

DOSSIER D'INITIATION A LA PROGRAMMATION GRAPHIQUE

En suivant ce dossier **pas à pas**, vous apprendrez à piloter le robot mBot et bien d'autres.

En route !

Présentation :

Il s'agit d'un robot Mbot du commerce.

Il est doté d'une carte électronique programmable. Celle-ci peut être programmé grâce à un ordinateur soit par fil (port USB), soit par Wifi.

On utilisera le logiciel « Mblock » pour créer nos programmes. Le langage graphique utilisé est le scratch.



SOMMAIRE

PRÉSENTATION DU MATÉRIEL	p1
LOCALISER LES CAPTEURS ET ACTIONNEURS	p2
L'ACCUEIL DU LOGICIEL DE PROGRAMMATION	p3
LA MÉTHODE POUR CRÉER LE PROGRAMME	p4
VOS PREMIERS PAS DANS LA PROGRAMMATION.....	p5
RAPPEL DU CONTENU DES BIBLIOTHÈQUES COURAMMENT UTILISÉES ...	p6
LA DÉMARCHE POUR ENVOYER LE PROGRAMME DANS LE ROBOT	p7
LES PROGRAMMES A RÉALISER 1 / 3	p8
LES PROGRAMMES A RÉALISER 2 / 3	p9
LES PROGRAMMES A RÉALISER 3 / 3	p10

PRÉSENTATION DU MATÉRIEL

Le robot mBot interagit avec son environnement en fonction du programme qu'on lui implante. Pour cela, il est capable de collecter des informations grâce à ses capteurs et de réaliser des actions grâce à ses actionneurs.

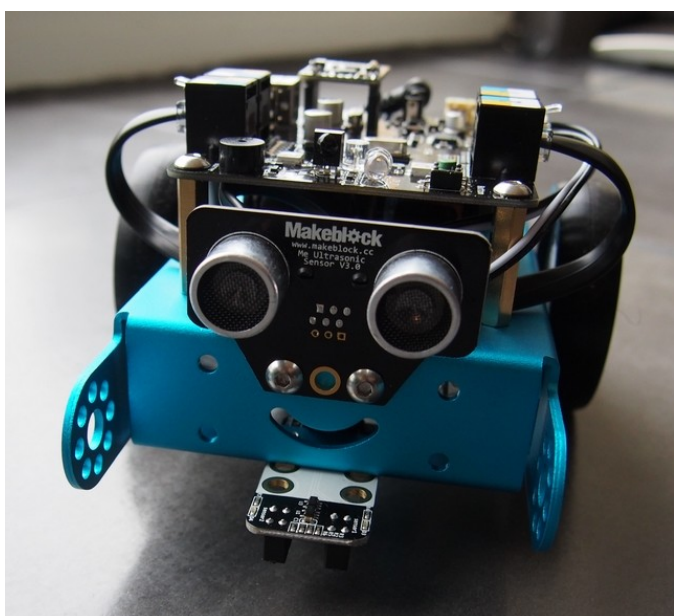
Actions et actionneurs :

- le robot est capable de **se déplacer** : il est équipé de deux moteurs indépendants qui pilotent chacun une roue motrice.
- il peut **émettre des sons** grâce à un buzzer.
- il peut **émettre de la lumière** grâce à 2 DEL RGB dont la couleur est paramétrable.

Boutons et capteurs :

Pour interagir avec son environnement et y recueillir des informations, on retrouve sur le robot :

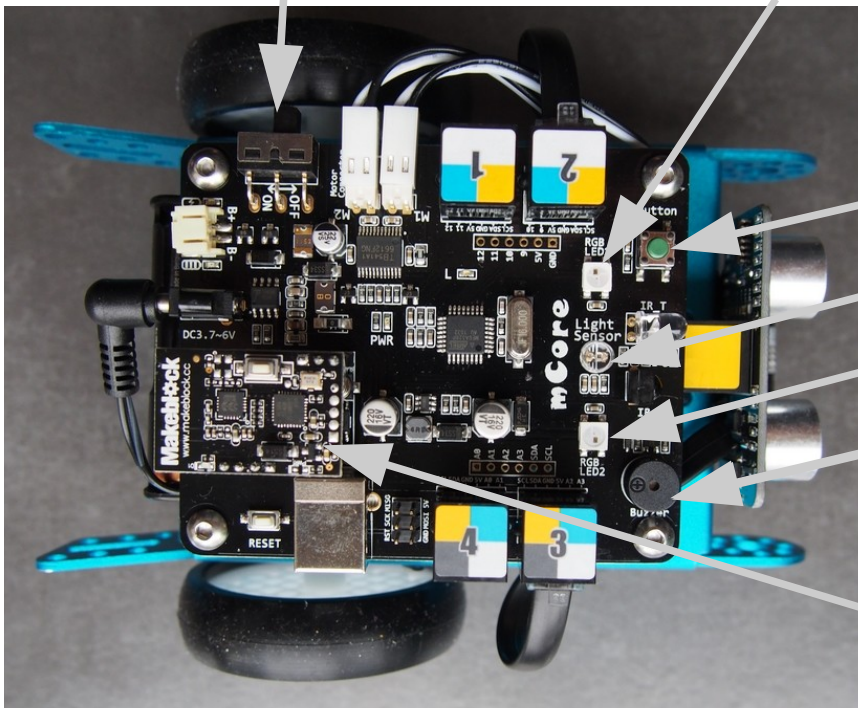
- un module Wifi qui permet de recevoir les ordres émis par l'ordinateur.
- un capteur de luminosité qui le renseigne sur la luminosité ambiante.
- un module à ultrasons qui lui permet de « voir » les obstacles à l'avant et d'en connaître la distance.
- un module de suivi de ligne au sol.
- un bouton paramétrable (on peut donc choisir la fonction qu'on veut pour ce bouton)
- un bouton de mise sous tension ON / OFF.



