

Présentation du projet

MBot entre dans un garage collectif. Il est face à une allée centrale ; Cinq box consécutifs identiques sont situés sur sa droite, cinq autres box identiques consécutifs sont sur sa gauche. Les box occupés sont fermés.

La mission est de programmer MBot pour qu'il se gare, en marche avant, dans le premier box disponible. Si tous les box sont occupés, MBot ressort.

MBot est équipé d'un détecteur d'obstacle (placé devant, ses yeux), de capteurs de suivi de ligne (placés sous son châssis, à l'avant), de leds et d'un avertisseur sonore.

Ce projet, pour être mené à bien, est découpé en plusieurs phases, en mathématiques et en technologie.

Avant de programmer concrètement MBot dans ce garage, il convient de simuler la situation sous scratch, afin d'éviter des accidents, d'économiser les piles, etc ...

Pour coller le plus possible à la programmation réelle du robot MBot, toutes les instructions et tous les blocs de contrôle de scratch ne seront pas à disposition pour cette simulation.

Simulation - Phase 1 : Box de droite uniquement, déplacement sans suiveur de ligne

Mission :

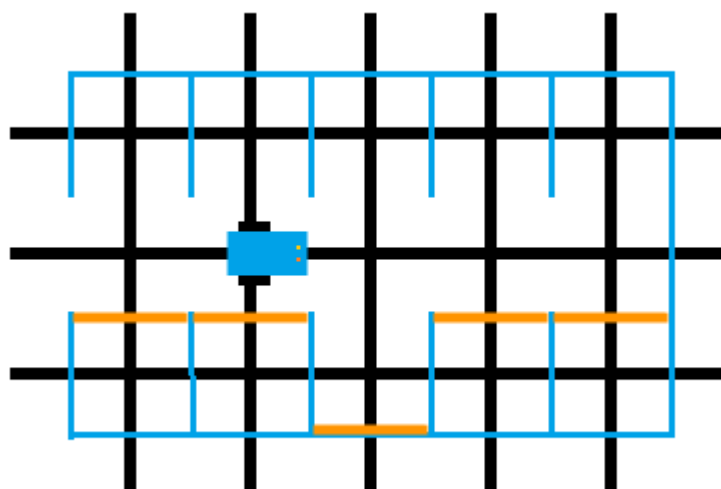
MBot se gare, en marche avant, dans le premier box disponible **sur sa droite**. Ce peut être n'importe lequel des 5 box, le premier des cinq disponible à droite. Si tous les box de droite sont occupés, MBot ressort.

A cette phase, on se préoccupe uniquement des box de droite. On n'utilise pas le suiveur de ligne.

• **Rédiger un algorithme***, en langage naturel.

Pour cet algorithme écrire des instructions comme

- « Se placer à l'entrée du parking »,
- « Aller à l'intersection suivante »,
- « Détecter un obstacle »,
- « Pivoter d'un quart de tour »,
- « Pivoter d'un demi tour », etc ...



*Un algorithme est une suite structurée d'instructions simples.

• **Valider l'algorithme en le testant à la main sur une des maquettes papier/voiture de la classe.**

• **Faire valider l'algorithme par le professeur.**

Un coup de pouce est à disposition en cas de besoin pour mieux comprendre la consigne.

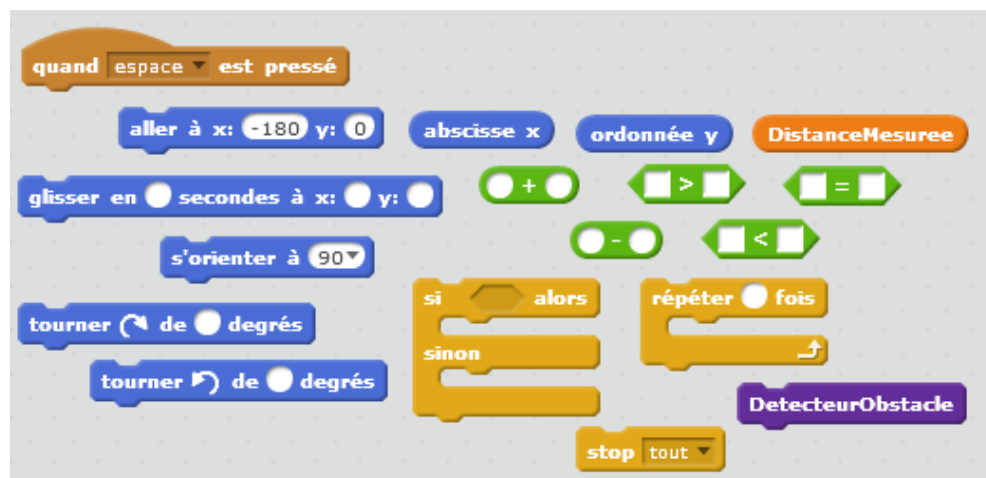
Maintenant, écrire le script de MBot. Pour cela, ouvrir le fichier « MBot_Phase1_eleve ».

