

Pilotage par ordinateur


L'organigramme



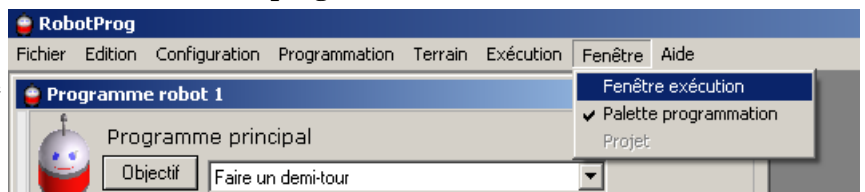
L'objectif de la séance est de réussir à programmer un petit robot virtuel en utilisant des organigrammes afin qu'il réalise des tâches simples.

10 exercices de programmation sont proposés. 4 d'entre eux sont notés

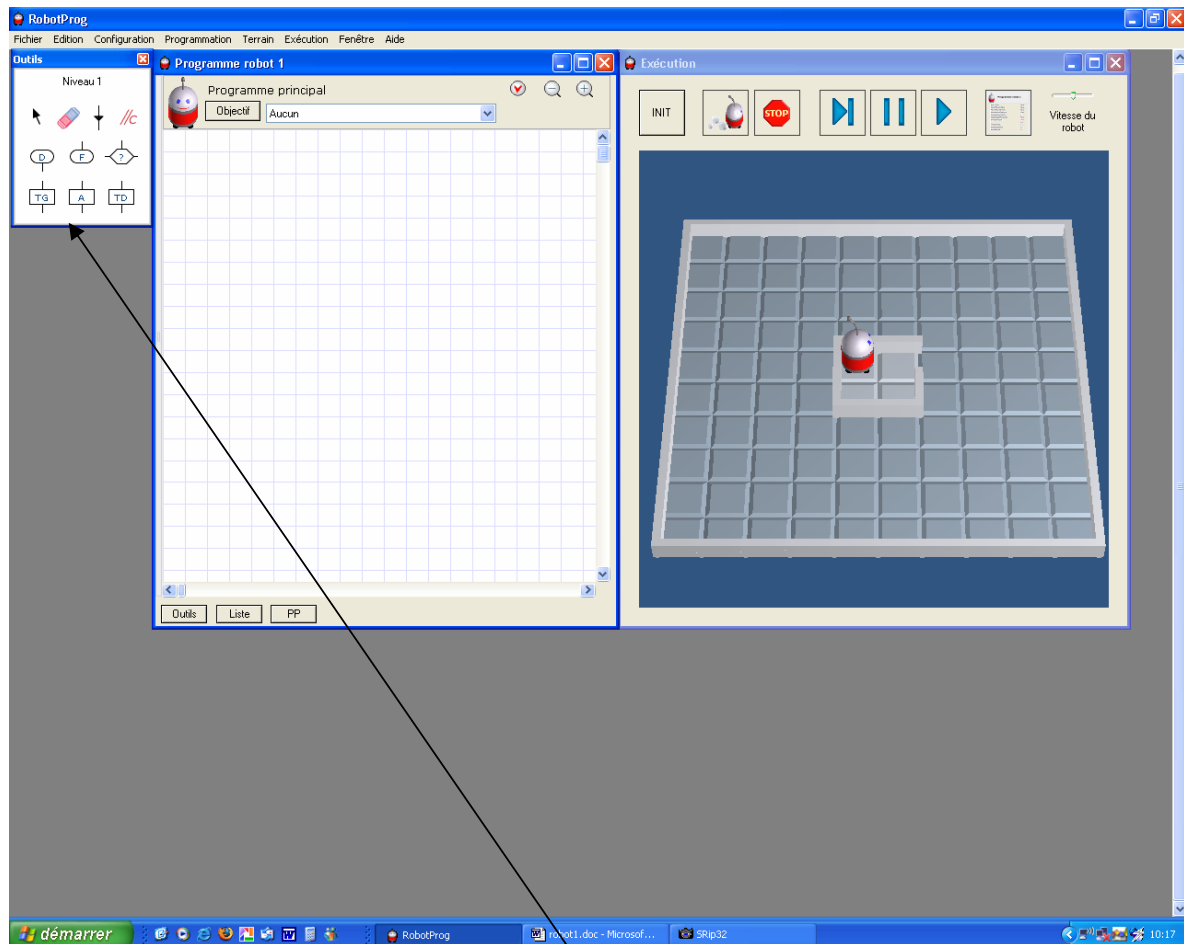
I Lancement et configuration du programme :

1- Clique Démarrer/Programmes/3èmes/Automatismes/  **Robotprog.**

2- Dans le menu **Fenêtre**, sélectionne **Fenêtre exécution** afin de faire apparaître le terrain d'évolution du robot.



