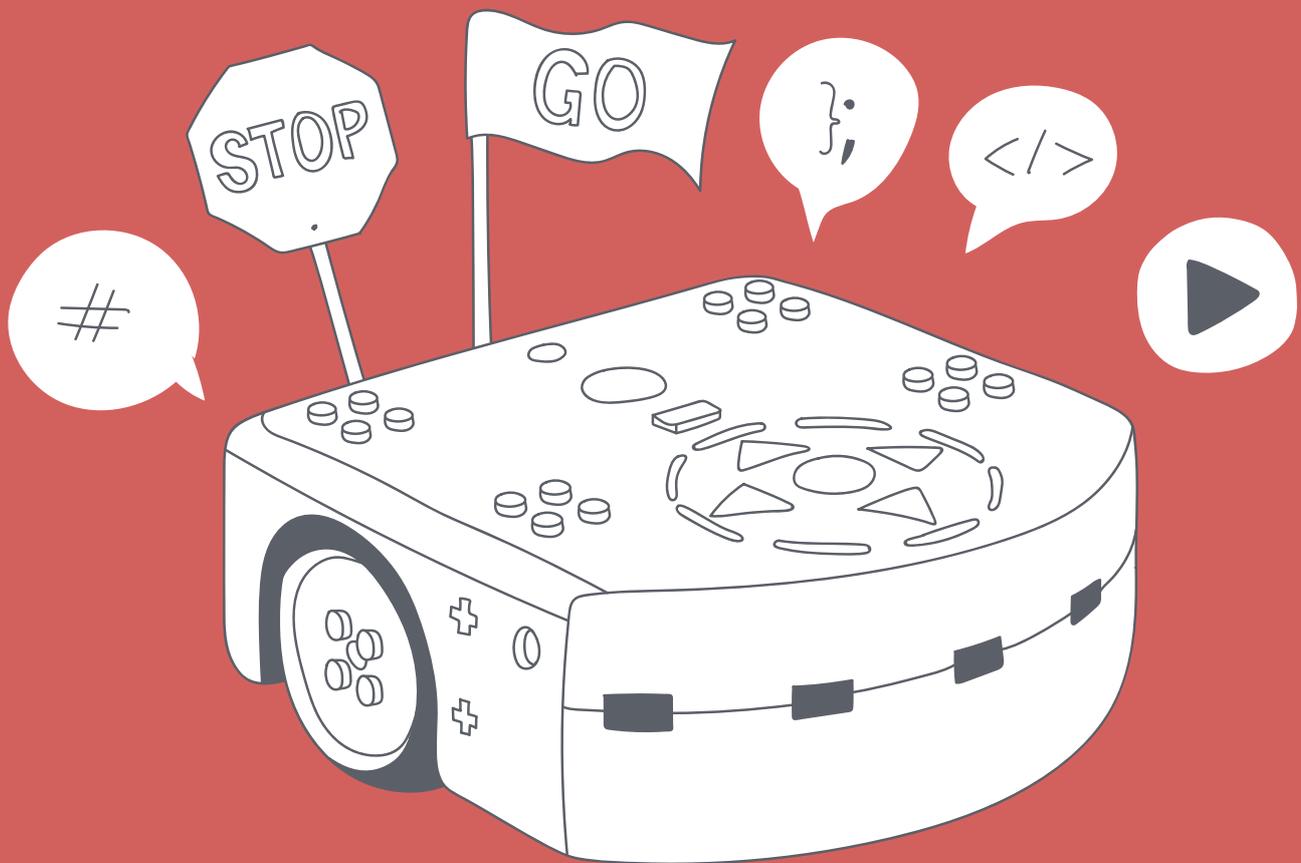


# PROGRAMMATION SÉQUENTIELLE ET ÉVÉNEMENTIELLE AVEC LE ROBOT THYMIO





## PLAN D'ÉTUDES ROMAND

### EN 22 — S'approprier les concepts de base de la science informatique...

3 ... en utilisant différentes machines et en découvrant le fonctionnement des réseaux

4 ... en créant, en exécutant, en comparant et en corrigeant des programmes

#### Algorithme et programmation

Création et comparaison de programmes avec des séquences, des tests conditionnels et des boucles à l'aide d'un langage de programmation visuel pour résoudre des problèmes simples.

#### Machines, systèmes, réseaux

Identification des composants principaux (*processeur, mémoire, dispositifs d'entrée/sortie...*) de différents types de machines (*ordinateur, tablette, robot...*) et de leurs fonctions

#### Liens disciplinaires

L1 21 – Compréhension de l'écrit

MSN 22 – Nombres ; MSN 25 – Modélisation

SHS 21 – Relation Homme-Espace



## INTENTIONS PÉDAGOGIQUES

Le but de cette activité est triple :

- mettre en évidence deux styles de programmation : la programmation séquentielle et la programmation événementielle ;
- montrer qu'il est nécessaire de choisir le langage informatique à utiliser en fonction de ce que l'on veut faire, tous n'ayant pas les mêmes caractéristiques.
- découvrir la programmation du robot Thymio avec Scratch, un autre langage de programmation.



## DESCRIPTION DE L'ACTIVITÉ

Cette activité se déroule en deux séances de 45 minutes.

La première séance débute par une découverte de quelques blocs de Scratch dédiés à Thymio. Puis, un programme est mis au point avec les capteurs de distance (ou capteurs de proximité) et les moteurs de Thymio. C'est l'occasion d'exposer une première différence entre Scratch et Thymio-VPL en ce qui concerne la gestion de la boucle événementielle. La fin de la séance est consacrée à l'élaboration d'un autre programme plus complexe permettant de comparer l'aisance de programmation entre Scratch et Thymio-VPL.

La seconde séance poursuit l'étude comparative de Scratch et Thymio-VPL au travers d'un programme simple, duquel émergent les notions

de programmation événementielle et séquentielle. La suite de la séance propose la programmation d'un jeu qui, outre l'aspect ludique pour les élèves, introduit la notion de programmation concurrente.

#### Prérequis :

- avoir des connaissances de base du robot Thymio, connaître ses composants tels que ses capteurs et actionneurs, savoir le programmer avec le langage Thymio-VPL ;
- connaître le langage de programmation Scratch (différent de Scratch Jr).

En particulier, il est crucial que les élèves aient fait deux scénarios de 7<sup>e</sup> : « Thymio VPL Les missions » et « Découverte de Scratch ».

*Remarque : les fiches sont destinées à être affichées numériquement à la classe pendant les explications.*

# Séance 1 - Programmer Thymio Avec Scratch

	<b>MODALITÉ</b>	En groupes de 2-3 élèves
	<b>MATÉRIEL</b>	<ul style="list-style-type: none"> <li>Fiche 1 : démarrer Scratch : les essentiels</li> <li>Fiche 2 : script du 1<sup>er</sup> programme en Scratch, Thymio-VPL et VPL3</li> <li>Fiche 3 : script du programme complet en Scratch, Thymio-VPL et VPL3</li> <li>Ordinateur avec le logiciel Thymio Suite (par groupes)</li> <li>Robot Thymio connecté à l'ordinateur grâce à un dongle ou à un câble USB (par groupes)</li> </ul>
	<b>DURÉE</b>	45 minutes



TEMPS 1.1

**PRÉSENTATION DES BLOCS  
SCRATCH POUR THYMIO**

15 minutes



## CONSIDÉRATIONS SUR SCRATCH POUR THYMIO

Scratch est à la fois un langage de programmation graphique par blocs, et une application qui permet de programmer avec ce langage.

