

Projet P.O.O. 2018 : Robot Aspirateur

April 24, 2018

Le projet devra être réalisé en Java 8 minimum, et utiliser JavaFx.

Projet à faire en **Binômes**.

Les trinômes peuvent être autorisés après accord de votre enseignant.

Les personnes seules qui n'arrivent pas à trouver de binôme s'en verront affecter un d'office par l'enseignant.

Les sources seront à remettre sous forme d'une archive zip à travers la plateforme communities.

Aucun rendu par mail ne sera accepté.

1 Introduction

Dans ce projet il s'agit d'écrire un programme de robot aspirateur autonome. Le robot est muni de capteurs et se déplace en fonction des informations qu'ils lui fournissent. Le robot autonome doit couvrir la pièce au maximum et en prenant le moins de temps possible. Il doit aussi être capable de générer une cartographie de la pièce. Le programme doit au moins exécuter trois threads pour les classes Piece, Robot et Base.

2 La pièce

Les informations de la pièce sont contenues dans un fichier décrit section 2.1. Il faut créer un thread spécifique pour la classe Piece connaissant la description de la pièce après avoir chargé le fichier. Le robot ne connaît pas à l'avance la description de la pièce mais va la reconstituer en l'explorant et grâce à ses capteurs. La pièce est représentée par une matrice de zones carrées unitaires. Vous êtes libres de la manière de l'implémenter. Il faudra gérer les exceptions d'accès et de dépassement des limites données (dimensions de la pièce et cases "négatives"). On va considérer la pièce close et carrée ou rectangulaire. En conséquence les bords de la pièce ne seront pas spécifiés dans les fichiers de description.

2.1 Fichiers décrivant les pièces

Une pièce est décrite par un fichier texte. Elle est découpée en carrés unitaires. Le fichier correspond à une matrice où chaque paire de caractères correspond à un carré unitaire (ligne par ligne et colonne par colonne). Liste des paires de caractères / types de zone possibles :

- "BB" : la base, obligatoire et une et une seule
- "00" à "09" : sol différent de la moquette avec une quantité de poussière de 0 à 9
- "OO" : obstacle
- "VV" : vide (escalier)
- "T0" à "T9" : tapis ou moquette avec une quantité de poussière de 0 à 9

Pour tester le programme, nous fournirons des "pièces" à aspirer, avec des murs, des obstacles (tables, chaises, meubles, escaliers, poutres...) et une information de volume de poussière à chaque endroit accessible.

3 Le robot

Spécifications du robot (classe Robot) :

- il fonctionne sur batterie d'une capacité donnée
- il a une réserve de poussière d'une taille donnée
- quatre directions de déplacement sont possibles : haut, bas, gauche et droite
- le robot a 4 capteurs de collision (haut, bas, gauche, droite) et un de détection du vide (pour détecter d'éventuels escaliers)
- il a une base de recharge des batteries servant aussi à vider la réserve de poussière
- il doit générer une cartographie de la pièce au fur et à mesure
- il doit être capable de retourner à la base quand le niveau des batteries est faible ou quand le réservoir est plein
- il dispose d'une puissance d'aspiration fixe exprimée en quantité de poussière aspirée en un passage, l'aspiration est toujours active sauf quand il est sur la base ou quand le réservoir à poussière est plein

Prévoyez des méthodes d'accès et de modifications des spécifications, ainsi que différents constructeurs. Le robot peut retourner autant de fois qu'il le veut sur la base pour recharger ses batteries et décharger sa poussière. À chaque tentative de mouvement le robot consomme une unité d'énergie de sa batterie.

À son démarrage ou s'il change de direction il consomme deux fois plus d'énergie. S'il est sur de la moquette ou un tapis il consomme 1,5 fois plus d'énergie. Si l'aspiration est active l'aspirateur consomme en plus une autre unité d'énergie. Chaque tentative de mouvement doit durer 0,25 seconde qu'elle réussisse ou non. Si l'énergie de la batterie est insuffisante pour faire le mouvement alors il est annulé et la charge de la batterie devient 0. Si la charge de la batterie est nulle alors plus rien ne se passe. Il faut alors afficher une alerte la batterie est vide (dans une fenêtre d'alerte de l'interface graphique).

Quand le robot arrive à une position praticable, il aspire la poussière présente jusqu'à hauteur de sa puissance maximale et dans la limite de la place disponible dans sa réserve de poussière. Par exemple s'il a une puissance d'aspiration égale à 4 et qu'il arrive sur une case avec un indice de poussière égal à 9, il ne pourra aspirer que 4 quantités de poussière et il en restera 5 à cet endroit. Le robot devra repasser sur cette case pour pouvoir totalement la dépoussiérer. Par ailleurs, s'il ne reste dans sa réserve que 2 quantités de poussière de livres, il n'aspirera que 2 quantités de poussière pour remplir sa réserve et il restera 7 quantités de poussière sur cette zone. Lorsque le robot a exploré toute la pièce (c'est-à-dire quand la pièce est entièrement cartographiée) et aspiré toute la poussière, il retourne à la base et un avertissement ménage terminé doit s'afficher pour annoncer la fin (via l'interface graphique).

