

Didacticiel de prise en main de Snap!

Part 1: Contrôler le robot Ergo

Durant cette séance, nous allons apprendre à faire bouger notre robot Ergo en utilisant le langage de programmation Snap! et coder notre premier programme permettant de lui donner des consignes en utilisant les touches du clavier.

Pour communiquer avec le robot Ergo, on utilise un langage spécifique composé d'un ensemble de signes et symboles, qu'on appelle "langage de programmation".

En Snap! un programme est composé de plusieurs scripts. Un script est un ensemble de blocs qu'on emboîte ensemble afin de donner des instructions.

A. Commencer sa première application

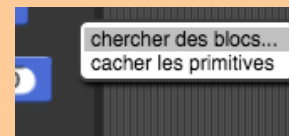
Pour chercher des blocs dans Snap!, vous pouvez effectuer une recherche :

- par catégorie (chaque catégorie a une couleur)



- par mots clés (☐ Clic droit sur la boîte à outil de gauche ☐ chercher des blocs) :

- ☐ vous pouvez utiliser un mot qui se trouve dans le bloc que vous voulez (exemple : "quand")
- ☐ Vous pouvez utiliser le mot clé "robot" afin de sélectionner seulement les blocs spécifiques aux robots Poppy



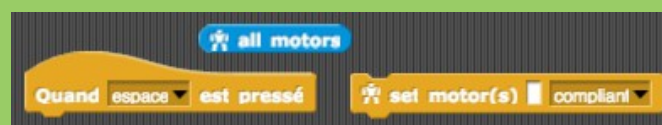
A vous de faire !

1. Vous allez commencer par créer les deux scripts ci-dessous afin de pouvoir mettre Poppy dans des positions spécifiques :

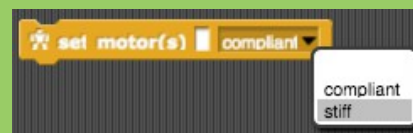


Vous pouvez suivre les étapes ci-dessous :

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 emboîter des blocs ensemble : sélectionnez le bloc et déposez-le à l'endroit désiré. La bordure blanche indique que les blocs vont s'emboîter si vous relâchez le clique de la souris.



d. Faites de même pour le deuxième script

Vous pouvez copier/coller les blocs et scripts : ☐ clic droit sur le bloc à copier ☐ "dupliquer"



2. Activez les programmes :

a. en pressant ☐ (la flèche du haut du clavier)

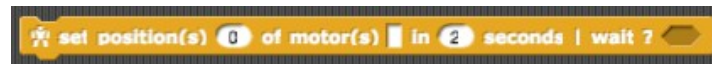
b. puis ☐ (la flèche du bas du clavier)

3. Qu'est ce que cela signifie lorsque l'on dit qu'un robot est en mode "compliant" et en mode "Stiff" ?

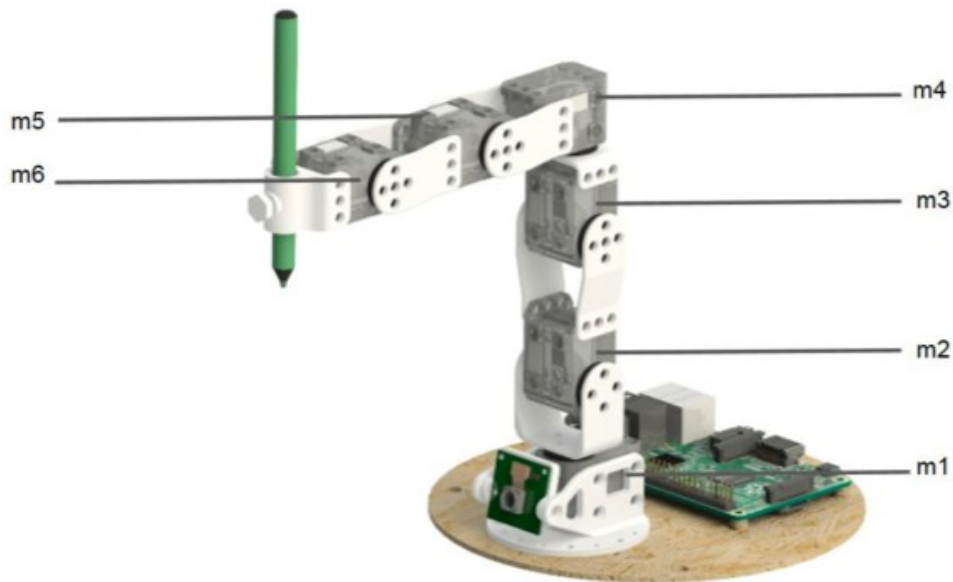
4. Activez plusieurs fois ces scripts et manipulez Poppy manuellement pour le mettre dans différentes positions.

