












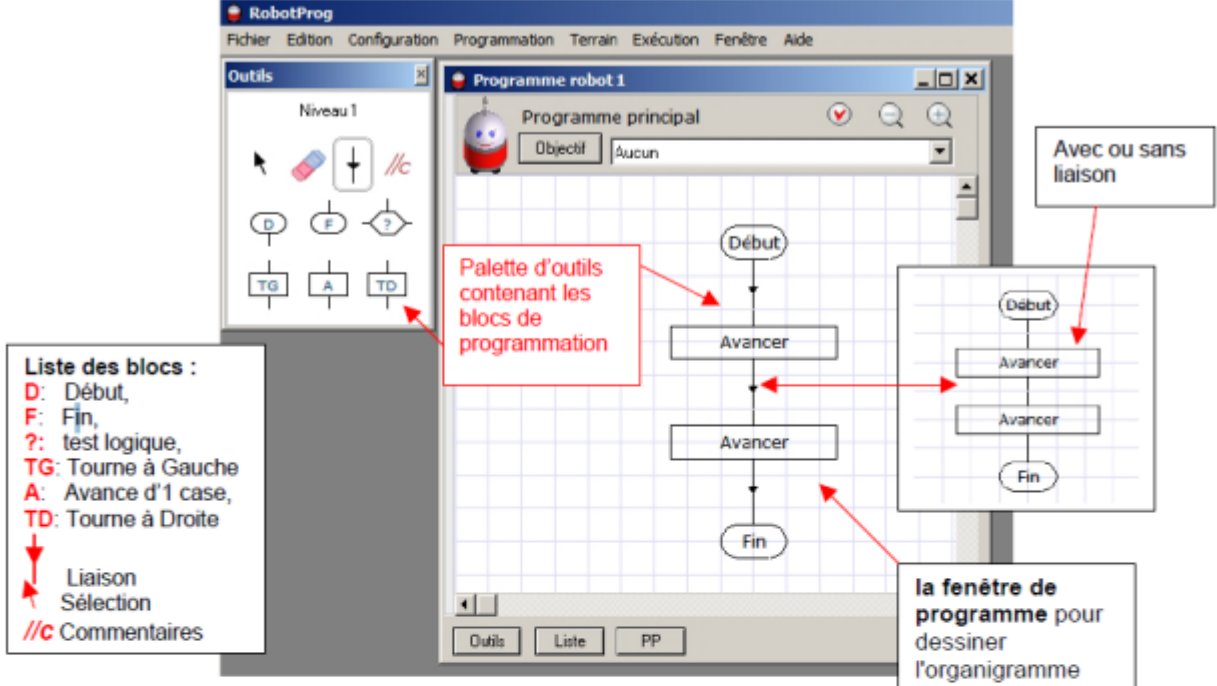


Date : .../.../.....	DEBUTER LA PROGRAMMATION AVEC ROBOTPROG	Classe :						
Thème : Langage informatique		Nom :						
		Prénom :						
Mise en situation : <ul style="list-style-type: none"> Vous êtes technicien dans une société de domotique et vous devez vous familiariser avec de nouveaux robots programmables qui seront intégrés par la suite dans les diverses solutions domotiques (ménage, aide à la personne ...) que vous proposerez à vos clients. Vous programmerez le robot pour le tester ce module dans différentes situations simples 								
Objectifs : <ul style="list-style-type: none"> Programmation 								
Matériels : <ul style="list-style-type: none"> logiciel RobotProg 								
Ressources : <ul style="list-style-type: none"> logiciel RobotProg 								
Savoirs associés	Compétences visées	Résultats attendus	Evaluation					
								
S2 – 2 Traitement de l'information	C1-1 Appréhender la mise en œuvre d'un projet simulé ou réel d'installation d'un système.	-Le rôle de tout ou partie des éléments répertoriés est énoncé - Les traces d'échange entre équipements sont exploitées						
S2 - 2 Les fonctions logiques de base en programmation	C3-3 Effectuer les tests nécessaires à la validation du fonctionnement des équipements							
S4 - 2 : Langage de programmation	C4-4 Installer, configurer les éléments du système et vérifier la conformité du fonctionnement.							
<div> <div>  Trop d'erreurs </div> <div>  Réussite partielle </div> <div>  Non-évaluable </div> </div> <div> <div>  En cours d'acquisition </div> <div>  Réussite totale </div> <div>  Absent </div> </div>								

1.Démarrez le logiciel RobotProg.

