

TUTORIEL

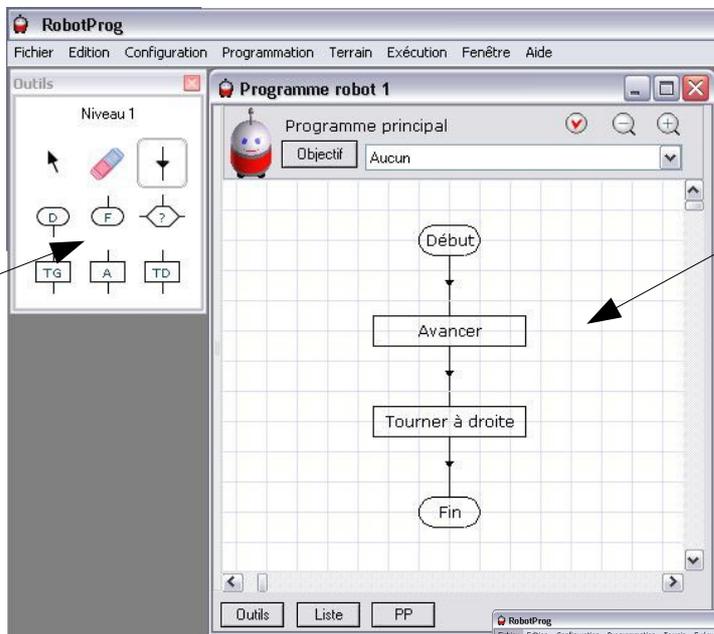
PROGRAMMER R2D2 AVEC ROBOTPROG



Exercice 1 : Mon premier programme

Les blocs permettent de construire le programme

Il faut obligatoirement un bloc Début et un bloc Fin

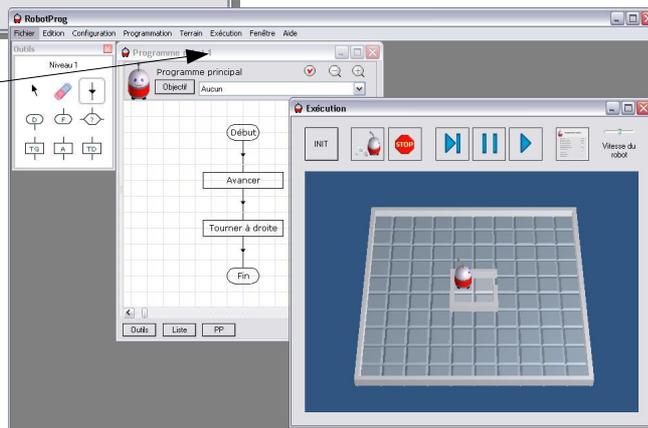


Ce qu'il faut :
- Logiciel RoboProg

Le programme est écrit sous la forme d'organigramme

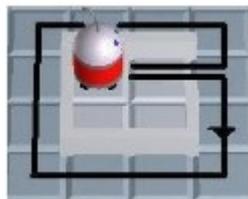
- D: Début
- F: Fin
- ?: test logique
- TG: Tourner à Gauche
- A: Avancer
- TD: Tourner à Droite

Dans Fenêtre/exécution, Lancer l'exécution du programme

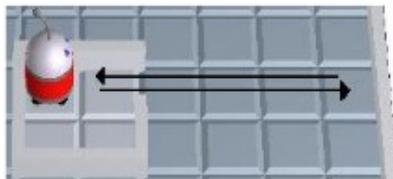


Exercice 2 : Le tour de la maison

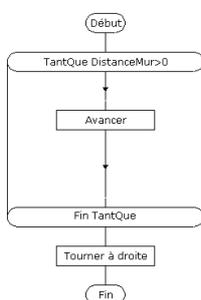
Le robot doit faire le tour de la maison.



Exercice 3 : Aller au mur Est et revenir



On peut utiliser des boucles TANT QUE :
DistanceMur > x.



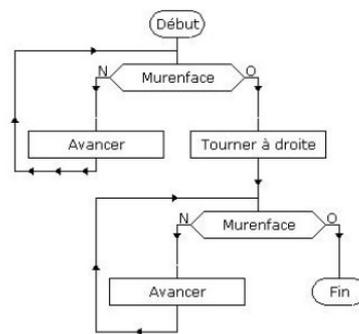
Exercice 4 : Le tour du terrain

Le robot doit faire le tour du terrain.