LOCALISER LES CAPTEURS ET ACTIONNEURS

Bouton de mise sous tension ON / OFF

LED RGB1



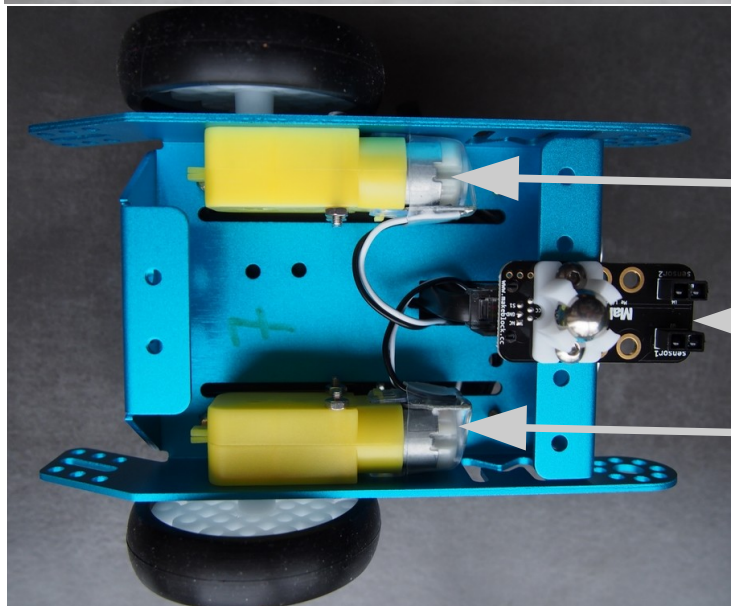
bouton

Capteur de luminosité

LED RGB2

Buzzer

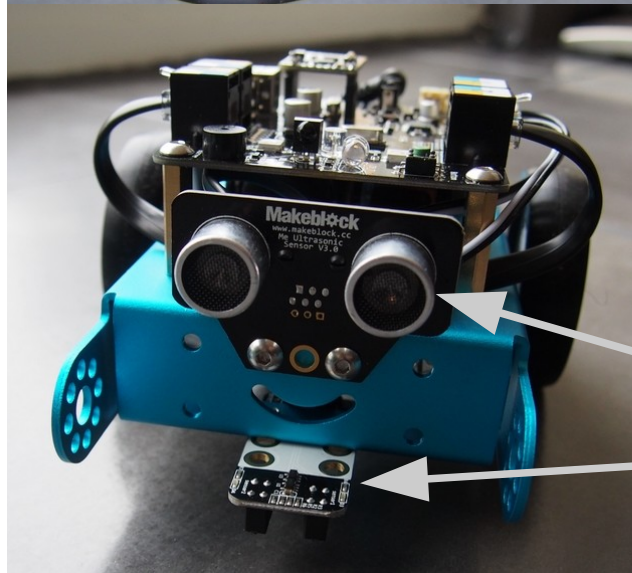
Module wifi ou bluetooth



Moteur de la roue droite

Module de suivi de ligne

Moteur de la roue gauche



Module à ultrasons

Module de suivi de ligne

L'ACCUEIL DU LOGICIEL DE PROGRAMMATION

1- Lancez le logiciel mBlock qui se trouve dans : Démarrer / programmes / technologie / mBlock



2- Dans « choix de la carte », sélectionnez mBot puis dans « choix des extensions », sélectionnez Makeblock.



