

1, 2, 3, codez ! - Activités cycle 2 - Séances 3.7 et 3.8: Parcours d'obstacles

Résumé	Les élèves doivent reproduire le mode « explorateur » de Thymio jaune. Au cours d'une première séance, ils écrivent le programme. Dans une seconde séance, ils testent leur programme sur un labyrinthe réel.
Notions	<p>« Machines »</p> <ul style="list-style-type: none">• Les machines qui nous entourent ne font qu'exécuter des "ordres" (instructions)• En combinant plusieurs instructions simples on peut effectuer une tâche complexe <p>« Langages »</p> <ul style="list-style-type: none">• On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine.• Un bug est une erreur dans un programme. <p>« Robot »</p> <ul style="list-style-type: none">• Un robot est une machine qui peut interagir avec son environnement• Un robot possède un ordinateur qui décide quelles actions faire dans quelles situations <p>« Algorithmes »</p> <ul style="list-style-type: none">• Un test dit quelle action effectuer quand une condition est vérifiée

Matériel	Par groupe <ul style="list-style-type: none"> • Un Thymio • Un ordinateur disposant du logiciel VPL • (facultatif) Fiche 27
Lexique	Bug
Durée :	2 séances de 1 heure

Mise en situation

L'enseignant rappelle aux élèves que Thymio était livré avec des modes pré-programmés. Il leur propose un défi : réussir à reprogrammer par eux-mêmes un équivalent (simplifié) du mode « jaune » du Thymio. Les élèves se [souviennent](#) qu'il s'agit du mode explorateur, où Thymio avance en esquivant les obstacles.

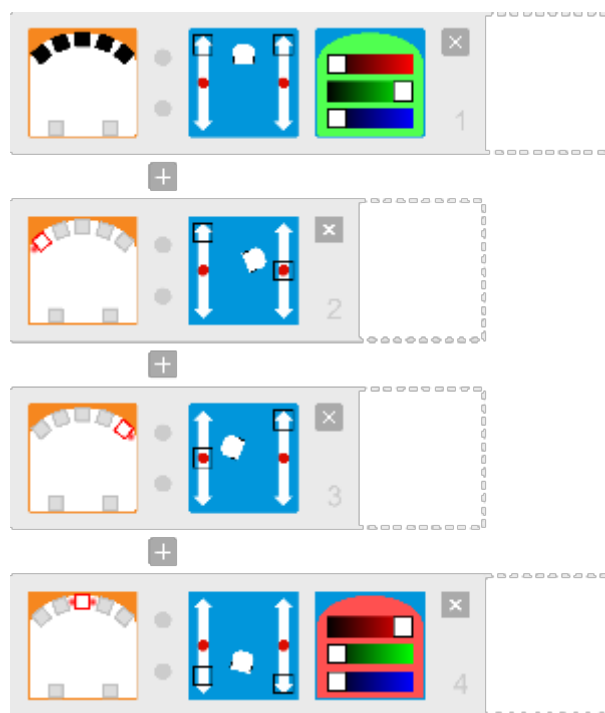
Défi : reproduire un Thymio explorateur (par groupes)

Selon l'aisance de la classe et l'âge des élèves, ce défi peut prendre plusieurs formes. Pour les élèves les plus autonomes, l'enseignant peut garder pour lui la [Fiche 27](#) et l'utiliser comme aide-mémoire. Inversement, la programmation peut être plus guidée si l'enseignant distribue aux élèves cette même fiche.

En groupes ou en classe entière, il faut d'abord arriver à conceptualiser les diverses étapes à programmer: *par défaut, que fait Thymio ? S'il détecte un obstacle à sa droite, que doit-il faire ? Et à gauche ? Devant lui ?* Puis il faut utiliser VPL pour programmer le robot, et tester si le programme marche en jouant avec Thymio sur la table.

Il est fort probable que la session arrive à sa fin après l'écriture des programmes. Le test « grandeur nature » aura donc lieu à la session suivante.

Un exemple de programme correct est :



Expérimentation : un vrai test pour notre Thymio (par groupes)

La classe prépare maintenant un grand labyrinthe constitué d'obstacles dont la hauteur est d'au minimum de 6cm. Tous les groupes vont tester leur programme simultanément : les robots vont interagir avec le labyrinthe, et interagir entre eux.

Si cela s'y prête, le sol du labyrinthe peut être muni d'une surface où dessiner : chaque groupe pourra insérer un feutre dans le trou prévu à cet effet sur le capot de Thymio. Cela permettra après coup de visualiser les chemins parcourus par les différents robots au cours de l'expérience.

Les groupes chargent le programme qu'ils ont conçu à la séance précédente, et laissent leur robot parcourir le labyrinthe. Ils peuvent améliorer leur programme au fur et à mesure des défauts observés : l'enseignant introduit alors le terme « bug » pour décrire ces dysfonctionnements.

Conclusion et traces écrites

La classe synthétise collectivement ce qui a été appris au cours de cette séance :

- *En combinant plusieurs instructions simples on peut effectuer une tâche complexe, comme parcourir un labyrinthe*
- *Un bug est une erreur dans un programme.*

Source URL: <http://www.fondation-lamap.org/fr/page/34511/1-2-3-codez-activites-cycle-2-seances-37-et-38-parcours-dobstacles>