Ne pas oublier d'enregistrer de suite le fichier dans le répertoire personnel « Maths_Pgm_Scratch » sous le nom « Mbot_Phase1_VosPrenoms », puis de le sauvegarder régulièrement.

Pour écrire ce script, bien lire toutes les informations qui suivent et les commentaires du fichier scratch.

Instructions autorisées sous scratch (autant de fois que nécessaire) :

Attention, lorsqu'une valeur est précisée dans l'instruction, elle ne peut pas être changée !



Trois coups de pouce à disposition en cas de besoin.

Les box, le quadrillage de lignes noires

Les box sont des carrés de 25 cm de côté. Les lignes noires forment un quadrillage carré de côté 25 cm. Ainsi, deux intersections consécutives sont distantes de 25 cm (de « centre » à « centre »).

Sous scratch, le garage et MBot sont représentés à l'échelle. 25 cm sont représentés par 60 pixels.

MBot se déplace à **vitesse constante**, sa vitesse moyenne est de 100 mm.s^{-1} .

A cette phase, il se déplace sur les lignes noires et n'utilise pas ses capteurs de suivi de ligne.

Sous scratch, il est initialement placé au point de coordonnées (-180;0).

Le détecteur d'obstacle.

C'est un capteur de distance à ultrason ;

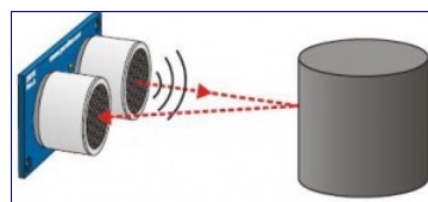
Il fonctionne avec une partie émetteur d'onde et une partie récepteur d'onde.

Cette onde se déplace dans l'air (comme le son), elle est renvoyée quand elle rencontre un obstacle.

Comme la vitesse du son est connue (340 m.s^{-1}), il est possible de déterminer la distance entre MBot et un obstacle en mesurant le temps écoulé entre l'envoi de l'ultrason et la réception de l'écho. Ces calculs sont gérés par les composants technologiques de MBot. Ainsi, en exploitant la valeur de ce composant, MBot peut détecter et éviter des obstacles.

Sous scratch, cette valeur est disponible dans la variable DistanceMesurée.

La variable Distancemesurée vaut 30 si la porte du box est fermée.



Pour valider cette phase, la tester plusieurs fois en appuyant à chaque test sur la « barre espace », de manière à pouvoir vérifier dans tous les cas : MBot doit pouvoir se garer dans n'importe lequel des cinq box ou ressortir. Faire valider cette phase par le professeur.

Simulation - Phase 2 : Hors du garage, déplacement avec suiveur de ligne

L'objectif de cette phase est de comprendre le déplacement de MBot utilisant ses capteurs de suivi de ligne. Pour cette étape, les box de garage sont oubliés, MBot se déplace sur une ligne sinueuse.

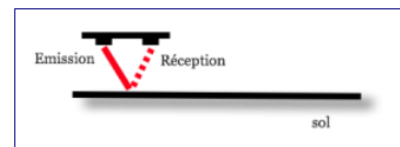
Les capteurs de suivi de ligne

Un capteur de suivi de ligne est un capteur de contraste.

Il est constitué d'une LED émettrice et d'un phototransistor.

Comme les couleurs foncées et les couleurs claires réfléchissent différemment

la lumière, le capteur détecte s'il se trouve au dessus d'une couleur foncée ou d'une couleur claire.



MBot possède deux capteurs de suivi de ligne, il est possible de déterminer le type des couleurs situées sous chacun des deux capteurs. **Un code couleur est créé et disponible dans un composant technologique de MBot**, Valeurs de ce code et interprétations :

0 indique foncé à gauche et foncé à droite,

2 indique clair à gauche et foncé à droite,

1 indique foncé à gauche et clair à droite

3 indique clair à gauche et clair à droite

Ainsi, en lisant et exploitant le code de ce composant, MBot peut se repérer sur une ligne noire, il sait alors s'il est sur la ligne, à cheval sur la droite de la ligne, à cheval sur la gauche ou en dehors de la ligne.