Zone dans laquelle vous allez saisir votre programme



Vous trouvez ici les différentes bibliothèques d'instructions pour concevoir vos programmes. En ce qui nous concerne, nous utiliserons essentiellement :

- Evènements
- Contrôle
- Opérateurs
- Pilotage

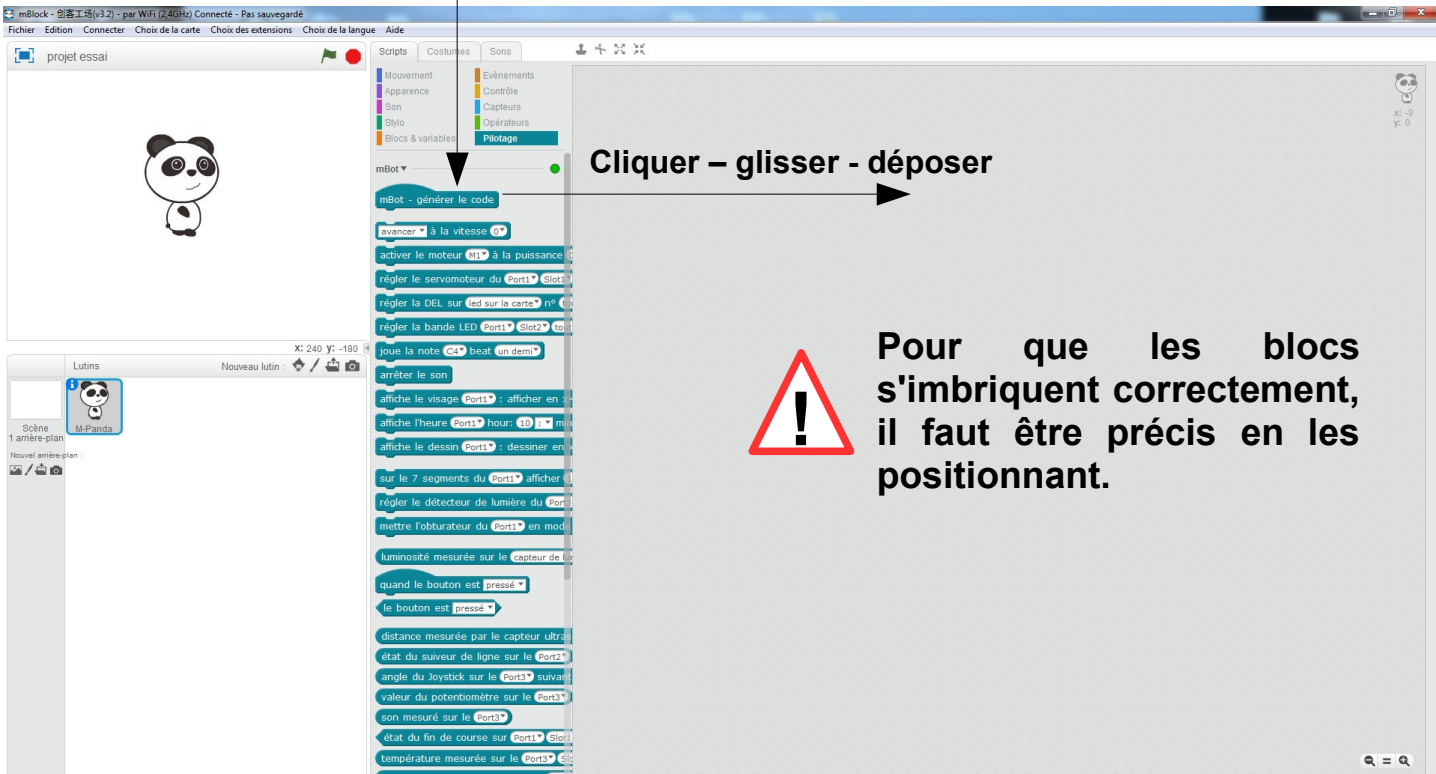
3- Pour créer le programme, il suffit de glisser les éléments de la bibliothèque dans la zone de création du programme. Attention, il faut parfois les déposer avec précision pour que cela s'imbrique.

LA MÉTHODE POUR CRÉER LE PROGRAMME

1- Je sélectionne la bonne bibliothèque d'instruction



2- Une fois les différentes instructions affichées, je choisis celle qui m'intéresse et la glisse dans la zone de droite.

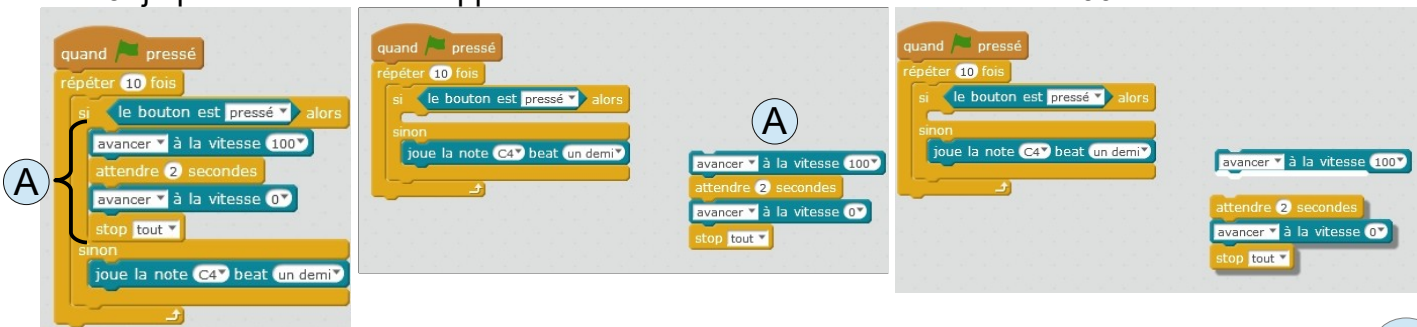


SUPPRIMER UNE INSTRUCTION :

Pour supprimer une instruction non souhaitée, il suffit de la re-glisser dans la bibliothèque à gauche. Attention, il faut parfois procéder par étapes car les instructions imbriquées viennent ensemble.