Il a été développé par l'équipe de Mitchel Resnick, directeur du laboratoire Lifelong Kindergarten au MIT (Massachusetts Institute of Technology). Tandis que le logiciel est aujourd'hui traduit en 70 langues, le langage Scratch s'est hissé parmi

les 20 langages de programmation les plus populaires au monde, en 2020. Ce logiciel est en open source, c'est-à-dire gratuit, modifiable et réutilisable.

Scratch est disponible sous forme d'application à installer sur tablette ou ordinateur, mais fonctionne aussi en ligne par le biais du lien [78-S2-01](#).

Une version AsebaScratch a été créée en ajoutant des blocs Scratch dédiés au robot Thymio, pour le faire agir et gérer les capteurs.



« Connectez le robot à l'ordinateur, démarrez le logiciel Thymio Suite et sélectionnez le langage de programmation Scratch pour contrôler le robot Thymio. »

Afficher numériquement la [Fiche 1](#) à la classe.

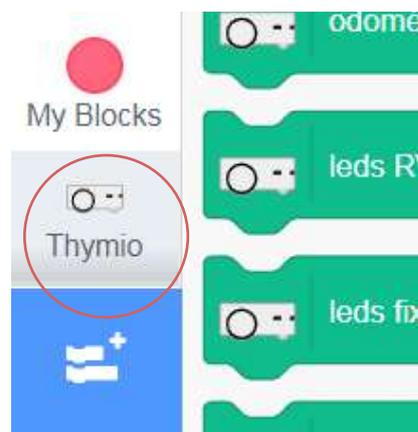
Puis, sélectionner le robot (1) et démarrer Scratch (2) (étape 2 de la Fiche 1).

Dans la barre des catégories de blocs à gauche, la dernière en bas concerne les blocs spécialement créés pour Thymio (voir figure ci-contre).

Une fois cliqué sur l'icône Thymio, une série de blocs verts apparaît, il y en a 46.

Ils permettent de commander les actionneurs (moteurs, lumière, etc.), de récupérer les données captées par le robot et envoyées à Scratch (capteurs, micro, température, etc.).

Dans cette séance, nous utilisons seulement trois blocs (étape 3 de la Fiche 1).



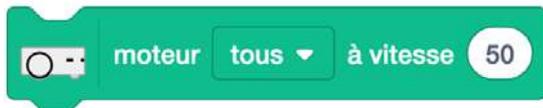
**CONSIGNES BLOCS MOTEUR :**

Avant de donner la consigne, demander aux élèves ce qu'il faudrait faire pour que le robot avance et dans un deuxième temps, pour qu'il s'arrête. Une fois que la plupart des élèves ont trouvé, donner la réponse.



« Placez ces trois blocs dans la zone centrale de script sans les assembler.

Cliquez sur le bloc,



Thymio avance.

Cliquez sur le bloc,



Thymio s'arrête. »

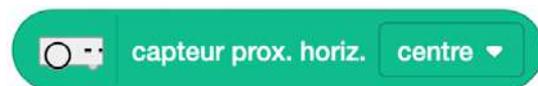
L'option [tous] du premier bloc permet de choisir les deux moteurs ou seulement un des deux moteurs, droite ou gauche.

La valeur 50 correspond à la vitesse de Thymio. Elle peut varier de 150 à -150 (valeur négative pour reculer).

Il est nécessaire de comprendre que ce bloc se contente de mettre en route les roues (aux mêmes vitesses). Les roues continueront à tourner, jusqu'à ce qu'un autre bloc vienne modifier ou arrêter le mouvement, comme le bloc « stop moteur » par exemple.

**CONSIGNES BLOC CAPTEUR :**

« En s'assurant qu'il n'y a rien devant Thymio, cliquez sur le bloc



Quelle valeur relevez-vous ? »

Réponse : 0.

« Puis, collez un doigt sur le capteur avant (celui du milieu) du robot, et cliquez à nouveau sur le bloc.

Quelle valeur relevez-vous ? »

Réponse : entre 4000 et 5000. Si on refait le test avec le doigt à 1 cm du capteur, une valeur plus faible s'affiche.

En résumé, plus un obstacle est **proche**, plus le capteur renvoie une **valeur importante**.

L'option [centre] permet de sélectionner un des sept capteurs de distance ou de proximité horizontaux de Thymio, cinq devant et deux à l'arrière.

**DOCUMENTATION BLOCS**

Pour une documentation complète des blocs, se référer au livret Thymio et Scratch disponible avec ce lien : [🔗 78-A9-01](#), page 68.



TEMPS 1.2

## UN PREMIER SCRIPT EN SCRATCH

15 minutes



## SCRIPT ET PROGRAMME

Dans le cadre de Scratch, nous adoptons la convention suivante : un script désigne un ensemble de blocs liés entre eux verticalement, qui débute par un bloc de type événement (forme de chapeau). Scratch permet l'exécution de plusieurs scripts en même temps. Le terme programme désigne alors l'ensemble des scripts exécutés, c'est-à-dire ceux figurants dans l'onglet Code de Scratch.

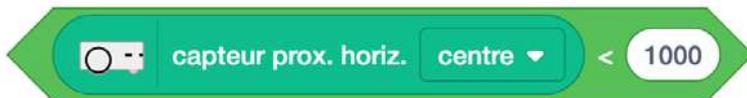


« En utilisant à la fois les blocs des catégories “Événement”, “Contrôle”, “Opérateurs” et ceux de Thymio, programmez le comportement suivant :

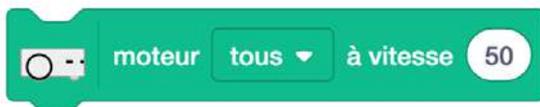
1. quand le drapeau vert est cliqué, si le capteur de proximité avant (du milieu) ne détecte pas d'obstacle, alors le robot avance ;
2. si au contraire, il capte quelque chose, Thymio s'arrête ou reste à l'arrêt. »

Laisser aux élèves un temps de réflexion, ensuite leur demander ce qu'ils ont trouvé.

