

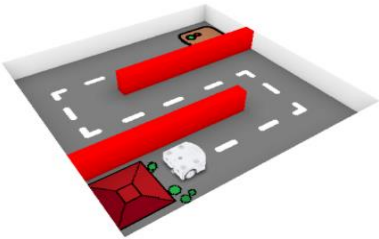
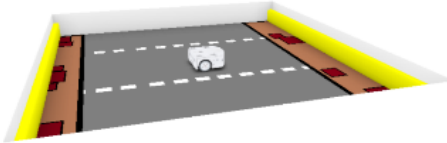
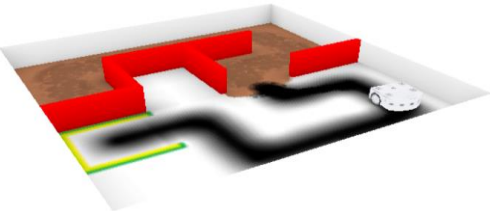


Image	Exercice	But de l'exo	Nombre de bugs
	1. Garden	Thymio doit faire le tour du jardin pour recuperer la clé	2
	2. Labyrith	Thymio doit suivre le tracé du labyrinth	2
	3. Buttons	Thymio doit répondre aux ordres des boutons	2
	4. Sensors	Thymio doit se fier à ses capteur pour faire des allers retours	2
	5. Parking	Thymio doit d'abord s'orienter avec les murs puis suivre la ligne noire et s'arreter sur la place verte	4

Possibilité de combiner	Erreur	Explication	Difficulté du bug	Longueur du code
Non, trop d'info	<u>Wrong timing</u>	Le robot tourne trop tôt. Le participant doit adapter le timing de la ligne droite	Moyen	20 (court)
	<u>Wrong turn</u>	Le robot tourne à droite au lieu de tourner à gauche. Le participant doit changer un signe moins	Facile	20 (court)
	<u>Invalid syntax</u>	Un symbol égal est manquant au step 10	Facile	29 (moyen)
Oui	<u>Wrong pivot</u>	Au step 14, le robot tourne à droite au lieu de tourner à gauche. Le participant doit juste changer l'appel de fonction	Facile	
Oui	<u>Missing global</u>	La variable motor_right_target a oublié d'être déclarée comme variable globale	Moyen	
	<u>Wrong button</u>	Le bouton backward a été remplacé par forward	Moyen	28 (moyen)
	<u>Wrong index</u>	Le programme essaie d'accéder à un index out of bound pour les capteurs (7 au lieu de 6)	Moyen	
Oui	<u>Wrong symbol</u>	Le signe ">" est utilisé au lieu du signe "<" pour la detection des capteurs	Difficile	21 (court)
Oui	<u>Wrong ground</u>	Le seuil pour detecter la ligne est trop haut car le sol est trop foncé, il faut le baisser	Difficile	40 (long)
	<u>Wrong turn</u>	Thymio se rapproche des murs ou s'éloigne des lignes, il faut changer un signe -	Moyen	
	<u>Add stop mode</u>	Il faut rajouter mode = STOP à la fin pour que le robot s'arrête	Difficile	

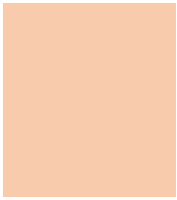
Scotch on sensor

Un bout de scotch bloque un
des capteurs au sol, il faut
l'enlever

Difficile

Taches de debuggage

Qualité du code	Temps	Motivation extrinsèque	Réplicabilité	Type de code
Bon	8 minutes	Il doit récupérer la clé qui est de l'autre côté du jardin, puis revenir à la maison	En VPL ou en scratch, changer les timing	Evenementiel
	2 minutes		En VPL ou en scratch, changer la direction	
	5 minutes	Il doit trouver la sortie du labyrinthe	En scratch seulement, car architecture séquentielle, et non pas événementielle	Séquentiel
	Bon	9 minutes	Thymio ne doit pas se cogner contre les murs, et arriver à l'usine des robots	Pas possible en VPL ou en scratch - > propre à python
Possible en VPL et scratch				
Mauvais	6 minutes	Thymio doit transporter des bouts de bois de la droite à la gauche	En VPL ou en scratch, referer au mauvais senseur	Evenementiel
			En VPL ou en scratch, mauvais seuil	
Moyen	10 minutes			
	5 minutes	Thymio doit passer le test ultime des robots : il doit s'échapper du labyrith, puis être capable de suivre la ligne noire au sol, puis s'arreter sur la place verte	Possible en VPL (preuve) et en scratch, utiliser les lumières pour les etats	Evenementiel sous forme de FSM
	10 minutes			



