

Structurer les connaissances

Ecrire, mettre au point (tester, corriger) et exécuter un programme

CYCLE 4

>

» Début de cycle
» Milieu de cycle
» Fin de cycle

Lors de la **conception** d'un **système programmable**, le concepteur va devoir **écrire un programme** pour commander le système **en fonction du comportement attendu**.

- Le concepteur écrit un **algorithme à partir du cahier des charges...** Puis, il crée **des programmes** à l'aide d'un **logiciel de programmation par blocs** pour **commander le système réel**.



Exemple : un robot évitant un obstacle

Si la distance détectée par le robot est inférieure à 10 cm, alors le robot devra s'arrêter, sinon il continue d'avancer.



Logiciel : Scratch



Logiciel : mBlock

Ces **logiciels** possèdent des **menus**, où sont stockés des **instructions**, qui permettront de **concevoir un programme** et enfin **vérifier le comportement attendu** en l'exécutant.



- Comment écrire, mettre au point un programme pour vérifier le comportement attendu ?**

Pour écrire un programme, plusieurs étapes sont nécessaires :

- vérifier les attentes du cahier des charges** : tenir compte des **fonctions de service**.
- rédigé un algorithme** : **des phrases** qui permettent de respecter **les fonctions de service demandées**.
- repérer les capteurs** et les **actionneurs** à utiliser
- repérer les instructions** sur le **logiciel** de programmation
- mettre au point** le programme : **assembler les instructions**.
- exécuter le programme**
- vérifier le comportement attendu** : Si cela ne correspond pas au comportement attendu. Il faut corriger le programme.
- adapter le programme** pour obtenir le **comportement attendu**

Pour **écrire, mettre au point, exécuter un programme** commandant un **système réel** programmable et **vérifier le comportement attendu**, il faut donner des **instructions** à ce système, en utilisant des **langages** compréhensibles à la fois par le **concepteur** et par le **système programmé**. Plusieurs étapes sont nécessaires : on **rédigé un algorithme**, que l'on met au point avec un **logiciel de programmation par blocs** et on vérifie le **comportement attendu** du système réel.

Structurer les connaissances

Déclenchement d'une action par un évènement, instructions conditionnelles

CYCLE 4


 » Début de cycle
 » Milieu de cycle
 » Fin de cycle

Pour résoudre différents problèmes et rendre les systèmes toujours plus autonomes, on utilise un algorithme avec des instructions qui peuvent être exécutées si des évènements sont détectés par des capteurs.

- Dans un algorithme ou un programme, les conditions qui déclenchent des actions à partir d'évènements sont des instructions conditionnelles. Elles permettent d'exécuter les instructions si les conditions sont vraies ou pas.

Différents évènements peuvent être détectés comme :

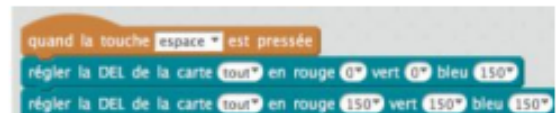
- la variation d'une grandeur physique (changement de luminosité, de chaleur, de couleur...)
- le déplacement d'un objet mesuré par un capteur du système
- la présence ou non d'une route
- ou simplement un évènement initial, permettant de lancer le programme.



- Les instructions peuvent être conditionnées par l'apparition d'évènements initiaux ou détectés par des capteurs.

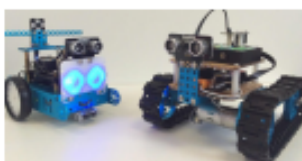
- Déclenchement d'une action par un évènement dit initial : "Quand évènement"

Exemple : « quand touche espace est pressée »



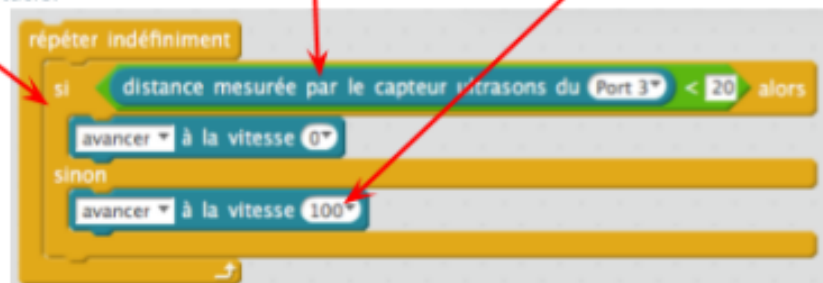
- Déclenchement d'une action par un évènement conditionnel : Si... Alors...Sinon

Exemple : le robot mBot doit détecter un obstacle.



*Si obstacle Alors
arrêt
Sinon avancer
Si obstacle Alors arrêt...*

Le robot mBot doit laisser la priorité, en détectant l'apparition d'un autre véhicule. Il s'arrête quand il détecte un obstacle devant lui et se remet en route quand il n'y a plus d'obstacle.



Dans un algorithme ou un programme, l'exécution des instructions peut être conditionnée par l'apparition d'un évènement. Celles-ci sont des instructions conditionnelles. Dans ce cas, l'instruction s'exécute SI l'évènement a lieu, SINON une instruction différente pourra aussi se réaliser.