Lancez RobotProg en double-cliquant sur l'icône 



Liste des blocs :
D: Début,
F: Fin,
?: test logique,
TG: Tourne à Gauche
A: Avance d'1 case,
TD: Tourne à Droite
 ↳ Liaison
 ↳ Sélection
 //c Commentaires

Palette d'outils contenant les blocs de programmation

Avec ou sans liaison


la fenêtre de programme pour dessiner l'organigramme

1 Dessiner un organigramme

- Pour choisir un **bloc** dans la palette d'outils, **cliquer** sur le bloc ; puis **déplacer** le pointeur de souris dans la fenêtre de programme ; puis **cliquer** de nouveau pour y placer le bloc
- Pour effacer un bloc, **cliquer** sur la **gomme** puis **cliquer** sur le bloc à effacer.
- Pour **lier** les blocs entre eux, **cliquer** sur l'**outil liaison** dans la palette , puis **cliquer** sur une sortie de bloc, **déplacer** la souris et **cliquer** sur l'entrée du bloc suivant.
- On peut aussi joindre directement un bloc à un autre au moment du placement mais on verra plus tard qu'il est préférable d'utiliser les liaisons. (voir ci-dessus)



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

Exercice 1	Reproduire l'organigramme ci-dessus. Vérifier la validité de l'organigramme en cliquant sur le bouton 
-------------------	--

Validation par enseignante

2 Affichage du terrain du robot

Pour visualiser l'exécution du programme, il faut faire apparaître le terrain.
Cliquer sur le menu **Fenêtre > Fenêtre Exécution ...**

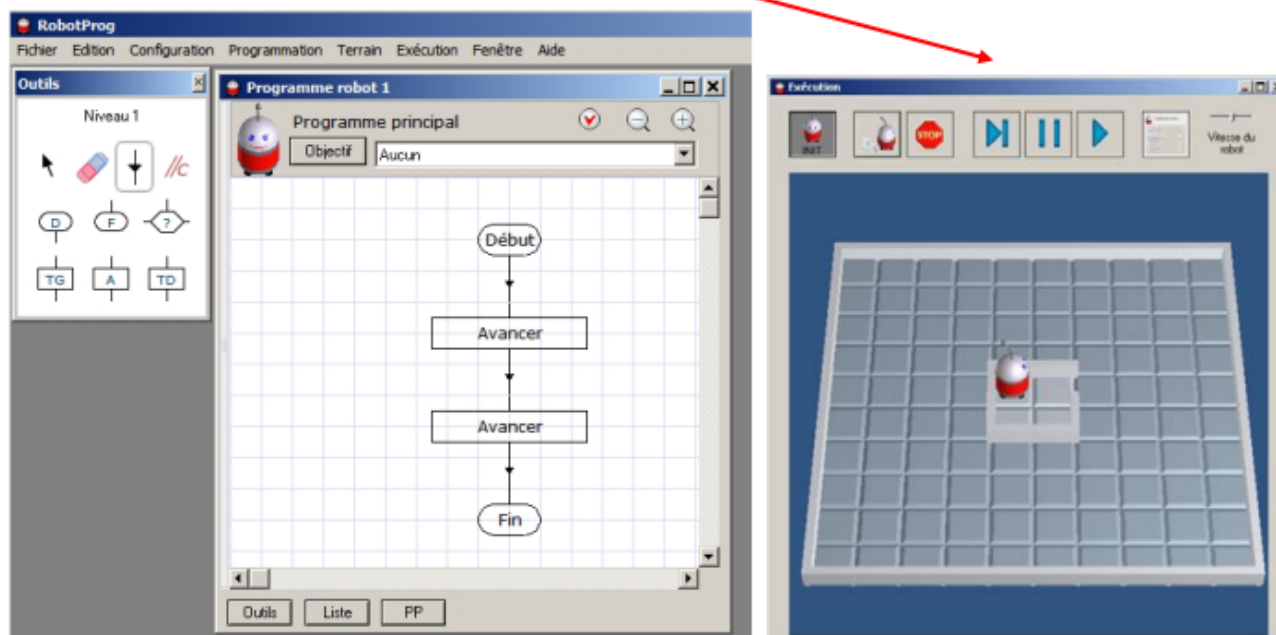



Figure 2

3 Initialisation du programme

Cliquez sur le bouton **INIT** (ou **Exécution > Initialisation**). Le programme créé d'après l'organigramme est alors vérifié .

