**Tâches de débogage avec Thymio**

**Projet de Semestre en Robotique**Ludovic Pujol – Master EPFL

***LECON (20 min)***

*Cette section contient une leçon brève sur Thymio et l’interface de programmation que vous allez utiliser, Thonny. Le but de cette leçon est de vous familiariser avec les concepts principaux de la programmation de Thymio. Il ne vous sera pas demander de coder un algorithme entier, mais plutôt de comprendre les quelques lignes de code qui vous seront présentées. N’hésitez pas à poser des questions en cas de doute.*

***Thymio ? Qu’est-ce que c’est ?***

Thymio est un robot à deux roues, utilisé pour son aspect éducatif. Des programmes peuvent lui être envoyés à partir d’un ordinateur. Il peut se déplacer, afficher des couleurs, capter les obstacles autour de lui et capter la réflectivité du sol.

Boutons

Capteurs de   
proximité frontaux

Capteur au sol

Capteurs de proximité arrières

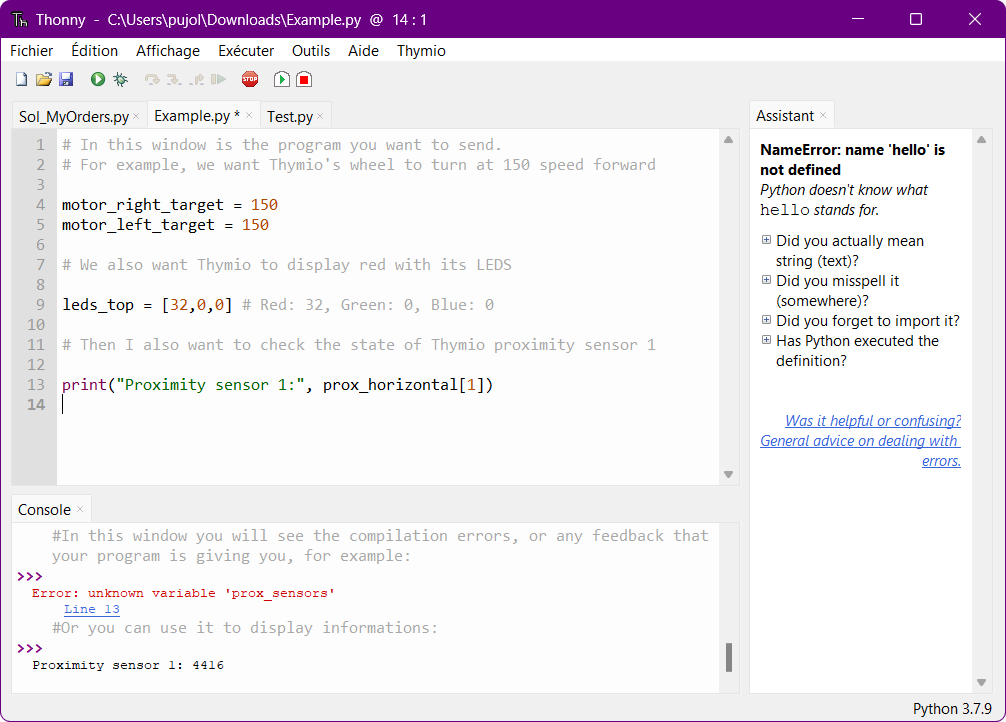
LEDS

***Comment communiquer avec Thymio ?***

En utilisant une clé USB, l’interface Thymio Suite détecte le Thymio connecté. On peut ensuite utiliser différents outils de programmation comme VPL3 ou Scratch pour écrire des programmes et les envoyer à Thymio. Dans notre cas, nous allons utiliser Thonny qui est l’interface Python pour Thymio.

***Comment utiliser Thonny ?***

«

Vous utiliserez Tonny pour les exercices. Les fenêtres principales sont l’éditeur (1) ou vous pouvez modifier votre code avant de l’envoyer à Thymio, et la console (2) ou des feedbacks du compilateur ou de votre programme y sont affichés. Une fois le programme prêt, vous pouvez l’envoyer et l’exécuter avec la flèche vert (3), ou l’arrêter en appuyant le carré rouge (4). Vous pouvez ouvrir de nouveaux fichiers (5), changer (6) entre eux, ou les fermer.

