
	Projet Informatique Algo/C	
	ISEN CIR1	

Déplacement d'un robot en 2D

Ce projet a pour objectif la réalisation en langage C d'une simulation de déplacement (en 2D) d'un robot dans un environnement inconnu.

L'environnement est un appartement comportant plusieurs pièces. Le robot connaît sa position ainsi que la position de la porte de sortie, mais le plan de l'appartement lui est inconnu.

Le comportement du robot sera extrêmement simple: il est muni uniquement d'un capteur tactile à l'avant, c'est-à-dire qu'il ne détecte un obstacle que lorsqu'il le rencontre au cours d'un déplacement.

Cahier des charges :

- Le plan de l'appartement est « maillé », c'est-à-dire qu'il se compose de « cases » qui sont soit libres, soit occupées (par un obstacle fixe: mur, ou mobile: autre robot). Le cahier des charges rend obligatoire un plan de 50x50 mailles. C'est au programmeur de fournir un plan (qui peut changer d'une exécution à l'autre).
- Le robot occupe une maille et une seule.
- Le robot aura plusieurs caractéristiques dont un nom et une « vitesse maxi ». Le cahier des charges impose que la vitesse maxi soit inférieure ou égale à 10. Il connaît à tout instant sa position (x,y) dans le « plan ».
- Les déplacements du robot sont « discrets », c'est-à-dire que le pas de déplacement minimum est d'une maille (distance=1). Il peut se déplacer dans toutes les mailles voisines, y compris en diagonale (pour laquelle distance= $\sqrt{2}$), donc il y a 8 directions possibles : N, NE, E, SE, S, SW, W, NW. Les déplacements se font selon un mode « tour à tour » : à chaque tour le robot peut se déplacer d'une distance inférieure ou égale à sa « vitesse maxi ». S'il rencontre un obstacle, il est bloqué et devra attendre le tour suivant pour se déplacer à nouveau (logiquement dans une autre direction). Une seule direction est autorisée pendant un tour (pas de déplacement en « zig zag »).
- Le robot « sait » qu'il rencontre un obstacle quand la maille suivante, dans sa direction de déplacement, est occupée (ce qu'il saura en interrogeant le « plan »). Ceci simule un capteur tactile à l'avant du robot.
- Le robot, au début de chaque tour (avant son déplacement) pourra obtenir (en interrogeant le « plan ») l'écart ($\Delta x, \Delta y$) avec la position de la sortie.
- Toute amélioration de l'« intelligence » du robot est la bienvenue (par exemple, mémorisation de la position des obstacles fixes), mais ceci n'est pas obligatoire.

- Une interface graphique est demandée (cf. remarques en fin de ce texte). Le code correspondant à cette interface doit absolument être séparé du code concernant le comportement des robots : on doit pouvoir changer de type d'interface graphique, sans intervenir dans le reste du code.
- Des améliorations de l'interface sont possibles : musique ou sons par exemple, ... mais pas obligatoires.
- **Bonus** : en plus de ce robot « tactile », vous pouvez concevoir un autre robot différent (les deux seront présents simultanément sur le plan) : un robot plus complexe muni de capteurs « visuels » (il peut déterminer la direction et la distance de tous les obstacles « visibles », ceux-ci étant supposés opaques).

Contraintes de programmation :

- Le programme devra bien sûr être conçu selon une décomposition « procédurale » (en fonctions) et les règles de bonne programmation (pas de variables globales, nom des variables explicites,...) devront être respectées.
- Votre code doit être commenté et lisible (indenté, aéré,...).

Livraisons et tests demandés :

- Un rapport de fin de projet devra décrire le fonctionnement global du programme, donner quelques informations sur les techniques mises en œuvre ainsi que sur résolution des problèmes techniques les plus difficiles rencontrés en cours de projet.
- Les sources et fichiers nécessaires à la compilation et l'installation devront être fournis.
- Une démonstration du programme aura lieu en fin de projet. Le programme sera jugé sur le nombre de tours nécessaire pour que les robots trouvent la sortie sur un plan d'appartement non connu à l'avance.

Déroulement : projet de 26h (8 séances de 2 à 4h)

- Questions générales sur le sujet lors de la 1^{ère} séance
- Présentation d'avancement intermédiaire au cours de la 4^{ème} ou 5^{ème} séance
- Démonstration lors de la dernière séance

Remarques sur l'interface graphique

La librairie graphique 2D à privilégier est la *SDL*, pour laquelle on trouve de très bons tutoriels (Site du Zéro particulièrement). Cependant, si vous connaissez bien une autre librairie graphique au moins équivalente en fonctionnalités, vous pouvez l'utiliser.

Dans le pire des cas, si vous êtes un peu « trop justes » pour développer une interface graphique, vous devrez au moins permettre de visualiser votre plan et votre robot en mode « texte ».