3.1 Les capteurs

Le robot est autonome et utilise ses capteurs pour se déplacer et créer une cartographie de la pièce. Les capteurs sont reliés au robot et ont un bit d'état qui change s'ils détectent ou non un obstacle. À chaque tentative de mouvement du robot, les capteurs interrogent la pièce et changent leur état en fonction de la zone visée. Si un des capteurs de collision ou de détection du vide devient actif alors le robot ne peut pas avancer sur la case visée et il reste à sa position. Dans ce cas le robot doit essayer une autre direction jusqu'à ce qu'il en trouve une praticable. Grâce aux tentatives infructueuses et en fonction du capteur l'ayant averti d'un obstacle, le robot détecte les obstacles et peut en connaître la nature (obstacle ou vide).

3.2 La base de recharge

La base (classe Base) sert à recharger la batterie du robot et à en vider le réservoir de poussière. Elle est aussi le point de départ du robot dans la pièce. La base a un capteur de présence du robot. Ce capteur est actif tant que le robot est positionné dessus. Quand le capteur change

d'état pour devenir actif, la base émet un signal au robot pour qu'il se mette en recharge et vide son réservoir de poussière (ce sont des opérations simultanées). Quand la batterie du robot est pleinement chargée et son réservoir de poussière vidé, le robot peut redémarrer et continuer son ménage. Le temps qu'il faut au robot pour accomplir sa tâche sera chronométré. Ce temps doit aussi inclure les durées de chaque recharge et vidange de la poussière (actions parallèles mais à détailler) : Il faut 5 secondes pour recharger le robot. Il faut 2 secondes pour vider le réservoir de poussière.

4 Interface graphique

Il faut également prévoir une interface graphique de commande / dépannage. On demande d'une part d'afficher les informations :

- le temps écoulé depuis le premier démarrage du robot
- la longueur cumulée du trajet parcouru par le robot
- le nombre de fois que le robot est allé à la base (sans compter le démarrage sur la base au départ)
- l'état de la batterie
- l'état du réservoir à poussière
- la quantité totale de poussière aspirée
- l'affichage de la cartographie générée

Il faut d'autre part permettre certains contrôles manuels :

- arrêt de la commande automatique et pilotage manuel du robot à la place avec des boutons directionnels et les flèches du clavier
- un bouton d'arrêt immédiat et d'ordre de retour à la base

Dans l'interface graphique, prévoir une barre de menu pour quitter et accéder à certains réglages via une fenêtre d'options dédiée :

- capacité de la batterie
- taille du réservoir à poussière

4.1 Représentation graphique de la pièce

Vous devrez représenter la pièce graphiquement au lieu d'afficher les paires de caractères décrites section 2.1. Cette représentation servira à afficher la pièce générée par le Robot dans la fenêtre principale et la pièce "vraie" dans une fenêtre à part, telle qu'elle est dans l'objet Piece. Pour cela, vous utiliserez des images carrées (16×16 ou 24×24 pixels pour ne pas faire trop grand). L'assemblage de ces images formera la pièce. Pensez à faire des représentations différentes des zones accessibles en fonction de la quantité de poussière encore présente (soit avec l'affichage d'un chiffre, soit avec un dessin représentant plus ou moins de poussière)

5 Modalités de notations

Le projet est à rendre via Moodle par groupe de deux. Les sources devront être rendus sous la forme d'une archive zip. Vous y inclurez un rapport faisant le bilan des fonctionnalités implémentées et dans lequel vous justifierez vos choix. Une soutenance aura également lieu (date à fixer) pendant laquelle votre binôme présentera son projet via une démonstration et une phase de questions de vos chargés de TD. Si votre projet présente les fonctionnalités de base (listées ci-dessous) vous aurez une note aux alentours de 13-14. Les fonctionnalités avancées vous apporterons des points supplémentaires.

Fonctionnalités de base:

- Interface graphique [/5pt]
- Contrôle manuel du robot [/4pt]
- Exceptions [/2pt]
- Threads [/1pt]

Fonctionnalités avancées:

- Ajouter des tests unitaires (pour la partie modèle du robot uniquement) [/2pt]
- Automatiser les déplacements du robot (I.A.): vous pouvez vous contenter de patterns de déplacement simple (en zig zag, en spirale), l'idée est d'ajouter un calcul du coût du plus court chemin pour rentrer à la base et que le robot retourne à la base par ce chemin de lui-même, quand c'est nécessaire.) [/3pt]
- Documentation (javadoc) [/1pt]

Éléments attendus dans le rapport : [/2pt]

- Introduction décrivant le projet (l'idée est de montrer que vous avez bien compris le projet.).
- Présenter les fonctionnalités implémentées en indiquant les divers bugs qu'il est possible de rencontrer, s'il y en a.
- Décrire l'arborescence du code source, en indiquant pour chaque fonctionnalité, dans quelle fichier, classe ou fonction elle a été implémentée. Indiquer les commandes pour compiler et exécuter le programme.
- Indiquer vos choix de conception, ou algorithme, en les justifiant.
- Indiquer la répartition des tâches au sein du binôme.
- Indiquer les difficultés rencontrées.
- Conclusion.