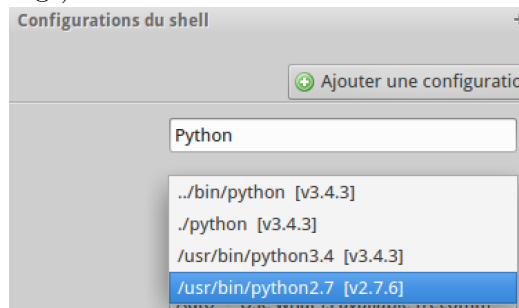


Modélisation et robotique

Exercices 7

1 Programmer Thymio

Question A. Préparatifs. Ouvrir la machine virtuelle, double cliquer sur Mise À Jour. Connecter le Thymio en USB. Il faut que la machine virtuelle ait accès en USB au Thymio. Pour cela, le mieux est de créer une règle de capture de l'USB dans la configuration Virtual Box de votre machine virtuelle (Ports/USB/ajouter). Faites vous aider. Dans la machine virtuelle, lancer la commande `asebamedulla "ser:device=/dev/ttyACM0"`. Votre Thymio est maintenant connecté et prêt à communiquer par programme avec votre machine virtuelle. Dans Pyzo, choisissez un shell Python 2.7 (c'est dans les Paramètres avancés/Advanced settings).



Question B. Premiers pas. Pour votre premier programme avec Thymio, nous allons exploiter le capteur de température. Vous pouvez en lire plus sur les capteurs sur cette page : <https://www.thymio.org/fr:thymioapi>

Les évènements n'iront que dans un sens, du Thymio vers votre programme. Créer un nouveau programme Python.

Pour communiquer avec le Thymio, il vous faut importer `pythymio` (`import pythymio as pt`), et créer un bloc `with pt.thymio(evts1) as Thym:` où `evts1` est une liste d'événements internes au Thymio que l'on souhaite écouter. Pour cet exercice, choisissez juste `["temperature"]`. Dans cette instruction, `Thym` est un surnom choisi librement pour parler du Thymio. À l'intérieur de ce bloc il faut créer une fonction qui traitera les événements en provenance du Thymio. Appelons cette fonction `progression`. Nous attacherons ensuite cette fonction à la boucle événementielle du Thymio en utilisant l'instruction : `Thym.loop(progression)`.

La fonction `progression` va être appelée à chaque événement du Thymio jusqu'à ce qu'on dé-

cide d'arrêter le programme avec Ctrl-C.

Cette fonction `progression` recevra en entrée trois arguments : un identifiant d'événement (que nous n'utiliserons pas), le nom de l'événement préfixé par `fwd.`, et une liste contenant les arguments de l'événement. Dans le cas de l'événement `fwd.temperature` nous recevrons une liste avec une seule valeur, la température actuelle en degrés Kelvin. Cet événement est déclenché à une fréquence fixe de 1Hz, ce qui est parfait pour effectuer un affichage. Affichez la température lue par le Thymio à chaque réception de l'événement `fwd.temperature`.

Question C. Filtrer par événement. La liste `evts1` ne contient qu'un seul événement. Que se passe t'il si vous ajoutez à cette liste d'autres événements, par exemple `button.center`? Faites en sorte que seul l'événement `fwd.temperature` déclenche un affichage. Affichez également "bouton du milieu" à chaque fois que votre progression reçoit l'événement `fwd.button.center`. Quel est le problème avec l'événement `button.center`?

Question D. Accéléromètre 1. Modifier votre programme précédent pour n'écouter que l'événement interne `acc` avec lequel nous recevrons la liste des trois valeurs lues par l'accéléromètre. Cet événement est déclenché à une fréquence fixe de 16Hz. C'est un peu trop rapide pour un affichage. Nous allons donc utiliser un compte à rebours pour ne faire l'affichage des valeurs que toutes les secondes. Il vous faudra utiliser un compteur `delay` de façon à attendre 16 fois que l'événement ait été déclenché avant un nouvel affichage.

Vous aurez besoin d'une variable globale, que la fonction `progression` devra pouvoir modifier. Vous rappelez vous comment faire?

Comme cette situation est appelée à se reproduire, plutôt que de multiplier les variables globales, créez en une seule qui contiendra tout l'état du programme entre deux appels à `progression`. Pour cela créez un dictionnaire `state` et utilisez des clés différentes pour les différentes valeurs que vous voulez stocker. Par exemple, `state["delay"]` pourra contenir un entier valant initialement 15, décrémenté à chaque événement `fwd.acc` et qui,

lorsqu'il atteindra zéro, déclenchera l'affichage puis se remettra à 15.