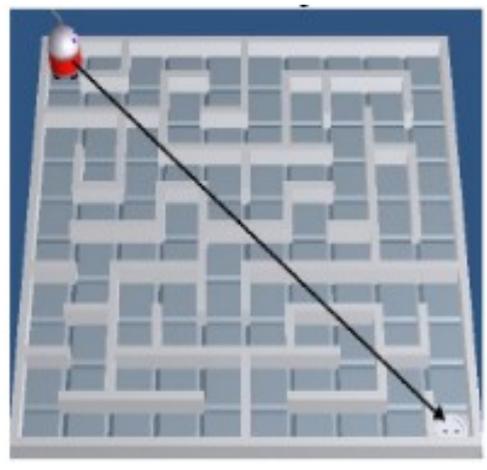
Différents tests SI peuvent être utilisés :

- MurEnFace,
- MurADroite,
- MurAGauche.



Exercice 5 : Le labyrinthe

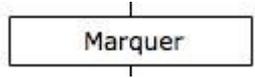
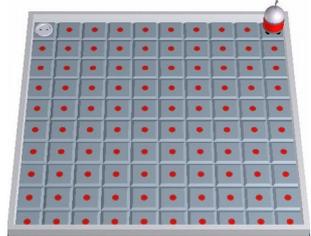
Le robot doit trouver la sortie !!!
 Télécharger le fichier labyrinthe.bog
 Choisir Terrain/Modifier
 Choisir Terrain/Ouvrir
 Le robot s'arrête quand il est sur une prise.



Toucher une prise
 Le robot détecte la prise en la touchant.

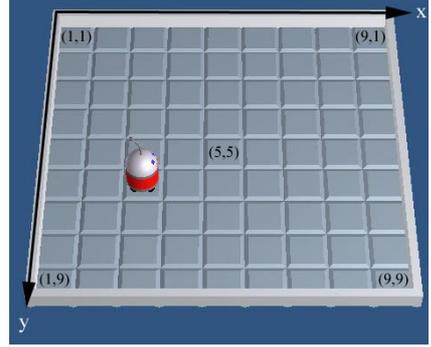
Exercice 6 : L'aspirateur

le robot doit passer dans chacune des cases du terrain pour y aspirer la poussière. Il marquera chacune des cases de son passage en utilisant le bloc « Marquer ».

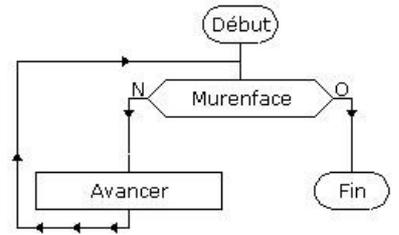


configurez le niveau 6 : Menu -> Configuration -> niveau 6

Télécharger le fichier aspirateur1.bog
 Choisir Terrain/Modifier
 Choisir Terrain/Ouvrir
 le robot aspirateur est au départ dans le coin nord-ouest (case 1,1) sur sa prise, Le robot s'arrête quand il est sur une prise.
 le robot s'arrêtera quand il sera sur la case du coin nord-ouest (10,1), (conditions xrobot=10 et yrobot=1 toutes les deux vérifiées),



Aller au mur EST

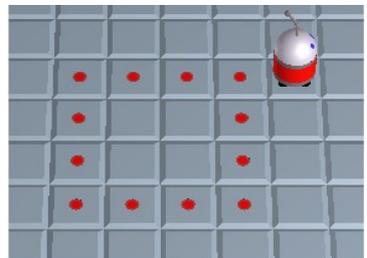
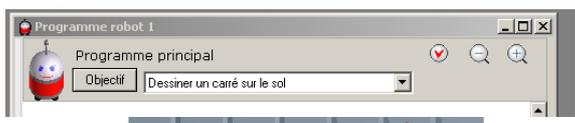


Les sous programmes

On peut créer des sous programmes pour simplifier :
 Programmation/Nouveau SousProg
 On appelle le sous programme avec le bloc SP dans le programme principal.

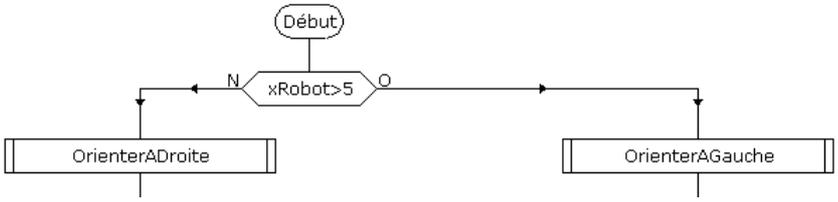
Exercice 7 : Le carré

Le robot doit dessiner un carré dans un terrain de 9x9. On ne sait pas où se trouve le robot au début.



