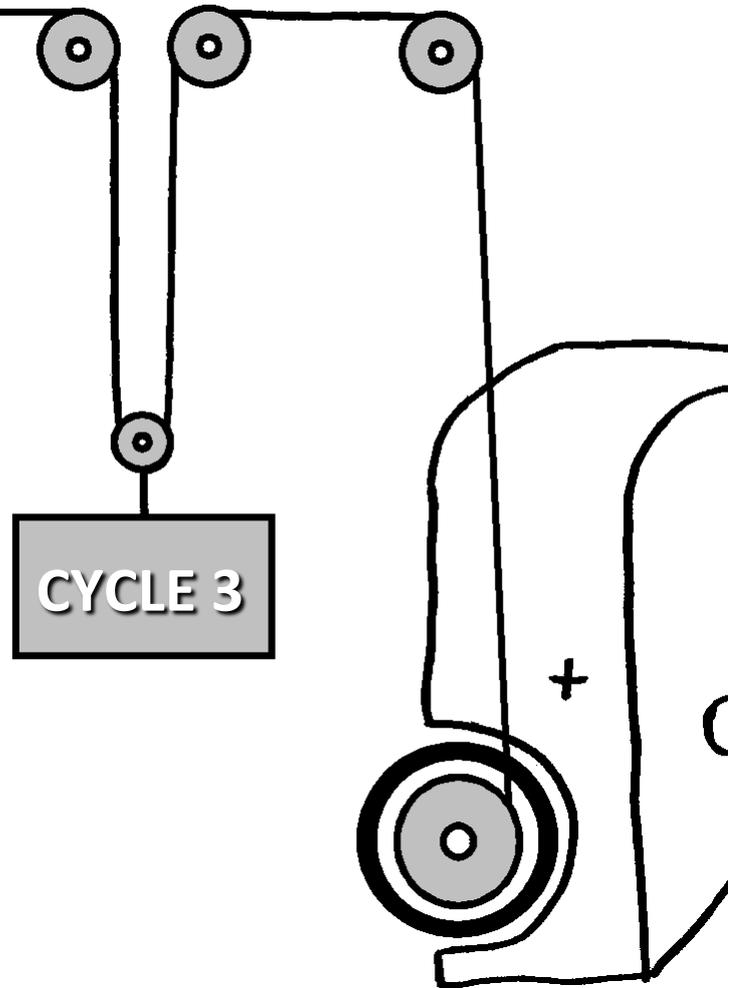


LA ROBOTIQUE PEDAGOGIQUE



- Les robots pédagogiques
- Découverte du robot Thymio
- Séquence d'apprentissage
- Fiche de lecture
- Fiches de recherche
- Défis programmation
- Complément pour les enseignants



Les robots pédagogiques

Qu'est-ce qu'un robot ?

Avant de s'interroger sur les robots pédagogiques, on peut se demander ce qu'est véritablement un robot. Nous connaissons le robot trancheur de légumes, le robot aspirateur, tondeur de pelouse, le robot soudeur des chaînes de fabrication de voitures. Ces différents appareils ne se ressemblent guère et il n'est pas facile de les regrouper sous une même définition !

A l'origine, robot vient du tchèque « robota » qui signifie tout simplement travail. Il fut utilisé pour la première fois dans une pièce de théâtre qui mettait en scène une fabrique d'hommes artificiels chargés de remplacer les humains pour les travaux les plus pénibles. Le terme fut ensuite démocratisé par le grand écrivain américain de science-fiction Isaac Asimov (cf. extrait de *Sally* en annexe 3).

La définition la plus générale d'un robot pourrait être la suivante : « *objet accomplissant une tâche de façon autonome* »

La robotique pédagogique

Un robot pédagogique est tout simplement un robot utilisé dans un but éducatif. Il puise sa légitimité pédagogique dans les théories constructivistes de Jean Piaget et Seymour Papert : « *L'enfant acquiert connaissances et compétences en agissant spontanément sur le réel et sur son environnement, d'autant plus facilement qu'il construit pour faire voir aux autres.* »

C'est en application de ces principes que S. Papert développe le langage LOGO. Mais la technologie disponible à son époque ne permet pas à LOGO d'agir véritablement sur le monde réel. La tortue LOGO exécute des déplacements programmés, trace des dessins, mais est incapable de renvoyer des informations sur son environnement, au contraire des robots actuels. C'est ce dernier élément qui devra vous guider dans le choix d'un robot pédagogique. Il existe en effet principalement deux types de robots pédagogiques :

- Les robots « actifs » qui se contentent de rouler en suivant des instructions programmées de façon linéaire. Ils sont destinés aux cycles 1 & 2 pour une approche simplifiée.
- Les robots « interactifs » qui disposent de capteurs réagissant à l'environnement. Ceux-ci permettent la réalisation de machines autonomes (Thymio est compatible avec le système « LEGO Technic ») ainsi qu'une programmation plus riche. Ils sont destinés aux cycles 3 et 4.

Quel robot pédagogique choisir ?

L'apprentissage du code informatique fait maintenant partie des programmes de l'école primaire. Les éditeurs ont développé des outils et des langages informatiques qui permettent de s'initier à la programmation sur son ordinateur ou sur une tablette. Ces outils se limitent principalement à la programmation de déplacements sur un quadrillage virtuel représenté à l'écran. Où sont donc passés le réel et l'action sur l'environnement chers à Piaget et Papert ?

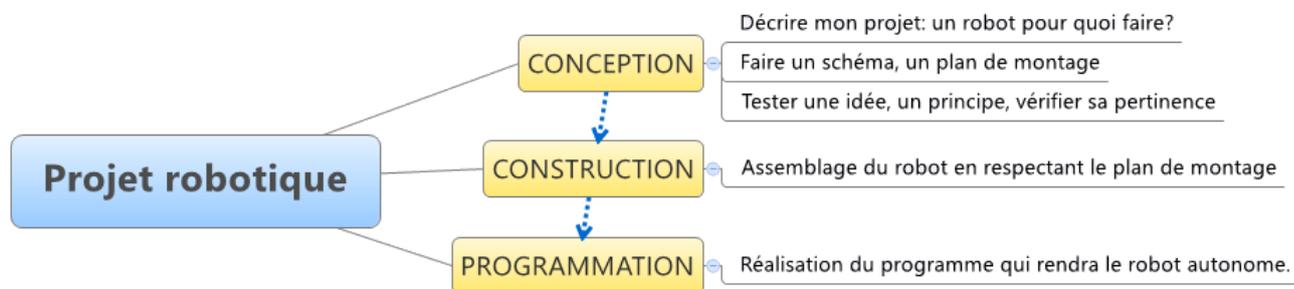
La robotique pédagogique permet de donner du sens à cet apprentissage en agissant sur de vrais objets, dans de vraies situations. Les élèves ne sont plus uniquement confrontés à un exercice intellectuel, mais à un vrai problème dont la programmation informatique n'est qu'une partie de la solution.

Le robot pédagogique que vous choisirez devra donc être aussi universel que possible. Il ne devra pas se limiter aux déplacements habituels mais permettre la construction d'objets et de machines autonomes. Il devra aussi fournir des informations sur son environnement à l'aide de capteurs intégrés.

La robotique dans les programmes du cycle 3

Décrire le fonctionnement d'objets techniques, leurs fonctions et leurs constitutions	
<ul style="list-style-type: none">» Besoin, fonction d'usage et d'estime.» Fonction technique, solutions techniques.» Représentation du fonctionnement d'un objet technique.» Comparaison de solutions techniques : constitutions, fonctions, organes.	Les élèves décrivent un objet dans son contexte. Ils sont amenés à identifier des fonctions assurées par un objet technique puis à décrire graphiquement à l'aide de croquis à main levée ou de schémas, le fonctionnement observé des éléments constituant une fonction technique. Les pièces, les constituants, les sous-ensembles sont inventoriés par les élèves. Les différentes parties sont isolées par observation en fonctionnement. Leur rôle respectif est mis en évidence.
Repérer et comprendre la communication et la gestion de l'information	
<ul style="list-style-type: none">» Environnement numérique de travail.» Le stockage des données, notions d'algorithmes, les objets programmables.» Usage des moyens numériques dans un réseau.» Usage de logiciels usuels.	Les élèves apprennent à connaître l'organisation d'un environnement numérique. Ils décrivent un système technique par ses composants et leurs relations. Les élèves découvrent l'algorithme en utilisant des logiciels d'applications visuelles et ludiques. Ils exploitent les moyens informatiques en pratiquant le travail collaboratif. Les élèves maîtrisent le fonctionnement de logiciels usuels et s'approprient leur fonctionnement.

