser la couleur d'un capteur en gris. Qu'est-ce-qui se passe dans une règle si un capteur est laissé gris ?

Cela veut simplement dire au robot qu'il ne doit pas tenir compte de ce que le capteur détecte pour décider s'il applique la règle ou pas.



Est-ce-que quand on écrit un programme, le robot va toujours faire les mêmes mouvements ?

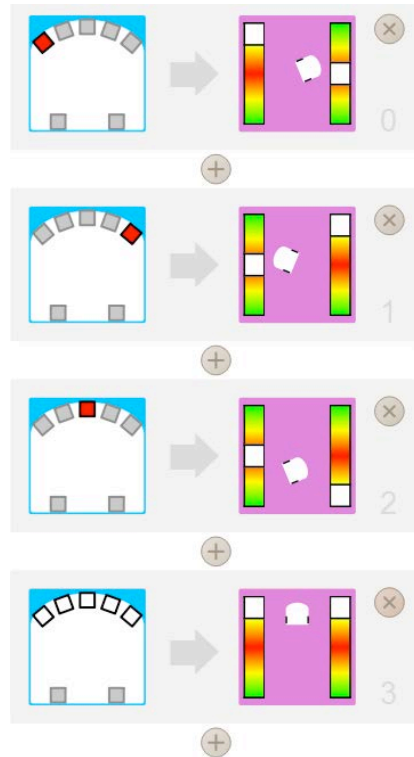
Non, parce que dans le programme, on lui dit seulement quoi faire quand il détecte des évènements. Mais l'ordre et le détail des mouvements qu'il fait dépend de l'ordre des évènements qu'il va détecter, et ça on ne peut pas le prévoir à l'avance. Par exemple, imaginons qu'on donne le programme suivant au Thymio :

Règle 1: "SI tu détectes un objet sur ta gauche, ALORS tourne à droite"

Règle 2: "SI tu détectes un objet sur ta droite, ALORS tourne à gauche"

Règle 3: "SI tu ne détectes un objet devant toi, ALORS recule vers la gauche"

Règle 4: "SI tu ne détectes rien, ALORS avance tout droit"



Exemple de programme VPL
à 4 instructions

Avec ces règles, si au début il n'y a aucun obstacle autour de lui il va avancer tout droit, puis s'il rencontre ensuite un obstacle sur la droite, il va tourner à gauche.

Au contraire, avec exactement les même règles, s'il y a dès le départ un obstacle sur sa gauche, il va commencer par tourner à droite, et si ensuite il n'y a plus d'obstacle, il va avancer tout droit.

Cela permet de s'apercevoir que le robot, d'une certaine manière, peut adapter son comportement à son environnement. Par exemple, pour le faire avancer sur un parcours sans toucher les bords, pas besoin de prévoir à l'avance tous les détails du parcours. Ces quatre règles vont lui permettre de s'en sortir sur la plupart des parcours.

Mais est-ce-que si on utilise le même programme, et qu'on met le robot dans le même parcours, avec la même position de départ, il va faire exactement les mêmes mouvements ?

On peut penser que oui, mais en réalité il ne va pas faire exactement les mêmes mouvements. Parfois même, il va finir par suivre une trajectoire très différente. C'est parce que ces capteurs et ses moteurs ne sont pas parfaits. Par exemple, il peut arriver qu'un capteur ne détecte pas bien un obstacle ou qu'un moteur ne soit pas assez précis parce que la batterie du robot est usée ou que ça glisse sur le parcours. Au début, cela peut provoquer des petites différences de réaction du robot, mais ces différences peuvent s'accumuler et faire qu'à un moment sa trajectoire devient très différente. Néanmoins, si l'on choisit bien les règles, le robot va souvent réussir la tâche si le parcours n'est pas trop compliqué.

Cela veut dire que le robot peut produire des mouvements compliqués et biscornus, voire difficiles à prévoir, même si on lui a donné des instructions bien précises ?

