

Prise en mains de Snap!

Unit 1 : contrôler Poppy avec les touches de son clavier

1. Commencer son premier programme en Snap!

Vous allez coder un programme permettant de contrôler le robot Poppy avec les touches de votre clavier d'ordinateur.

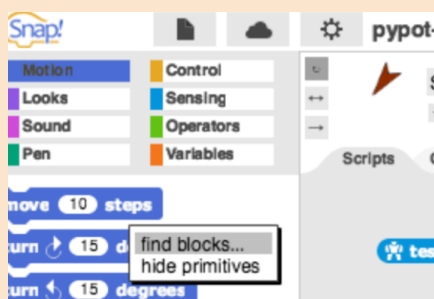
Pour communiquer avec le robot Poppy on utilise un langage spécifique composé d'un ensemble de signes et symboles, qu'on appelle "langage de programmation".

Les mots clés utilisés sont souvent des mots anglais mais il n'est pas nécessaire de maîtriser cette langue. Voyez plutôt cela comme un langage codé. N'hésitez pas à regarder la traduction en français, dans un dictionnaire, pour vous aider à comprendre les mots clés.

A vous de faire !

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

- par catégorie (chaque catégorie à une couleur et une forme spécifique)
- par nom du bloc (⇒ Clic droit dans la partie de gauche ⇒ find blocks). Tapez "robot" pour découvrir les blocs spécifiques aux robots Poppy.



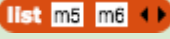
Puis, cliquez sur le bloc (ou l'ensemble de blocs) pour exécuter le programme.

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

 all motors

 all motors groups

motors in group base

- Quelles valeurs ces blocs rapportent-elles ? Quelle est la différence entre les trois blocs ?
- Testez le bloc  , que permet de faire ce bloc ?

2. Mettez Poppy en position “Curieux”

- Observez les deux programmes ci-dessous et repérez les différences (les deux programmes se ressemblent mais les valeurs sont différentes)
- Construisez les deux programmes :

Pour emboîter des blocs ensembles : 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.

Vous pouvez copier/coller les blocs : ⇒ click droit ⇒ duplicate

when up arrow key pressed

 set motor(s)  all motors compliant

when down arrow key pressed

 set motor(s)  all motors stiff

- Pressez  (la flèche du haut du clavier) puis, manuellement, manipulez le robot pour le mettre en mode “curieux” (à vous d’être imaginatif et d’inventer la position “curieux” !).
- Pressez  (la flèche du bas du clavier). Que se passe-t-il ? (essayer de manipuler le robot)

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

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

- Les blocs en forme oval (comme ) sont appelés “reporter” ; quand il est exécuté il rapporte une valeur.

- Au sommet du script se trouve un bloc “Hat” (chapeau), qui indique quand le scénario doit être effectué. Les noms de bloc “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.

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 les moteurs de la base en mode Stiff.
- b. Quand  (la fleche de gauche) est pressée alors mettre les moteurs m1 et m6 en mode Compliant.

2. Faire bouger le robot

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

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

A vous de faire !

1. Assurez-vous que les moteurs sont bien activés (Stiff) et exécutez ce bloc :



set position(s) 0 of motor(s) all motors in 2 seconds | wait ?

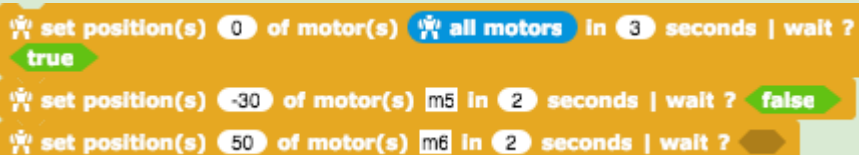
Ici on utilise la fonction 'set position(s)' (Elle accepte des valeurs de positions allant de -180° à +180°), suivi du/des nom(s) du/des moteur(s), puis de la durée en secondes.

2. Modifiez les différentes valeurs du bloc pour que le moteur M1 se mette en position 80 degrés en 3 secondes.

3. Exécutez les deux programmes ci-dessous et observez ce qui se passe :



```
set position(s) 0 of motor(s) all motors in 3 seconds | wait ? true
set position(s) -30 of motor(s) m5 in 2 seconds | wait ? true
set position(s) 50 of motor(s) m6 in 2 seconds | wait ?
```



```
set position(s) 0 of motor(s) all motors in 3 seconds | wait ? true
set position(s) -30 of motor(s) m5 in 2 seconds | wait ? false
set position(s) 50 of motor(s) m6 in 2 seconds | wait ?
```

4. Lorsqu'il y a deux blocs ou plus d'emboîtés, dans quel ordre s'effectuent les actions ? Que se passe-t-il quand "wait" est égal à "True" ?