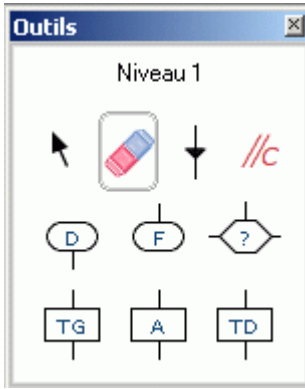
3- Tu dois obtenir l'écran suivant :






II Prise en main du logiciel :

Pour programmer ton robot, tu disposes de la fenêtre **Outils** qui contient tous les blocs nécessaires à la réalisation de tes organigrammes. Pour utiliser un bloc, il suffit de cliquer dessus, puis de cliquer dans la fenêtre **Programme** à l'endroit où tu veux le placer.

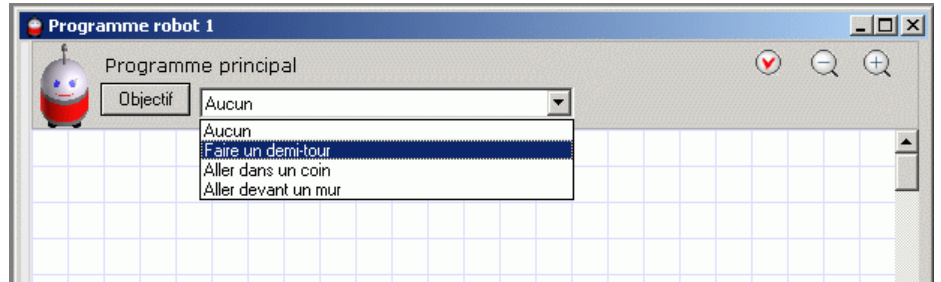
Pour l'effacer, il suffit de cliquer sur la **gomme**, puis sur le bloc à effacer.



- D** : Début
- F** : Fin
- TG** : Tourne à gauche
- A** : Avance
- TD** : Tourne à droite
-  : Effacer un élément
-  : Créer une liaison entre deux blocs
-  : Sélectionne un bloc (pour écrire dedans par exemple)

III Mon premier programme :

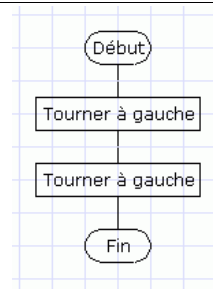
1. Dans la fenêtre programme, sélectionne **Faire un demi-tour** dans le menu déroulant.




*L'objectif de l'exercice **Faire un demi-tour** est le suivant :*

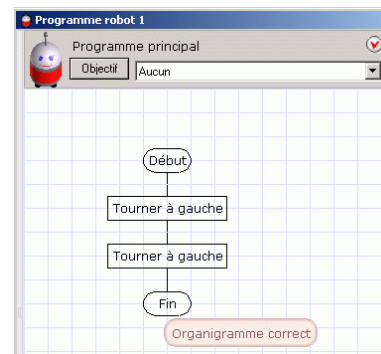
A la fin de l'exécution, le robot doit occuper la même case qu'au début et doit être dirigé dans la direction opposée à celle du début.

2. Reproduit à présent le programme ci-contre



3. Clique ensuite sur le bouton . Si l'organigramme est correctement réalisé, apparaît dans la fenêtre.

Organigramme correct





4. Dans la fenêtre **Exécution**, clique sur le bouton **Init** puis sur celui sur lequel figure le **petit robot**.



Le programme s'exécute. Le robot tourne deux fois sur lui-même et saute de joie car il a réussi son demi-tour.

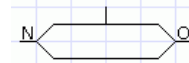
9- Clique n'importe où dans la fenêtre **Programme**.

Utilise l'outil flèche pour tout sélectionner  ou bien la commande Edition / Tout sélectionner

puis utilise la gomme pour tout effacer 

IV Tests et conditions logiques

Pour les exercices suivants, tu vas devoir utiliser le symbole .



et utiliser les mots-clé suivants :

- MurADroite (Y a-t-il un mur à droite ?)
- MurAGauche (Y a-t-il un mur à gauche ?)
- MurEnFace (Y a-t-il un mur en face ?)