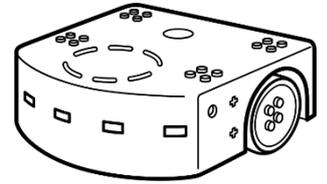
Projet type



Pour commencer

Avant de se lancer dans un projet complet, il convient de se familiariser avec le robot choisi et son mode de programmation. Pour cette découverte, j'ai choisi le robot Thymio qui répond aux critères vus plus haut. Il en existe d'autres bien sûr, notamment *MBot* fabriqué par MakeBlock.

Découverte du robot Thymio



1. Les modes préprogrammés

- Suiveur : Approchez la main d'un capteur, Thymio la suit.
- Explorateur : Thymio explore l'environnement en évitant les obstacles.
- Peureux : Approchez la main d'un capteur, Thymio la fuit.
- Attentif : Thymio réagit au son. Claquez dans vos mains et il change de direction.
- Pisteur : Tracez une piste noire de 4 cm de large, Thymio la suit.
- Obéissant : Pilotez Thymio à l'aide des 4 flèches.

2. Utilisation des modes préprogrammés

Les modes préprogrammés sont d'un intérêt limité pour l'apprentissage de la programmation. Mais ils sont très utiles pour mieux comprendre le fonctionnement du robot (cf. séance n° 2).

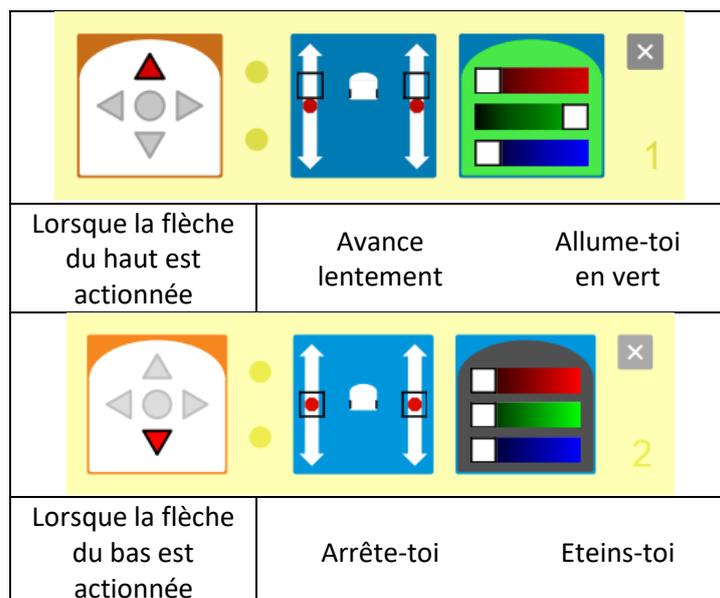
3. Pour programmer Thymio

- Au premier abord, la programmation du robot Thymio, bien que simple, naturelle et proche des comportements humains, peut paraître déroutante. Il s'agit en effet de programmation *événementielle*. Contrairement à une approche linéaire classique, la programmation événementielle consiste à prévoir les réactions au déclenchement des différents capteurs du robot Thymio. Si vous êtes un néophyte en programmation ou que vos connaissances se limitent à la tortue LOGO, lisez au préalable l'annexe 6 sur l'historique de la programmation.

- Pour programmer Thymio, le logiciel Aseba Studio propose 3 interfaces :

- ❖ *Une interface visuelle* (VPL = Visual Programming Language) composée de blocs logiques qu'il faut agencer en lignes de commandes. Cette interface est particulièrement bien adaptée à l'apprentissage de la programmation. Chaque ligne se décompose en 2 zones : à gauche l'évènement concerné, à droite les actions à réaliser lorsque cet évènement est déclenché. Cette interface comporte en outre 2 modes : *simplifié* (par défaut) et *complet* (en cliquant sur l'icône Professeur).

Le mode complet gère un plus grand nombre d'évènements. Exemple de programme réalisé en mode simplifié :



- ❖ Une interface de programmation par *blocs logiques*, appelée Blockly. Le programme ci-dessus s'écrit alors comme ceci :



- ❖ Une troisième interface *textuelle*, réservée aux programmeurs chevronnés :

```

onevent button.forward
  when button.forward == 1 do
    motor.left.target = 300
    motor.right.target = 300
    call leds.top(0,19,0)
  end

onevent button.backward
  when button.backward == 1 do
    motor.left.target = 0
    motor.right.target = 0
    call leds.top(0,0,0)
  end

```

4. Commencer à programmer Thymio

- Installez le logiciel « Aseba Studio pour Thymio »
Vous trouverez la dernière version ici : <https://www.thymio.org/fr:start>
- Connectez le *dongle* à l'ordinateur, puis démarrez le robot en appuyant quelques secondes sur le bouton central.
- Thymio et son dongle doivent clignoter de façon synchrone
- Lancez « Aseba Studio pour Thymio »
- Choisissez votre interface (en bas à gauche de l'écran)

5. Apprendre à programmer Thymio en classe

Voici une proposition de séquence-découverte qui sera détaillée dans les pages suivantes :

	Thème	Compétences travaillées
1	Qu'est-ce qu'un robot ? Comment fonctionne-t-il ?	Comprendre les notions d'algorithme et de programme. Débattre, argumenter.
2	Découverte du robot Thymio. Analyse des modes préprogrammés.	Observer et agir sur le réel, manipuler, expérimenter. Décrire un objet technique. Compléter un plan, une représentation symbolique. Travailler en collaboration.
3	Programmer Thymio (niveau 1).	Construire un programme informatique simple. Expérimenter.
4	Programmer Thymio (niveau 2).	Travailler en collaboration.

Une fois les bases acquises, les élèves pourront se lancer dans la construction de machines autonomes plus complexes.

Les robots

1. Qu'est-ce qu'un robot ? Où en trouve-t-on ? A quoi servent-ils ?

- L'objectif de ce débat est de se mettre d'accord sur une définition du robot :
« *Machine autonome qui exécute une tâche programmée* »
- Il pourra être précédé par la lecture – individuelle ou collective – d'un extrait de la nouvelle de science-fiction d'Isaac Asimov : « Sally » (Annexe 3). En 1953, l'auteur imagine les voitures du futur, autonomes et disposant de leur propre « personnalité » que nous appelons aujourd'hui Intelligence Artificielle.
 - A quel moment du récit comprend-on que Sally n'est pas humaine ?
 - Une telle voiture existe-t-elle actuellement ?
 - Les robots actuels ne sont pas aussi « intelligents ». Connaissez-vous des exemples de robots ou de machines exécutant automatiquement un travail ou une tâche ?
- A partir des propositions des élèves, on pourra mettre en place une classification, par exemple en complétant collectivement un tableau comme celui-ci :

Famille	Exemples	Fonction
Robots industriels	Assembleur de voitures	Remplacer l'homme pour les travaux pénibles ou répétitifs
Robots domestiques	Robot ménager Robot aspirateur Robot tondeuse	Aider l'homme pour les tâches quotidiennes et fastidieuses.
Robots explorateurs	Robots sous-marins Sondes spatiales Exploration planétaire	Aller où l'homme ne peut pas aller.
Robots jouets	Thymio	S'amuser, apprendre à programmer un robot
Robots humanoïdes	Robots compagnons	Imiter l'homme
Robots médicaux	Robots chirurgicaux	Aider les médecins à soigner et à opérer les malades
Robots de surveillance et de communication	Satellites	Aider les hommes à communiquer partout dans le monde. Etudier et surveiller la Terre. Observer le ciel et les étoiles.