5 minutes



Types de bug	Compétence évaluée	Comportement d
		Expert
Logique	Introduction: voir si ils cherchent à comprendre le code avant de faire des modifications pour ne pas inserer de nouveaux bugs (sourtout avec un code etranger)	Run puis cherche à comprendre le code avant de corriger l'erreur
Logique		Run puis sait directement car familié avec le code
Syntaxique	Capacité à utiliser la console	Run, s'aperçoit un bug syntactic, corrige directement
Logique	Capacité de recherche de bug dans un code séquentiel	Run, se doute que le robot va tourner à droite, se focalise sur les "pivot_right"
Compile	Capacité de généralisation (copier le model de motor_left à motor_right)	Run, se rend compte de ce qui manque, se base sur motor_left_target pour corriger
Logique	Capacité de regarder le code dans son ensemble	Run, manipule thymio, se rend compte que forward est pourtant bien défini: l'erreur vient d'ailleur
Run-time	Capacité à utiliser la console et la leçon dans un code brouillon	Run, lit la console, se rappelle (ou va voir la leçon) pour corriger 7 en 6 pour le capteur arrière (bonus: met le code au propre)
Logique	Capacité de logique et débogage d'état, manipulation du robot	Run, se rend compte que Thymio ne change pas d'état, manipule Thymio, va voir la leçon, corrige
Logique	Deboggage d'état	Run, se rend compte que Thymio n'est pas dans le bon état, cherche ce qui fait basculer Thymio d'état, print(ground) et corrige
Logique	Capacité de logique et de manipulation	Run, est dans le bon état, mais cherche à se rapprocher des murs, manipulation du robot pour s'en rendre compte
Logique	Capacité d'anticipation de l'algorithme et de débogage d'état	Run, appuie sur le bouton, mais ne change pas d'état. Anticipe ce que fait l'algorithme et se rend compte que le mode STOP n'est jamais appelé

Environnemental

Capacité de réfléchir en
dehors du programme

Run, et se rendent compte que Thymio ne réagit pas
correctement aux capteur, `print()` les capteurs du sol
pour se rendre compte que `=0`, manipule le thymio
pour voir le bout de scotch

e debuggage attendu

Novice

Remarques

Run puis change des lignes au hasard sur ce qui lui paraît logique

Run puis doit chercher un peu. Retourne à la leçon pour voir comment tourner à gauche

Run, s'étonne que ça ne fonctionne pas, puis lit la console

2 versions: un grand labyrinth, plus complexe et plus intéressant, et un plus petit, plus rapide, plus facile à debugger

Run, peut se douter que le robot va tourner à droite, suit le code en suivant les mouvements du robot, sans se focaliser seulement sur `pivot_right()`

Run, se rend compte avec plus de difficulté de ce qu'il manque, retourne voir le court pour voir comment déclarer `motor_right_target`

Run, manipule Thymio, forward est pourtant bien défini, mais s'entête à vouloir modifier forward quand même

Run, s'étonne que ça ne fonctionne pas, puis lit la console, retourne voir la leçon, essaie de comprendre quel chiffre mettre

Run, n'a pas cette logique de bug d'état, change des trucs au hasard. Relis la leçon pour voir un truc similaire, et corrige en recopiant la leçon sans comprendre forcément
Run, peut avoir du mal à se rendre compte que l'état n'est pas bon, ne va pas utiliser de `print`, aura vraiment du mal à trouver l'erreur, va bcp utiliser la leçon pour voir si le code est bien écrit

Run, ne manipule peu/pas le robot, a vraiment du mal à voir d'où peut venir l'erreur, change des lignes au hasard jusqu'à ce que ça marche

2 versions: avec des variables, et harcodé. Une novice m'a dit qu'elle préférerait avoir les valeurs bruts (magic numbers / harcodé)

Run (si n'a pas déjà burn out), appuie sur le bouton mais continue. Ne va pas chercher à anticiper l'algorithme, va peut être essayer de modifier le code en insérant des nouveaux bugs

Vont difficilement comprendre que l'erreur puisse ne pas provenir du programme, mais de l'environnement. Il faut avoir le reflex de print et avoir assez confiance pour se dire que les valeurs sont dues à un défaut du robot et pas du programme



Debugging strategies -
identification, representation,
selection, application, evaluation