

TP N°7

1. Compétences à acquérir

Suite à ce TP, vous serez en mesure de :

- Mettre en application les connaissances acquises dans les TP précédents.
- Utiliser un algorithme d'exclusion mutuelle pour résoudre un problème réel.

2. Introduction

Ce TP est une continuité des tests de connaissances des TP 1 et 2 et dans lesquels nous avons remarqué l'existence d'un problème de collisions lors du déplacement des robots mobiles, cela est dû aux faits que les robots mobiles ne communiquent pas entre eux.

L'objectif de ce TP consiste à utiliser un algorithme d'exclusion mutuelle pour contrôler le déplacement des robots mobiles tout en évitant les collisions.

3. Enoncé

Nous désirons développer une application pour simuler le comportement des robots mobiles sur un plan orthonormé à 4 directions de dimensions ($h \times l$). Chaque robot est considéré comme un agent qui reçoit une liste de paramètres nécessaire à son fonctionnement. Chaque robot doit récupérer ses paramètres et les convertir selon un format adéquat pour les stocker dans des variables. Nous supposons que :

- Chaque robot est initialement placé dans une case du plan de déplacement (voir Figure 1) ayant les coordonnées (x_{actual} , y_{actual}).

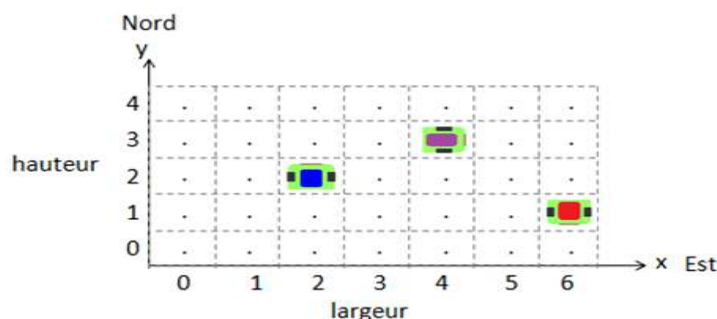


Figure 1 : Plan de déplacement des robots mobiles.

- Chaque robot possède une direction initiale d_{actual} parmi les valeurs : « Nord », « Sud », « Est » et « Ouest ».
- Chaque robot effectue un nombre infini de déplacements et ne doit pas sortir du plan de déplacement.
- Chaque robot effectue un seul déplacement à la fois selon les 4 directions possibles (voir Figure 2).

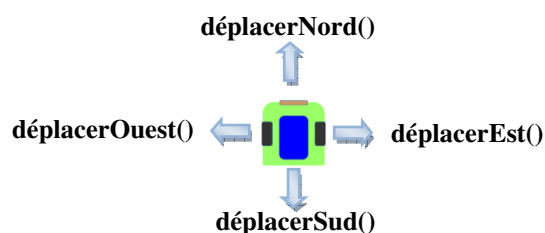


Figure 2 : Déplacements possibles d'un robot mobile.

- Le déplacement d'un robot s'effectue d'une façon aléatoire c-à-d un nombre aléatoire entre 0 et 3 doit être généré tel que :
 - Si le 0 est généré alors le robot fait un déplacement dans la direction Est (la nouvelle direction = Est)
 - Si le 1 est généré alors le robot fait un déplacement dans la direction Ouest (la nouvelle direction = Ouest)
 - Si le 2 est généré alors le robot fait un déplacement dans la direction Nord (la nouvelle direction = Nord)
 - Si le 3 est généré alors le robot fait un déplacement dans la direction Sud (la nouvelle direction = Sud)
- Chaque robot reste un temps fini dans une case.
- Lorsqu'un robot se déplace vers une nouvelle case, son ancienne case sera libérée une fois arrivé dans la nouvelle case.
- Chaque robot possède les méthodes suivantes :
 - déplacerEst() : incrémente de 1 la valeur de x.
 - déplacerOuest() : décrémente de 1 la valeur de x.
 - déplacerNord() : incrémente de 1 la valeur de y.
 - déplacerSud() : décrémente de 1 la valeur de y.
 - tournerDroite() : tourner le robot à droite (si la direction = « Nord » alors elle devient « Est », si la direction = « Est » alors elle devient « Sud », ...). Cette méthode sera appelée dans les méthodes (déplacerEst(), déplacerOuest(), déplacerNord(), déplacerSud()) si l'ancienne et la nouvelle direction ne sont pas identiques (voir Figure 3).

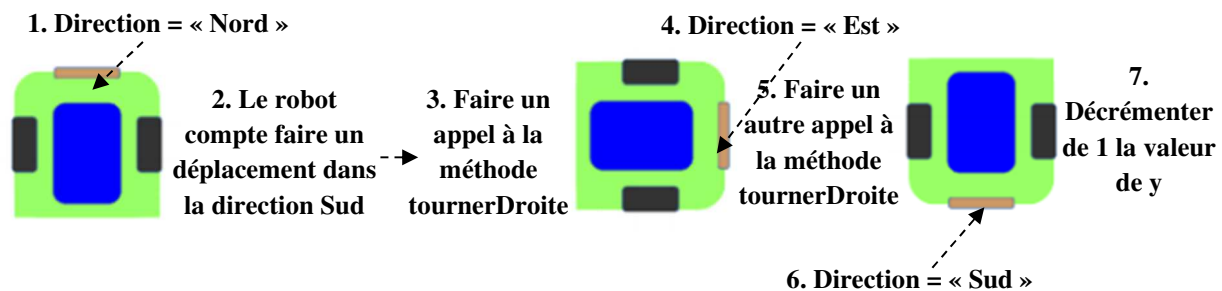


Figure 3 : Exemple d'utilisation de la méthode tournerDroite (direction initiale = « Nord », nouvelle direction = « Sud »).

4. Travail demandé

1. Proposer un protocole qui permet aux robots de se déplacer sans collisions en utilisant un algorithme d'exclusion mutuelle.
2. Tracer un graphe qui représente les comportements de chaque Robot et les relations qui existent entre les différents comportements
3. Quel(s) est(sont) le(s) message(s) échangé(s) entre les Robot ? Donner leurs contenus.
4. Quels sont les arguments qui seront passés à chaque Robot ? Donner leurs significations.
5. Implémenter la classe Robot (la méthode setup et les différents comportements).
6. Exécuter le programme en utilisant au minimum 3 robots.
7. Analyser le contenu de la console. Que remarquez-vous ?
8. Quel est le problème qui peut apparaître ? Proposer une solution à ce problème.