2. Comment fonctionnent-ils ?

- Cette partie du débat devra faire émerger les quatre principaux constituants de base de tout automate ou robot :
 - Une source d'énergie (électricité, carburant, ...). Celle de Thymio est une batterie rechargeable comme celles des téléphones portables.
 - Des capteurs (boutons, sondes, caméras, micros, ...) que l'on peut comparer aux différents sens du corps humain. Ils permettent à Thymio de s'informer sur son environnement.
 - Des actionneurs (moteurs, ampoules, générateurs de sons, écrans, ...). Ils permettent à Thymio d'agir et d'effectuer certaines actions.
 - Un programme de fonctionnement. Celui de Thymio est stocké dans une mémoire numérique du même type que celle qui équipe les ordinateurs.

Quelques exemples de robots et d'automates



Le robot Thymio

1. Les élèves sont répartis en groupes. Chaque groupe dispose d'un robot Thymio.
2. Observation collective visuelle du robot et identification des différents éléments. Il s'agit de retrouver les éléments de base d'un robot découverts lors de la première séance : capteurs, actionneurs.
 - La fiche proposée en [Annexe 1](#) permet de localiser les différents éléments.
3. Mise en route et explication collective du choix des différents modes.
4. Les élèves testent librement les différents modes, en groupes. Pour le mode « pisteur », vous pouvez utiliser la piste de l'annexe 5 en l'agrandissant au format A3.
5. Il s'agit maintenant d'explicitier le fonctionnement de 4 modes, en complétant la fiche de l'annexe 2. Pour chacun des modes, les groupes vont devoir décrire le comportement du robot à l'aide de phrases logiques du type SI ... ALORS ([fiche annexe 2](#)). Cette activité est une bonne introduction à la programmation logique du robot Thymio. Elle permettra :
 - D'apprendre à distinguer un évènement d'une action.
 - De les associer par un enchaînement logique : SI ... ALORS ...
 - D'associer les évènements aux capteurs physiques du robot.

Programmer Thymio

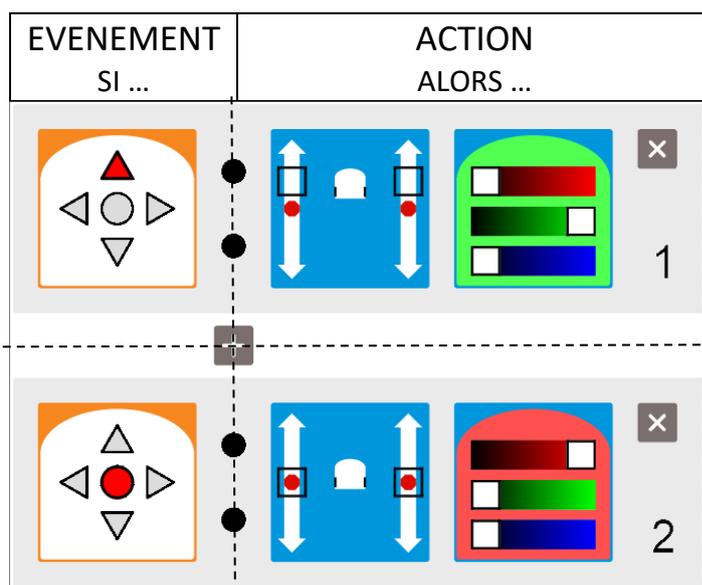
L'analyse des modes préprogrammés a permis d'aborder la programmation du robot Thymio sous la forme d'instructions « SI ... ALORS ... »

Les élèves vont maintenant découvrir l'environnement de programmation grâce aux défis Thymio. Imprimez recto-verso les pages de l'annexe 4, puis découpez et plastifiez les cartes obtenues.

Chaque carte propose un défi avec :

- Au recto : le programme à réaliser
- Au verso : une aide pour la réalisation

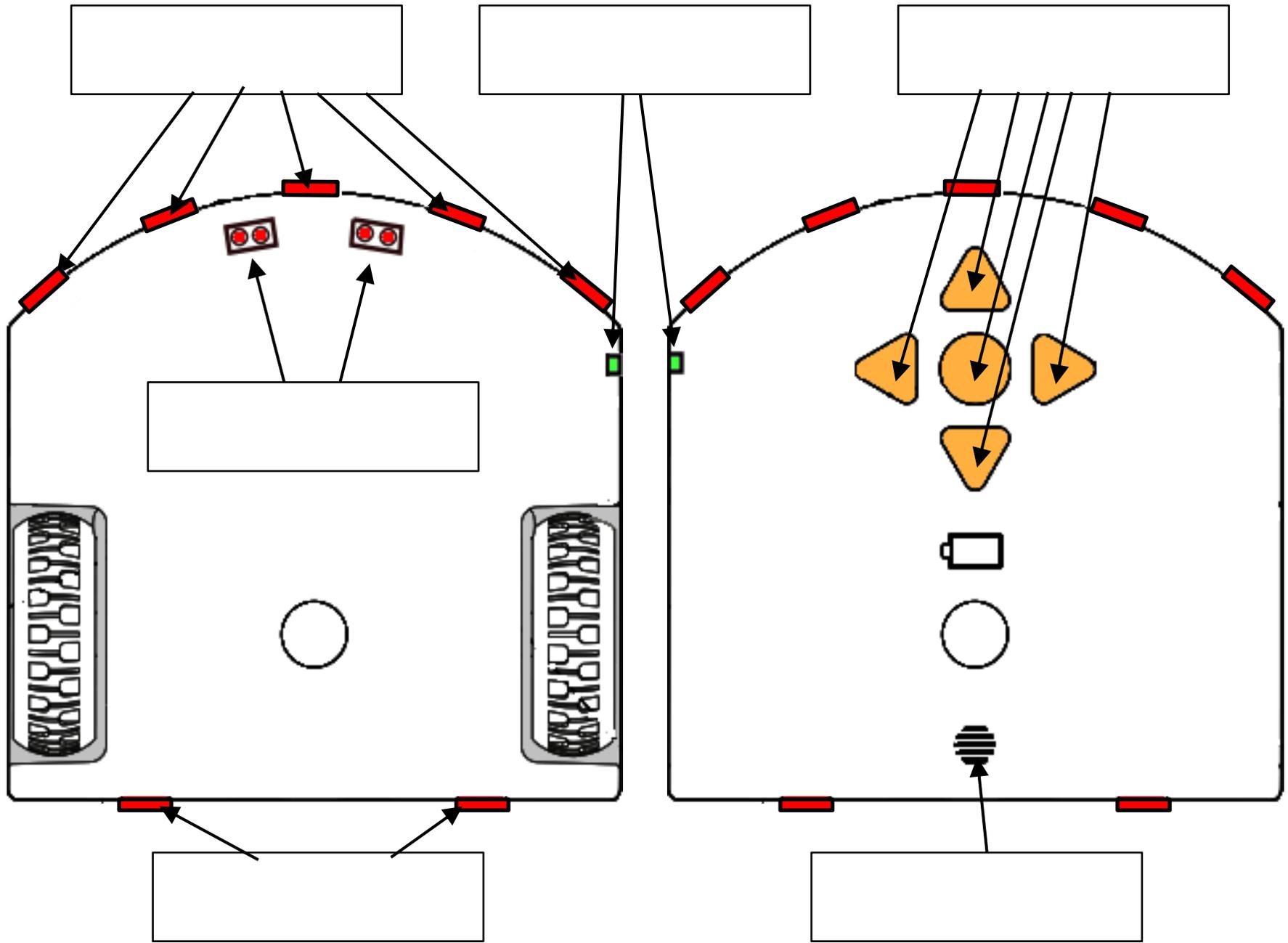
1. Commencez par une présentation en réalisant collectivement un ou deux défis de niveau 1.
2. Distribuez les cartes restantes aux différents groupes : chaque groupe tente de réaliser un défi puis présente son programme à la classe.



Avant d'aborder les défis du niveau 2, il sera nécessaire de revenir à une présentation collective des instructions du niveau Avancé (chronomètre et accéléromètre) et d'en expliquer le fonctionnement.

