

Terminologie sur la programmation

Pour ce livret, il s'agit de préciser les sens donnés à certains termes dans l'environnement de Scratch.

- **Bloc** : terme de Scratch. Ce sont les formes géométriques glissées dans la zone de code susceptibles de déclencher une action en cliquant dessus.
- **Programme** : représente l'ensemble des blocs et variables présents dans la zone de code centrale. Cela correspond aux éléments qui seront sauvegardés par le menu Fichier / Sauvegarder sur votre ordinateur.
- **Script** : ensemble des blocs collés entre eux, ou sous un bloc chapeau comme l'illustre la figure 1. Un script est donc l'équivalent d'une fonction gestionnaire d'événements telle que l'on peut en rencontrer dans d'autres langages. Un programme est donc composé d'un ou plusieurs scripts.
- **Code** : terme général pouvant désigner dans Scratch une association quelconque de blocs.
- **Sous-programme** : il s'agit d'un bloc de couleur rouge appartenant à la rubrique Mes Blocs (voir figure 2). L'utilisateur peut ajouter des blocs normaux sous un tel bloc. Dans le programme principal on utilisera la forme (voir figure 3) pour déclencher l'exécution de ce bloc.
- **Programmation séquentielle** : les blocs sont exécutés les uns à la suite des autres, en tenant compte des test Si. et des boucles. Un bloc donné n'est exécuté que lorsque le bloc précédent a terminé son exécution. C'est la forme la plus classique de programmation, illustrée ici par l'exécution d'un script unique.
- **Programmation événementielle** : dans ce cas, une boucle, invisible pour l'utilisateur scrute en permanence les événements susceptibles d'arriver : frappe du clavier, action sur la souris, arrivée des mesures des capteurs de Thymio Si le programmeur le désire, il peut prendre en compte, avec les blocs chapeau par exemple, un événement et déclencher l'exécution d'un script en réponse à cet événement.
- **Programmation concurrente** : dans ce type de programmation, plusieurs scripts peuvent s'exécuter simultanément. C'est le mode de fonctionnement naturel d'un robot. Par exemple, un robot doit pouvoir faire démarrer ses moteurs, les accélérer ou les arrêter tout en jouant de la musique ou en changeant les couleurs de ses LEDs.

La programmation événementielle génère de manière naturelle une programmation concurrente.

(figure 1)



(figure 2)



(figure 3)