Exemple : pour supprimer le bloc « avancer à la vitesse 100 »

- 1- je saisis le bloc (A) par sa première instruction et le je le sors du programme
- 2- je saisis « attendre 2 secondes » et je le dissocie des autres instructions pour isoler l'instruction que je veux effacer.
- 3- je peux maintenant supprimer l'instruction « avancer à la vitesse 100 »



VOS PREMIERS PAS DANS LA PROGRAMMATION

1- Votre programme commencera par :

mBot - générer le code

Si vous souhaitez que le robot enregistre votre programme pour l'exécuter ensuite. (on pourra mettre « attendre jusqu'à – le bouton est pressé » pour que le programme démarre à l'appui sur le bouton du robot.)

2- Vous devez ensuite choisir la structure de base de votre programme. Dans un premier temps, nous utiliserons deux types de structures :

répéter indéfiniment

La **boucle infinie**

OU

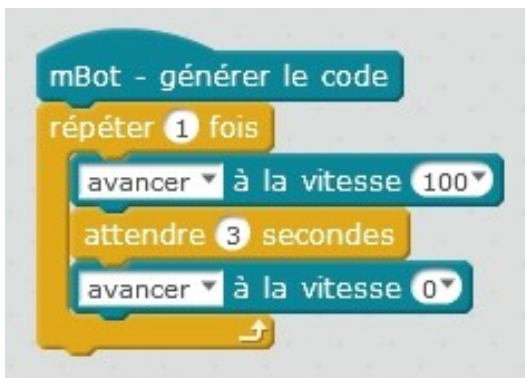
répéter 10 fois

La **boucle avec compteur** (qui s'exécute un nombre de fois déterminé)

3- Ensuite vous pouvez enchaîner les instructions dans l'ordre souhaité.

EXEMPLES DE PROGRAMMES SIMPLES :

Le robot doit se déplacer en avant à la vitesse 100 pendant 3 secondes puis s'arrêter.



Réalisez ce programme, téléversez-le dans le robot en suivant les instructions page 7 et testez-le !

Le robot doit se déplacer en avant à la vitesse 100 pendant 3 secondes puis s'arrêter 3 secondes. Le programme se répète 2 fois.

Complétez ce programme et testez-le sur le robot.



REMARQUE : les actionneurs ne s'arrêtent pas tout seul. Il faut **TOUJOURS** leur donner l'ordre de s'arrêter (à l'exception du buzzer car on détermine la durée de la note jouée)

RAPPEL DU CONTENU DES BIBLIOTHÈQUES COURAMMENT UTILISÉES.

Le marqueur ● indique que vous aurez à utiliser cette commande dans les programmes que vous devrez créer.

Bibliothèque PILOTAGE

Bibliothèque CONTRÔLE

Bibliothèque OPÉRATEURS

LA DÉMARCHE POUR ENVOYER LE PROGRAMME DANS LE ROBOT

Vous allez créer tous les programmes proposés dans l'ordre.

1- Tout d'abord, dans « **Ordinateur / Groupes / votre groupe de techno / travail / mBot** », créez un dossier que vous nommerez « **mBotprénoms** »

Par exemple, si ce sont les programmes créés par Mathieu et Olivia, leur dossier s'appellera Mbotmathieuolivia

Tous vos programmes seront enregistrés dans ce dossier au fil de votre travail.

2- Créez le programme demandé

3- Poser le robot sur un support, branchez-le à l'ordinateur et vérifiez sur le robot que le commutateur est bien sur « **ON** ».

3- Allez dans l'onglet « **Connecter** », « **par port série** », choisissez le bon port COM puis assurez-vous que l'instruction « **connecter** » est bien activée.

4- Une fois le programme prêt à être testé, cliquez sur « édition », « mode arduino ». A ce moment-là, à droite apparaît votre programme en langage C. Vous n'avez plus qu'à cliquer sur « **téléverser dans l'arduino** ».

5- Un message s'affichera pour vous prévenir que le téléversement est terminé.



ATTENTION : une fois le téléversement terminé, le robot exécute le programme ! Tenez le en l'air pour qu'il ne saute pas de la table !

Édition / mode arduino

Téléverser dans l'arduino

LES PROGRAMMES A RÉALISER 1 / 3

Programme n°1 : l'objectif est de manipuler tous les actionneurs

Il ne s'exécutera qu'une seule fois.
Le robot avance durant 3 secondes à la vitesse 100.
Puis il s'arrête (on peut activer chaque moteur à la vitesse 0).
Les Led s'allument en vert pendant 2 secondes puis s'éteignent. Le robot émet alors le son G7 durée « tout ».

