



Écrire un programme :

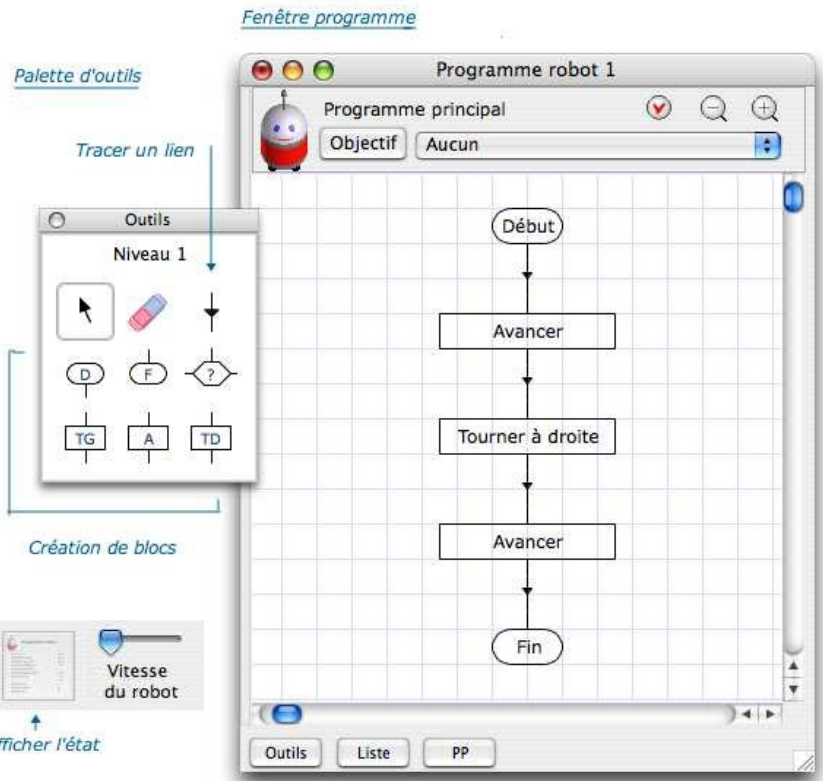
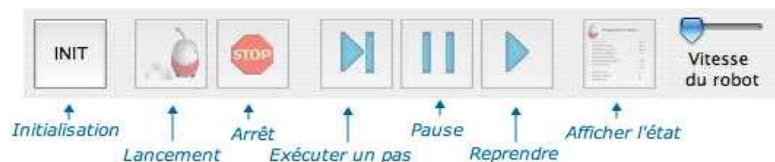
Définition : Un programme est une suite d'instructions logiquement ordonnées qui comporte un début et une fin.

Pour construire l'organigramme représenté ci-contre :

- pour chaque bloc de l'organigramme : prenez le bloc dans la palette et cliquez ensuite dans la fenêtre de programme pour y placer le bloc ;
- pour lier les blocs entre eux, choisissez l'outil lien dans la palette ; cliquez sur une sortie de bloc, déplacez la souris et cliquez sur l'entrée du bloc suivant.

Un organigramme doit contenir un bloc début et un seul pour indiquer où le programme doit commencer, et un ou plusieurs blocs fin.

Affichage du terrain du robot : (Menu Fenêtre > Exécution)



Le terrain apparaît dans une autre fenêtre avec des boutons en haut permettant de contrôler l'exécution.

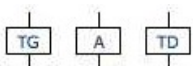
Initialisation du programme : (Bouton INIT ou menu Exécution > Initialisation)

Le programme est alors vérifié : si le programme ne contient pas d'erreur, vous pouvez alors changer la position et la direction initiale du robot en cliquant dessus ; le bouton INIT affiche alors une image du robot à initialiser.

Si le programme contient une erreur, vous ne pourrez pas lancer l'exécution, vous devez d'abord corriger l'erreur.

Lancement du programme : (Bouton Lancement ou menu Exécution > Lancer)

Les déplacements du robot :



Pour déplacer le robot, vous disposez de trois commandes : **Tourner à gauche**, **Avancer**, **Tourner à droite**. Ces commandes sont écrites dans des blocs de forme rectangulaire disponibles dans la palette d'outils

Exercices

Prenez une feuille de classeur et écrivez en en-tête « Premiers programmes avec Robotprog ». Pour chaque exercice, vous noterez son numéro ainsi que son énoncé.

Quand vous avez terminé l'organigramme, appelez le professeur pour le faire vérifier. Quand celui-ci est validé, lancez le programme (raccourcis « RobotProg réseau » sur le bureau) et saisissez l'organigramme de votre solution.

(1) Écrire un programme qui fasse faire demi-tour au robot.

(2) Écrire un programme qui fasse se déplacer le robot jusqu'au mur délimitant le terrain.

(3) Que se passe-t-il si le robot n'a pas la même position de départ ?

Remarque : Pour faire demi-tour il suffit de tourner deux fois de suite du même côté.