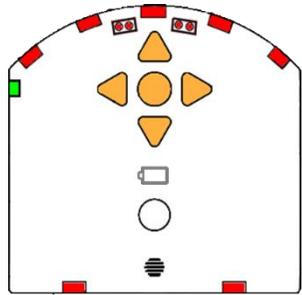
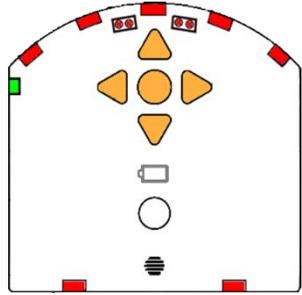
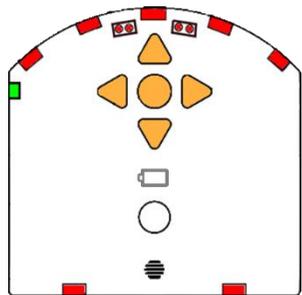
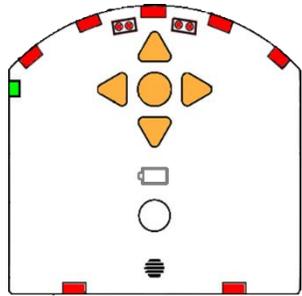
Si vous ne disposez que d'un seul robot, il faudra vous organiser en faisant tourner plusieurs ateliers. Exemple :

Groupe 1	Programmation du robot Thymio.
Groupe 2	Programmation de déplacements avec TuxBot sur ordinateur ou tablette.
Groupe 3	Atelier technologique : plan et/ou construction d'un robot



DESSOUS

DESSUS

MODE	SI  ALORS	CAPTEURS UTILISÉS
<p>VERT</p>  <p>AMICAL</p>	<p>S'il détecte un objet devant lui <input type="radio"/></p> <p>S'il détecte un objet à droite <input type="radio"/></p> <p>S'il arrive au bord d'une table <input type="radio"/></p>	
<p>ROUGE</p>  <p>PEUREUX</p>	<p>S'il détecte un objet devant lui <input type="radio"/></p> <p>S'il détecte un objet à droite <input type="radio"/></p> <p>Si on tapote son dos <input type="radio"/></p> <p>S'il détecte un objet derrière lui <input type="radio"/></p>	
<p>JAUNE</p>  <p>EXPLORATEUR</p>	<p>S'il détecte un objet devant lui <input type="radio"/></p> <p>S'il détecte un objet à droite <input type="radio"/></p> <p>S'il détecte un objet à gauche <input type="radio"/></p> <p>S'il détecte un objet derrière lui <input type="radio"/></p> <p>S'il arrive au bord d'une table <input type="radio"/></p>	
<p>VIOLET</p>  <p>OBÉISSANT</p>	<p>Si on appuie sur la flèche avant <input type="radio"/></p> <p>Si on appuie sur la flèche arrière <input type="radio"/></p> <p>Si on appuie sur la flèche droite <input type="radio"/></p> <p>Si on appuie sur la flèche gauche <input type="radio"/></p>	

SALLY

Isaac Asimov

Sally arrivait par la route du lac, alors je lui fis signe et l'appelai par son nom. J'aimais toujours voir Sally. Je les aimais toutes, comprenez-moi bien, mais Sally était la plus jolie. Cela ne faisait aucun doute. Elle accéléra un peu quand j'agitai la main, pour montrer qu'elle était heureuse de me voir aussi. Je me tournai vers l'homme debout à côté de moi.

— Voilà Sally, M. Gelhorn, lui dis-je.

Il me sourit, hocha la tête puis demanda :

— Vous pourriez me parler un peu de votre Ferme que voici, si ça ne vous dérange pas trop ?

J'entendis Sally arriver derrière moi et je tendis la main. Elle se glissa dessous et je sentis l'email dur et lisse de son aile, tout tiède.

— Une belle automobile, estima Gelhorn.

C'était une façon de parler. Sally était une décapotable de 2045 avec un moteur électronique. Elle avait la ligne la plus belle, la plus élancée que j'aie jamais vue. Depuis cinq ans, elle était ma préférée et je lui avais consacré et ajouté tout ce dont je pouvais rêver. Pendant tout ce temps, il n'y avait jamais eu un être humain derrière son volant. Pas une seule fois.

— Sally, dis-je en la caressant tendrement, je te présente M. Gelhorn.

Le ronronnement des cylindres de Sally monta d'un ton. Je tendis l'oreille avec attention, pour guetter le moindre bruit insolite.

— Vous donnez des noms à toutes vos voitures ? demanda Gelhorn.

— Certainement. Les voitures ont chacune leur personnalité, n'est-ce pas ? Les coupés sont tous masculins, et les décapotables féminines.

Gelhorn souriait de nouveau.

— Et maintenant, M. Folkers, est-ce que je pourrais vous parler en particulier ?

— Bien sûr. Cela ne vous ennuie pas que Sally nous accompagne, n'est-ce pas ? demandai-je.

— Pas du tout. Elle ne peut pas répéter ce que nous dirons, hein ?

Il rit de sa propre plaisanterie et caressa la calandre de Sally. Elle emballa son moteur et Gelhorn retira vivement sa main.

— Elle n'est pas habituée aux étrangers, expliquai-je.

Nous nous assîmes sur le banc, sous le grand chêne. C'était l'heure chaude de la journée et les voitures étaient sorties en force, au moins trente d'entre elles. De loin je pus voir que Jeremy jouait à son petit jeu habituel. Il arrivait sans bruit derrière une autre automobile puis accélérât d'un coup et doublait à toute vitesse en faisant hurler ses freins. Deux semaines plus tôt, il avait provoqué un accident et j'avais dû sévir en coupant son moteur pendant deux jours.

— Eh bien, M. Gelhorn, fis-je, si vous me disiez ce qui vous amène ?

— D'accord, Jake. Combien de voitures avez-vous ici ?

— Cinquante et une. Elles sont toutes en parfait état de marche et équipées d'un cerveau électronique qui réagit beaucoup plus vite qu'un cerveau humain. On monte, on tape sa destination et on laisse la voiture prendre le chemin qu'elle veut.

Quel chemin accompli, quand on pense aux premières voitures autonomes qui avaient encore un volant au cas où quelque-chose aurait été de travers !

— Cinquante et une voitures ! s'exclama Gelhorn. Il vous a fallu beaucoup de temps pour monter cette ferme ?

— C'est sûr, M. Gelhorn. Trente-trois ans. Tenez ! Regardez donc Sally.

Sally était si propre ! Un insecte avait dû mourir sur son pare-brise ou un grain de poussière était tombé, alors elle se mettait au travail. Un petit tube était sorti et aspergeait la vitre. Le produit s'étalait sur la surface et, aussitôt, de petits balais-éponges se mettaient en place pour chasser l'eau dans la petite

rainure qui la faisait couler par terre. Pas une goutte n'éclaboussa son capot vert pomme. Balais et tube rentrèrent et disparurent.

- Je n'ai jamais vu une automatique faire ça ! s'écria Gelhorn.
- Non, sans doute. C'est un système que j'ai installé spécialement sur nos voitures. Elles sont propres. Elles n'arrêtent pas de polir leurs vitres. Elles aiment ça. Les décapotables sont très coquettes.
- Je peux vous dire comment trouver de l'argent, si ça vous intéresse.
- Ça m'intéresse toujours. Comment ?
- N'est-ce pas évident, Jake ? N'importe laquelle de vos voitures vaut cinquante mille au minimum, vous avez dit.
- Et alors ?
- Vous n'avez jamais pensé à en vendre quelques-unes ? Laissez-moi vous expliquer la situation. Je sais où trouver un stock de carrosseries. Je sais aussi où je peux vendre des automatiques à un bon prix. Tout ce qu'il me faut, c'est des moteurs. Vous voyez la solution ?
- Non, M. Gelhorn.
- Vous êtes un mécanicien expert en automotobiles, Jake. Vous pourriez démonter un moteur et le placer sur une autre voiture et personne ne remarquerait la différence.
- Ce ne serait pas très moral.
- Vous ne feriez pas de mal aux voitures.
- Allons, allons, M. Gelhorn, un moment. Les moteurs et les carrosseries ne sont pas deux choses séparées. C'est une unité. Ces moteurs sont habitués à leur propre carrosserie. Ils ne seraient pas heureux sur une autre voiture.
- Bon, d'accord, je veux bien. Vous avez parfaitement raison, Jake. Ce serait comme si je prenais votre cerveau pour le mettre dans le crâne de quelqu'un d'autre. Oui ? Vous pensez que vous n'aimeriez pas ça ?
- Je ne crois pas que ça me plairait, en effet.
- Mais si je prenais votre cerveau pour le mettre dans le corps d'un jeune athlète ? Hein, Jake ? Vous n'êtes plus un jeunot. Si vous aviez le choix, est-ce que vous n'aimeriez pas avoir de nouveau vingt ans ? C'est ce que j'offre à certains de vos moteurs électroniques. Ils seront placés dans des carrosseries neuves. Les tout derniers modèles.

