

# Comment développer une IA en Prolog pour le jeu Rasende Roboter ?

par Fabrice Lauri

11 Décembre 2011

L'IA à programmer pour le jeu *RasendeRoboter* doit être implémentée dans le fichier "**decision.pl**" présent dans le répertoire "**CognitiveSystems/prolog/00/**".

Ce fichier contient deux prédicats, *init/1* et *move/2* décrits ci-dessous.

Le prédicat *init/1* est consulté une seule fois lors du chargement du jeu. Il est recommandé d'effectuer l'initialisation des variables globales dans ce prédicat.

Le prédicat *move(+GameConfiguration, -Movements)* est constitué de :

- une liste instanciée de 13 éléments nommée *GameConfiguration*, de la forme **(TL,TR,BL,BR, TargetId, BlueRobotPosition, GreenRobotPosition, YellowRobotPosition, RedRobotPosition)**, telle que :
  - **TL, TR, BL et BR** sont des booléens, renseignant sur la configuration initiale du plateau. Il y a donc 16 configurations initiales possibles. Chaque configuration se distingue d'une autre par la disposition spécifique des murs et des cibles.
  - **TargetId** indique le numéro de la cible selon la codification suivante :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
																

- **BlueRobotPosition, GreenRobotPosition, YellowRobotPosition, RedRobotPosition** sont les coordonnées (x,y) respectivement du robot bleu, vert, jaune et rouge.
- Une liste de mouvements *Movements* = [R1,D1, R2,D2, ..., Rn,Dn] à instancier, telle que :
  - chaque Ri représente le robot à déplacer, selon la codification ci-dessous :
    - 0 : robot bleu
    - 1 : robot vert
    - 2 : robot jaune
    - 3 : robot rouge
  - chaque Di représente la direction que devra suivre le robot Ri, selon la codification ci-dessous :
    - 1 : droite
    - 2 : haut
    - 3 : gauche
    - 4 : bas

Démarche à suivre pour déterminer une solution :

1. Ne considérer qu'une seule configuration initiale du plateau (disposition des murs+cibles).
2. Utiliser un A\* pour déterminer la séquence de mouvements des robots permettant à l'un d'eux d'atteindre la cible sélectionnée au hasard.
3. Selon votre cahier des charges, un état dans A\* peut être :
  - un tuple de la forme **(BlueRobotPosition, GreenRobotPosition, YellowRobotPosition, RedRobotPosition, Target)** si vous permettez à n'importe quel robot de se déplacer pour atteindre la cible. Dans ce cas, il est important de noter qu'il n'y a pas qu'un seul état final mais un ensemble d'états finaux. Les informations en commun pour ces états finaux sont la cible

à atteindre et la position du robot de la couleur de la cible. La position des autres robots est a priori inconnue...

- un tuple de la forme (**Target, RobotPosition**) où seul le robot de la couleur de la cible est invité" à se déplacer. Dans ce cas, étant donné que les autres robots resteront immobiles, il n'est pas nécessaire de stocker leur position dans un état. Leurs positions peuvent être stockées dans une ou des variables globales qui pourront être consultées au besoin.
4. Une heuristique admissible pour ce problème peut être la distance de Manhattan.