UNE FOIS LE PROGRAMME CRÉÉ, PENSEZ BIEN A L'ENREGISTRER !

Programme n°2 : faire fonctionner les 2 moteurs indépendamment

Il ne s'exécutera qu'une seule fois.
Le robot se déplacera avec un moteur à la vitesse 150 et l'autre à la vitesse 50, le tout durant 4 secondes. Ensuite les moteurs s'arrêtent.
Normalement, le robot fait un rond.

Programme n°3 : Utiliser une boucle, une condition et le module à ultrasons de détection d'obstacles

Le but est que le robot se déplace en ligne droite avec ses lumières vertes allumées. Il doit s'arrêter et s'allumer en rouge quand il arrive à 30 cm d'un obstacle. Si on supprime l'obstacle, il repart.

C'est une **boucle infinie** donc tout commencera par: →

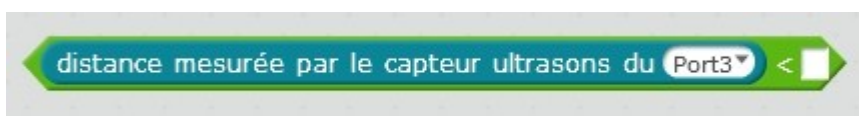
Le principe :

Jusqu'à ce qu'un obstacle soit à moins de 30 cm, le robot avance à la vitesse 100 avec ses Leds allumées en vert (on mettra une petite temporisation de 0,1s à la fin, c'est -à-dire « attendre 0,1s »).

Quand il détecte un obstacle à moins de 30 cm, il doit s'arrêter et allumer ses Leds en rouge puis attendre une seconde.

Astuce 1 : il va falloir aller chercher le comparateur < dans « opérateurs » et y coller la distance mesurée que vous trouverez dans « pilotage ».

Astuce 2 : il faudra utiliser « répéter jusqu'à » que vous trouverez dans l'onglet « contrôle ».



LES PROGRAMMES A RÉALISER 2 / 3

Programme n°4 : l'objectif est de faire fonctionner le robot à la télécommande avec les flèches

Il faut que le robot avance à 100 avec la flèche du haut, recule à 100 avec la flèche du bas, tourne à droite avec la flèche de droite et à gauche avec la flèche de gauche. (pour tourner, on mettra un moteur à 100 et l'autre à -100). Si rien n'est pressé, le robot est à l'arrêt.

Remarque : l'instruction concernant les boutons de la télécommande se trouve parmi les dernières instructions de la bibliothèque « pilotage ».

Astuce 1 : utiliser des conditions si / alors / sinon imbriquées.

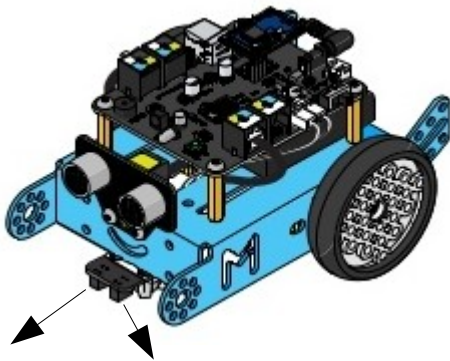
Programme n°5 : Utiliser le capteur de luminosité

Il s'exécutera en continu.

Le robot allume ses 2 Leds en vert 255 si la luminosité est inférieure à 250 (on mettra une temporisation de 0,2 s) , sinon elles sont éteintes.

Astuce : on utilisera le même comparateur < que dans le programme 3 mais cette fois-ci on ira chercher la valeur de la luminosité mesurée.

Programme n°6 : Robot suiveur de ligne



La Led de droite renvoie la valeur 1 si elle quitte la piste noire ou 0 si elle est au-dessus du noir.

La Led de gauche renvoie la valeur 2 si elle quitte la piste noire ou 0 si elle est au-dessus du noir.

On utilisera la piste fournie avec le robot pour mettre en œuvre ce programme. Il s'agit d'un circuit noir sur fond blanc.

En fonction de ce que voit le robot, il renvoie une valeur qui peut aller de 0 à 3.

Il existera donc 4 situations possibles :

- le robot est sur la piste.
- le robot est un peu trop à gauche.
- le robot est un peu trop à droite.
- le robot n'est plus du tout sur la piste.

Réalisez un programme qui permet au robot de faire des tours de piste en suivant la ligne noire, aussi vite que possible.