③

④

⑤

⑥

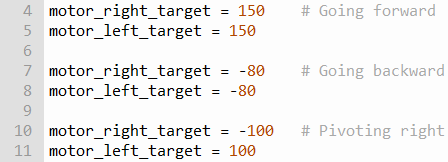
① EDITEUR

② CONSOLE

***Quelles commandes envoyer ?***

Thonny a des variables prédéfinies pour vous aider à manipuler les composants de Thymio.

**Les roues :** pour que Thymio se déplace, vous devez assigner la vitesse de rotation des deux roues (gauche et droite) à une valeur entre -500 et +500. Thymio se déplacera à une vitesse constante, jusqu’a qu’une autre vitesse cible soit envoyée. Ouvrez le fichiermotors.py *pour tester les roues.*



**Connaitre la valeur d’une variable :** vous pouvez utiliser la fonction print() pour afficher la valeur d’une certaine variable. Par exemple vous pouvez connaitre la dernière valeur envoyée à la roue droite en affichant motor\_right\_target pour être sûr que qu’elle est bien fixée à -100.



**LEDS** : pour afficher les LEDS d’une certaine couleur, vous devez choisir une valeur entre 0 et 32 pour la composant rouge, verte et bleue. Ouvre le fichier leds.py *pour essayer.* Par exemple, si je veux que Thymio soit coloré en orange :

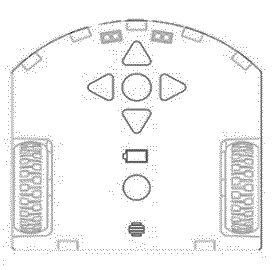


**Boutons** : tant qu’un certain bouton est appuyé, les variables suivantes seront True. Si le bouton n’est pas appuyé, il sera False par défaut. Par exemple, si j’appuie sur la flèche avant et celle de droite :



**Capteurs de proximité :** si un des sept capteurs de proximité détecte un obstacle, la valeur du capteur correspondant (*voir Figure* 1) augmentera de 0 (pas d’obstacle) jusqu’à 4000 (obstacle très proche). Par exemple pour savoir si les murs devant et derrière sont loin vous pouvez afficher les valeurs du second et sixième capteur. Ouvrez le fichierprox.py *afficher les valeurs des capteurs.*





⑤

⑥

④

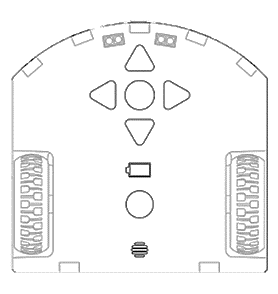
③

②

①

⓪

**Fig. 1 : capteurs de proximité**



⓪

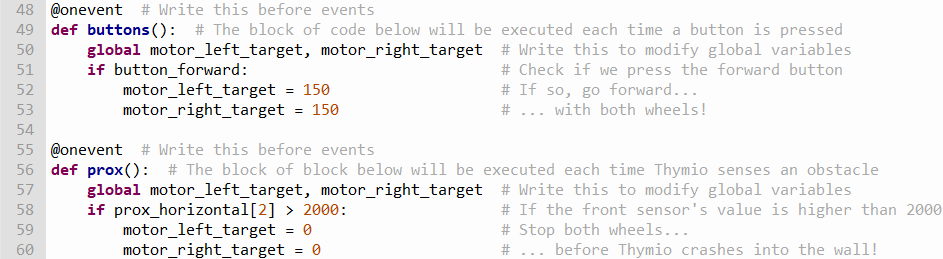
①

**Fig. 2 : capteurs au sol**

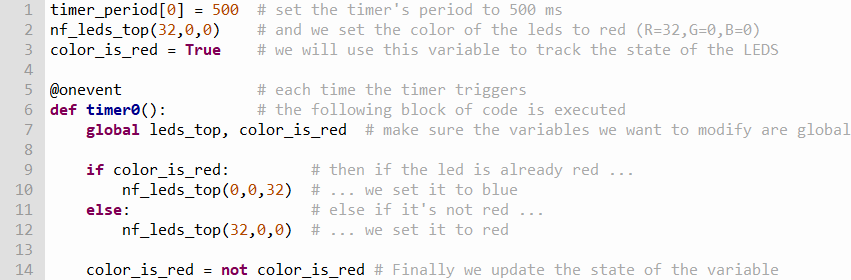
**Capteurs au sol :** Thymio a deux capteurs au sol à l’avant (*voir Figure 2*). Leurs valeurs changent de 0 (sol complétement noir mat) à 1000 (sol complètement blanc et réflecteur).



