

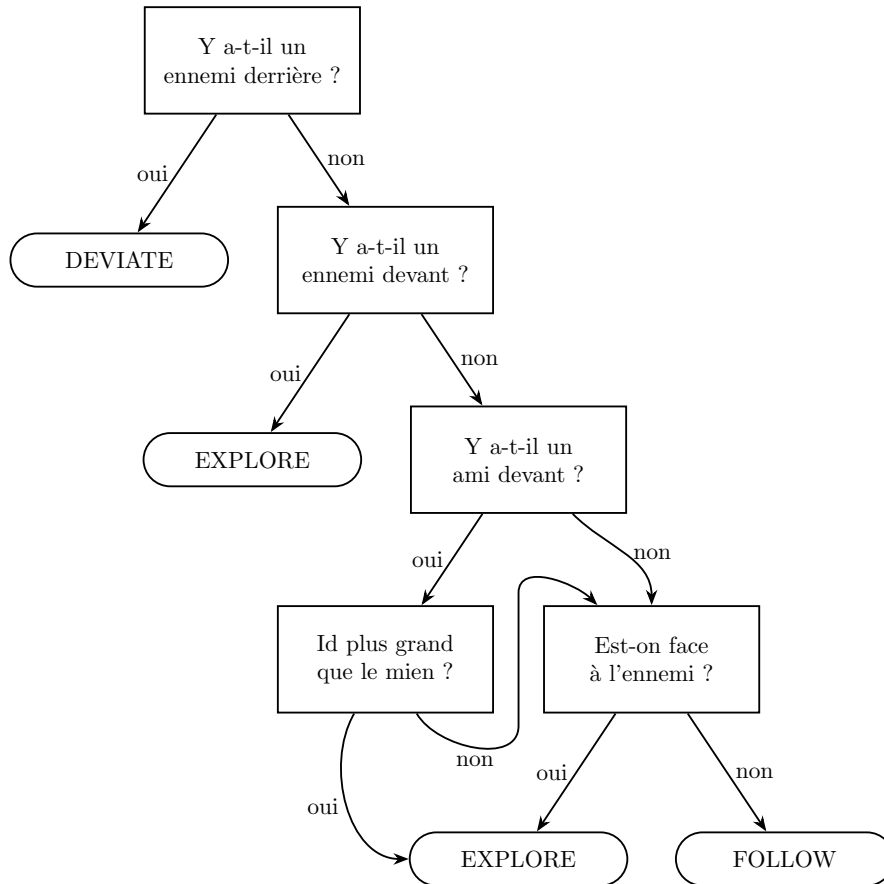
3I025 — Projet

Ariana CARNIELLI et David HERZOG

On a deux versions du joueur, une première que l'on appelle A basée sur un arbre de décision simple avec des paramètres optimisés par l'algorithme génétique et une seconde, appelée B, basée sur une arbre de décision plus complexe.

L'arbre de décision de A a trois comportements de base : explorer l'espace en évitant murs et autres robots (EXPLORE), suivre un robot ennemi (FOLLOW) et essayer de dévier d'un robot ennemi qui nous suit (DEVIATE). Dans les comportements EXPLORE et FOLLOW, on regarde aussi si le robot est bloqué et fait une manœuvre pour le débloquent si nécessaire. L'arbre A est donné dans la figure suivante :

L'optimisation par l'algorithme génétique utilise 44 paramètres, correspondant à deux types de robots, chacun à 22 paramètres. Ces paramètres comprennent 8 coefficients multipliant le vecteur des capteurs pour chacun des comportements EXPLORE et FOLLOW, 2 coefficients déterminant le comportement DEVIATE, un coefficient pour la vitesse de FOLLOW et un autre pour l'angle à partir du quel on suit un robot, un coefficient pour le bruit dans l'exploration et un dernier coefficient déterminant combien de pas sur la même position déterminent un état bloqué.



L'arbre de décision B se décompose en trois parties, I, II et III, et contient les comportements de base suivants :

- GO_FOWARD : aller tout droite
- AVOID_WALLS : éviter des murs et ignorer des robots
- FOLLOW_ENNEMY : suivre un robot
- AVOID_MATES : éviter des robots amis
- EXPLORE : éviter des robots et des murs
- ALONG_WALLS : longer un mur
- STOP : s'arrêter
- TURN_RIGHT : tourner à droite
- Arrière : marche arrière

La partie I détermine si le robot se déplacera en longeant les murs, ce qui n'est fait qu'au maximum par un robot (celui d'id 2). La partie II détermine si le robot doit suivre un ennemi, ce qui n'est fait qu'au maximum par deux robots (ids 0 et 2). Finalement, la partie III essaie de se débarrasser des ennemis qui nous suivent et éviter les situations de blocage. Des derniers tests permettent de donner des ajustements fins au comportement du joueur suivant les murs, en utilisant la stratégie TURN_RIGHT.

