Apprendre à programmer Ergo Jr en Snap!

Partie 1 : Contrôler Poppy Ergo Jr

Vous allez apprendre à faire bouger le robot Poppy Ergo Jr en utilisant le langage de programmation Snap!. A la fin de la séance vous pourrez appliquer vos connaissances avec le défi *Ergo Jr joue au chamboule-tout*.

Un **langage de programmation** permet d'écrire un programme informatique, qui est une suite d'instructions à exécuter. Ceci permet de donner des comportements à un robot.

Snap! est un langage de programmation avec lequel on assemble des blocs d'instructions. Un assemblage de blocs s'appelle un script.

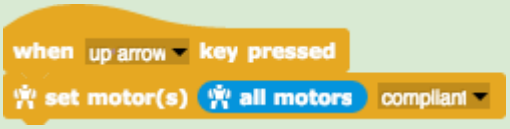
1. Votre premier programme

Pour trouver des blocs dans Snap!, vous pouvez chercher :

- Par couleur/catégorie (chaque catégorie a une couleur)
- Par mots clés (⇒ *Clic droit* sur la partie de gauche ⇒ *find blocks*) et saisir :
  - Un mot se trouvant dans le bloc souhaité (exemple : *when*)
  - Le mot clé *robot* afin de séléctionner seulement les blocs spécifiques au robot Ergo Jr

A vous de jouer !

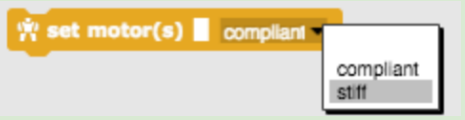
1. Créez les deux scripts ci-dessous afin de pouvoir mettre Ergo Jr dans des positions spécifiques :



Pour cela :

a. Sélectionnez et déposez les trois blocs dont vous avez besoin sur l'espace de travail central :

b. Cliquez sur les listes et choisissez les bonnes valeurs :

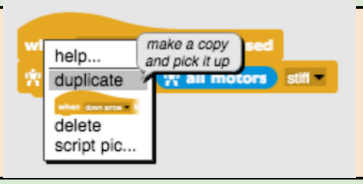


c. Assemblez les blocs ensemble :

Pour emboiter des blocs : sélectionnez le bloc et déposez-le à l'endroit désiré avec la souris. La bordure blanche indique que les blocs vont s'emboiter.

d. Faites de même pour le deuxième script.

Vous pouvez copier/coller les blocs et scripts : *Clic droit* puis *duplicate*



2. Activez les deux scripts (une bordure blanche apparaît autour d'un script activé) :

a. Appuyez sur ⇩ *flèche* du bas du clavier puis manipulez le robot. Que remarque-t-on ?

b. Appuyez sur ⇧ *flèche du haut* du clavier puis manipulez le robot. Que remarque-t-on ?

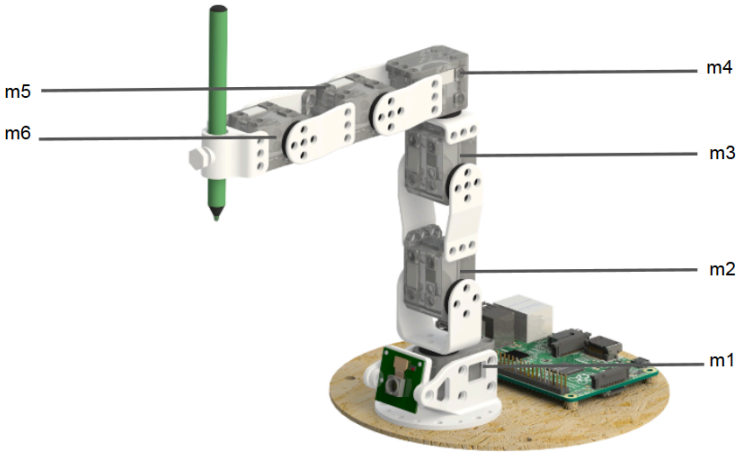
3. Qu'est-ce que le mode *compliant* du robot ? Et le mode *stiff* ?

4. Activez ces scripts et manipulez Ergo Jr pour le mettre dans différentes positions.
Des idées : donnez-lui l'air curieux, timide, content. A vous d'imaginer !

2. Faire bouger Ergo Jr à l'aide de ses moteurs

On utilise le bloc suivant pour faire bouger Ergo Jr, moteur par moteur

Voici le schéma du robot avec le nom de chacun des moteurs :



Il est **nécessaire** d'activer les moteurs (**stiff**) du robot avant de pouvoir le **faire bouger** avec Snap!

A vous de jouer !