**Evénements :** Enfin, il est possible de faire réagir Thymio à certain évènements (un obstacle, un bouton, un choc ou un minuteur). Par exemple dans le programme ci-dessous, Thymio avance si on appuie sur la flèche avant, puis s’arrête s’il détecte un obstacle devant. Ouvrez le fichierevents.py *pour essayer le programme*.



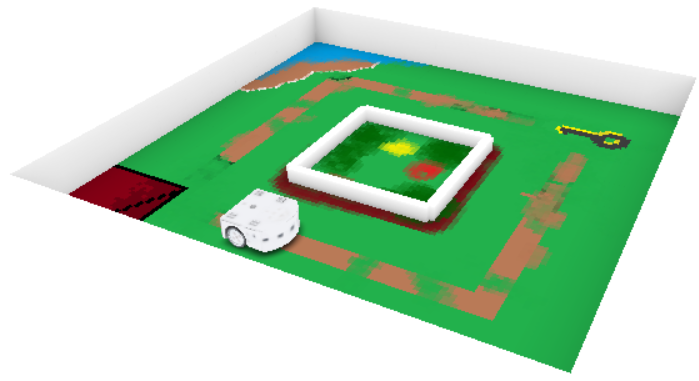
**Les timers :** des minuteurs peuvent être utilisés, et à la fin du minuteur, l’évènement correspondant se déclenche. Pour s’en servir, on doit donner sa période en milliseconde. Le bout de code ci-dessous fait alterner les couleurs de Thymio entre bleu et rouge chaque seconde.



Une fois que vous avez fini d’écrire votre code, vous pouvez envoyer le programme à Thymio en utilisant la flèche verte sur Thonny. Thymio exécutera chaque ligne du code, donc si son comportement ne correspond pas à celui attendu, l’erreur provient surement du programme. Si l’erreur est syntaxique, la console vous le fera surement savoir. Sinon vous devrez revoir la logique du programme que vous envoyez à Thymio.

***EXERCICES (1 heure)***

*Dans cette partie, il vous sera demander de trouver et corriger des bugs dans 5 petits programmes de difficulté croissante. Le but de ces exercices n’est pas que vous trouviez tous les bugs, mais plutôt d’étudier vos comportements et méthodes pour les corriger. Les deux premiers exercices sont pour vous pour vous familiariser avec Thymio. Les deux suivant sont un peu plus challengeant. Le dernier est considéré comme difficile, ne vous inquiétez pas si vous n’arrivez pas à résoudre tous ses bugs.*

Exercice 1 – Autour du jardin (10 min)

* *Ouvrez le dossier* Exercise 1 – Around the Garden

Thymio est enfermé dehors chez lui. Faites-lui suivre un carré parfait autour du jardin pour qu’il puisse récupérer ses clés et revenir à la maison.

Etape 1 (8 min)

* *Ouvrez le fichier* Bug1\_Garden.py

On dirait que Thymio n’arriver pas à atteindre les clés. Pouvez-vous deviner ce qu’il ne va pas dans les instructions qu’on lui donne ? Il n’y a qu’une erreur.

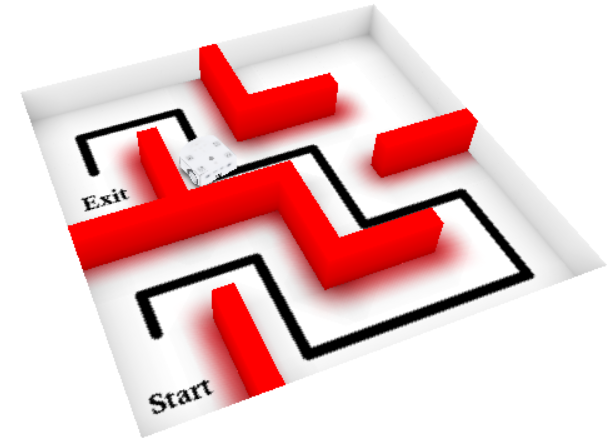
Etape 2 (2min)

* *Ouvrez le fichier* Bug2\_Garden.py

On dirait que Thymio a des problèmes pour s’orienter. Devinez-vous ou on a fait l’erreur ? Il n’y en a qu’une seule.