} Écriture sans espaces

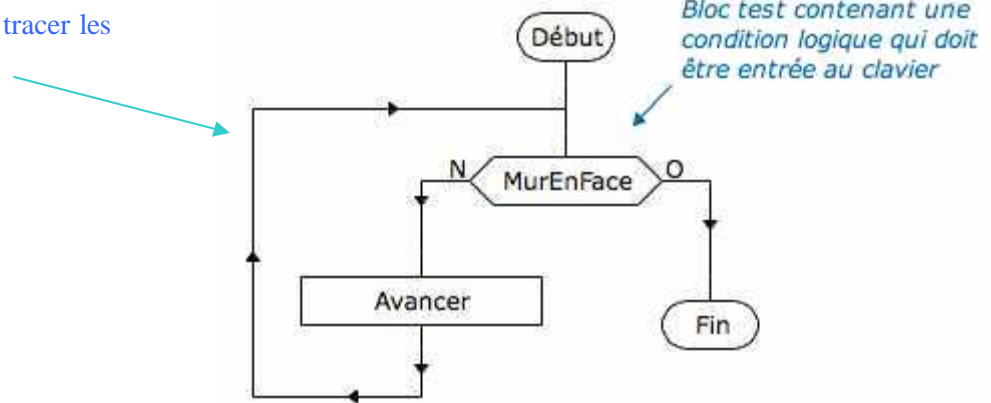
1. Aller devant un mur :





Objectif. construire un programme permettant d'aller jusqu'à un mur.

- Sélectionne **Aller devant un mur** dans le menu déroulant à côté du bouton **Objectif**
- Construis l'organigramme représenté ci-dessous

Utilise l'outil  pour tracer les liaisons



- Lance son exécution en cliquant  puis 

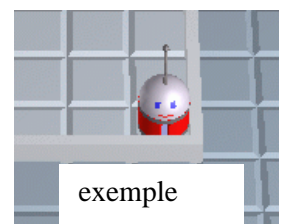
A la fin de l'exécution, le robot doit se trouver sur une case voisine d'un mur et doit faire face au mur. La case finale doit être différente de la case initiale.

2. Aller dans un coin :

Objectif. Depuis une situation initiale quelconque, amener le robot dans un coin extérieur.
La case finale doit être différente de la case initiale




Dans la fenêtre **Programme**, efface l'organigramme précédent et sélectionne **Aller dans un coin** dans le menu déroulant à côté du bouton **Objectif**.



- Construis l'organigramme qui te permet d'atteindre l'objectif



La méthode pour aller dans un coin peut être la suivante :

1. le robot va d'abord devant un mur → c'est le programme précédent

2. Il tourne à gauche ou à droite  ou 
3. puis il longe le mur, 
4. jusqu'à ce qu'il arrive en face d'un autre mur. → programme précédent

- Lance son exécution en cliquant  puis 

Tu montreras ton travail au professeur. Ce 3^{ème} programme vaut 4 points

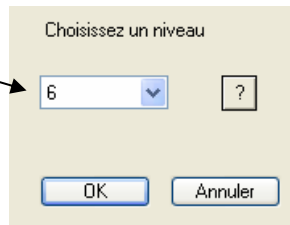
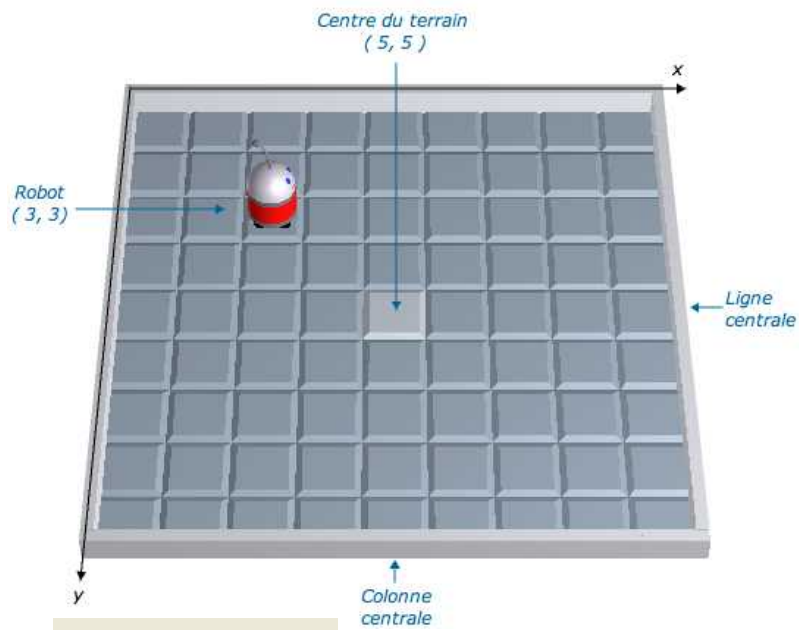
Tu effaceras le programme *pour pouvoir passer à la suite...*