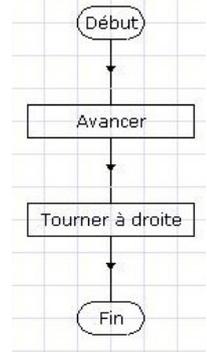
L'orientation du robot

Les tests xRobot>5 et yRobot>5 permettent de savoir où se situe le robot afin de l'orienter à droite ou à gauche pour lui faire faire le carré

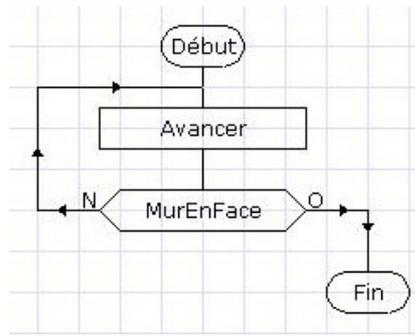


Les blocs à utiliser

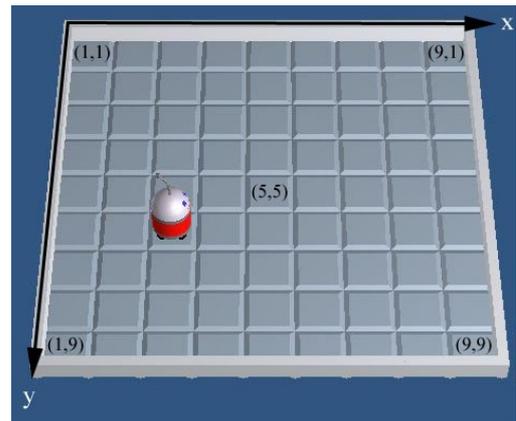
Tout programme ou sous programme doit avoir un début et une fin.



Les **tests** ont deux résultats : oui ou non.
 MurEnFace
 MurADroite
 MurAGauche
 Robotsuruneprise

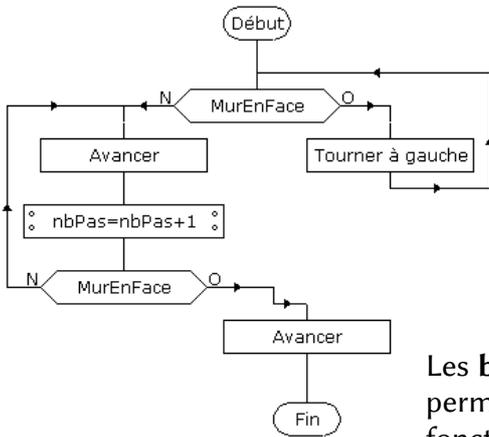
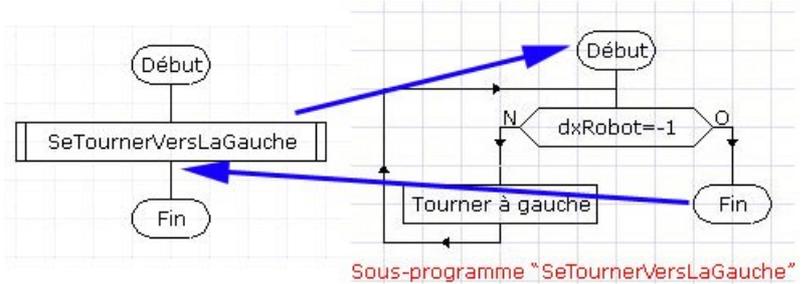


La **position** du robot est donnée par les deux mots-clé xRobot et yRobot. Pendant l'exécution, xRobot et yRobot ont les valeurs x, y de la case occupée par le robot. Dans l'exemple ci-dessus, xRobot = 3 et yRobot = 6

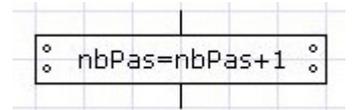


La **direction** suivant laquelle le robot est orienté est donnée par les deux mots-clé dxRobot et dyRobot.
 si le robot est tourné vers la droite du terrain: dxRobot vaut 1 et dyRobot vaut 0

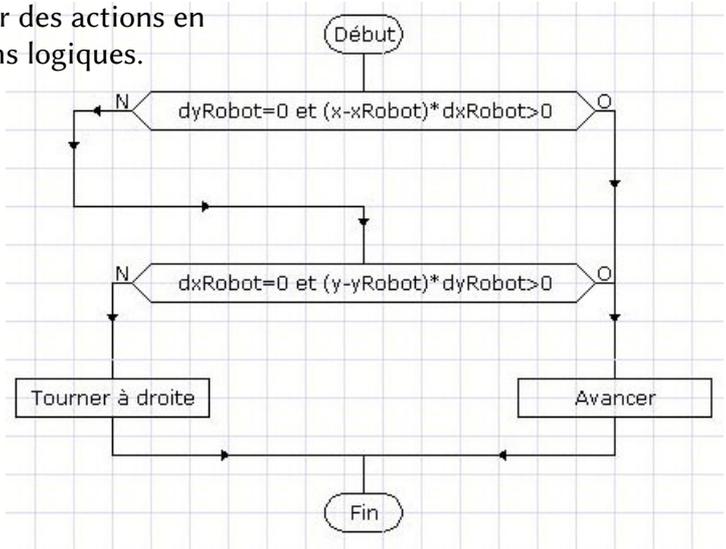
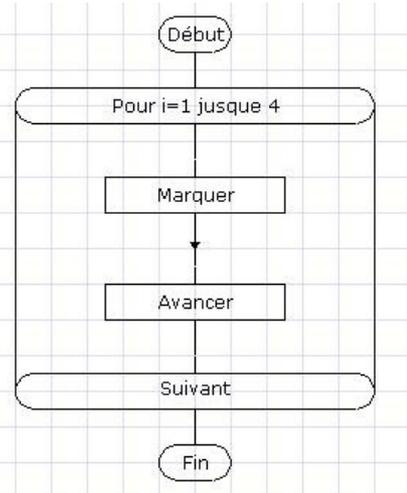
Les **sous programmes** permettent d'alléger le programme principal.



