ENSTA IP PARIS

THALES

TP3: Déminage

Cédric Buron

7 décembre 2020

Introduction

Dans ce TP, nous allons étudier un système inspiré du *Mars Explorer* de Luc Steels un peu modifié. La problématique est la suivante : un endroit impénétrable aux ondes a été miné, et le but est de le déminer. Pour cela, on emploie une équipe de robots capables de détecter les mines et de les désarmer. Puisqu'il est impossible de communiquer avec l'intérieur de l'endroit, les machines doivent constituer un système autonome et efficace. Les agents décrits dans le cadre de ce TP sont basés sur les principes des agents réactifs et permettent de résoudre ce problème.

L'environnement est composé de trois éléments principaux :

- les mines sont les objets que le robot doit détecter. Une fois une mine détectée, le robot se place aux mêmes coordonnées qu'elle et le détruit,
- les obstacles sont des éléments qui ne doivent pas être détruits. De plus, un robot ne doit jamais se retrouver sur un obstacle.
- des sables mouvants, qui ralentissent les agents (ils vont alors à la moitié de leur vitesse normale)

Enfin, les robots doivent s'éviter. On supposera qu'ils ne doivent jamais se croiser.

Description basique du robot

Le robot est doté d'un certain nombre de capacités primaires : il peut se déplacer, détecter un obstacle ou un autre robot à éviter, détecter une mine et la détruire. Voici les comportements attendus par le robot :

Se déplacer lorsque le robot est à la recherche de mines, il se déplace de manière rectiligne, à une vitesse `speed` et selon un angle `angle`. Selon une probabilité fixée par avance (`PROBA_CHGT_ANGLE`), le robot peut changer de direction. Dans ce cas, son nouvel angle est un angle aléatoire choisi. Si un agent commence son tour dans des sables mouvants, il se déplace à la moitié de sa vitesse.

Éviter un autre robot Lorsqu'un ou plusieurs autres robots se trouve à portée de détection, le robot calcule son futur mouvement et vérifie si les robots à portée pourraient entrer en collision avec lui. Puisqu'il ne connaît que la position des autres robots et non leur angle, il considère qu'une collision est possible si sa position au tour suivant ou n'importe quelle position intermédiaire entre sa position actuelle et sa position future se trouvent à une distance inférieure à la vitesse maximale de l'autre robot. Si c'est le cas, il modifie son angle jusqu'à trouver un angle où il n'y ait pas de collision.

Éviter un obstacle / les bords de l'environnement Si un robot détecte un ou plusieurs obstacles (y compris les bords de l'environnement), il procède au même calcul, mais il considère qu'une collision est possible si sa future position est dans l'obstacle. Pour simplifier, on ne prendra pas en compte le fait qu'un robot puisse traverser un obstacle.

Détruire une mine Lorsqu'un agent se trouve sur une mine, il la désamorce (dans mesa, il la retire du `model`). Un agent peut détruire une mine et se déplacer le même tour.

Se diriger vers une mine Lorsqu'un agent détecte une mine, il se dirige vers la mine à sa vitesse, sans prendre en compte ni modifier son angle.

Le robot est considéré comme poctuel (il n'a pas d'épaisseur). Tous les robots ont la même vitesse et le savent, ils ont donc la connaissance de la vitesse des autres robots, ce qui leur permet de calculer leur rayon d'action. Notez que dans le code, les angles sont calculés en radiants.

Question 1– Quelle architecture vous paraît la plus à même de traiter ce problème ?

Question 2– Il est capital que les robots ne rentrent pas en collision les uns avec les autres. Cela prime sur chacun de leurs déplacement. Proposer un ordre de priorité pour les comportements décrits ci-dessus qui respecte cette contrainte et permette aux robots d'accomplir leur mission (NB, il est possible que plusieurs comportements doivent être fusionnés)

Implémentez l'ordre que vous proposé précédemment. Ajoutez une métrique représentant le cumul des mines désamorcées à chaque tour. Enregistrez ce graphe et joignez le à votre TP. Lancez la simulation une dizaine de fois et donnez le temps moyen de désamorçage de toutes les mines.

Question 3– Quels principes des agents réactifs sont ici respectés ? Lesquels ne le sont pas ? Justifiez.

Communication indirecte

Pour faire communiquer nos robots, nous allons utiliser l'environnement. Nous allons utiliser des balises (`marker`), comme Steels le décrit pour le *Mars Explorer*. Les robots ont deux types balises. Un agent dépose une balise `DANGER` lorsqu'il sort de sables mouvants, afin que les autres agents n'y pénètrent pas, et une balise `INDICATION`, déposée lorsque l'agent démine, et qui indique dans quelle direction il est allé. Un agent qui détecte une balise `DANGER` fera demi-tour. Un agent détectant une balise `INDICATION` modifie son angle afin de se déplacer à 90° dans une direction ou l'autre par rapport à l'angle indiqué sur la balise. Un robot détectant une balise se déplace jusqu'à elle, la ramasse et fait demi-tour.

Plus formellement, voici la description des comportements attendus pour le robot :

Faire demi-tour Lorsqu'un agent détecte une balise `DANGER`, il fait demi-tour.

Tourner à 90° Lorsqu'un agent détecte une balise `INDICATION`, il se dirige à 90° de la direction indiquée par la balise

Déposer une balise Lorsqu'un agent vient de sortir des sables mouvants, un agent dépose une balise `DANGER`; lorsqu'un agent démine, il dépose une balise `INDICATION` dont la `direction` indique la direction dans laquelle se dirige l'agent. Attention, pour éviter que le robot ne soit influencé par la balise qu'il vient de déposer, un compteur de tour est initialisé à `speed/2` durant lequel le robot ignore les balises qu'il voit.

Question 4– Commentez ce dernier point au regard des principes de l'architecture réactive.

Question 5– Proposez une nouvelle manière d'organiser ces comportements et justifiez.

Question 6– Relancez 10 simulations et incluez un graphe dans le rendu du TP. Quel est le temps moyen de désamorçage des mines ? Commentez.

Question 7– Ajoutez un reporter permettant de suivre le nombre de tours passés dans les sables mouvants. Comment les balises `DANGER` influencent-elles le temps moyen passé dans les sables mouvants ?

Question 8– (Bonus) Ajouter le mécanisme suivant : on suppose désormais que les agents sont capables de transmettre un signal leur permettant de savoir où se trouvent les autres. Lorsqu'un agent modifie son angle aléatoirement, faites en sorte qu'il se tourne de manière à maximiser l'angle entre sa nouvelle direction et la direction envers chacun de ses 2 pluss proches voixins. Relancez 10 expérimentations. Qu'observez-vous ?

Si certains éléments du framework mesa vous ont posé problème, merci de l'indiquer à la fin du fichier `reponses.md`