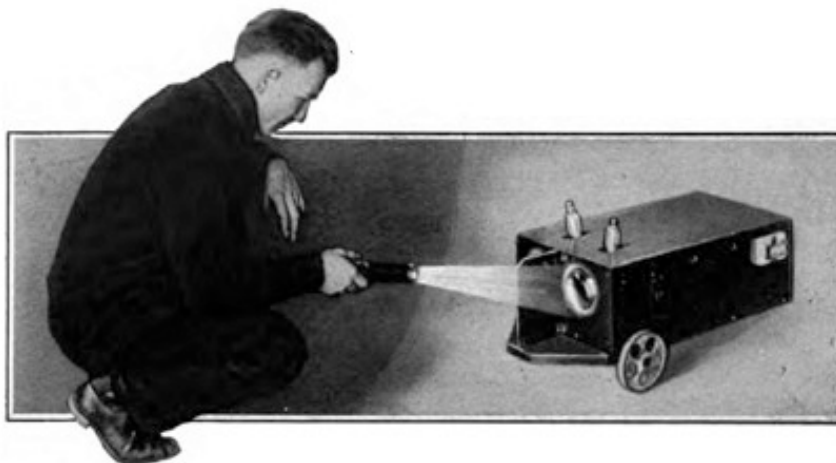
Oui, tout à fait. Cela ne veut pas dire que le robot ne vas pas réaliser ce qu'on voulait qu'il fasse (par exemple avancer dans le parcours sans toucher les murs), mais que le programme, en interaction avec l'environnement, peut l'amener à produire des mouvements parfois biscornus pour y arriver.



Exemple de trajectoire
de robot à deux roues

Un peu les mêmes d'ailleurs que ceux qu'on observe chez certaines espèces d'insectes, qui cherchent la lumière tout en évitant les obstacles. Quand on regarde par exemple les moustiques et les papillons de nuit voler autour d'un lampadaire, on les voit faire des trajectoires très compliquées et qui ne sont jamais exactement les mêmes.

Les scientifiques croyaient jusqu'au début du 20ème siècle que pour expliquer ces trajectoires, il fallait soit que les insectes aient un cerveau très puissant, soit qu'une sorte d'"esprit mystérieux" les fasse bouger (mais sans vraiment savoir ce que cet "esprit mystérieux" pouvait être). C'est grâce à deux ingénieurs (Hammond et Miessner) qui en 1912 ont construit un robot très simple, qui s'appelait "Séléno", et fonctionnait un peu comme le Thymio, qu'on s'est aperçu que des connexions très simples entre les capteurs et les moteurs (les muscles pour les insectes) permettaient de produire ces comportements qui semblaient compliqués. Construire des robots, ça peut nous aider à comprendre comment fonctionnent certains animaux.



Quand on donne au robot des règles comme celles dont on a parlé, le robot ne fait que réagir à des événements qui viennent juste d'arriver. Comment faire pour qu'il tienne compte d'événements qui ont eu lieu bien avant dans le passé ? Peut-il avoir de la mémoire ?

C'est vrai, quand on donne au robot des règles comme "SI tu détectes un obstacle sur ta droite, ALORS tourne à gauche", c'est une sorte de réflexe, et le robot n'a pas besoin de mémoire pour les appliquer.

Mais comme on l'a vu, un robot contient un mini ordinateur, et un ordinateur ça a de la **mémoire**. Cette mémoire, ce sont des sortes de petites cases qu'il peut allumer ou éteindre, pour se souvenir s'il a détecté tel ou tel événement. Par exemple, le robot peut allumer une de ces cases pour se souvenir qu'il a détecté un obstacle sur sa droite.



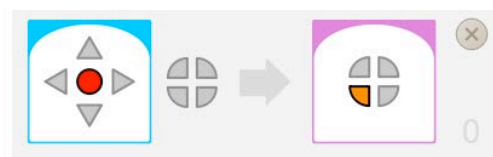
Avec le langage de programmation VPL, on peut dire au robot, avec des cartes spéciales où il y a le dessin de ces petites cases, d'allumer ou d'éteindre certaines d'entre elles dans les instructions qu'on lui donne. Par exemple, on peut construire une règle pour le robot qui dit: "SI tu détectes qu'on appuie sur ton bouton rond, ALORS allume ta petite case numéro 3".