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.

5. Avec le bloc que nous venons d'explorer : faites faire un mouvement au robot Poppy signifiant "bonjour" (tout est possible, c'est un robot, c'est à vous de déterminer le code robotique de politesse) et faites en sorte qu'il s'active lorsque la touche "a" est pressée. Pour cela, nous vous conseillons de suivre les étapes suivantes :

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

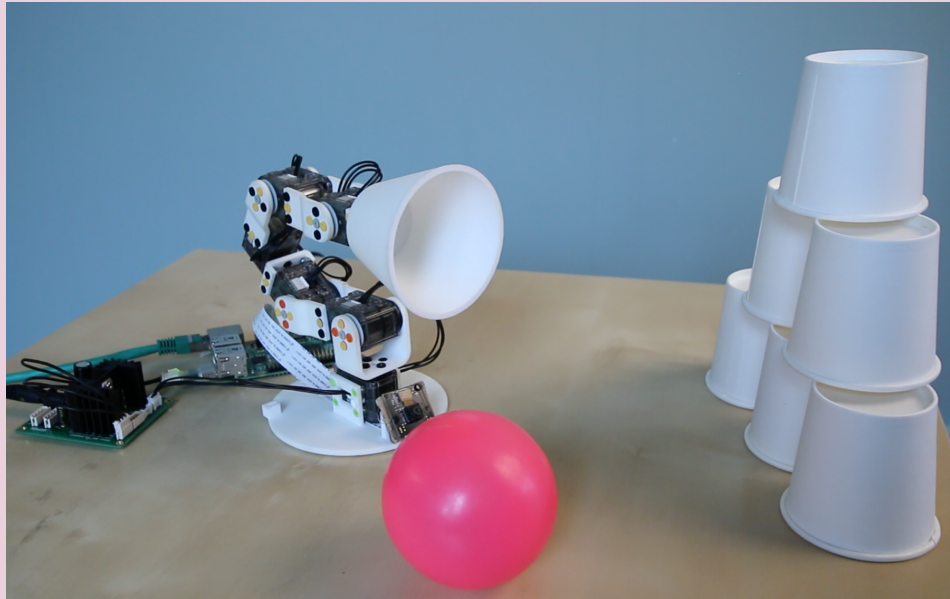
Défi robotique : Poppy chamboule-tout

Matériel :

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

Objectif :

- contrôler la position et la vitesse des moteurs du robot pour lancer la balle et faire tomber le chamboule-tout. Vous pouvez explorer plusieurs manières de faire !
- il y a de nombreuses manières possibles de lancer la balle. Combien pourriez-vous en trouver ?



3. Utilisez le bloc aléatoire pour faire danser Poppy

Nous allons voir comment créer des mouvements aléatoires et faire danser Poppy.

A vous de faire

1. Cherchez le bloc  et exécutez-le plusieurs fois pour l'essayer.
2. Quelle est la valeur que le bloc "random" rapporte ?
3. Modifiez le bloc  pour qu'il rapporte une valeur entre -80 et 80
4. Programmez Poppy Ergo pour qu'il bouge :
 - a. le moteur "m1" (pour Poppy Torso : l_elbow_y) à une position "random" avec une intervalle de -80 ; 80
 - b. puis le moteur "m3" : déterminez l'intervalle des positions où le moteur peut bouger sans que ça pose problème (vous pouvez vous aider du bloc )
 - c. répétez ce mouvement (le moteur "m1" à une position random puis le moteur m2 à une position random) trois fois successivement.
4. Faites la même chose mais en utilisant le bloc . Quel est l'intérêt de ce bloc ?

Défi robotique : Poppy l'otarie

Matériel :

- une balle légère
- poppy ergo avec l'abat jour

Objectif :

Mettez une balle dans l'abat jour et faites faire à Poppy une danse aléatoire. Bien sûr, sans faire tomber la balle !

Utilisez au moins 3 moteurs de votre choix. Vous pouvez augmenter la difficulté en augmentant le nombre de moteurs utilisés.

Vous pouvez suivre ses étapes :

- a. choisissez les moteurs que vous souhaitez utiliser pour la création de la danse
- b. pour chaque moteur déterminez les positions randoms que vous souhaitez utiliser : vérifiez l'intervalle des positions où le moteur peut bouger sans que ça pose problème (faites particulièrement attention au moteur m3)
- c. faites bouger les moteurs **les uns après les autres** (wait? = true) dans les positions randoms que vous avez choisis et testez
- d. répétez le programme

Pour aller plus loin...

Temps libre avec le robot, avec ce que vous avez appris, innovez !