J'éclatai de rire.

- Ça ne tient guère debout, M. Gelhorn. Certaines de nos voitures sont vieilles, peut-être, mais elles sont bien soignées. Personne ne les conduit. Elles ont le droit de faire ce qu'elles veulent. Elles sont à la *retraite*, M. Gelhorn. Je ne voudrais pas d'un corps de vingt ans si, pour cela, je devais travailler dur pendant tout le restant de ma nouvelle vie, sans jamais avoir assez à manger... Qu'est-ce tu en penses, Sally ?

Les deux portières de Sally s'ouvrirent et se refermèrent avec un claquement étouffé.

- Qu'est-ce que c'est que ça ? s'exclama Gelhorn.
 - C'est le rire de Sally.
- Il se força à sourire. Il pensait sûrement que je plaisantais, que c'était une mauvaise blague. Il insista :
- Soyez raisonnable, Jake. Les voitures sont faites pour être conduites. Elles ne sont probablement pas heureuses si on ne les conduit pas.
 - Sally n'a pas été conduite depuis cinq ans. Elle m'a l'air assez heureuse.
 - Je me le demande !
- Il se leva et marcha lentement vers Sally.
- Alors, Sally, qu'est-ce que tu dirais de faire un petit tour ?
- Le moteur de Sally s'emballa. Elle recula.
- Ne la bousculez pas, M. Gelhorn, conseillai-je. Elle est assez nerveuse.
 - Du calme, doucement, Sally, dit Gelhorn.
- Il bondit et saisit la poignée de la portière. Elle ne bougea pas, naturellement.
- Ça s'est ouvert il y a une minute ! cria-t-il.
 - Verrouillage automatique, dis-je. Elle tient beaucoup à préserver son intimité.

Il lâcha la porte et recula de trois ou quatre pas, puis, rapidement, si vite que je ne pus l'arrêter, il courut et sauta dans la voiture. Il prit Sally par surprise parce que, en tombant assis, il coupa le contact avant qu'elle puisse le verrouiller.

Pour la première fois depuis cinq ans, le moteur de Sally s'arrêta.

Je crois que je poussai un cri, mais Gelhorn avait tourné la manette sur « Manuel » et l'avait verrouillée ainsi. Il mit le moteur en marche. Sally se ranimait mais elle n'avait plus aucune liberté d'action. Il démarra...

Défis-programmation pour Thymio (Niveau 1)

1

- Lorsqu'on appuie sur la flèche du haut, Thymio tourne en rond sans s'arrêter.

2

- Lorsqu'on appuie sur la flèche du haut, Thymio s'allume en vert.
- Lorsqu'on appuie sur la flèche du bas, Thymio s'allume en bleu

3

- Lorsqu'on approche la main à l'avant-centre de Thymio, il joue un son.

4

- Lorsque Thymio détecte un objet devant lui à droite, il s'allume en rouge.
- Lorsqu'il détecte un objet derrière lui à gauche, il s'allume en jaune.

5

- Lorsqu'un objet s'approche à l'arrière-droite, Thymio joue un son.

6

- Lorsqu'on frappe dans les mains, Thymio s'allume en vert.
- Lorsqu'on appuie sur le bouton central, il s'éteint.

7

- Lorsque Thymio détecte un objet à l'arrière, il avance.
- Lorsque Thymio détecte un objet à l'avant-centre, il s'arrête.

8

- Lorsqu'un objet s'approche à l'arrière-gauche, Thymio joue un son et s'allume en vert.
- Lorsqu'on appuie sur le bouton central, il s'éteint.

9

- Lorsque l'on tapote Thymio, il émet une musique.

10

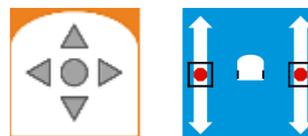
- Lorsque l'on tapote Thymio, il s'allume en blanc et joue une musique.
- Lorsqu'on appuie sur le bouton central, il s'éteint.

Défis-programmation pour Thymio (Niveau 1)

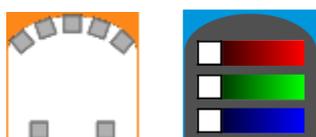
2 lignes, en utilisant :



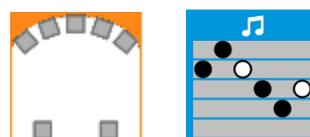
1 ligne, en utilisant :



2 lignes, en utilisant :



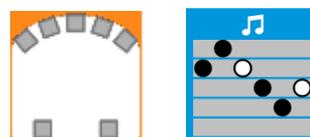
1 ligne en utilisant :



2 lignes en utilisant :



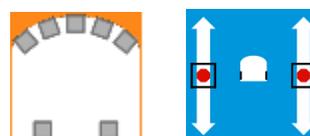
1 ligne en utilisant :



2 lignes en utilisant :



2 lignes en utilisant :



2 ligne en utilisant :



1 ligne en utilisant :



Défis-programmation pour Thymio (niveau 2)

1

- Lorsqu'on appuie sur la flèche du haut, Thymio tourne en rond sans s'arrêter.
- Pour l'arrêter, il faut appuyer sur le bouton central.

2



- Lorsqu'on appuie sur le bouton central, Thymio s'allume en vert pendant 3 secondes, puis s'éteint.

3



- Lorsqu'on appuie sur la flèche du haut, Thymio clignote en rouge et jaune.
- Lorsqu'on appuie sur le bouton central, il s'éteint.

4

- Lorsque Thymio détecte un objet droit devant **ET** un objet derrière lui, il s'allume en rouge et émet une alarme..
- Lorsque les objets disparaissent, il s'éteint.

5



- Lorsqu'un objet s'approche à l'arrière-droite, Thymio s'allume en jaune pendant 3 secondes puis s'éteint.
- Lorsqu'un objet s'approche à l'arrière-gauche, Thymio s'allume en violet pendant 3 secondes puis s'éteint.

6



- Lorsqu'on frappe dans les mains, Thymio s'allume en vert.
- Lorsqu'on frappe à nouveau, il s'éteint.

7

- Lorsque Thymio détecte un objet à l'avant-droite, il *pivote* à gauche (sans avancer).
- Lorsque Thymio détecte un objet à l'avant-gauche, il *pivote* à droite (sans avancer).
- Lorsqu'il ne détecte rien, il s'arrête.

8

- Chacun des 5 boutons (flèches et rond central) fait entendre une note différente.

9



- Lorsque Thymio penche à gauche, il s'allume en jaune.
- Lorsqu'il penche à droite, il s'allume en bleu.
- Lorsqu'il est à l'horizontal il reste éteint.

10



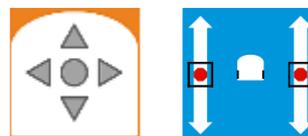
- Lorsqu'on appuie sur la flèche du haut, Thymio avance pendant 1 seconde, puis s'arrête.
- Lorsqu'on appuie sur la flèche du bas, il recule pendant 1 seconde, puis s'arrête.