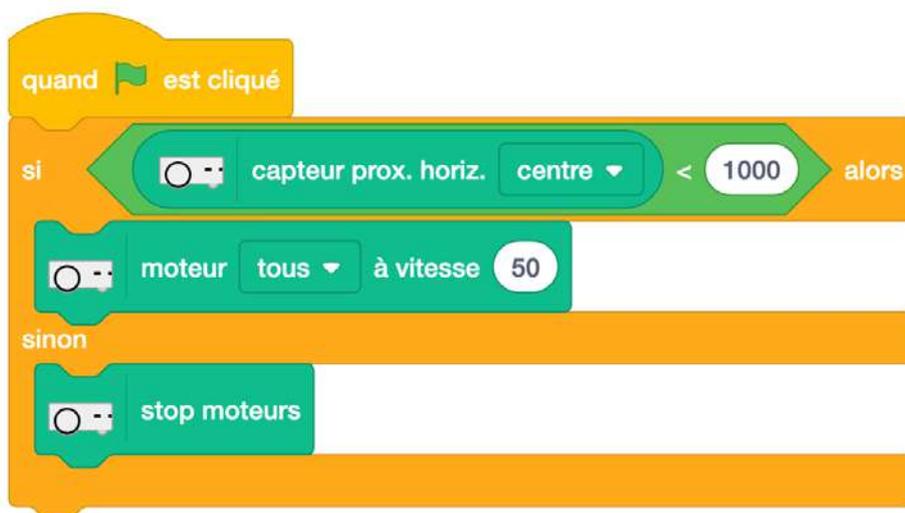
Réponse : la condition de la ligne 1 (ne capte rien) se traduit ainsi :



et l'action « avancer par » :



Il est probable que les élèves produisent le script suivant (code 1) :



Ce script ne fonctionne pas car le test du capteur central n'est exécuté qu'une seule fois, au tout début de l'exécution du script. Une fois ce test effectué, l'exécution du programme se termine. Thymio est alors livré à lui-même et ne communique plus avec Scratch. S'il n'y a pas d'obstacle, ses moteurs se mettent en route, Thymio continue à avancer.

Il faut ajouter le bloc « répéter indéfiniment » pour créer une boucle qui renouvelle l'exécution du bloc de test en permanence (figure 1, [Fiche 2](#)). Les élèves modifient leur script et observent que Thymio se comporte comme attendu.

#### Comparaison avec Thymio-VPL :

On ne demande pas aux élèves de réaliser le script en Thymio-VPL (figure 2, Fiche 2), on se contente de l'afficher, soit pour le commenter, soit pour demander aux élèves de l'expliquer.

La programmation en Thymio-VPL du comportement précédent utilise deux lignes.

Thymio-VPL ne montre pas de boucle entourant les tests de capteurs. Cette boucle existe pourtant, mais elle est cachée. Les deux lignes sont exécutées plusieurs dizaines de fois par seconde.



### DIFFÉRENCE ENTRE THYMIO-VPL ET SCRATCH

Une différence majeure entre Thymio-VPL et Scratch est mise en évidence. Dans Thymio-VPL, la boucle qui consulte les événements est non accessible au programmeur (boucle implicite) alors qu'avec Scratch, elle est sous le contrôle de ce dernier (boucle explicite). Cette différence a des conséquences importantes sur les fonctionnalités proposées par les deux langages. Certaines seront découvertes par la suite.



#### TEMPS 1.3

#### UNE SUITE AU PROGRAMME PRÉCÉDENT

15 minutes

Complétons le programme précédent par le comportement suivant :



« **Si, au cours de l'exécution du programme précédent, on appuie sur le bouton central, Thymio change de comportement ; il tourne sur lui-même tout en se colorant en rouge et en émettant un son. De plus, il ne réagit plus à la présence ou non d'un obstacle.** »

À moins que les élèves soient des utilisateurs avertis de Scratch, l'élaboration de ce complément mérite une assistance. Les questions suivantes, posées aux élèves, permettent d'accompagner et de les guider dans la réalisation du programme demandé. Pour chaque question, il est nécessaire de laisser aux élèves un temps de réflexion avant de donner la solution.

#### Question 1 :

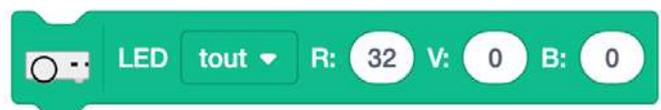


« **Dans les nouveaux blocs pour Thymio, quels sont ceux qui permettent de l'allumer en rouge et de jouer un son ?** »

Pour favoriser la recherche des élèves, on peut les autoriser à déposer les blocs envisagés dans la zone de script (sans les coller) et à cliquer dessus pour les tester.

Réponse :

• *bloc lumière :*



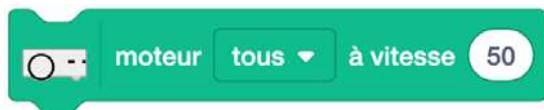
• *bloc son :*



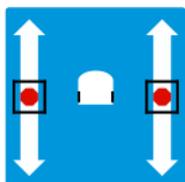
## Question 2 :



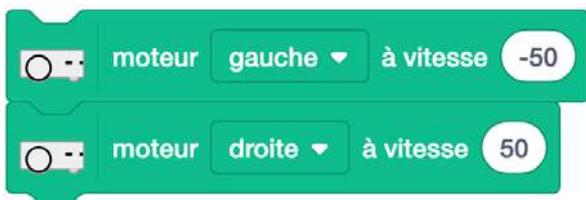
« Pour que Thymio tourne sur lui-même, il faut utiliser deux fois le bloc :



Mais quels moteurs choisir et quelles vitesses leur donner ? Pensez au bloc de Thymio-VPL. Le robot Thymio a deux moteurs, un pour chaque roue. »



Réponse : il faut utiliser deux fois le bloc « moteur », un des moteurs tourne dans le sens « avancer », l'autre dans le sens « reculer ».



## Question 3 :

« Les blocs de ce nouveau comportement peuvent-ils s'insérer dans la boucle déjà écrite ? »

Réponse : non, car le comportement vis-à-vis d'un obstacle doit être oublié. Comme expliqué dans la consigne plus haut, le robot Thymio ne réagit plus à la présence ou non d'un obstacle. Il faut donc placer les nouveaux blocs après la boucle et par conséquent l'interrompre.

## Question 4 :

« Sachant que le nouveau comportement doit commencer quand on appuie sur le bouton central de Thymio, par quel bloc doit-on remplacer le bloc « Répéter indéfiniment » ? »

Réponse : par le bloc



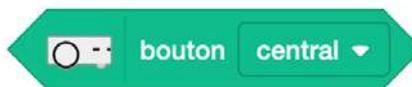
## Question 5 :

« Ce nouveau bloc doit recevoir un bloc hexagonal du type :



Parmi ces blocs, lequel doit-on choisir ? »

Réponse :



« Compléter le programme précédent pour réaliser le programme complet. »

Solution sur la [Fiche 3](#).

## Comparaison avec Thymio-VPL :

En Thymio-VPL, on ne peut pas sortir de la première boucle car on n'y a pas accès. Il faut donc trouver une astuce pour programmer le 2<sup>e</sup> comportement dans la même boucle.

C'est possible en utilisant la notion d'état, voir le programme donné sur la Fiche 3.

## Séance 2 - Programmation séquentielle, programmation événementielle

	<b>MODALITÉ</b>	En groupes de 2-3 élèves
	<b>MATÉRIEL</b>	<ul style="list-style-type: none"> <li>• Fiche 4 : solution programme du carré (Scratch et Thymio-VPL)</li> <li>• Fiche 4.1 : solution programme du carré (Thymio-VPL3)</li> <li>• Fiche 5 : script de la séance 1 avec le bloc « avancer »</li> <li>• Fiche 6 : jeu de la chaussette</li> <li>• Fiche 6.1 : jeu de la chaussette « corrigé »</li> <li>• Ordinateur avec le logiciel Thymio Suite (par groupes)</li> <li>• Robot Thymio connecté à l'ordinateur grâce à un dongle ou à un câble USB (par groupes)</li> </ul>
	<b>DURÉE</b>	45 minutes

Au cours de cette séance, d'autres différences importantes entre Scratch et Thymio-VPL sont mises en évidence, ce qui permet de distinguer deux types (deux paradigmes en langage technique) de programmation, la programmation événementielle et la programmation séquentielle (ou procédurale).