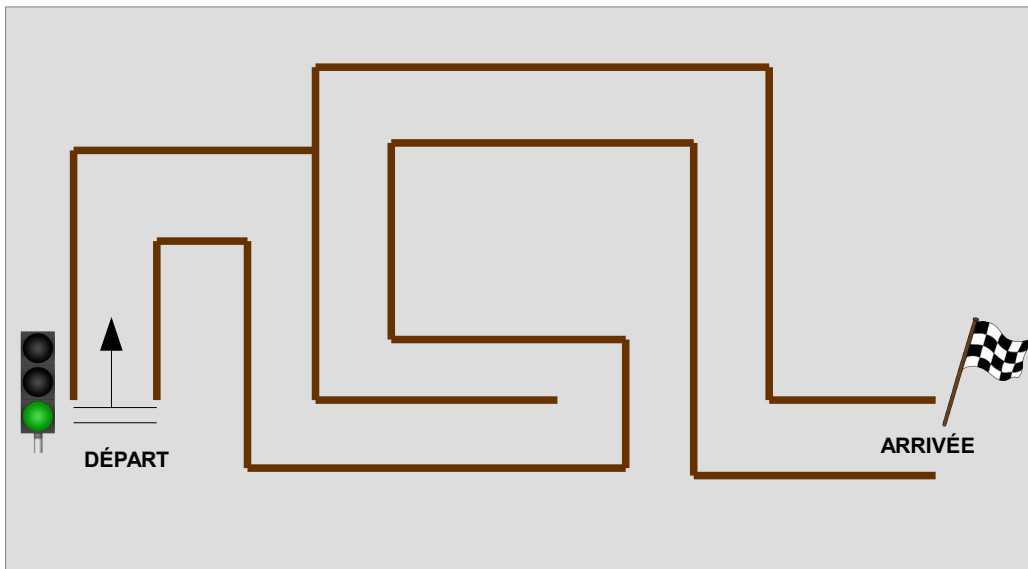
Astuce : il faudra utiliser des « si » imbriqués.

Programme n°7 : Le labyrinthe !

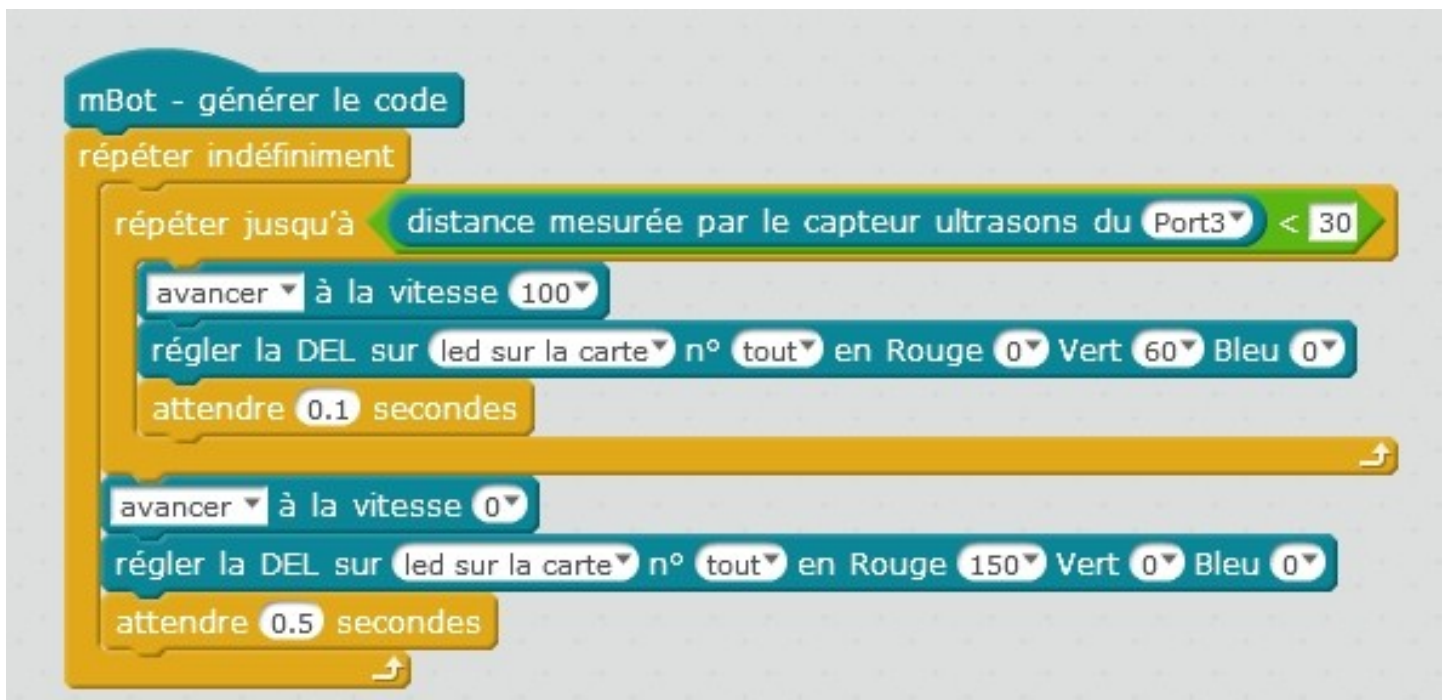
Votre robot est posé sur un parcours labyrinthe ! A chaque fois qu'il arrive juste devant un mur, il doit choisir s'il part à gauche ou à droite en explorant les 2 options . Objectif : sortir du labyrinthe tout seul !