V) POSITION ET DIRECTION DU ROBOT

Une case du terrain est repérée par ses coordonnées **x, y**. x et y sont des nombres entiers positifs.

La case arrière gauche a pour coordonnées **(1, 1)**, la case arrière droite **(9,1)** pour le terrain représenté ci-contre.

Pour les exercices suivants tu vas devoir changer de niveau. Pour cela, utilise le menu Configuration/Niveau puis sélectionne le niveau 6



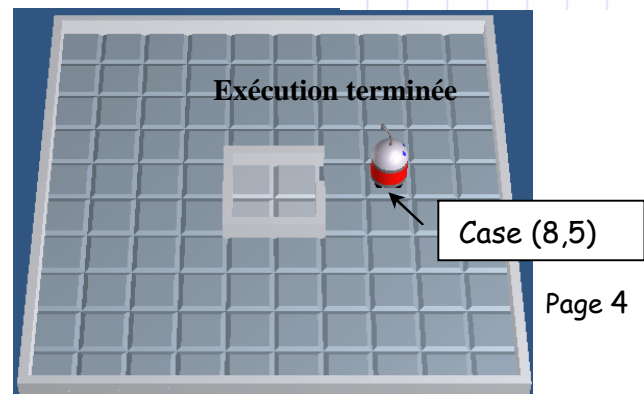
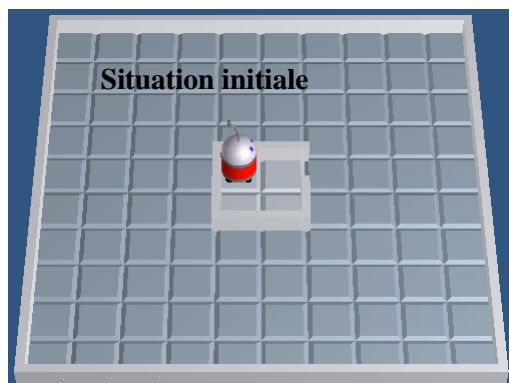
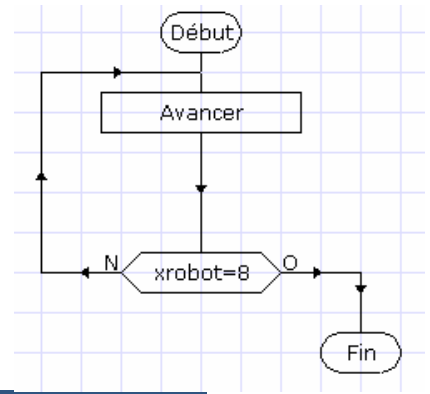
1. Déplacer le robot jusqu'à une case précise :

Position du robot :

La position du robot est donnée par les deux mots-clef **xRobot** et **yRobot** utilisables dans un programme. Pendant l'exécution, xRobot et yRobot ont les valeurs x, y de la case occupée par le robot.

- Construis l'organigramme représenté ci-contre et lance son exécution

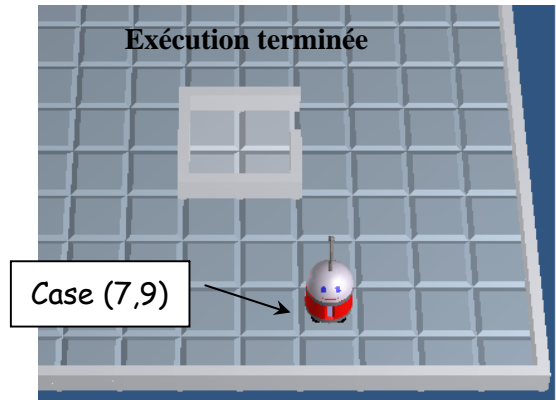
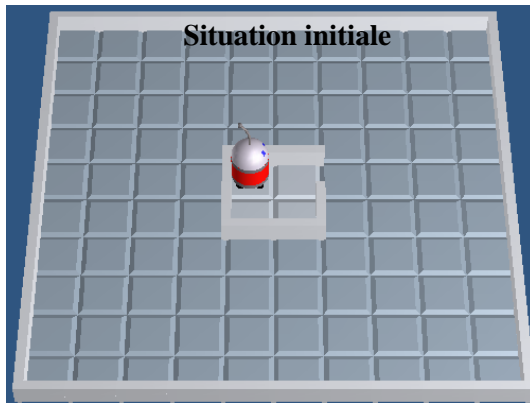
Explications : Depuis sa position initiale, le robot avance jusqu'à la case ayant pour valeur x= 8