### TEMPS 2.1

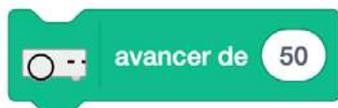
### DESSINER UN CARRÉ

15 minutes

Il s'agit de réaliser un programme simple. Dans la cavité centrale de Thymio, on ajoute un crayon qui permet de dessiner. Sur une feuille blanche, Thymio doit réaliser la tâche suivante : dessiner un carré.

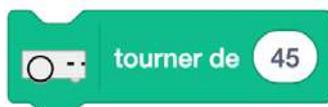
Pour exécuter cette tâche, il faut deux nouveaux blocs de Scratch (à tester par les élèves) :

- bloc « avancer » :



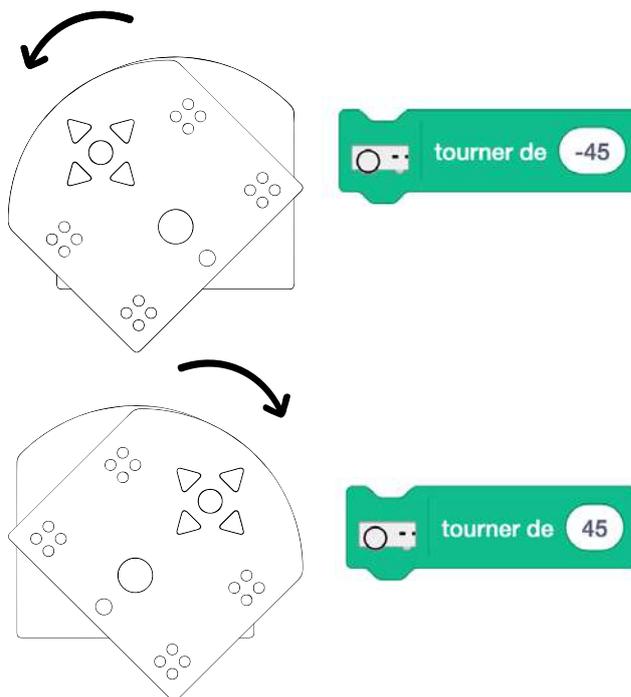
le nombre correspond à la distance à parcourir, exprimée approximativement en mm ;

- bloc « tourner » :



le nombre correspond à l'angle de rotation, exprimé approximativement en degré.

Si l'angle est positif, le robot tourne à droite. Si l'angle est négatif, il tourne à gauche. La rotation se fait sur place, sans avancer.





« Assemblez le script pour que Thymio dessine un carré. »

Réponses possibles, [Fiche 4](#) :

Les réponses A et B sont correctes. D'un point de vue optimisation, la B est préférable car elle évite d'utiliser plusieurs fois les mêmes blocs. Néanmoins, pour prévenir d'éventuelles confusions dans l'utilisation des boucles, on valide sans remarque particulière le premier script (A).

Dans ce programme, l'enchaînement des actions effectuées par Thymio ne nécessite pas d'événement pour se dérouler, à l'exception du bloc :



qui déclenche l'exécution du script.

Un tel type de programmation est qualifié de **séquentiel** (séquence = suite).

Il n'est pas possible de faire un tel programme en Thymio-VPL car les actions sont obligatoirement déclenchées par un événement.

Thymio-VPL est un type de programmation **événementielle**, comme le rappelle la figure ci-dessous.

The screenshot shows the Thymio-VPL programming environment. On the left, under the 'Événements' (Events) tab, there are three event blocks: a directional pad, a semi-circle with segments, and a rectangle with two small squares. The first block is selected. In the center workspace, a yellow block is being assembled, consisting of the selected event block, a 'déclenche' (triggers) block with two yellow dots, and an action block (a blue block with two vertical arrows and a robot icon). On the right, under the 'Actions' tab, there are three action blocks: a blue block with two vertical arrows and a robot icon, a grey block with three horizontal bars (red, green, blue), and another grey block with three horizontal bars (red, green, blue) and a robot icon. The interface has a window title bar with '○○○' on the left and '-□x' on the right.



## REMARQUE IMPORTANTE



On observe facilement que le carré n'est pas carré. Il faut bien avoir en tête qu'un robot n'est pas un algorithme désincarné dans une machine, mais un dispositif composé d'électronique, de fils, de moteurs, de frottements, et que son comportement ne saurait être parfait.

Un robot davantage perfectionné et coûteux tracerait un carré plus réaliste (avec conservation des propriétés d'un carré) mais il resterait toujours des imperfections de tracé. Dans la programmation d'un robot performant, une grande partie du temps est consacrée à la correction des petites erreurs commises par le matériel.

On peut néanmoins simuler un comportement séquentiel avec Thymio-VPL en utilisant un événement particulier, le timer. Il s'agit d'un chronomètre qui déclenche un événement quand il s'arrête.

À titre informatif, voir les programmes en Thymio-VPL et Thymio-VPL3, Fiches 4 et 4.1

(il faut neuf lignes pour un carré, il en faudrait 21 pour un décagone).



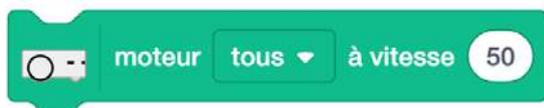
TEMPS 2.2

DIFFÉRENTS BLOCS POUR LE MOUVEMENT

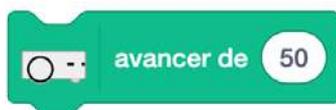
10 minutes

Scratch propose deux blocs pour faire avancer Thymio :

- le bloc :



- le bloc :



1<sup>re</sup> différence : avec le bloc « moteur », Thymio ne s'arrête pas. Avec le bloc « avancer », Thymio s'arrête au bout de 5 cm (environ).

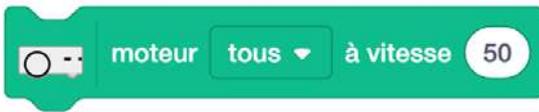
Seconde différence : quand Thymio se déplace avec le bloc « avancer », il ne prend pas en compte les données de ses capteurs. Il reste « sourd » aux événements. Si un obstacle se trouve devant, il le percute.

Ce comportement est généralisable à tous les blocs du type « avancer » ou « tourner ».



