

DEVOIR à Rendre le 14/01/2023

EXERCICE 1 : Planification jointe

Trois robots évoluent dans une grille avec des destinations distinctes. Pour éviter des collisions, une planification calcule un plan joint pour les deux robots qu'ils doivent exécuter.

1. Donner la formalisation de ce problème en termes d'états, actions, et de possible fonctions heuristiques en se fondant sur les fonctions CUF.
2. Implémenter un algorithme de planification à base de A* pour calculer un plan joint, en prenant l'exemple de la grille ci-dessous. Le programme doit afficher la position de chaque robot à chaque action du plan d'actions.

R1				G3	
				G2	
R2	R3				
		G1			

EXERCICE 2 : Algorithmes minmax

1. Implémenter l'algorithme minmax sur une structure d'arbre et tester le sur l'exemple donné ci-dessous.
2. **OPTIONNEL : implémenter l'algorithme alpha/beta pour ceux qui le souhaitent (points de bonus).**

EXERCICE 3 : ROS

Un système de surveillance d'un site sensible est composé d'un drone D et de trois robots terrestres R1, R2 et R3. Le drone survole le site et envoie des messages avec 3 digits xxx (par exemple 010) permettant d'informer sur l'état des trois zones du site pour dire à vérifier on non. On suppose que les robots évoluent dans une grille et que les positions et les zones sont des cellules de la grilles.

Ecrire un programme en ROS permettant au drone de publier les messages en 3 digits et que les robots traitent de la manière suivante : C'est le robot R1 qui choisit en premier le site à vérifier et publie un message de second tour à trois digits après vérification que le R2 traitera de la même manière et publie un message de troisième tour que le robot R3 traitera.

Par exemple, si le drone publie 111, le robot R1 traite le message et choisit, par exemple le site 2 à vérifier, il publiera le message 101 (car il a vérifié le site) que le robot R2 traitera par exemple choisira le site 1 à vérifier, il publiera le message 001 car il a vérifié le site 1 et le robot R3 traitera le message 001. Si le message reçu est 000, les robots ne font rien.

Complément :

Les étudiants ayant programmé des algorithmes vus en TD, peuvent les rendre pour des points de Bonus.

Procédure du rendu :

- 1. Vous devez rendre une archive Linux (seuls les archives .zip ou .gz sont acceptées) qui contient 4 répertoires : exercice1, exercice 2, exercice 3 et complément. Chaque répertoire doit contenir le(s) programme(s) et un README.txt expliquant le programme et comment l'utiliser.**
- 2. Sur E-campus, vous avez le sujet du devoir ainsi que le lien pour le dépôt de votre devoir au plus tard le 14 Janvier 2023.**