

Conception de notre projet

Organisation des fichiers

Nous avons séparé l'ensemble des fichiers en différentes catégories dans le dossier src :

- enumerator : contenant les enum Direction (NORTH, SOUTH, EAST, WEST) et TypeLand (WATER, FOREST, STONE, FIELD, HABITATION)
- fire : contenant uniquement Fire, un fichier créant les feux et assurant la gestion de ceux-ci, notamment la diminution et la propagation
- gui : contenant l'ensemble des fichiers pour le bon fonctionnement de l'interface graphique
- io : contenant la gestion des données de lecture et de dessin pour l'interface graphique
- map : contenant la définition de la carte (ensemble de Box), l'état actuel de la carte ainsi que la recherche des plus courts chemins avec AStar
- robot : contenant tous les types de robots ainsi que le robot principal CaptainRobot
- simulation : contenant toute la gestion du lancement de la simulation ainsi que l'avancement des événements durant celle-ci
- fichier main pour lancer la simulation et la terminer

Spécifications sur l'implémentation

Vu que nous avons implémenté le projet à notre manière, certaines notions ou implémentations sont expliquées ci-dessous :

-

Gestion des erreurs et tests

Nous avons, tout au long du projet, ajouté des tests afin d'éviter tout problème :

- Vérification que l'on reste dans la carte en aval d'un déplacement (lance une erreur si un robot sort de la carte)
- Vérification des fichiers donnés en argument lors du lancement de la simulation (lance une erreur si les arguments sont incorrects)
- Vérification des terrains lors des déplacements des robots (certains ne peuvent pas traverser des terrains spécifiques)
- Vérification de la décroissance de l'intensité des feux au fil du temps

Optimisations

Nous avons voulu optimiser certaines parties de notre code afin d'alléger le nombre de calculs lors de la simulation :

- Lorsqu'un robot n'a aucun chemin vers le moindre feu (il est donc inutile), nous le laissons de côté et ne faisons plus de calcul sur celui-ci

- Nous avons préféré implémenter A Star pour calculer les plus courts chemins au lieu de Dijkstra pour gagner en complexité temporelle; notre heuristique H est la suivante:

$$H_r(x,y) = \frac{|abs(r) - x| + |ord(r) - y|}{v(r)}$$

avec (x, y) les coordonnées de la case visée, $abs(r)$ et $ord(r)$ respectivement l'abscisse actuelle du robot r et l'ordonnée actuelle de r, et $v(r)$ la vitesse actuelle de r

- Les robots font un nouveau calcul du feu le plus proche lorsque l'un est éteint pour éviter tout trajet inutile si un autre robot se dirigeait vers ce feu