Objectif : depuis sa situation initiale case (5,5), déplacer le robot jusqu'à la case (7,9)

Note : la fonction Avancer ne devra être utilisée que 2 fois !

Tu utiliseras les fonctions xRobot et yRobot et Avancer



Tu montreras ton travail au professeur. Ce 5^{ème} programme vaut 4 points

2. Donner une direction au robot :

Direction du robot :

La direction suivant laquelle est orienté le robot est donnée par deux mots-clef **dxRobot** et **dyRobot** utilisables dans un programme. Les valeurs de dxRobot et dyRobot correspondent à la variation de xRobot et yRobot quand le robot avance d'une case devant lui :

- si le robot est tourné vers la droite du terrain : **dxRobot = 1**



orientation est

- si le robot est tourné vers la gauche du terrain : **dxRobot = - 1**



orientation ouest

- si le robot est tourné vers l'avant du terrain : **dyRobot = 1**



orientation sud

- si le robot est tourné vers l'arrière du terrain : **dyRobot = - 1**

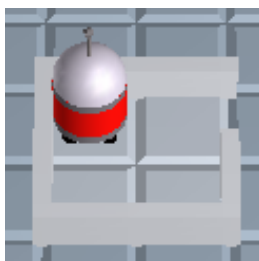


orientation nord

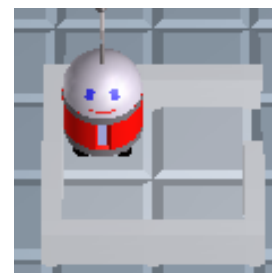
Remarque : dxRobot et dyRobot n'ont comme valeurs possibles que 0, 1 ou -1. L'une des deux valeurs est nulle et l'autre non nulle.

Objectif : depuis une situation initiale quelconque (il y a 4 directions possibles), le robot doit avoir comme direction l'avant du terrain

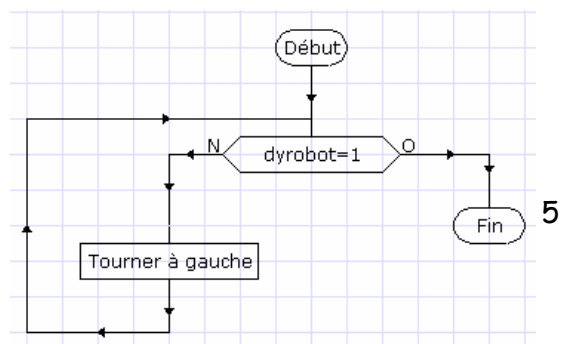
Situation initiale possible (obtenue par clics de souris sur le robot)



Exécution terminée



- Efface le programme précédent (si ce n'est pas déjà fait)
- Construis l'organigramme représenté ci-contre et lance son exécution

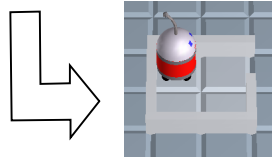


Explications : Ici, l'organigramme utilise le mot-clef dyRobot.

Le robot est tourné vers l'avant si dyRobot a la valeur 1

Si dyRobot ne vaut pas 1, on fait tourner le robot d'un quart de tour et on recommence le test.