B. Faire bouger les moteurs de Poppy

Ici on va apprendre à faire bouger Poppy, moteur par moteur, en utilisant le bloc



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 les faire bouger avec Snap!

À vous de faire !

1. Assurez-vous que tous les moteurs sont bien activés (Stiff) et mettez tous les moteurs en position de base (qui est la position où chaque moteur est à 0 degrés) en cliquant sur ce bloc avec votre souris pour l'exécuter :

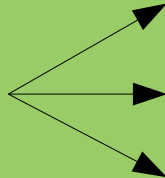


Ici on utilise le bloc 'set position(s)' : il accepte des valeurs de positions en degrés, il est conseillé de respecter un intervalle de -90 ; 90, suivi du/des nom(s) du/des moteur(s), puis de la durée en secondes qu'il va mettre pour atteindre cette position. Concernant, la valeur "Wait!" nous verrons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

2. Maintenant mettez le moteur "m1" dans la position 30 degrés en 2 secondes.




3. Cherchez les blocs ci-dessous et exécutez-les chacun leur tour à la place de « m1 » dans le bloc précédent (pensez à remettre tous les moteurs à 0° avant de tester un nouveau bloc) :




a. Quelles valeurs ces blocs rapportent-ils ?

Les blocs ont des formes différentes, chaque forme correspond à une catégorie spécifique. Nous en avons vu trois :

☐ Les blocs de forme ovale (comme ) sont appelés "reporter" ; quand ils sont exécutés, ils rapportent une valeur.

☐ Au sommet du script peut se trouver un bloc "Hat" (chapeau), qui indique quand le scénario doit être effectué.


Les noms de blocs "Hat" commencent généralement par le mot "Quand" (ex : );

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  (la fleche de droite) est pressée alors mettre tous les moteurs en position 0 degrés en 3 secondes.

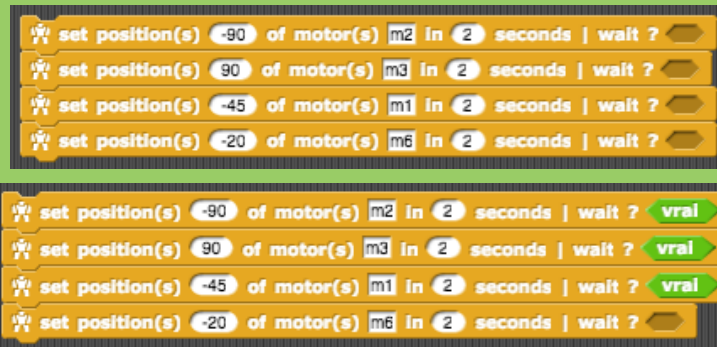
b. Quand  (la fleche de gauche) est pressée alors mettre les moteurs m1 et m4 en position 60 degrés en 2 secondes.

C. Créer des mouvements

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

À vous de faire !

1. Exécutez les deux programmes ci-dessous et observez ce qui se passe (pensez à remettre les moteurs à 0° entre les deux tests) :



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

b. Que se passe-t-il quand "wait" est égal à "Vrai" ? Que se passe-t-il quand "wait" est vide ou égal à "Faux" ?

Les lignes de code s'exécutent de façon quasi instantanées ; parfois, même si la position demandée dans la ligne précédente n'a pas été atteinte.

Le bloc "Wait" permet d'attendre que le moteur ait atteint la position voulu avant d'exécuter la commande suivante.

2. Avec le bloc que nous venons d'explorer, réaliser un programme pour que le robot Poppy s'approche du porte-conteneur et faites en sorte qu'il s'active lorsque la touche "a" est pressée.

Conseils :

- ☐ Choisissez les moteurs que vous souhaitez utiliser pour la création du mouvement.
- ☐ Faites faire le mouvement choisi au robot (en mode compliant) et observez le mouvement de chaque moteur.
- ☐ Vous pouvez vous aider du bloc ci-dessous pour connaître 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 le complexifier au fur et à mesure.

Vous pouvez ensuite créer le mouvement de saisie du conteneur et du dépôt dans la zone de stockage.