Si le programme ne contient pas d'erreur, 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 devrez d'abord corriger l'erreur.

Il faut maintenant positionner correctement le robot dans une case de départ.

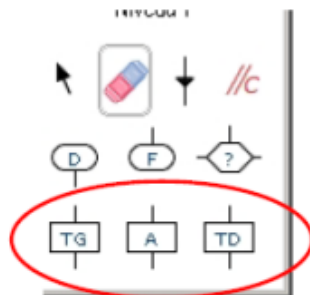
- Pour positionner le robot sur une case, cliquez sur la case choisie.
- Pour le tourner d'un quart de tour, cliquez sur le robot.

4 Lancement du programme

- Cliquez sur le bouton ou bien choisissez le menu **Exécution > Lancer**. (Impossible de cliquer sur ce bouton si vous n'avez pas tout d'abord initialisé le programme.)
- Initialisez et lancez votre programme. (Un rectangle rouge qui se déplace dans la **fenêtre de programme**.)
- Essayez plusieurs positions de départ du robot et plusieurs orientations (G/D).
- Dans votre espace de travail (2SN\2SN1GA ou 2SN1GB), créez un dossier **Robotprog-NOM** et
- Sauvegardez votre programme (**menu Fichier**) et sous le nom **PROG1**
- Quittez le programme (**menu Fichier-FERMER**).

5 Les déplacements du robot

Pour déplacer le robot, on dispose de trois commandes :



Avancer (A) = avance d'une case devant lui

Tourner à droite (TD) = 1/4 de tour à Droite

Tourner à gauche (TG) = 1/4 de tour à Gauche



Si le robot se heurte au mur, on dit qu'il y a une erreur d'exécution : le programme s'arrête.

Exercice 2a

Le robot doit parcourir le trajet de la figure 3 ci-dessous. Écrivez le programme. Sauvegardez ensuite votre programme sous le nom **Trajet1**.

Trajet à reproduire :

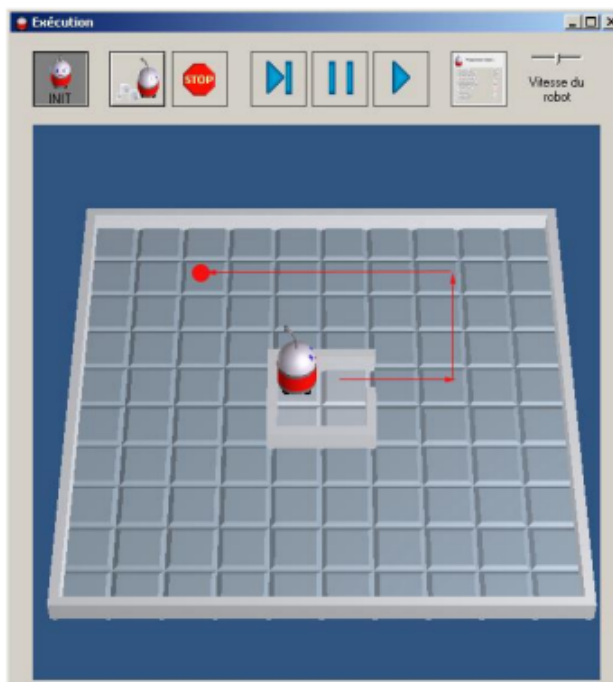
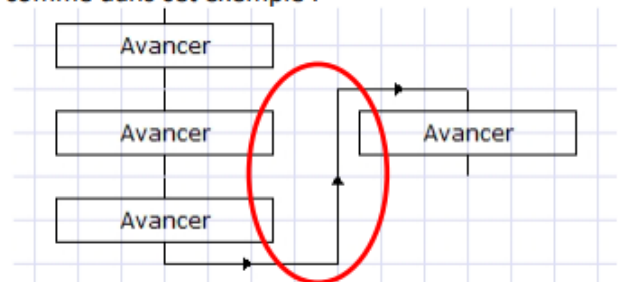


Figure 3

Remarque :

Si la fenêtre est trop petite, utilisez les liaisons comme dans cet exemple :



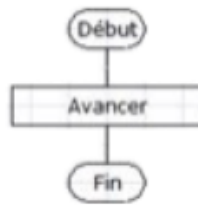
Exercice 2b

- Écrivez un nouveau programme où le robot avance de trois cases, fait demi-tour, et retourne à sa case de départ, en position initiale exacte. Exécutez-le pour vérifier. (Attention, lisez bien la consigne).
- Sauvegardez ensuite votre programme sous le nom de **Trajet2**.