Ainsi, si le robot détecte qu'on appuie sur le bouton, il allume la case et plus tard il peut tout simplement regarder si elle est éteinte ou allumée pour savoir si quelqu'un a déjà appuyé sur le bouton. Et en fonction de cela, il peut choisir différentes actions. Par exemple, on pourrait donner au robot le programme suivant;

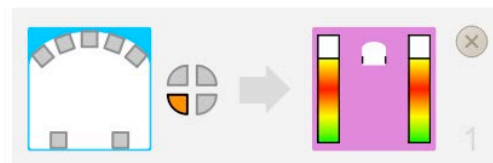
Règle 1: "SI on appuie sur ton bouton rond, ALORS allume la petite case numéro 3"

Règle 2: "SI la case numéro 3 est allumée, ALORS avance tout droit"

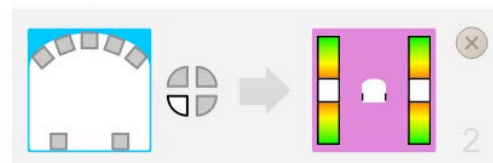
Règle 3: "SI la case numéro 3 est éteinte, ALORS arrête les moteurs"



Si on appuie sur le bouton, quel que soit l'état de la mémoire
Alors on allume la case mémoire en bas à gauche



Quel que soit ce que Thymio détecte, si la case mémoire est allumée
Alors le Thymio avance

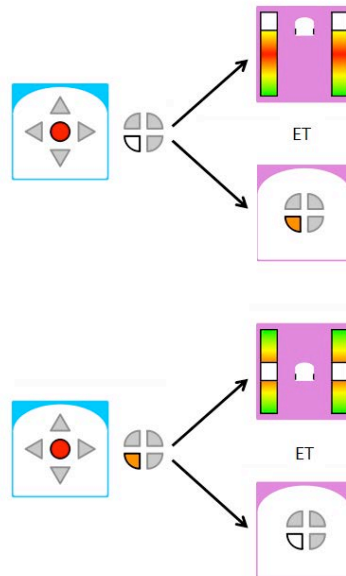


Quel que soit ce que Thymio détecte, si la case mémoire est éteinte
Alors le Thymio reste sur place

Autre exemple, avec un programme qui fait que le bouton rond sert d'interrupteur pour déclencher le mouvement si le robot est arrêté, et l'arrêter s'il est en mouvement :

Règle 1: "SI tu détectes qu'on appuie sur ton bouton rond ET que la case 3 est éteinte, ALORS avance tout droit ET allume la case 3"

Règle 2: "SI tu détectes qu'on appuie sur ton bouton rond ET que la case 3 est allumée, ALORS arrête les moteurs ET éteint la case 3"

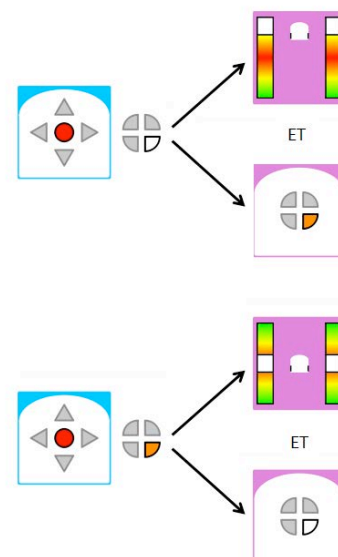


Ici, on a choisi d'utiliser la case numéro 3, et elle permet au robot de se souvenir si sa dernière action était d'avancer ou de s'arrêter. Et on a aussi choisi de dire que si elle est allumée, alors cela voulait dire que sa dernière action était "avancer", et que si elle était éteinte, cela voulait dire que sa dernière action était "s'arrêter". C'est une sorte de code secret (mais qui n'est pas secret), et on aurait pu en choisir un autre. Par exemple, on aurait pu choisir d'utiliser la case 4, et que "case éteinte" soit le code qui veut dire "ma dernière action était d'avancer", et "case allumée" soit le code qui veut dire "ma dernière action était de s'arrêter".

Cela aurait donné:

Règle 1: "SI tu détectes qu'on appuie sur ton bouton rond ET que la case 4 est allumée, ALORS avance tout droit ET éteint la case 4"

Règle 2: "SI tu détectes qu'on appuie sur ton bouton rond ET que la case 4 est éteinte, ALORS arrête les moteurs ET allume la case 4"



Essaie, le robot va faire la même chose !