« Assembler et lancer le script suivant [Fiche 5](#) . »

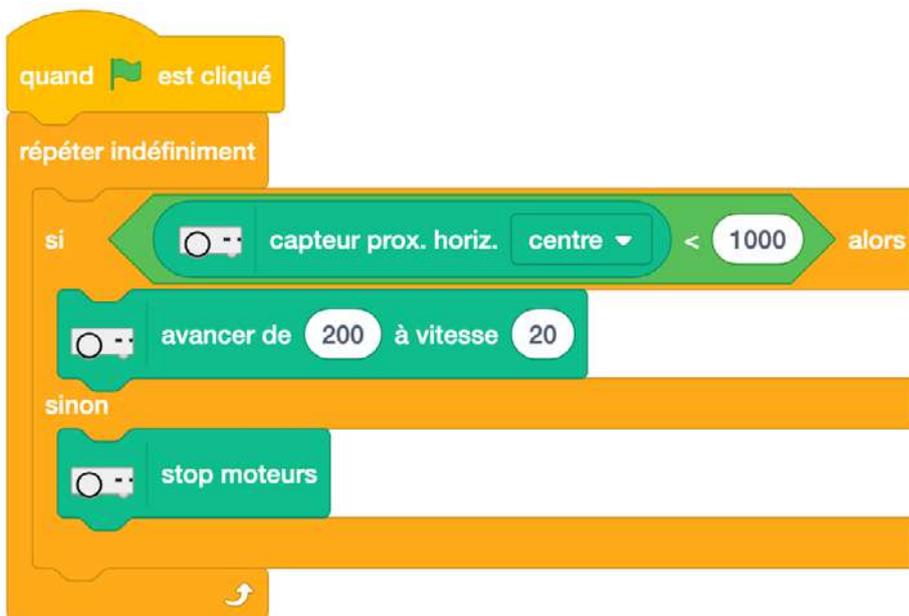
Ce script reprend celui de la séance précédente en remplaçant le bloc :



par le bloc :



Mettre la distance à 200 et la vitesse à 20.



On remarque un paramètre supplémentaire : la vitesse. Le temps mis par Thymio pour exécuter le bloc « avancer » est de 10 secondes car il parcourt 200 mm à la vitesse de 20 mm/s.

Pendant ce laps de temps, le robot ne réagit pas à son capteur central.

On peut considérer que les blocs « avancer » cachent un code de nature *séquentielle*, puisqu'ils s'exécutent quoiqu'il arrive.



## ROBOTIQUE ET PROGRAMMATION

Le type de **programmation événementielle** est parfaitement adapté aux fondamentaux d'un robot.

Un robot met en jeu en permanence une boucle sensori-motrice (boucle fermée avec feedback de l'environnement) dans laquelle un événement, après analyse, provoque (ou non) une action.

Si on programme un robot avec une **programmation séquentielle**, (comme le programme du tracé de carré), on n'utilise aucun capteur (on n'a pas besoin de tenir compte de l'environnement du robot pour effectuer la tâche) et le robot n'a aucune initiative. Dans ces conditions, le robot se comporte comme un **automate** et non comme un robot qui prendrait en compte son environnement.



TEMPS 2.3

## LE JEU DE LA CHAUSSETTE

20 minutes

Cette séance s'achève par la programmation d'un jeu qui permet de découvrir une autre possibilité de Scratch, absente de Thymio-VPL : la **programmation concurrentielle**.

Jusqu'à présent, dans les programmes un seul script était utilisé, chacun étant repéré par un bloc de type chapeau. Cependant, il est possible de faire plusieurs scripts à la fois. Ils s'exécutent de manière indépendante, simultanée ou non. Chacun démarre à la suite de l'événement correspondant à son bloc chapeau.

## SCRIPT 1



## SCRIPT 2



## SCRIPT 3



Le jeu de la chaussette est un jeu pratiqué dans le scoutisme. Deux adversaires sont face à face. Ils ont chacun dans leur dos une chaussette ou un foulard passé dans la ceinture ou le pantalon. Les deux adversaires se tournent autour pour attraper la chaussette de l'autre. Le premier qui y arrive a gagné.

Pour adapter ce jeu à deux robots Thymio, on remplace la chaussette par les capteurs arrière. Le robot qui voit un de ses capteurs arrière s'allumer (quelle que soit la raison) a perdu. Les deux robots sont contrôlés par les touches flèches du clavier. Chaque élève en contrôle un.

On place de préférence les deux robots dans une enceinte fermée, de sorte qu'un Thymio qui s'approche trop près d'un mur perde la manche (règles sur la [Fiche 6](#)).

Eu égard au temps limité, une partie du programme leur sera communiqué (Fiche 6) :

- le script pour marquer que le joueur a perdu en allumant des leds qui clignotent ;
- le script pour commander la rotation à gauche à l'aide de la touche flèche gauche du clavier.

Il reste ainsi quatre scripts à écrire par imitation du précédent :

- pour tourner à droite ;
- pour aller tout droit ;
- pour reculer ;
- pour s'arrêter.

Le jeu se déroule soit entre deux joueurs, soit en mode arène où tous les robots sont mis ensemble. Le combat s'achève quand il ne reste qu'un seul robot. L'enseignant veille à ce qu'un Thymio qui perde (leds qui clignotent) soit retiré du jeu.



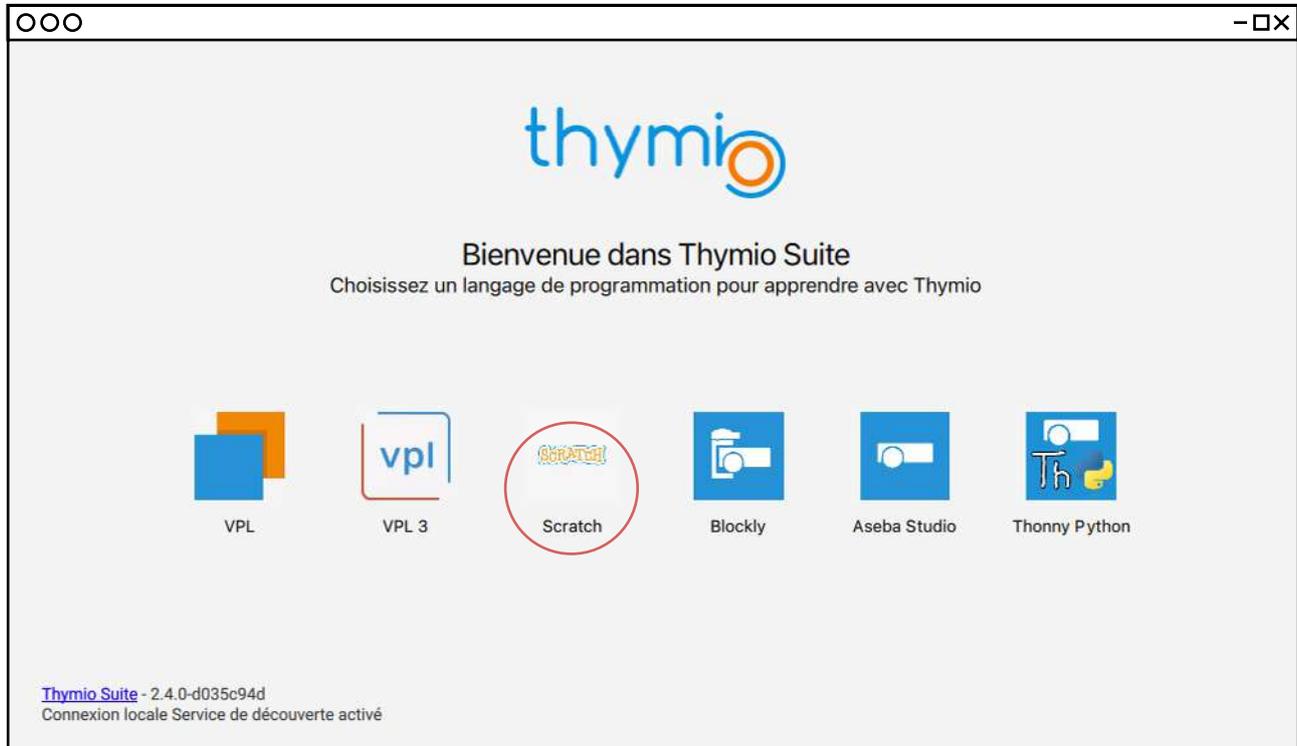
« En vous aidant de la **Fiche 6**, écrire le programme du jeu de la chaussette. »

Afficher la Fiche 6 et la commenter.  
Solutions sur la Fiche 6.1.