Défis-programmation pour Thymio (niveau 2)

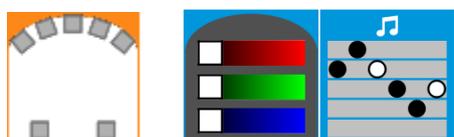
2 lignes, en utilisant :



2 lignes, en utilisant :



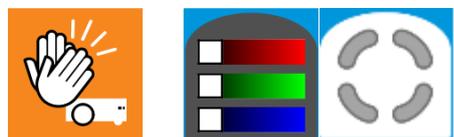
2 lignes, en utilisant :



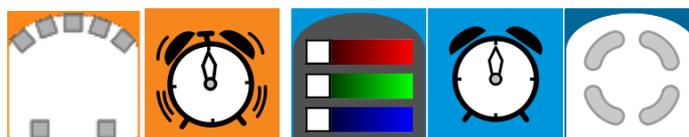
4 lignes, en utilisant :



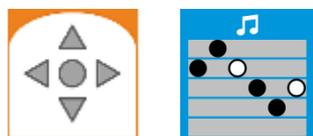
2 lignes en utilisant :



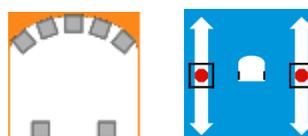
3 lignes en utilisant :



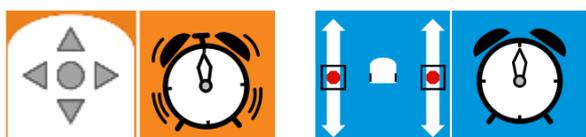
5 lignes en utilisant :



3 lignes en utilisant :



3 lignes en utilisant :

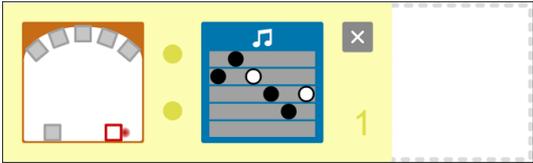
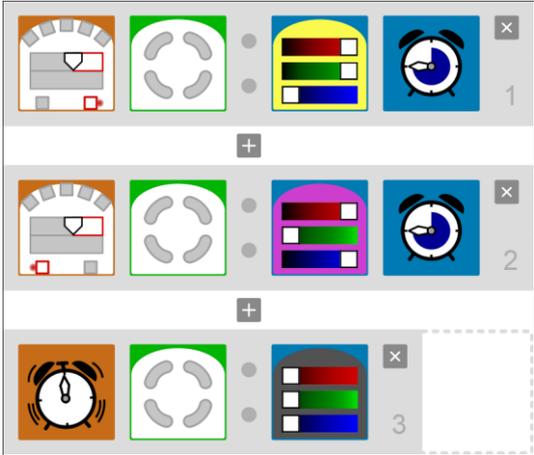
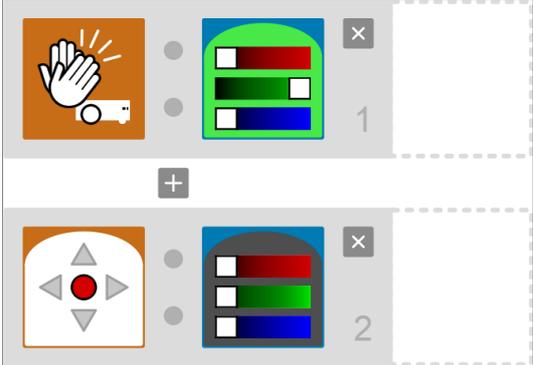
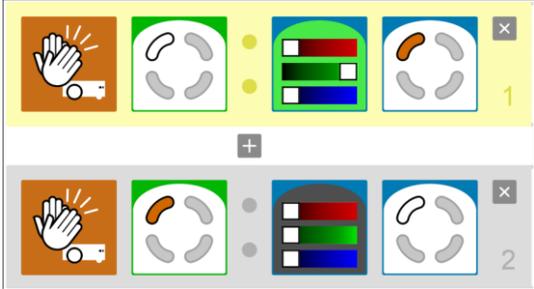
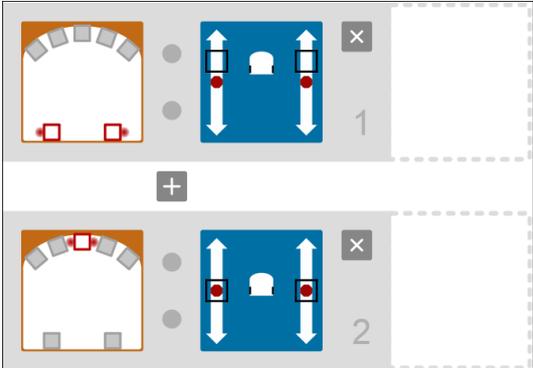
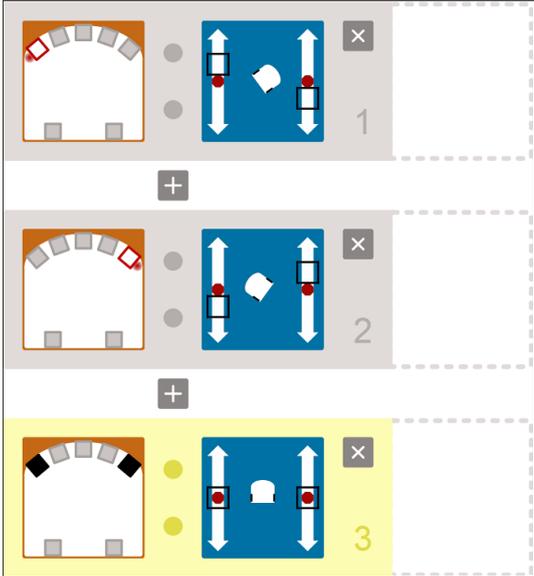


3 lignes en utilisant :