Validation par enseignante:

4. Vous allez devoir réaliser un organigramme « Avancer d'un pas ». Pour cela, suivez les étapes ci-dessous.

4.1 Réalisez l'organigramme ci-dessous afin que RobotProg avance d'un pas :



- 4.2 Réalisez maintenant l'organigramme afin que le robot avance de 5 pas (jusqu'au mur). Faites ensuite un copier-coller de votre organigramme dans le cadre ci-contre.



1

5. Réalisation de l'organigramme « **Le bloc Test** ». Suivez les étapes ci-dessous.

Le terrain peut être modifié, **mais avant l'exécution d'un programme.**

On peut ajouter ou retirer des murs à l'intérieur du terrain (mais pas ceux l'entourant).

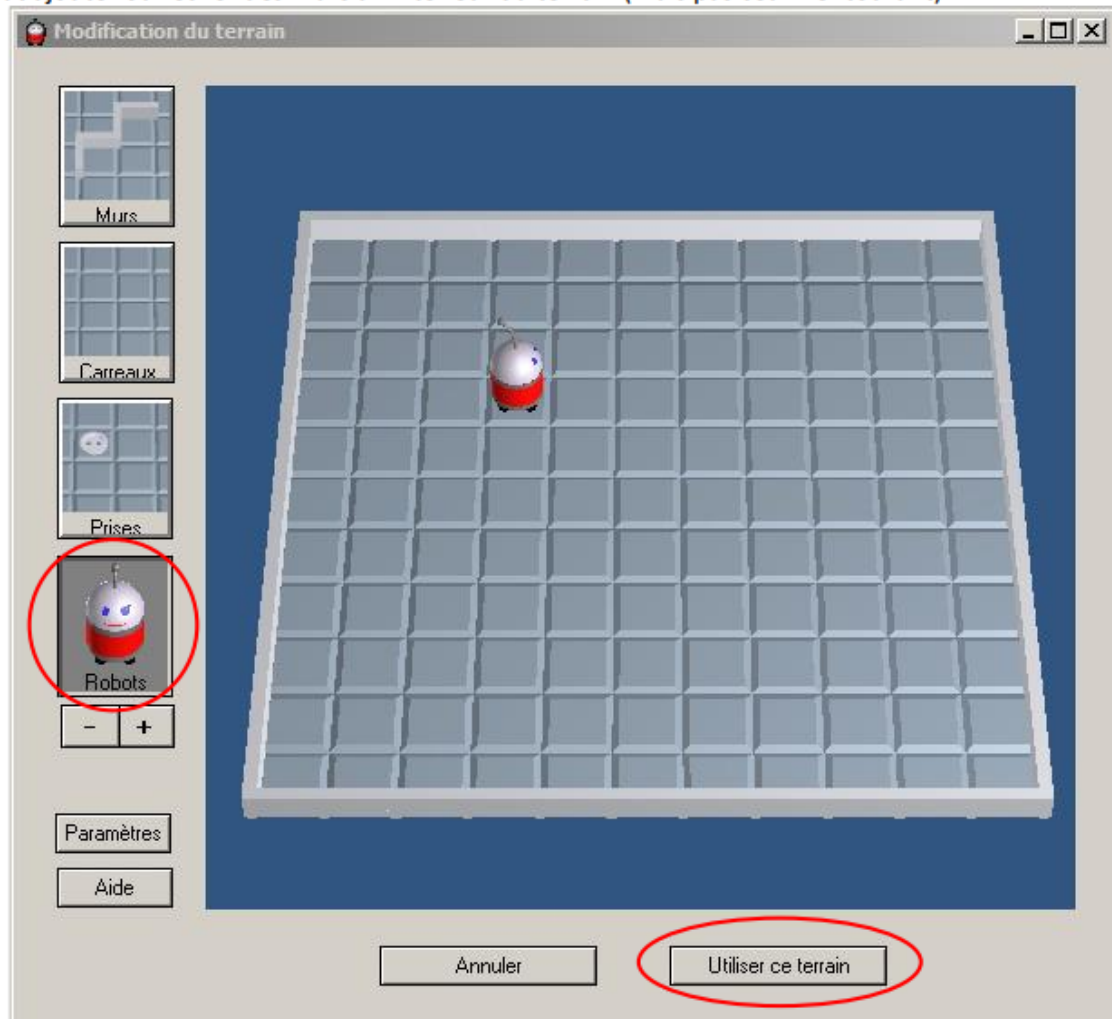


Figure 4

1. Menu **Terrain** > **Modifier**. La **fenêtre d'édition de terrain** apparaît.
2. Menu **Terrain** > **Nouveau** pour créer un nouveau terrain. La fenêtre des paramètres du terrain apparaît.
3. Choisir une largeur et une hauteur de 11 cases
4. Choisissez 1 seul robot et une énergie de 1000.
5. Validez en cliquant le bouton **OK**.
6. Cliquez sur le bouton **Robots** puis cliquez sur le bouton **+** en dessous : un robot apparaît sur le terrain.
7. Cliquez sur une case pour placer le robot à sa position initiale pour le début de l'exécution.
8. Enregistrer (**Terrain-Enregistrer sous**) sous le nom **Terrain11x11SansObstacles**
9. Fermez la fenêtre en cliquant sur le bouton Utiliser ce terrain. Si vous affichez le terrain avec le menu **Fenêtre** > **Fenêtre exécution**, vous constaterez que le terrain a bel et bien été modifié.