Sous scratch, cette valeur est disponible dans la variable Port2. Elle vaut 0, 1, 2 ou 3 selon l'emplacement de MBot par rapport à la ligne noire (dessous, à cheval sur un bord, dehors).

Mission :

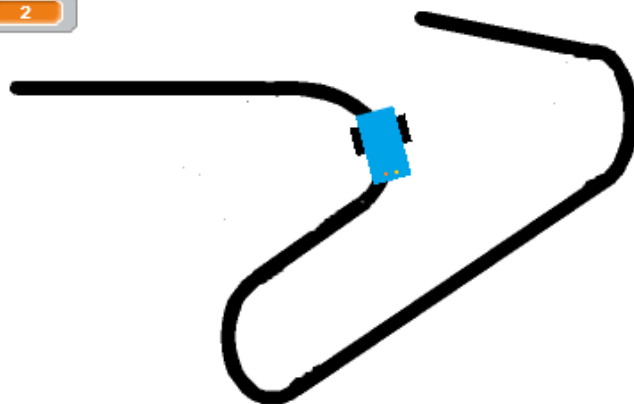
MBot suit la ligne noire en utilisant ses capteurs de suivi de ligne. On ne se préoccupe pas de respecter la vitesse de 100 mm.s^{-1} .

Ouvrir le fichier « MBot_Phase2_eleve » pour compléter le script de MBot.

Enregistrer de suite le fichier dans le répertoire personnel « Maths_Pgm_Scratch » sous le nom « Mbot_Phase2_VosPrenoms ».

Sauvegarder régulièrement.

Port2 **2**



Instructions autorisées sous scratch :



Deux coups de pouce à disposition en cas de besoin.

Tester le programme avec les deux arrières-plans (deux lignes sinueuses différentes à disposition).

Simulation - Phase 3 : Box de droite uniquement, déplacement avec suiveur de ligne

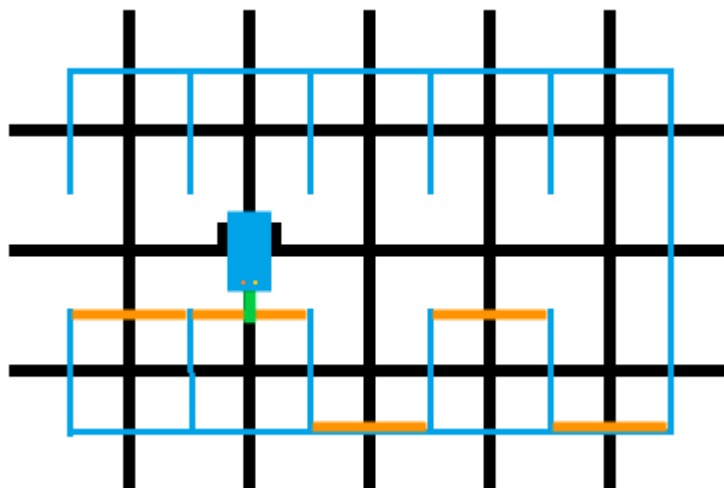
Mission :

MBot cherche à se garer, en marche avant, dans le premier box disponible sur sa droite, en utilisant son détecteur d'obstacle et son suiveur de ligne.

Si tous les box de droite sont occupés, Mbot ressort. On ne se préoccupe pas de respecter la vitesse de 100 mm.s^{-1} .

Il s'agit donc de refaire le script de MBot de la phase 1 en remplaçant les instructions « Glisser en 2,5 s ... » par un bloc « AllerAIntersection ».

Le bloc « AllerAIntersection » permet de se rendre d'une intersection à une suivante, avec suiveur de ligne, c'est à dire d'avancer de 60 pixels en utilisant la valeur du Port2.



Ouvrir le fichier « MBot_Phase3_eleve » pour compléter le script de MBot.

Enregistrer de suite le fichier dans le répertoire personnel « Maths_Pgm_Scratch » sous le nom « Mbot_Phase3_VosPrenoms ». Sauvegarder régulièrement.

Une fois fini, ajouter le son du verrouillage centralisé des portières. Pour cela, jouer le son « Hand clap » jusqu'au bout.