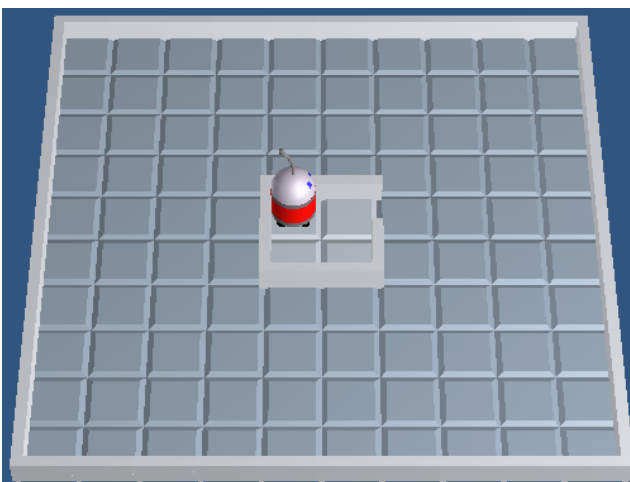
Objectif : depuis une situation initiale quelconque (il y a 4 directions possibles), le robot doit avoir comme direction la droite du terrain soit une orientation est



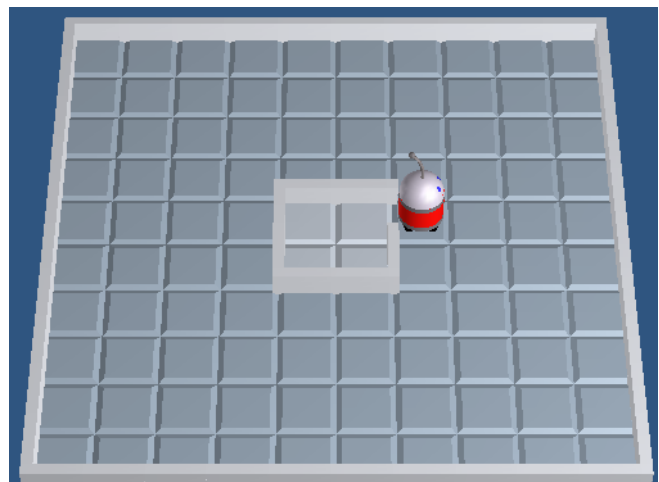
Tu montreras ton travail au professeur. Ce 7^{ème} programme vaut 3 points

3. **Case extérieure**

Situation initiale



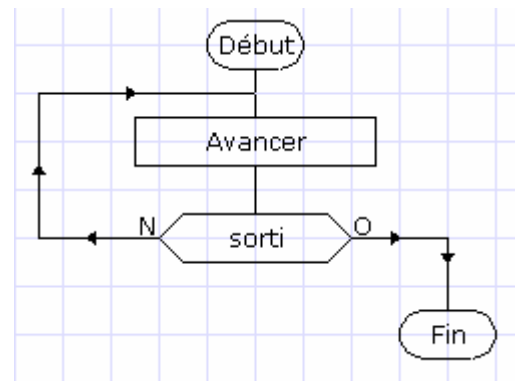
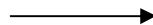
Exécution terminée



Tu vas utiliser le mot-clef : Sorti

Sorti : fonction logique retournant la valeur vrai si le robot est sur une case extérieure, la valeur faux autrement

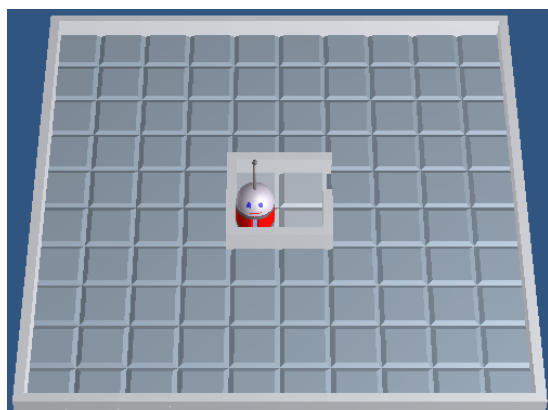
- Efface le programme précédent (si ce n'est pas déjà fait)
- Construis l'organigramme représenté ci-contre et lance son exécution



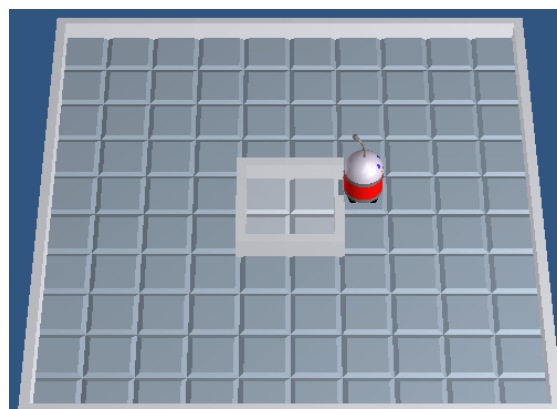
Objectif : depuis une situation initiale *quelconque* du robot dans sa zone, (16 possibilités: 4 cases et 4 directions) le robot doit se trouver à l'extérieur une fois l'exécution terminée. Tu utiliseras la fonction "Sorti"