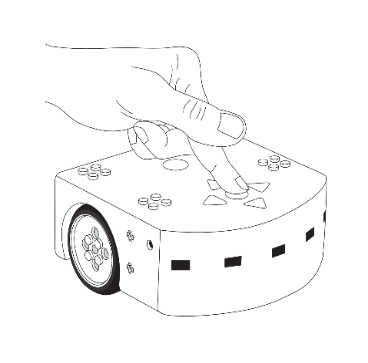
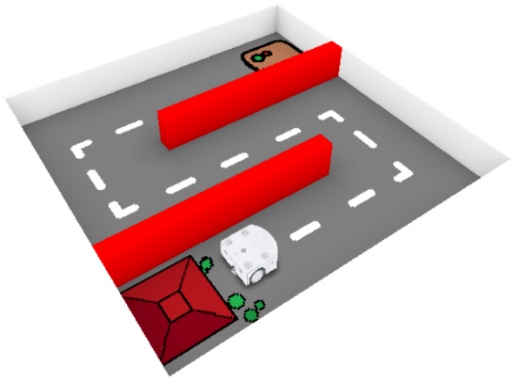
Exercice 2 –Le labyrinthe (5 min)

* *Ouvrez le dossier* Exercise 2 – The labyrinth *et le fichier* Bug\_TheLabyrinth.py

Thymio est perdu dans le labyrinthe. Vous savez ou est la sortie et vous lui donnez les instructions pour qu’il y parvienne. Cependant il y a des bugs dans le code. Arriverez-vous à les trouver ? Il y en a deux.

Exercice 3 – Suis mes ordres (6 min)

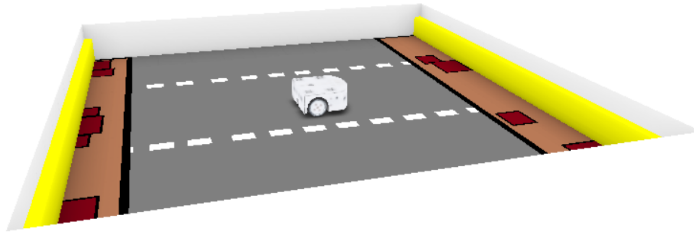
* *Ouvrez le dossier* Exercise 3 – Follow my orders *et le fichier* Bug\_MyOrders.py

Thymio est en retard pour son premier jour de travail, mais il ne sait pas où est le laboratoire. Vous devez le guider en utilisant les flèches sur sa tête. Il est enfin prêt à partir mais il semble que la communication entre vous n’est pas claire, et Thymio a du mal à comprendre vos instructions. Qu’est-ce qui ne va pas ? Il y a deux erreurs.

Exercice 4 – Le transporteur (9 min)

* *Ouvrez le dossier* Exercise 4 – The transporter *et le fichier* Bug\_TheTransporter.py

Vous devez transporter des bouts de bois d’un côté à l’autre de la route, et Thymio est là pour vous aider. Thymio avance et recul de manière périodique, et utilise ses capteurs pour se déplacer. Lorsqu’il est assez proche du mur de devant, il recul, et vice versa. Enfin… c’est ce qu’il devrait se passer si le code était écrit correctement. Pouvez-vous trouver les deux bugs qui s’y cachent ?



****Exercice 5 – Le test ultime (30 min)****

* *Ouvrez le dossier* Exercise 5 – The ultimate test *et le fichier* Bug\_FinalTest.py

Thymio s’est inscrit à l’évaluation robotique la plus prestigieuse, et a besoin de votre aide. Pour cette ultime tâche, Thymio devra utiliser ses capteurs de proximité, ses capteurs au sol, et ses boutons sur la tête.

Le parcours est divisé en trois parties. Pour la première, Thymio doit s’orienter en suivant les murs. Pour la seconde, il doit suivre la ligne noire au sol. Enfin vous devrez appuyer sur le bouton central de Thymio pour le faire s’arrêter sur la place verte à la fin de la ligne noire.

Malheureusement, il y quelques bugs dans le code. Le comportement de Thymio ne correspond pas à celui attendu. Thymio doit être prêt dans 30 minutes. Pouvez vous trouver et corriger les bugs ? Il y en a quatre au total.