Pour valider cette phase, la tester plusieurs fois en appuyant à chaque test sur la « barre espace », de manière à faire varier le nombre et la répartition des box ouverts.

Ne pas oublier de vérifier que MBot ressort lorsqu'aucun box n'est disponible à droite.

Sauvegarder.

Instructions autorisées sous scratch :

Celles des phases 1 et 2, SANS L'INSTRUCTION  !

Deux coups de pouce à disposition en cas de besoin.

A remarquer -(une fois la mission de la phase 3 accomplie) :

A cette phase, une fois MBot mis sur la ligne noire, le suiveur de ligne n'a pas grand intérêt. En effet sous scratch, le monde est sans imperfection : il n'y a pas de problème de puissance de piles, de plan légèrement incliné, d'adhésif noir abîmé ou mal posé, de grain de sable dans les engrenages de MBot etc...

C'est pourquoi un bloc « SimulationImperfection » a été créé (il n'est pas à modifier).

Placer :  dans le script de MBot simulant le suiveur de ligne comme indiqué ci-dessous

Tester à nouveau cette phase, en appuyant plusieurs fois sur la « barre espace ».

Faire valider cette phase par le professeur.



Simulation - Phase 4 : Hors du garage, tourner à une intersection avec suiveur de ligne.

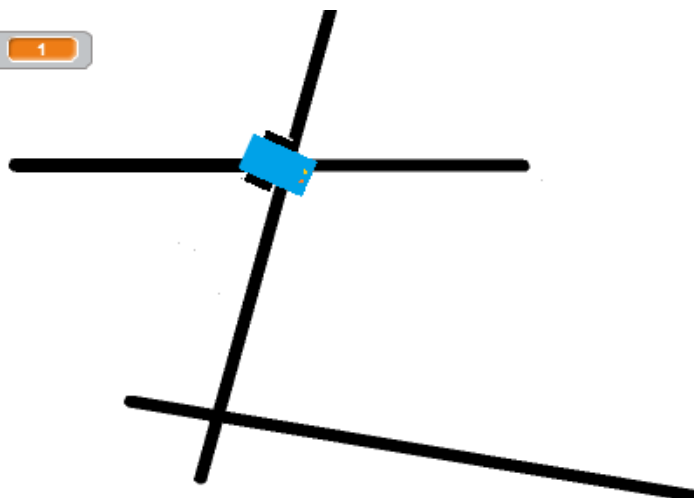
Pour cette phase, plus de box à nouveau!
En pratique, pour tourner à une intersection le vrai robot MBot ne dispose pas de

« Tourner à ... de ...° » .

Dans cette phase, MBot prendra à droite à la première intersection puis à gauche à l'intersection suivante, en utilisant son suiveur de ligne.

Pour cette simulation sous scratch, on utilisera tout de même « Tourner à droite de 2° » et « Tourner à gauche de 2° » pour remplacer les « Tourner à droite à la vitesse ... » et « Tourner à gauche à la vitesse ... » de Mblock.

Port2 1



Mission :

Ecrire le bloc « PrendreADroite ». Ce bloc permet à MBot de pivoter sur la droite lorsqu'il est sur une intersection, en utilisant le suiveur de ligne (c'est à dire la valeur de Port2).

Puis de même, écrire le bloc « PrendreAGauche », il permet à MBot de pivoter sur la gauche lorsqu'il est sur une intersection, en utilisant le suiveur de ligne.

Il n'y a que ces deux blocs à compléter. Les autres déplacements de MBot sur la ligne noire sont déjà programmés !

Ouvrir le fichier « MBot_Phase4_eleve » pour compléter le script de MBot.

Enregistrer de suite le fichier dans le répertoire personnel « Maths_Pgm_Scratch » sous le nom « Mbot_Phase4_VosPrenoms ».

Sauvegarder régulièrement.

Instructions autorisées sous scratch :



Rappel : Seule la valeur 2° est permise dans ces instructions de mouvement.

Un coup de pouce à disposition en cas de besoin.

Faire valider cette phase par le professeur.

Simulation - Phase 6 : Box de droite et de gauche, déplacement avec suiveur de ligne

MBot entre dans un garage collectif. Il est face à une allée centrale ; Cinq box consécutifs identiques sont situés sur sa droite, cinq autres box identiques consécutifs sont sur sa gauche. Les box occupés sont fermés.

Mission complète :

Programmer MBot pour qu'il se gare, en marche avant, dans le premier box disponible (de droite ou de gauche). Si tous les box sont occupés, MBot ressort.