Exercice 3	Construisez le terrain comme décrit ci-dessus (figure 4) et enregistrez-le (Terrain-Enregistrer sous) sous le nom Terrain11x11SansObstacles .
-------------------	---

Validation par enseignante:

--

7 Conditions logiques et tests

Objectif : Faire tester un obstacle au robot

7.1 Conditions logiques

Une condition logique est une expression donnant un résultat vrai ou faux.

On verra par la suite que l'on peut combiner différents mots-clefs avec les opérateurs logiques **Et**, **Ou**, **Non**.

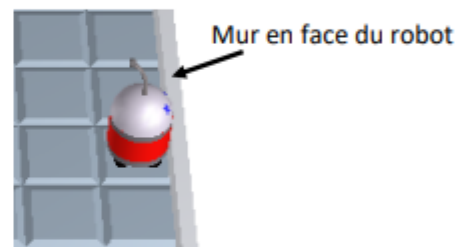
Par exemple : (*MurEnFace Et MurADroite*) **Ou** Non *MurAGauche*.

1. Ouvrez le fichier « Mots-clefs-ROBOTPROG.pdf »


Dans ce fichier, vous trouverez des mots-clefs pour le logiciel RoboProg, ces mots-clefs sont des définitions existantes et reconnus par ce logiciel RobotProg.

Par exemple :

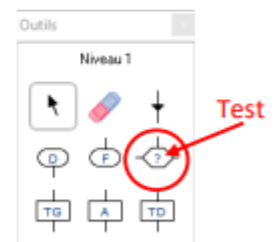
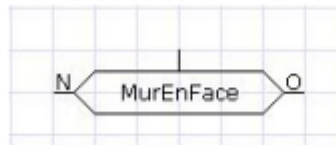
Le mot-clef « **MurEnFace** » signifie que le robot a atteint le mur en face de lui.




7.1 Bloc « Test »

Pour identifier le bloc « Test », placez votre souris sur le symbole  situé dans la fenêtre « outils ».

Dans l'organigramme, un bloc test a cette forme suivante :



Pour modifier le texte contenu dans un bloc test, il faut choisir l'outil sélection  et faire un double-clic sur le bloc.

Si le résultat est vrai, l'exécution se poursuit après la sortie marquée **O** (Oui signifie vrai) ;

Si le résultat est faux, l'exécution se poursuit après la sortie marqué **N** (Non ou faux).

Dans l'exemple ci-contre, on utilise la condition logique MurEnFace. (À écrire dans le bloc).