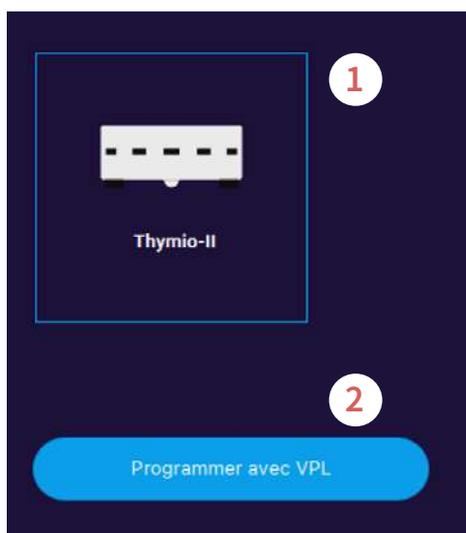
# Démarrer Scratch : les essentiels



## ÉTAPE 1

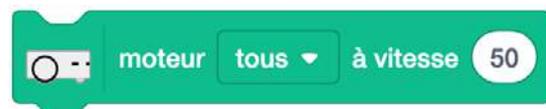


## ÉTAPE 2

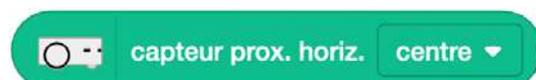


## ÉTAPE 3

- 2 blocs actionneur situés au début de la liste :



- 1 bloc capteur situé à la fin de la liste :