Les **variables** sont définies par le programmeur.



Les **boucles POUR** et **TANT QUE** permettent de répéter des actions en fonction de conditions logiques.



Langage du robot

Mot-clef	Commentaire
Début	Premier bloc exécuté dans un programme ou sous-programme
Fin	Fin de l'exécution du programme ou du sous-programme courant
Lire	Instruction d'entrée permettant d'entrer une valeur de variable au clavier Exemple : Lire maVariable
Afficher	Instruction de sortie permettant d'afficher des textes et/ou des valeurs de variables Exemple : Afficher "La valeur de A est :", A
TantQue	Utilisé dans l'en-tête d'une boucle TantQue Ce mot -clef doit être suivi d'une expression logique. Exemple : TantQue Non MurEnFace
FinTantQue	Fin d'une boucle TantQue
Pour	Utilisé dans l'en-tête d'une boucle Pour. Exemples d'instructions Pour : Pour k = 1 jusque 10 Pour i = 5 jusque 25 pas 5
Jusque	Utilisé dans l'en-tête d'une boucle Pour
Pas	Utilisé dans l'en-tête d'une boucle Pour
Suivant	Fin d'une boucle Pour
Avancer	Commande robot : le robot avance d'une case en avant
TournerADroite	Commande robot : le robot tourne à droite en restant dans la même case
TournerAGauche	Commande robot : le robot tourne à gauche en restant dans la même case
Marquer	Commande robot : le robot dessine une marque sur la case qu'il occupe
Démarquer	Commande robot : le robot efface la marque sur la case qu'il occupe
Recharger	Commande robot : si le robot est sur une prise d'énergie, son énergie est remise à la valeur maximum
PrendreBallon	Commande robot : le robot prend le ballon, si le ballon est posé dans la case occupée par le robot
PoserBallon	Commande robot : si le robot tient le ballon; il le pose dans la case où il se trouve
LancerBallon	Commande robot : si le robot tient le ballon, il le lance à trois case devant.
MurEnFace	Fonction logique retournant la valeur vrai si le robot est en face d'un mur
MurADroite	Fonction logique retournant la valeur vrai s'il y a un mur à droite du robot
MurAGauche	Fonction logique retournant la valeur vrai s'il y a un mur à gauche du robot
Sorti	Fonction logique retournant la valeur vrai si le robot est sur une case extérieure, la valeur faux autrement
RobotSurUnePrise	Fonction logique retournant la valeur vrai si le robot est sur une case contenant une prise d'énergie
CaseMarquée	Fonction logique retournant vrai si le robot est sur une case avec une de ses marques dessinée
CaseDevantOccupée	Fonction logique retournant la valeur vrai si la case devant le robot est occupée par un autre robot.
BallonSurLeSol	Fonction logique retournant la valeur vrai si le ballon est posé sur le sol.
DistanceMur	Fonction retournant le nombre de cases entre le robot et le mur le plus proche devant le robot
Energie	Fonction retournant un nombre représentant l'énergie restante du robot
xRobot	Fonction retournant la position x du robot sur le terrain
yRobot	Fonction retournant la position y du robot sur le terrain
dxRobot	Fonction retournant la direction du robot suivant l'axe x
dyRobot	Fonction retournant la direction du robot suivant l'axe y
xBallon	Fonction retournant la position x du ballon sur le terrain
yBallon	Fonction retournant la position y du ballon sur le terrain
Et	Opérateur logique
Ou	Opérateur logique
Non	Opérateur logique
Vrai	Constante logique
Faux	Constante logique
=	Opérateur d'affectation figurant dans une instruction d'affectation
()	Parenthèses utilisables dans les expressions numériques et logiques
+ - * /	Opérateurs numériques produisant un résultat numérique avec des opérandes numériques
= > >= < <=	opérateurs de comparaison produisant un résultat logique avec des opérandes numériques