Si le robot est en face d'un mur, Alors le programme ira vers Oui et donc s'arrêtera (Fin), Sinon le programme ira vers Non et donc le robot avancera et recommencera le test jusqu'à ce qu'il rencontre un mur.

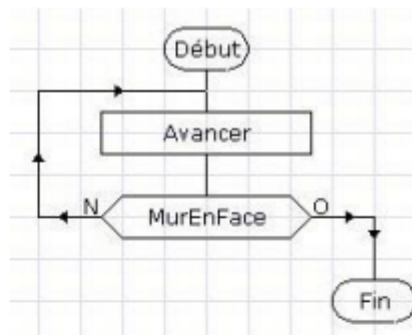



Figure 5

Exercice 4a	<p>Objectif : Aller jusqu'à un mur et s'arrêter Vous utiliserez le terrain Terrain11x11SansObstacles. Réalisez l'algorithme de la figure 5. Le programme doit fonctionner quelles que soient la position et la direction initiale du robot, sans erreur d'exécution. Enregistrer le programme sous le nom TestMur</p>
--------------------	---

Exécution pas à pas :

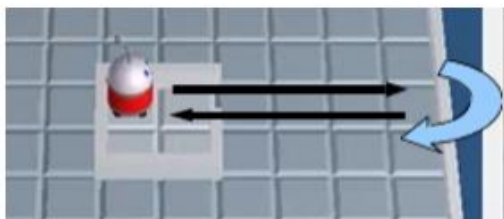
Visualisation de l'exécution pas à pas Pendant l'exécution d'un programme, vous pouvez cliquer sur le bouton Pause dans la fenêtre exécution, puis faire exécuter les instructions les unes après les autres en cliquant sur le bouton Exécution pas à pas . Vous trouverez la liste complète des mots-clefs dans le document accessible par le menu Aide > Résumé du langage du robot.

Exercice 4a	<p>Objectif : Modifier le programme TestMur pour Aller dans un coin et s'arrêter Vous utiliserez le terrain Terrain11x11SansObstacles. Le programme doit fonctionner quelles que soient la position et la direction initiale du robot, sans erreur d'exécution. Testez plusieurs orientations initiales du robot. Enregistrer le programme sous le nom TestCoin</p>
--------------------	--

Exercice 4a	<p>Objectif : Modifier le programme TestCoin pour créer un programme qui permet au robot d'aller dans les 4 coins et s'arrêter. Modifiez la position du robot et testez que quelle que soit sa position, il passe par les 4 coins. Enregistrer le programme sous le nom Test4Coins.</p>
--------------------	---

- 5.2 Réalisez un organigramme afin que RobotProg aille jusqu'au mur en utilisant un bloc test. Ensuite, insérez votre organigramme dans le cadre ci-dessous.

5.3 Réalisez un organigramme afin que RobotProg avance jusqu'au mur puis réalise un demi-tour et retourne à sa base. Ensuite, insérez votre organigramme dans le cadre ci-dessous.



A large empty rectangular box for drawing the flowchart.

6. Réalisation de l'organigramme « **La boucle TantQue** ». Suivez les étapes ci-dessous.

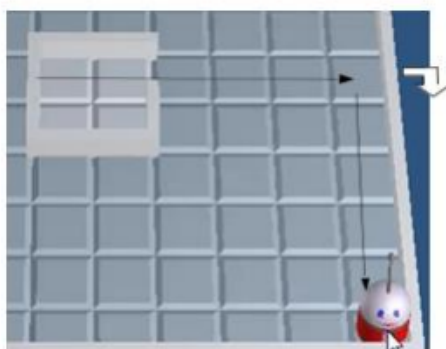
6.1 Dans RobotProg, vous devez passer au niveau 5 : Menu **Configuration** → **Niveau** → **5**

6.2 Pour compléter la boucle TantQue, vous devez ajouter deux mots-clefs. Voir le fichier « Mots-clefs-ROBOTPROG.pdf ».

TantQue

6.3 Réalisez un organigramme afin que RobotProg aille jusqu'au mur. Ensuite, insérez votre organigramme dans le cadre ci-dessous.

6.4 Réalisez un organigramme afin que RobotProg se déplace jusqu'au mur, tourne à droite et se rende jusqu'au coin comme indiqué sur la figure ci-dessous. Ensuite, insérez votre organigramme ci-dessous.



Validation par enseignante:

--