# Script du 1<sup>er</sup> programme en Scratch, Thymio-VPL et VPL3

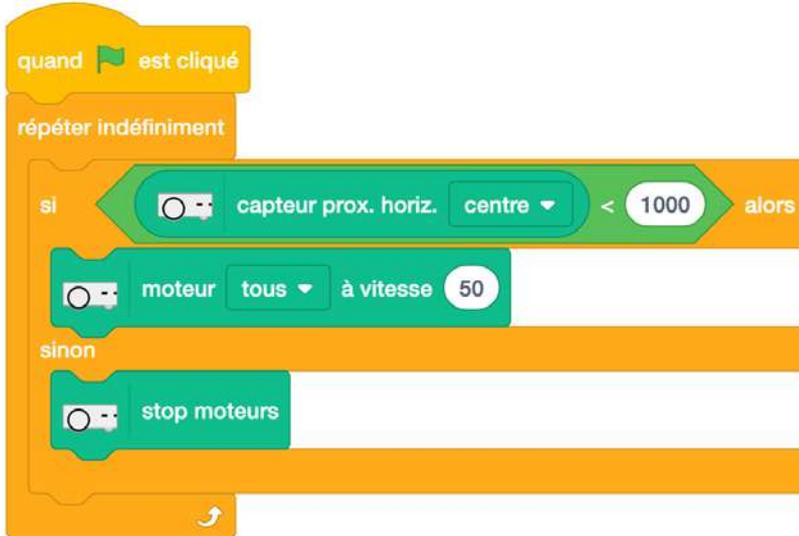


Figure 1 : bloc

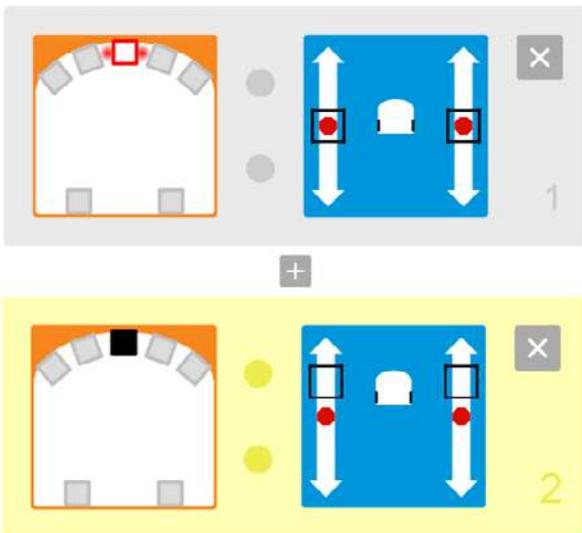


Figure 2 : Thymio-VPL

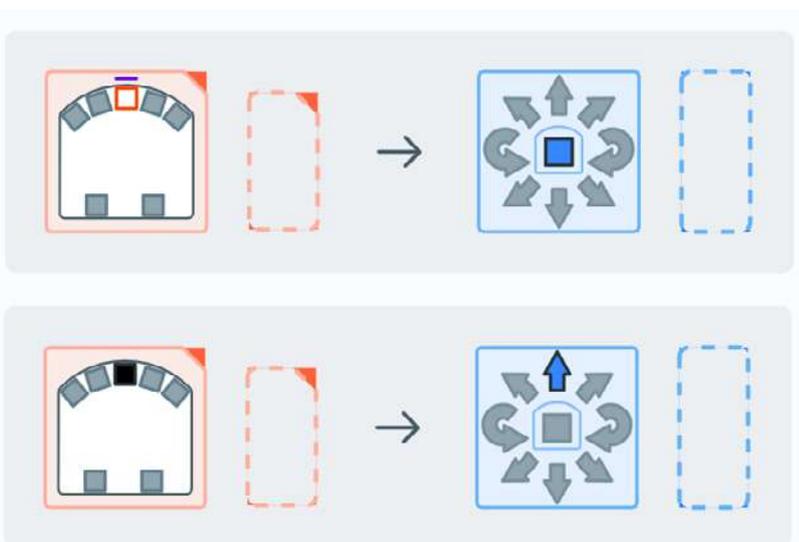


Figure 3 : Thymio-VPL3



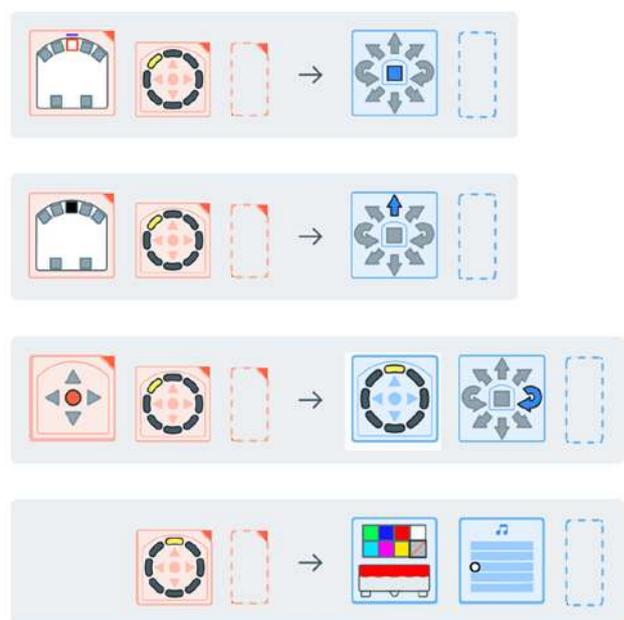
# Script du programme complet en Scratch, Thymio-VPL et VPL3

```

    quand est cliqué
    éteindre LED
    répéter jusqu'à ce que bouton central
    si capteur prox. horiz. centre < 1000 alors
    moteur tous à vitesse 50
    sinon
    stop moteurs
    répéter indéfiniment
    moteur gauche à vitesse -50
    moteur droite à vitesse 50
    LED tout R: 32 V: 0 B: 0
    jouer son système 5
  
```



Thymio-VPL



Thymio-VPL3



# Solution programme du carré

(Scratch et Thymio-VPL)



Réponse A :

```

    quand [drapeau] est cliqué
    avancer de 100
    tourner de 90
    avancer de 100
    tourner de 90
    avancer de 100
    tourner de 90
    avancer de 100
    tourner de 90
  
```



Réponse B :

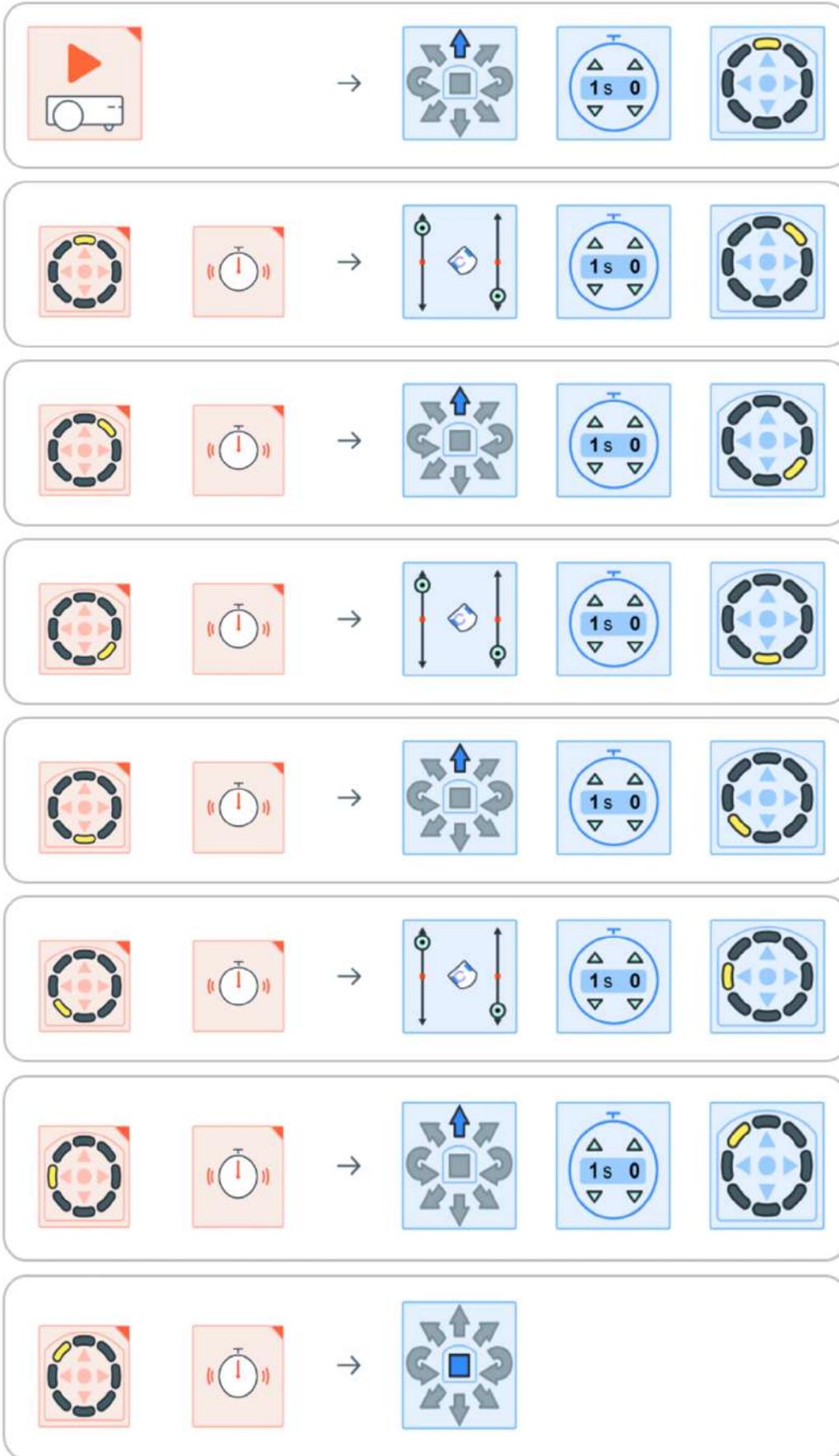
```

    quand [drapeau] est cliqué
    répéter 4 fois
    avancer de 100
    tourner de 90
  
```

PROGRAMMATION ÉVÉNEMENTIELLE	PROGRAMMATION SÉQUENTIELLE
Style (paradigme) de programmation dans lequel le code s'exécute seulement quand un événement a lieu.	Style (paradigme) de programmation dans lequel les instructions s'exécutent les unes après les autres, sans tenir compte d'éventuels événements.