1. **Assurez-vous que tous les moteurs soient bien activés (mode stiff)** et mettez tous les moteurs en position de base (qui correspond à la position où chaque moteur est à 0 degrés : aligné sur l'encoche) en cliquant sur le bloc suivant pour l'exécuter :

Ici on utilise le bloc 'set position(s)' : il accepte des valeurs de positions en degrés, il est conseillé de respecter un intervalle de [-90 ; 90], suivi du/des nom(s) du/des moteur(s), puis de la durée en secondes qu'il va mettre pour atteindre cette position. Concernant, la valeur *wait?* nous verrons son

utilisation plus tard.

2. Mettez le moteur *m1* dans la position 90 degrés en 2 secondes.

3. Cherchez les blocs ci-dessous et exécutez-les chacun leur tour :

a. Quelles valeurs ces blocs renvoient-ils ?

Les blocs ont des formes différentes, chaque forme correspond à une catégorie spécifique.

- Les blocs de forme ovale (comme 

) sont appelés *reporter* : quand il sont exécutés, ils renvoient une valeur.
- Au sommet du script peut se trouver un bloc *Hat* (chapeau), qui indique quand le scénario doit être exécutés. Les noms de blocs *Hat* commencent généralement par le mot *When* (exemple : 

); un script n'a pas obligatoirement un bloc *Hat*, mais sans ce bloc, le script sera exécuté seulement si l'utilisateur clique sur le script lui-même.
- Les blocs *command* (comme 

) correspondent à une action.

b. Modifiez le bloc *set position* pour mettre le moteur *m1* et le moteur *m6* dans la position -30 degrés en 2 secondes.

4. En vous aidant des blocs que nous venons de découvrir, construisez deux programmes correspondant aux instructions ci-dessous :

a. Quand  *fleche de droite* est pressée alors mettre tous les moteurs en position 0 degrés en 3 secondes.

b. Quand  *fleche de gauche* est pressée alors mettre les moteurs m1 et m4 en position 60 degrés en 2 secondes.

3. Créer des mouvements

Maintenant, nous allons faire bouger les moteurs pour créer des mouvements.

A vous de jouer !

1. Exécutez le script ci-dessous et observez ce qui se passe :

2. Remplacez **le deuxième** bloc `true` par le bloc `false`

   a. Lorsqu'il y a deux blocs (ou plus) d'emboités, dans quel ordre s'effectuent les actions ?

   b. Que se passe-t-il quand *wait* est égal à *true* ? Que se passe-t-il quand *wait* est égal à *false* ?

> Les lignes de code s'exécutent de façon quasi instantanées ; et même si parfois la position demandée dans la ligne précédente n'a pas été atteinte.
>
> La partie *wait* permet d'attendre que le moteur ait atteint la position voulue avant d'exécuter la commande suivante.

3. Avec les blocs que vous connaissez maintenant, faites jouer un mouvement à Ergo Jr signifiant *bonjour* quand on appuie sur la touche *b*.

Conseils :

- Choisissez les moteurs que vous souhaitez utiliser pour la création du mouvement.
- Faites jouer le mouvement choisi au robot (**en mode compliant**) et observez les actions de chaque moteur.
- Vous pouvez vous aider du bloc 

pour connaitre la position d'un moteur ciblé, et ainsi noter la valeur pour la réutiliser

ensuite.

- Programmez le mouvement moteur par moteur et testez à chaque fois le résultat de votre programme.

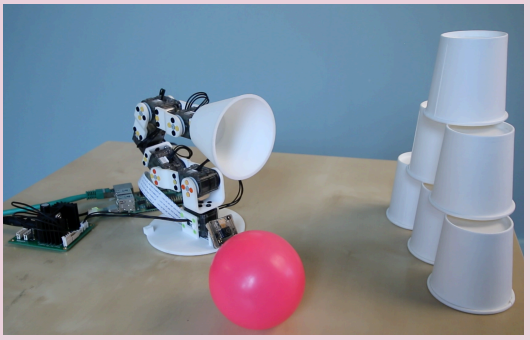
⇒ il est conseillé de commencer par un mouvement simple puis de l'enrichir au fur et à mesure.

N'hésitez pas à créer d'autres mouvements !

Défi robotique : Ergo-Jr joue au chamboule-tout

Matériel :

- Poppy Ergo Jr avec l'abat jour
- une balle légère
- des verres (carton ou plastique) pour faire un chamboule-tout



Objectif :

Contrôler la position et la vitesse des moteurs du robot pour lancer la balle et faire tomber le chamboule-tout.

Il y a de nombreuses manières possibles de lancer la balle. Combien pouvez-vous en trouver ?