Astuce : il va falloir réfléchir :)

Exemple de plateau d'évolution :



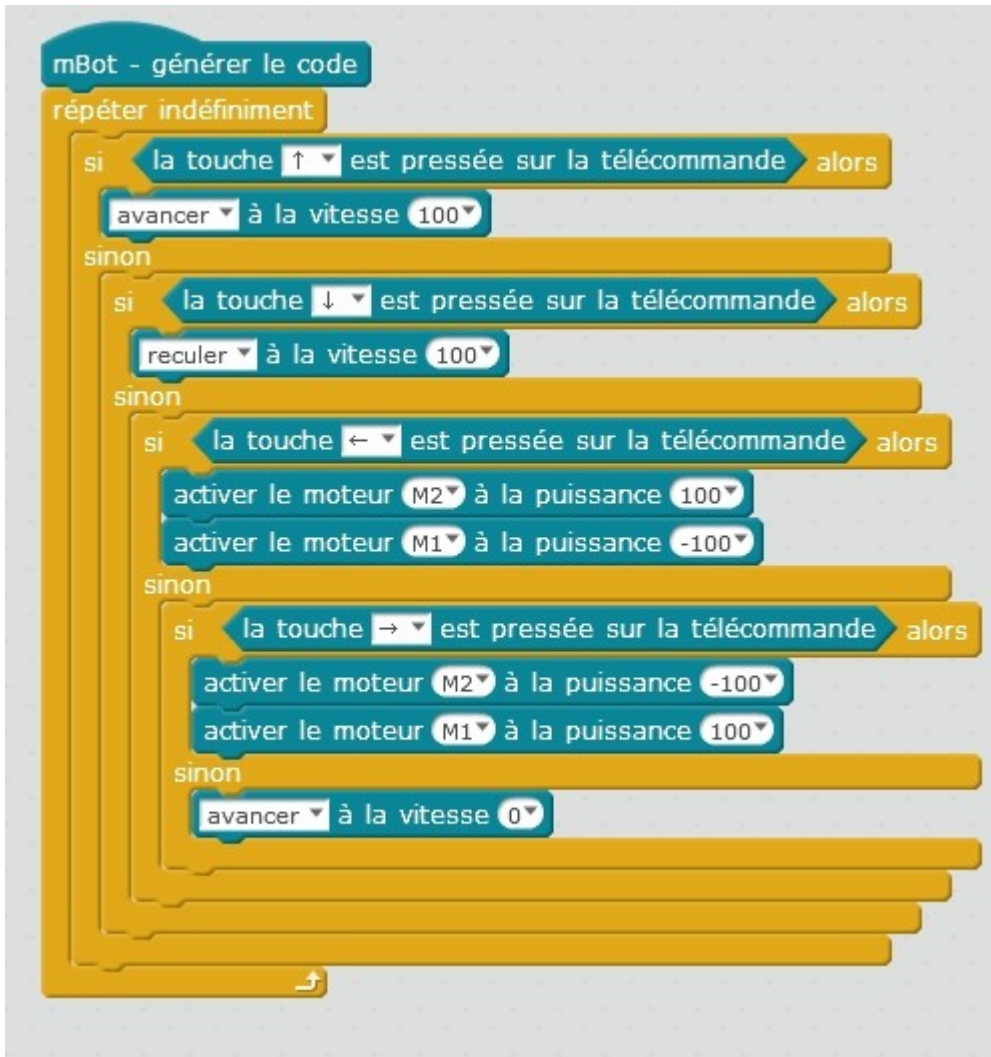
Programme n°3 corrigé



```
mBot - générer le code
répéter indéfiniment
  répéter jusqu'à distance mesurée par le capteur ultrasons du Port3 < 30
    avancer à la vitesse 100
    régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 60 Bleu 0
    attendre 0.1 secondes
  avancer à la vitesse 0
  régler la DEL sur led sur la carte n° tout en Rouge 150 Vert 0 Bleu 0
  attendre 0.5 secondes
```

The image shows a Scratch script for an mBot. It starts with a 'mBot - générer le code' block. The main logic is enclosed in an 'indefinitely repeat' loop. Inside this loop, there is a 'repeat until' block that checks the distance measured by the ultrasonic sensor on Port3 against the value 30. When the distance is less than 30, the following actions are performed: the mBot moves forward at a speed of 100, the LED on the card is set to 'tout' (all) with Red at 0, Green at 60, and Blue at 0, and it waits for 0.1 seconds. After the 'repeat until' block, the mBot moves forward at a speed of 0, the LED is set to 'tout' with Red at 150, Green at 0, and Blue at 0, and it waits for 0.5 seconds before the loop repeats.