Exemple

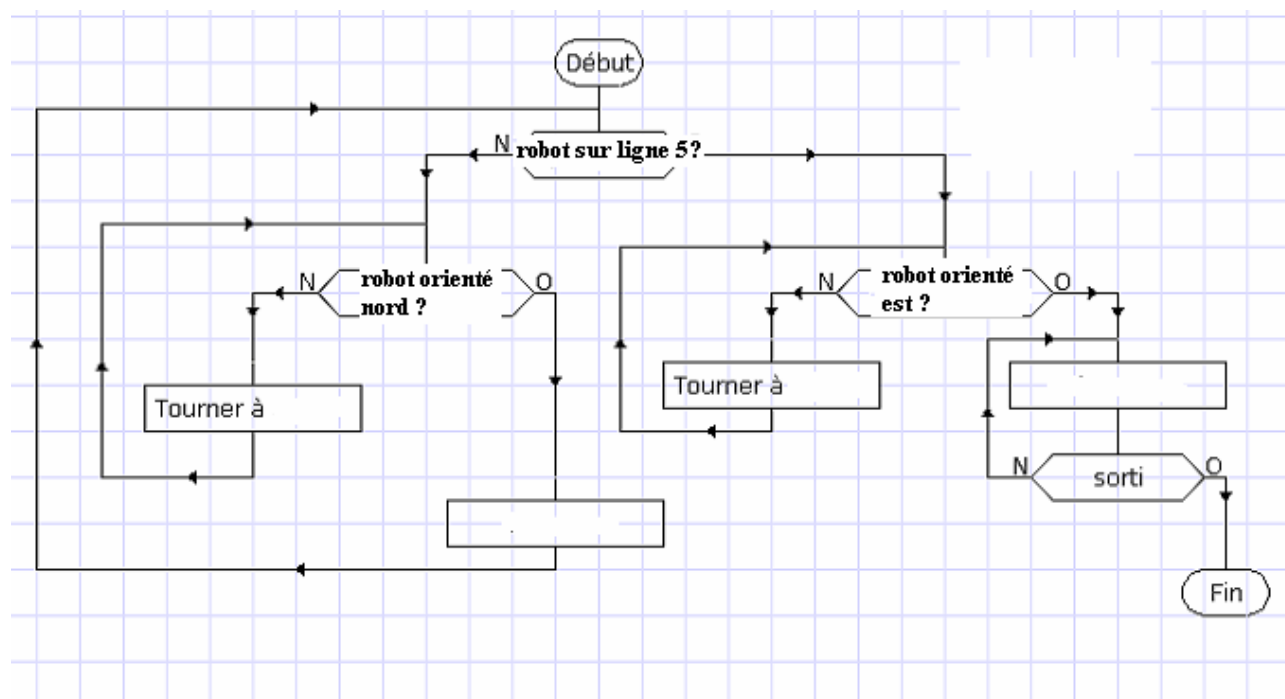
Situation initiale possible



Exécution terminée



Aide: teste d'abord la position du robot puis sa direction. Il faut qu'il soit sur la ligne 5 et orienté vers la droite pour pouvoir sortir. Voir ci-dessous :

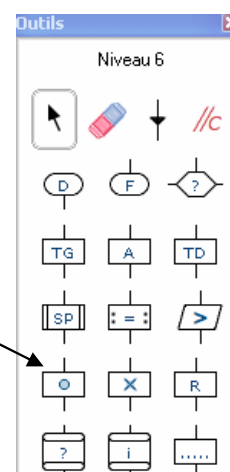


Tu montreras ton travail au professeur*. Ce 9^{ème} programme vaut 4 points

V LE ROBOT DESSINE

Objectif : faire dessiner par le robot une ligne de 4 cases.