Solutions des défis

	Niveau 1	Niveau 2
1		
2	 	
3		
4	 	

<p>5</p>		
<p>6</p>	<p>Défi n° 6</p> 	
<p>7</p>	<p>Défi n° 7</p> 	

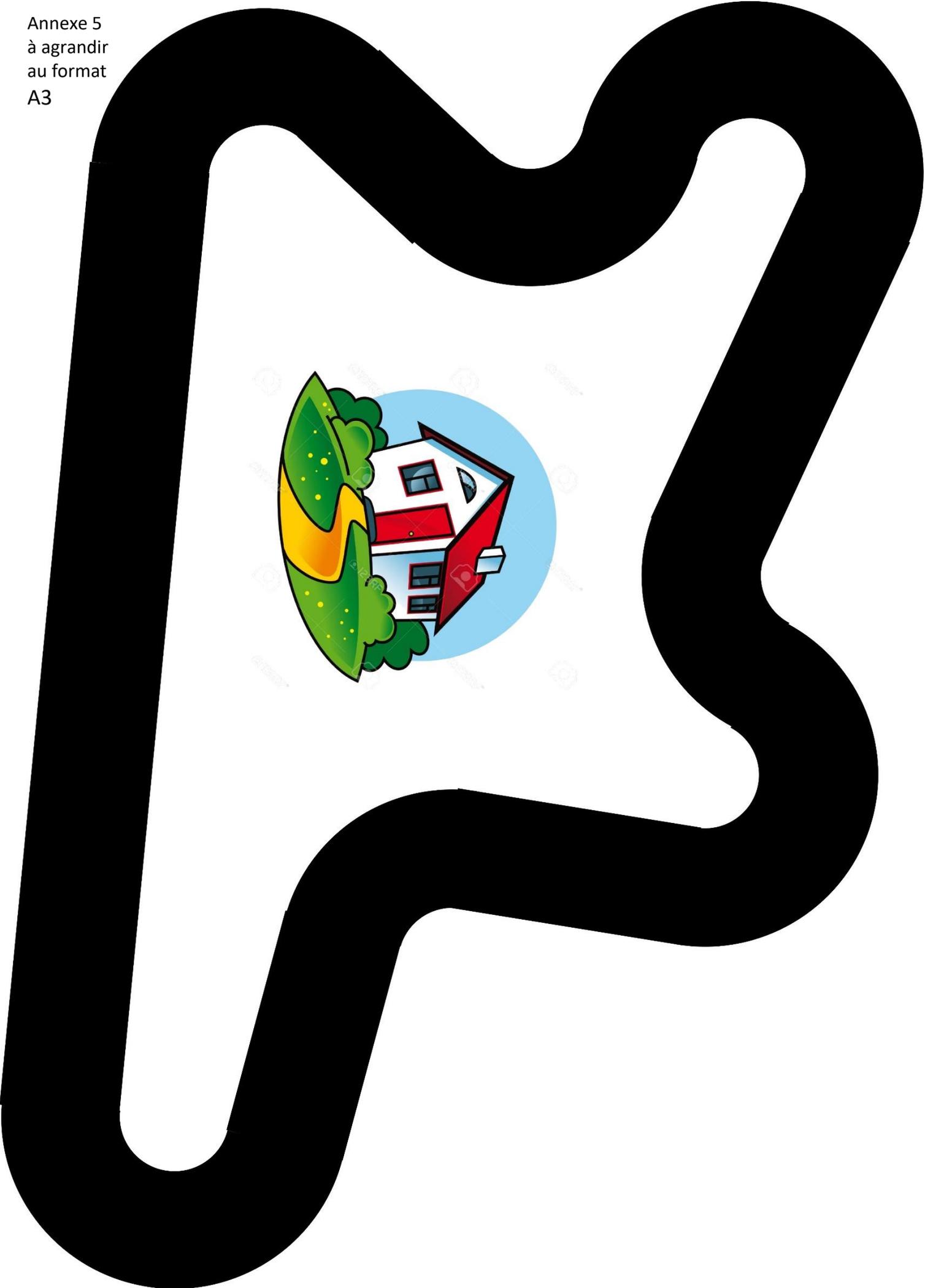
8

Défi n° 8

9

10

Annexe 5
à agrandir
au format
A3



Robots et programmation - Historique



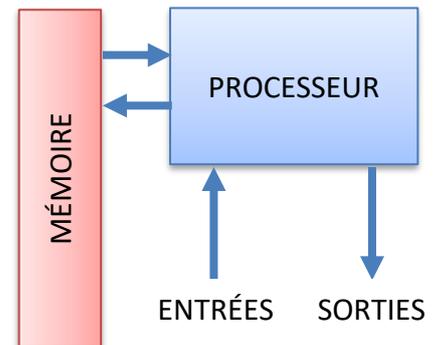
Comparé aux premiers ordinateurs, le petit robot Thymio est un véritable génie. Truffé de capteurs qui lui permettent de réagir à son environnement (capteurs infrarouges, boutons de commande, microphone, ...), il est capable d'exécuter des programmes écrits dans un langage clair et compréhensible par tous.

Pour comprendre les bases du fonctionnement de Thymio et mieux appréhender la notion de programme informatique, un petit retour en arrière s'impose.

Historique

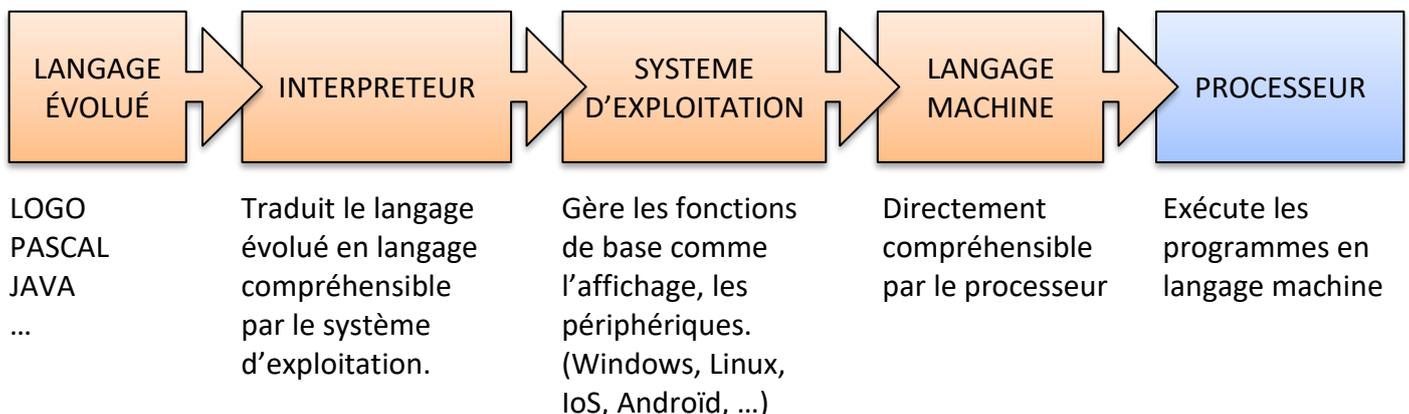
C'est à **John Von Neumann**¹ que l'on doit les principes généraux du fonctionnement d'un ordinateur : processeur pour les calculs, mémoire pour stocker programme et données, entrées et sorties pour la communication avec l'extérieur.

En se basant sur les travaux de Von Neumann, **Alan Turing**² participe au développement du premier véritable ordinateur, Mark 1, et développe la programmation. Il est aujourd'hui considéré comme l'inventeur de la programmation informatique. Il est aussi connu pour avoir proposé le « test de Turing ». Ce test consiste à mettre une personne en confrontation verbale à l'aveugle avec un ordinateur et un autre humain. Si la personne s'avère incapable de différencier l'ordinateur de l'humain, alors le logiciel utilisé par l'ordinateur peut prétendre à la dénomination d'intelligence artificielle (IA). Ça fait peur, mais ce n'est pas encore pour demain.



Programmation séquentielle

Un programme informatique, tel que défini par Turing, est une suite d'instructions compréhensibles par le processeur. Ces instructions sont exécutées séquentiellement. Elles sont écrites dans un langage purement mathématique appelé « langage machine » ou bien « langage assembleur ». Ce langage est constitué d'instructions travaillant uniquement sur des nombres contenus dans la mémoire, ou dans les registres du processeur : calculs arithmétiques, test logiques, déplacements des contenus de registres, etc. C'est un langage très éloigné du langage humain, qui ne comprend que les nombres, ce qui le rend très difficile à utiliser. C'est un peu comme si l'on devait décrire tous les calculs effectués par notre cerveau chaque fois qu'il déchiffre un mot par exemple. Pour compliquer encore la tâche, chaque famille de processeurs dispose de son propre langage assembleur. Pour faciliter la programmation des ordinateurs, on a donc créé des programmes intermédiaires chargés d'interpréter un langage « de programmation » en langage « machine ».



¹ John Von Neumann : Mathématicien et physicien américano-hongrois (1903 Budapest – 1957 Washington)

² Alan Turing : Mathématicien et cryptologue britannique (1912 Londres – 1954 Wilmslow)

On appelle ces programmes intermédiaires des « couches logicielles ».

Il existe de nombreux langages de programmation évolués : BASIC, LOGO, PASCAL, C, FORTRAN, JAVA, FLASH, JAVASCRIPT, SCRATCH, PHP, etc. Chaque langage a son propre vocabulaire et ses règles de syntaxe.

Ainsi, pour tracer un carré en LOGO, le programmeur écrira

```
REPETE 4 [ AVANCE 100 TOURNE-DROITE 90 ]
```

Qui sera, au bout de la chaîne traduit de façon automatique et transparente en quelque-chose comme :

```
movl $8, %edx
int $0x80
movl $1, %eax
```