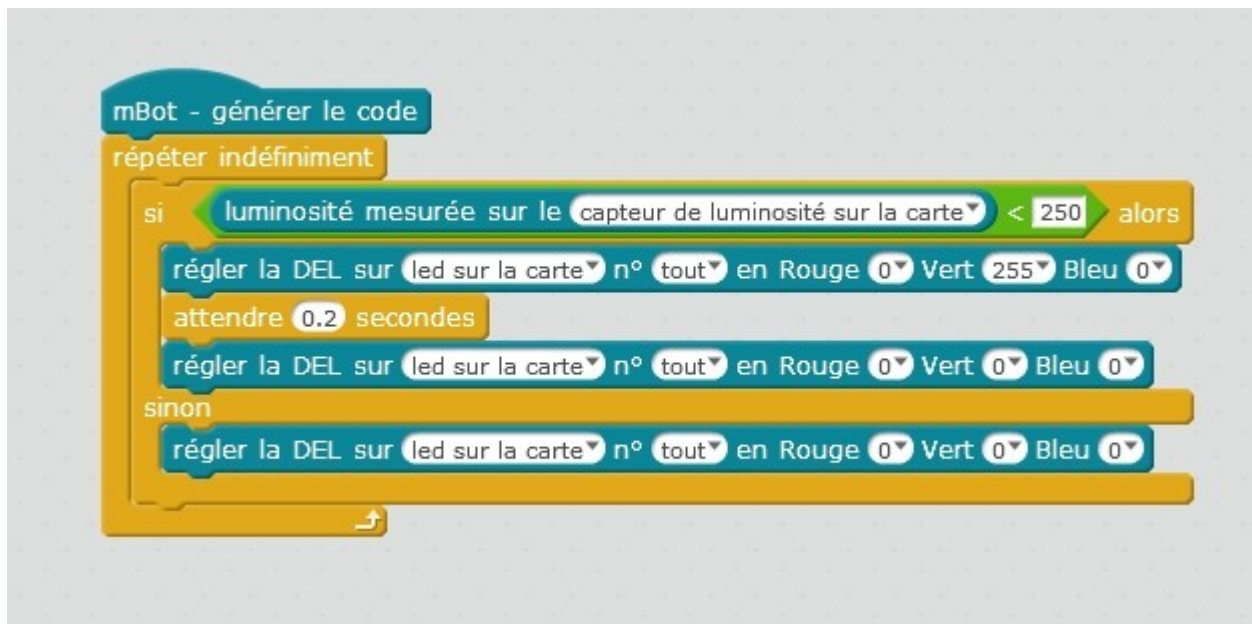
Programme n°4 corrigé



```
mBot - générer le code
répéter indéfiniment
  si la touche ↑ est pressée sur la télécommande alors
    avancer à la vitesse 100
  sinon
    si la touche ↓ est pressée sur la télécommande alors
      reculer à la vitesse 100
    sinon
      si la touche ← est pressée sur la télécommande alors
        activer le moteur M2 à la puissance 100
        activer le moteur M1 à la puissance -100
      sinon
        si la touche → est pressée sur la télécommande alors
          activer le moteur M2 à la puissance -100
          activer le moteur M1 à la puissance 100
        sinon
          avancer à la vitesse 0
```

The image shows a Scratch script for an mBot robot. It starts with a blue block 'mBot - générer le code'. Below it is a yellow 'répéter indéfiniment' (repeat forever) loop. Inside the loop, there are four conditional blocks: 1) 'si la touche ↑ est pressée sur la télécommande alors' followed by 'avancer à la vitesse 100'. 2) 'sinon' followed by 'si la touche ↓ est pressée sur la télécommande alors' followed by 'reculer à la vitesse 100'. 3) 'sinon' followed by 'si la touche ← est pressée sur la télécommande alors' followed by two blocks: 'activer le moteur M2 à la puissance 100' and 'activer le moteur M1 à la puissance -100'. 4) 'sinon' followed by 'si la touche → est pressée sur la télécommande alors' followed by two blocks: 'activer le moteur M2 à la puissance -100' and 'activer le moteur M1 à la puissance 100'. Finally, a 'sinon' block followed by 'avancer à la vitesse 0'. The script ends with a small arrow icon at the bottom of the loop.

Programme n°5 corrigé



```
mBot - générer le code
répéter indéfiniment
  si luminosité mesurée sur le capteur de luminosité sur la carte < 250 alors
    régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 255 Bleu 0
    attendre 0.2 secondes
    régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 0 Bleu 0
  sinon
    régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 0 Bleu 0
```

The image shows a Scratch script for an mBot. It starts with a 'mBot - générer le code' block. The main logic is enclosed in a 'répéter indéfiniment' (repeat forever) loop. Inside the loop, there is an 'if' block: 'si luminosité mesurée sur le capteur de luminosité sur la carte < 250 alors'. If the condition is true, the script performs three actions: 'régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 255 Bleu 0', 'attendre 0.2 secondes', and 'régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 0 Bleu 0'. If the condition is false (the 'sinon' branch), it performs one action: 'régler la DEL sur led sur la carte n° tout en Rouge 0 Vert 0 Bleu 0'. The script ends with a small arrow icon at the bottom of the loop block.