Question E. Gravité. Quelle est la valeur de l'attraction terrestre selon Thymio ?

Question F. Bouton(s) quitter. L'événement interne `buttons` a une fréquence de 100Hz. Attention, il y a de quoi bien occuper votre machine virtuelle si vous faites un affichage à chaque réception. Cet événement retourne une liste de 5 valeurs, 0 ou 1, qui encode le fait que les boutons sont pressés ou non. Utilisez la technique du délai employée précédemment pour ne faire un affichage que tous les 80 centièmes de seconde. Trouvez à quoi correspond chaque bouton, puis faites en sorte de quitter la boucle événementielle avec `Thym.stop()` lorsque on appuie simultanément sur le bouton gauche et le bouton droite.

Question G. Simon. Le jeu Simon consiste en observer une séquence de boutons tirée au hasard puis reproduire la même séquence. La difficulté progresse en augmentant la longueur de la séquence. Le but du jeu est d'atteindre la plus haute difficulté possible. Pour que le jeu soit agréable la séquence de bouton s'accompagne de sons et de couleurs, un son et une couleur différente par bouton. Le bouton central n'entrera pas dans la composition des séquences. Commencez par écrire une fonction qui prend en entrée un entier et retourne une séquence de lettres tirées parmi U, D, L et R (up/haut, down/bas, left/gauche, right/droite) de longueur l'entier reçu en paramètre. Testez votre fonction sur un programme ne nécessitant pas Thymio.

Question H. Simon 2. Vous allez utiliser Thymio pour montrer une séquence de boutons, en allumant des leds en direction des boutons (haut, bas, gauche, droite). Pour cela vous aurez besoin de créer des événements qui iront du programme au Thymio. Les événements qui vous intéressent sont : `circle.right`, `circle.left`, `circle.front`, `circle.back` et `circle.off`. Pour envoyer un événement au Thymio, par exemple `circle.off`, vous pouvez utiliser `Thym.send_event("circle.off")`. Vous pouvez ajouter du son et de la couleur à l'aide des événements : `chord.C3`, `chord.E3`, `chord.G3`, `chord.B3`, `sound.bad`, `sound.good`, `become.yellow`, `become.violet`,

`become.lightblue`, `become.green`, `become.blue`, `become.red`, `become.blank`.

Question I. Simon 3. Une fois que votre programme a choisi une séquence et l'a joué à une personne, cette personne doit reproduire la séquence à l'identique (sans toutefois en respecter le rythme). Cette personne peut se tromper ou réussir, issue qui ajustera la difficulté pour la fois suivante, après quoi le programme se met en attente. Avant de commencer la lecture d'une nouvelle séquence le programme attend que la personne appuie sur le bouton central. Imaginez un automate à cinq états "ready", "read", "compare", "won", "lost" et les transitions que vous aurez à faire entre ces états. Pensez à introduire des délais entre les étapes du programme. Complétez le programme.

Question J. Variante. Plutôt que de tirer une nouvelle séquence à chaque succès, rallongez la séquence précédente.

Question K. Accéléromètre 2. Utilisez Thymio comme télécommande pour vous déplacer dans gardenworld. Tenir Thymio droit vous fera avancer d'une case par seconde, le basculer vers l'avant le stoppera, le tourner comme un volant un peu incliné vers la gauche ou la droite le fera tourner dans la direction correspondante.

Question L. Code Konami 1 Écrire un programme qui allume tour à tour les leds du cercle autour des boutons pour afficher un code composé des flèches gauche, droite, haut et bas à l'utilisateur. Un code classique est haut, haut, bas, bas, gauche, droite, gauche, droite. Allumer chaque led pendant une seconde, et éteindre toutes les leds pendant 0,5s entre chaque indication. Vous utiliserez les événements `circle.right`, `circle.left`, `circle.front`, `circle.back` et `circle.off`.

Question M. Code Konami 2 Écrire un programme qui attend que l'utilisateur tape exactement le code choisi précédemment pour rendre le Thymio vert et signaler une réussite. Attention vous devrez faire en sorte que dès que la séquence saisie par l'utilisateur se termine par le code complet, le Thymio devient vert. Par exemple si le code était H, H, B, B, G, D, G, D, la séquence H, H, G, D, G, H, H, B, B, G, D, G, D, réussit.