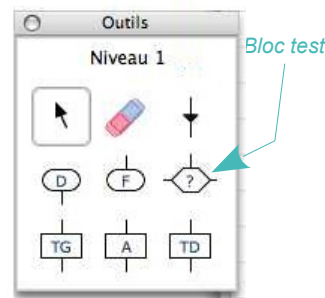


Les conditions :

Définition : Une condition est une expression logique dont le résultat ne peut être que vrai ou faux.

- On peut en particulier combiner différents mots-clé avec des opérateurs logiques tels que **Et**, **Ou**, **Non**. Dans RobotProg, les conditions sont entrées au clavier dans un « Bloc test ».

Exemple : **MurEnFace Et MurADroite**



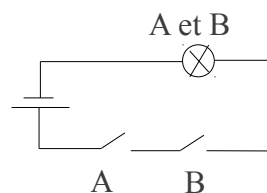
Définition : Un opérateur logique est une opération entre deux expressions logiques qui se définit par sa table de vérité.

L'opérateur ET

ET		A	
		V	F
B	V	V	F
	F	F	F

Le résultat de l'expression « A et B » n'est vrai que quand les deux variables logiques « A » et « B » sont vraies.

On pourrait comparer le « ET » à deux interrupteurs en série. Pour que le courant passe, il faut que les deux interrupteurs soient fermés.

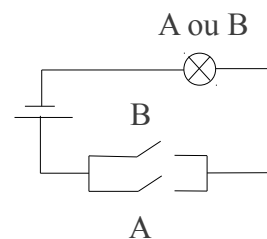


L'opérateur Ou

OU		A	
		V	F
B	V	V	V
	F	V	F

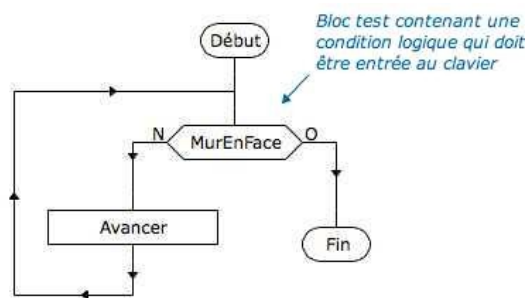
Le résultat de l'expression « A ou B » n'est fausse que quand les deux variables logiques « A » et « B » sont fausses.

On pourrait comparer le « OU » à deux interrupteurs en parallèle. Pour que le courant passe, il faut que l'un des deux interrupteurs au moins soit fermé.



- La condition logique contenue dans le bloc test est évaluée quand le bloc est exécuté. Le résultat peut être vrai ou faux. Si le résultat est vrai, l'exécution se poursuit après la sortie marquée O (oui ou vrai) ; si le résultat est faux, l'exécution se poursuit après la sortie marquée N (non ou faux).

- Dans cet exemple, la condition logique est MurEnFace, c'est un mot-clé du langage du robot qui fournit un résultat de type logique (vrai ou faux) en fonction de la position du robot au moment où elle est évaluée



Exercices

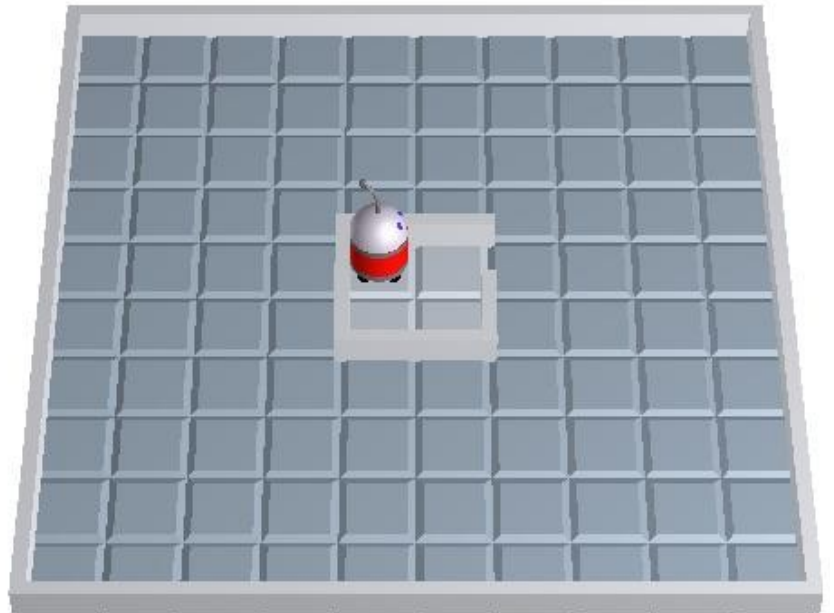
- Quelles sont les conditions pour que le robot soit dans un coin ?
- Écrire un programme qui fasse aller le robot dans un coin.
- Que se passe-t-il si le terrain n'est pas régulier ?



Exemples de terrain :

Terrain 1 :

- *dimensions 10x10*
- *sans obstacle*



Terrain 2 :

- *dimensions 10x10*
- *avec obstacles*