Interruption momentanée des programmes

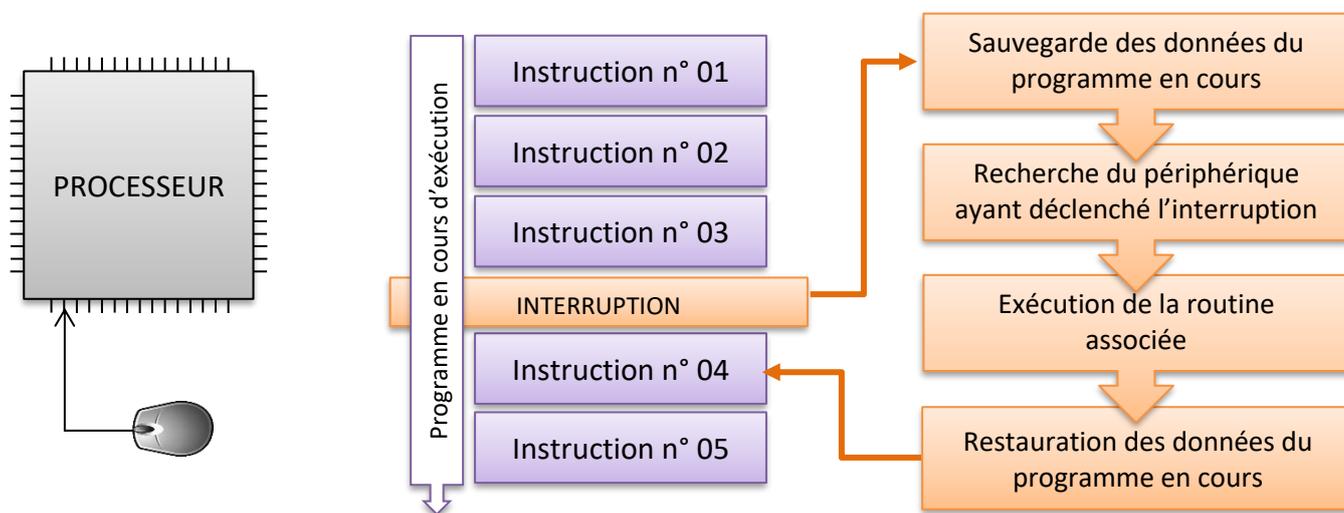
Un processeur ne peut exécuter qu'un seul programme à la fois³. Quand un programme est lancé, il faut attendre la fin de son exécution pour lancer un autre programme, ou bien l'interrompre.

- L'interrompre ? Oui mais comment ?
- Et bien en cliquant sur le bouton d'arrêt, tiens !
- Ah bon ? Mais comment le processeur sait-il que j'ai cliqué sur ce bouton ?
- Et bien grâce à un autre programme qui teste le bouton de la souris !
- Oui, mais les deux programmes ne peuvent pas fonctionner en même temps...
- Et bien alors, il faut que mon programme teste aussi la souris.

Et le clavier ? Et l'écran tactile ? Et les boutons de Thymio ? Et les capteurs infrarouges ? Et...

C'est naturellement impossible. Pour résoudre ce problème il faut avoir recours aux *interruptions*.

Le processeur comporte une broche spéciale, dénommée *ligne d'interruption*, qui permet d'interrompre temporairement le programme en cours d'exécution pour lancer une routine (petit programme) associée au périphérique qui a déclenché l'interruption.



Avec ce fonctionnement, le programmeur ne s'occupe que des *événements* qui l'intéressent et pour chacun d'eux, il devra écrire un programme à exécuter si l'évènement est *déclenché*.

La programmation événementielle

Elle consiste donc à programmer les réactions aux différents événements produits pas les capteurs ou les périphériques de notre robot. Si rien n'est prévu pour un événement, il passera inaperçu.

Voici un exemple de programme permettant au robot de se balader sur la table sans jamais tomber.

³ Il existe maintenant des processeurs « multicœurs ». Chaque cœur est en fait une unité arithmétique qui peut exécuter des calculs indépendamment des autres unités. Mais le processeur ne peut exécuter plusieurs programmes simultanément.

D'abord en programmation séquentielle, puis en utilisant les évènements :

Programme BALADE //version séquentielle

```

REPETE [
  AVANCE 1
  SI LE CAPTEUR DE DESSOUS EST DÉSACTIVÉ [
    RECOULE 2
    TOURNE DROITE 90
  ]
]
FIN
BALADE

```

On lance le programme

Après chaque pas, on teste le capteur pour savoir si l'on est au bord de la table. Il faut avancer lentement, pour ne pas risquer la chute entre 2 tests du capteur. Et si l'on veut utiliser plusieurs capteurs, le programme devient vite très lourd et le robot va encore ralentir.

Programme BALADE //version évènementielle

```

Programme principal
  REPETE [AVANCE 10000]
FIN
Fonction CAPTEUR-DESSOUS-DÉSACTIVÉ
  RECOULE 20
  TOURNE DROITE 90
FIN
BALADE

```

On lance le programme

Le programme principal se contente de faire avancer le robot indéfiniment. Il est très simple et va tourner en boucle.

Dès que le capteur sous le robot ne détecte plus rien, il génère un évènement et cette fonction est exécutée.

Avec la programmation évènementielle, il n'est plus besoin de tester en permanence l'état du capteur. *C'est le capteur lui-même qui déclenche l'exécution de la routine correspondante.* Cette routine ne sera donc exécutée que lorsqu'elle est utile. Le programme principal est allégé et il sera exécuté plus rapidement. Le robot peut foncer ! Enfin pas trop quand même...

La programmation orientée objet (POO)

La programmation évènementielle permet de se rapprocher un peu de la réalité. Comme dans la vraie vie, un évènement déclenche une action qu'il ne me reste plus qu'à définir et à programmer. Exemples :

Avec le robot	Dans la vraie vie
J'appuie sur un bouton => Le robot démarre.	On me tape sur l'épaule => Je me retourne.
Un capteur détecte une présence => Il s'arrête.	Un piéton va traverser => Je stoppe ma voiture.

La Programmation Objet va encore plus loin. Comme son nom l'indique elle permet de créer des objets (virtuels bien sûr) et de leur attribuer des propriétés, des actions (appelées méthodes), des évènements et de définir une réaction à chaque évènement. Une fois créé, un objet pourra être facilement décliné en autant d'exemplaires que nécessaire. Tous les exemplaires créés seront dits de la même *classe*.

Exemple : définissons la classe MAISON

Objet MAISON [

Propriétés : sol, murs, toit, porte, fenêtres ;

Méthodes : construire, démonter, repeindre, passer-aspirateur ;

Evènements : fuite => réparer-fuite, chaudière en panne => réparer-chaudière ;]

Toutes les méthodes et fonctions doivent bien sûr être détaillées. Il faudra écrire les programmes correspondant à construire, démonter, repeindre, passer-aspirateur, réparer-fuite, réparer-chaudière.

Nous pourrons ensuite décliner cette classe en plusieurs exemplaires :

```
Domicile : MAISON ; // Domicile est défini comme un objet de classe MAISON
```

```
Chez-Laurette : MAISON ; // Chez-Laurette est défini comme un objet de classe MAISON
```

Puis les construire :

```
Maison1.construire ;
```

```
Chez-Laurette.construire ;
```

Pour créer un chalet, inutile de tout redéfinir. Notre objet CHALET sera un *héritier* de l'objet MAISON. Nous lui ajouterons des propriétés et compléterons les méthodes si nécessaire :

Objet CHALET (MAISON) [

```
Propriétés : cheminée, terrasse ; // Nouvelles propriétés ajoutées
```

```
Fonction construire { // On surcharge la méthode construire pour ajouter la cheminée et la terrasse
```

```
    MAISON.construire ; // on appelle la méthode de l'ancêtre MAISON
```

```
    Ajouter cheminée ;
```

```
    Ajouter terrasse ;
```

```
}
```

```
]
```

Puis nous pourrons créer notre classe LOTISSEMENT :

Objet LOTISSEMENT [

```
Propriétés :
```

```
    Maison1 : MAISON
```

```
    Maison2 : MAISON
```

```
    Chalet1 : CHALET
```

```
    Chalet2 : CHALET
```

```
Fonction construire {
```

```
    Maison1.construire ;
```

```
    Maison2.construire ;
```

```
    Chalet1.construire ;
```

```
    Chalet2.construire ;
```

```
}
```

```
]
```

Et enfin créer autant de lotissements que nécessaire pour maintenir les effectifs dans nos écoles.

Il ne s'agit là que d'un survol de la programmation objet.

Si le sujet vous intéresse, vous trouverez sur Internet toute la documentation utile.

Notre robot est bien un objet

Pour finir, nous pouvons maintenant assimiler notre robot à un « objet informatique » :

- Il possède des propriétés : vitesse de rotation des moteurs, états des capteurs, des LEDS, ...
- Il possède des fonctions prédéfinies (méthodes) : allumage d'une LED, lancement d'un moteur, ... que nous pouvons utiliser dans notre programme.
- Il génère des événements : capteur activé ou désactivé, bouton enfoncé, ... que nous pouvons intercepter pour les associer à un comportement particulier.