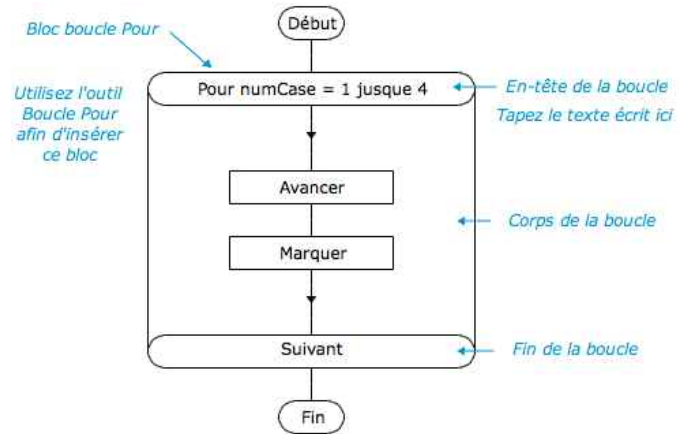
Pour dessiner une marque sur une case le robot utilisera l'instruction "Marquer" puis il devra avancer pour marquer la case suivante.



Pour éviter des organigrammes trop longs, surtout pour des actions qui doivent être répétées, tu utiliseras ici une boucle qui permet de répéter un nombre de fois donné une même partie de programme.

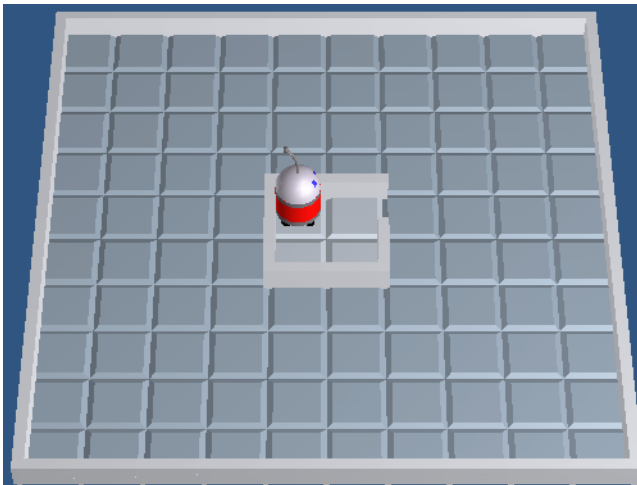
- Efface le programme précédent
- Construis l'organigramme représenté ci-contre et lance son exécution

Explications : depuis sa case de départ, le robot avance 4 fois et marque 4 cases

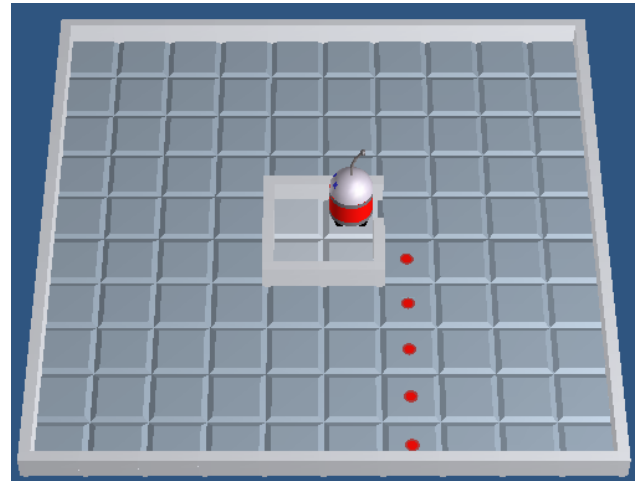


Objectif : depuis sa situation initiale (case 5,5 et orienté vers la droite, donc à l'est), le robot doit dessiner à l'extérieur un trait vertical de 5 cases puis revenir dans sa zone.

Situation initiale



Exécution terminée



Tu montreras ton travail au professeur. Ce 10^{ème} programme vaut 5 points

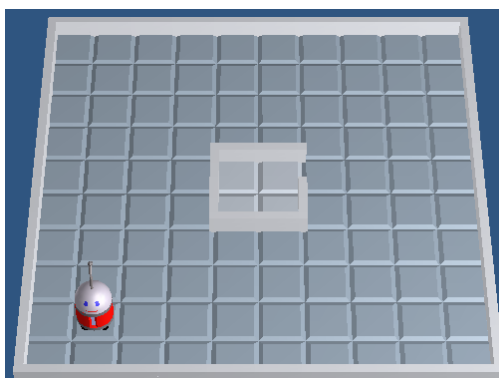
Pour les plus rapides :

Objectif : depuis une situation quelconque du robot hors de sa zone, ramener le robot à l'intérieur de sa zone

Tu montreras ton travail au professeur. Ce 11^{ème} programme vaut 20 points

Exemple

Situation initiale possible



Exécution terminée