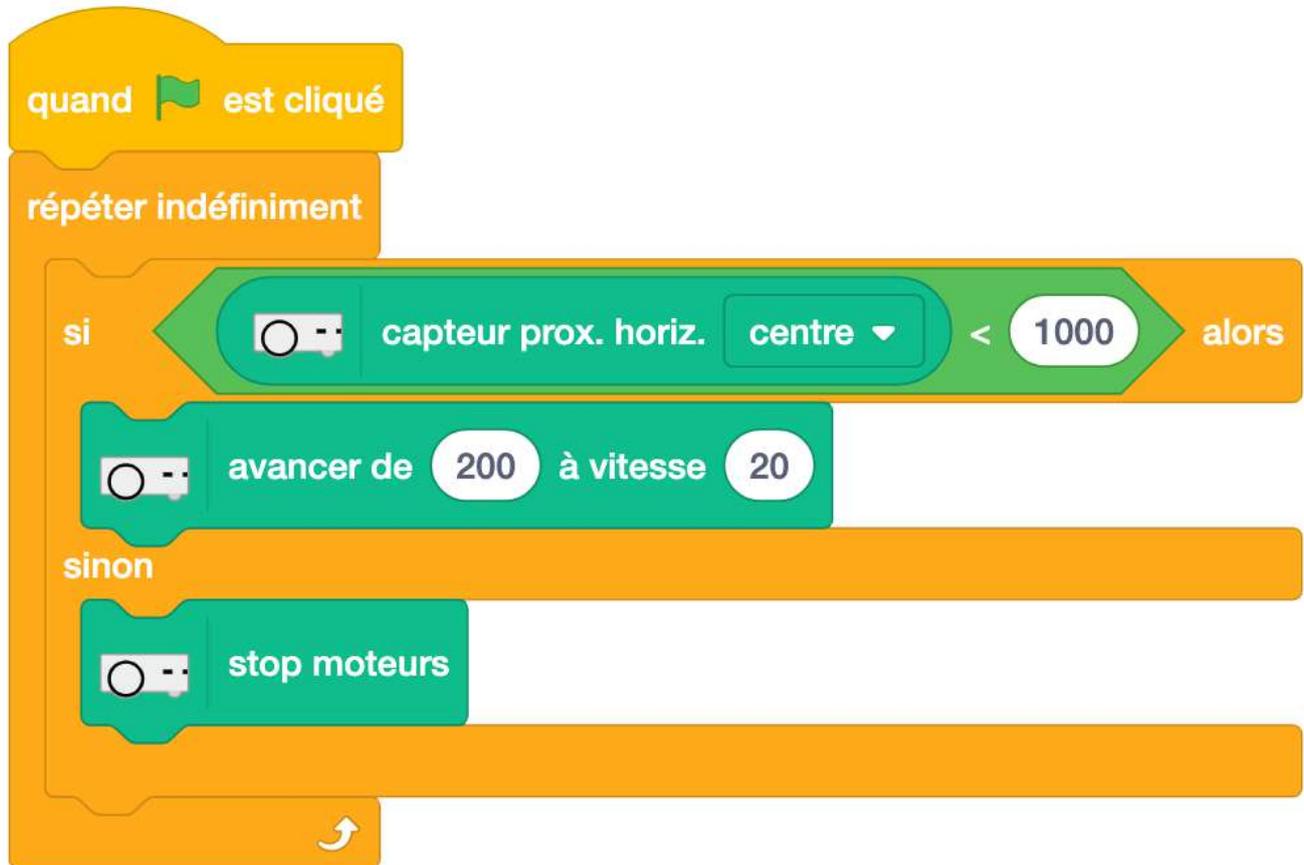


# Solution programme du carré (Thymio-VPL3)





# Script de la séance 1 avec le bloc « avancer »



# Jeu de la chaussette

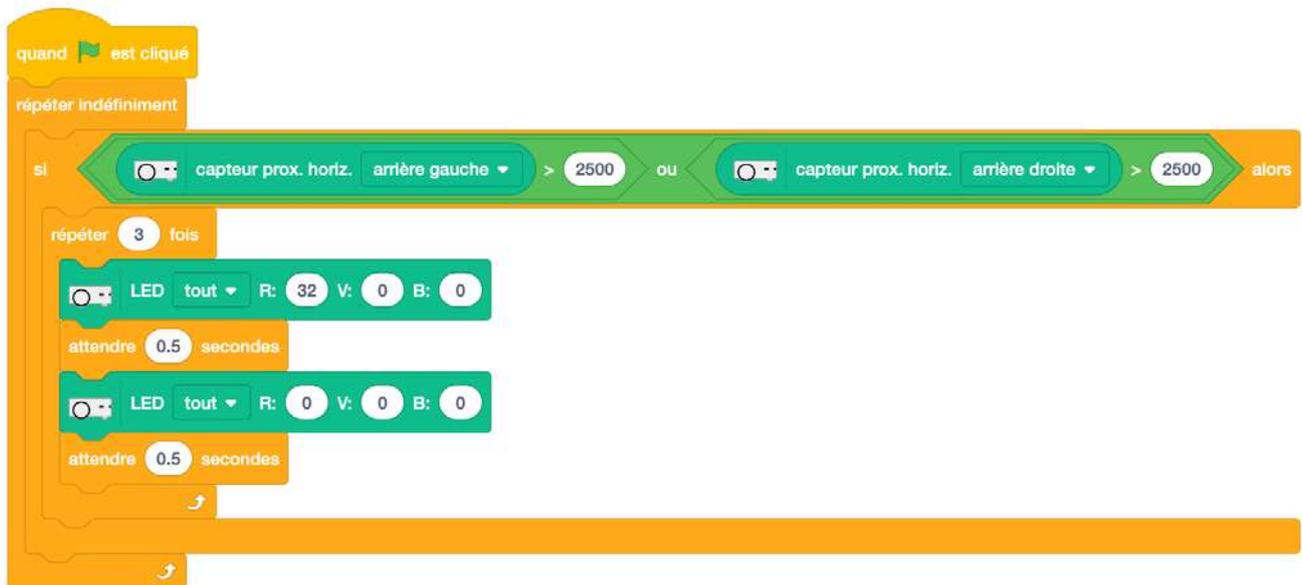


## RÈGLES :

Deux Thymio sont face à face. Chacun doit tenter de contourner son adversaire pour se rapprocher de ses capteurs arrière (les leds des capteurs s'allument alors).

Le joueur du Thymio dont Leds arrière s'allument a perdu. Attention, si la scène comporte des murs ou des obstacles qui allument les leds (en cas de recul), le joueur perd également.

Script pour signifier que le joueur a perdu (à recopier dans Scratch) :



Script pour commander la rotation à gauche à l'aide de la touche flèche gauche du clavier (à recopier dans Scratch) :



Il vous reste quatre scripts à écrire par imitation du précédent :

- pour tourner à droite ;
- pour aller tout droit ;
- pour reculer ;
- pour s'arrêter.

*Remarque : vous pouvez essayer d'autres valeurs pour les vitesses.*

CORRIGÉ

# Jeu de la chaussette



```

graph TD
    Start[quand est cliqué] --> Loop[répéter indéfiniment]
    Loop --> If[si]
    If --> And[capteur prox. horiz. arrière gauche > 2500 ou capteur prox. horiz. arrière droite > 2500]
    And --> Repeat[répéter 3 fois]
    Repeat --> LED1[LED tout R: 32 V: 0 B: 0]
    LED1 --> Wait1[attendre 0.5 secondes]
    Wait1 --> LED2[LED tout R: 0 V: 0 B: 0]
    LED2 --> Wait2[attendre 0.5 secondes]
    Repeat --> Repeat
    Loop --> Loop
  
```

```

graph TD
    Start[quand la touche flèche gauche est pressée] --> Motor1[moteur gauche à vitesse 15]
    Motor1 --> Motor2[moteur droite à vitesse 50]
  
```

```

graph TD
    Start[quand la touche espace est pressée] --> Stop[stop moteurs]
  
```

```

graph TD
    Start[quand la touche flèche haut est pressée] --> Motor[moteur tous à vitesse 50]
  
```

```

graph TD
    Start[quand la touche flèche bas est pressée] --> Motor[moteur tous à vitesse -50]
  
```

```

graph TD
    Start[quand la touche flèche droite est pressée] --> Motor1[moteur gauche à vitesse 50]
    Motor1 --> Motor2[moteur droite à vitesse 15]
  
```