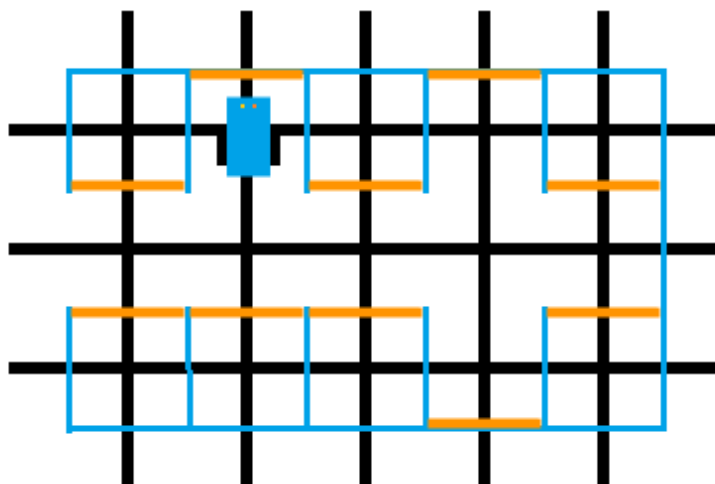
MBot est équipé d'un détecteur d'obstacle, de capteurs de suivi de ligne et d'un avertisseur sonore, il les utilise.

Enregistrer de suite **VOTRE** fichier

« Mbot_Phase5_VosPrenoms » sous

le nom de « Mbot_Phase6_VosPrenoms »

Suivre les consignes pour la mise à en place des portes de gauche pour la simulation (voir page suivante).



Consignes pour la mise à en place des portes de gauche pour la simulation

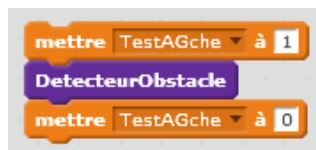
Une fois votre fichier enregistré sous le nom « MBot_Phase6_VosPrenoms ».

Dans la partie gauche de la fenêtre de travail de scratch, en partie basse, les lutins MBot, Porte 1, Porte 2, Porte 3, Porte 4, Porte 5, Détecteur... apparaissent.

- Cliquer gauche sur le lutin « Porte 1 », il devient bleuté.
- Cliquer droit sur le lutin « Porte 1 » et dupliquer : le lutin « Porte 6 » apparaît alors.
- Cliquer droit sur le lutin « Porte 6 »
- Modifier le script de « Porte 6 », en remplaçant les ordonnées par les ordonnées opposées (ainsi -32 devient 32 et -88 devient 88).
- Recommencer ainsi avec le lutin « Porte 2 » pour avoir le lutin « Porte 7 ».
- Recommencer ainsi avec le lutin « Porte 3 » pour avoir le lutin « Porte 8 ».
- Recommencer ainsi avec le lutin « Porte 4 » pour avoir le lutin « Porte 9 ».
- Recommencer ainsi avec le lutin « Porte 5 » pour avoir le lutin « Porte 10 ».

Désormais le fichier est prêt pour programmer la phase 6 !

Une particularité tout de même, quand l'instruction «DétecteurObstacle» sera utilisée pour détecter une éventuelle porte à gauche, il faudra placer juste avant l'instruction « Mettre TestAGche à 1 », il faudra placer juste après « Mettre TestAGche à 0 », comme montré ci-après.



Ceci est spécifique à cette simulation sous scratch et ne sera pas à faire avec MBlock.

Listes des coups de pouce à disposition pour ce projet.

Phase 1 - Coup de pouce 1 : Un exemple d'algorithme en langage naturel (dans un autre contexte).

Phase 1 - Coup de pouce 2 : Le script de MBot (phase 1) à compléter, à poursuivre.

Phase 1 - Coup de pouce 3 : Vitesse constante, déterminer le nombre de secondes du



Phase 1 - Coup de pouce 4 : Coordonnées de MBot dans une boucle « Répéter » avec



Phase 2 - Coup de pouce 5: Interprétation du code du suiveur de ligne.

Phase 2 - Coup de pouce 6 : Conseils pour le débogage.

Phase 3 - Coup de pouce 7 : Script à compléter, à finir.

Phase 3 - Coup de pouce 8 : Son pour le verrouillage centralisé des portières.

Phase 4 - Coup de pouce 9 : Tour sur soi-même, à une intersection, au suiveur de ligne, pour changer de direction.